Chapter 14
Hybrid Systems & Games

Synopsis This chapter begins the study of an entirely new model of cyber-physical systems: that of *hybrid games*, which combine discrete, continuous, and adversarial dynamics. While hybrid systems with their discrete and continuous dynamics have served us well in the analysis of cyber-physical systems so far, other cyber-physical systems crucially require an understanding of additional dynamical effects. Adversarial dynamics is relevant whenever choices in the system can be resolved by different players. This happens frequently in CPSs with multiple agents who may or may not agree on a common goal or who, even if they share a common goal, may act differently based on a different perception of the world. This chapter discusses the far-reaching consequences of this insight and advances hybrid programs to a programming language for hybrid games.

14.1 Introduction

Hybrid systems have served us well throughout this textbook as a model of cyber-physical systems [1, 3, 7, 11]. But contrary to what we simply pretended in Parts I and II, hybrid systems and cyber-physical systems are not the same. Hybrid systems can also serve as models of other systems that are not cyber-physical per se, i.e., they are not built as a combination of cyber and computing capabilities with physical capabilities. Some biological systems can be understood as hybrid systems, because they combine discrete activation of genes and continuous biochemical reactions. Or physical processes can be understood as hybrid if things happen at very different speeds. Then, there is a slow process about which a continuous understanding is critical as well as a very fast process in which a discrete abstraction might be sufficient. Just think back to the bouncing ball where a discrete understanding of the event of the bounce was more suitable even if a continuous deformation occurs, but at a much faster pace than the continuous falling due to gravity. None of those examples is particularly cyber-physical. Nevertheless, they can be naturally modeled as hybrid systems, because their fundamental characteristic is the interaction of dis-

A. Platzer, *Logical Foundations of Cyber-Physical Systems*,
https://doi.org/10.1007/978-3-319-63588-0_14

crete and continuous dynamics, which is exactly what hybrid systems are good for. Hybrid systems are a mathematical model of dynamical systems with mixed discrete and continuous dynamics, whether cyber-physical or not. Hence, despite their good match, not *all* hybrid systems are cyber-physical systems.

One important point of this chapter is that the converse is not true either. Not all cyber-physical systems are hybrid systems! The reason for that is *not* that cyber-physical systems lack discrete and continuous dynamics, but, rather, that they involve also additional dynamical aspects. It is a pretty common phenomenon in cyber-physical systems that they involve several dynamical aspects, which is why they are best understood as *multi-dynamical systems*, i.e., systems with multiple dynamical features [4–7, 9, 10, 12].

In a certain sense, applications often have a $+1$ effect on dynamical aspects. Your analysis might start out focusing on some number of dynamical aspects only to observe during the elaboration of the analysis that there is another part of the system for which one more dynamical aspect is relevant than was originally anticipated. The bouncing ball is an example to which a preliminary analysis might first ascribe an entirely continuous dynamics, just to find out after a while that the singularity of bouncing back from the ground can be more easily understood by a discrete dynamics. Whenever you are analyzing a system, be prepared to find one more dynamical aspect around the corner! That is yet another reason why it is useful to have flexible and general analysis techniques grounded in logic that still work even after a new dynamical aspect has been found.

Of course, it is not going to be feasible to understand all multi-dynamical system aspects at once in this chapter. But this chapter is going to introduce one absolutely fundamental dynamical aspect: *adversarial dynamics* [9, 12]. Adversarial dynamics comes from multiple players that, in the context of a CPS, interact on a hybrid system and are allowed to make their respective choices arbitrarily, in pursuit of their goals. The combination of discrete, continuous, and adversarial dynamics leads to *hybrid games*. Unlike hybrid systems, hybrid games allow choices in the system dynamics to be resolved adversarially by different players with different objectives.

Hybrid games are necessary in situations where multiple agents actively compete. The canonical situation of a hybrid game would, thus, be when two teams of robots play robot soccer, moving around physically in space, controlled according to discrete computer decisions, and in active competition to score goals in opposite directions on the field. The robots in a robot soccer match can't agree on the direction in which they try to get the ball rolling. This leads to a mix of discrete, continuous, and adversarial dynamics for truly competitive reasons.

It turns out, however, that hybrid games also come up for reasons of *analytic competition*, that is, where possible competition is assumed only for the sake of a worst-case analysis. Consider a robot that is interacting with another robot, let's call it the *roguebot*. You are in control of the robot, but somebody else is controlling the roguebot. Your objective is to control your robot so that it will not run into the roguebot no matter what. That means you need to *find some* way of using your control choices for your robot so that it makes progress toward its goal but will remain safe *for all* possible control choices that the roguebot might follow. After all, you

do not know exactly how the other roguebot is implemented and how it will react to your control decisions. That makes your robot play a hybrid game with the roguebot in which your robot is trying to safely avoid collisions. The roguebot might behave sanely and try to stay safe as well. But the roguebot's objectives might differ from yours, because its objective is not to get your robot to your goal. The roguebot rather wants to get to its own goal instead, which might cause unsafe interference whenever the roguebot takes an action in pursuit of its goal that is not in your robot's interest. If your robot caused a collision, because it chose an action that was incompatible with the roguebot's action, your robot would certainly be faulty and be sent back to the design table. And even when both robots perfectly agree on the same goal, their actions might still cause unintended interferences when their perception of the world differ. In that case the two robots could take conflicting actions despite pursuing the same goal, just because they each thought the state of the world was a little different. Just imagine a common goal of not colliding with a rule that whoever is further west moves even further to the west. Now if both robots think they are the one that is further west because their sensors tell them that, then they might still collide even if both really didn't mean to.

Alas, when you try to understand how you need to control your robot to stay safe, it can be instructive to think about what the worst-case action of a roguebot might be to make life difficult for you. When a test engineer is trying to demonstrate under which circumstance a simulation of your robot controller exhibits a faulty behavior, so that you can learn from the cases where your control does not work, they actually play a hybrid game with you. If your robot wins and stays safe, that is an indication of a good robot design at least in this scenario. But if the test engineer wins and shows an unsafe trace, then you still win even if you lose this particular simulation, because you learn more about the corner cases in your robot control design than when staring at simulation movies where everything is just fair-weather control.

This chapter is based on prior work [9], where more information can be found on logic and hybrid games. The most important learning goals of this chapter are:

Modeling and Control: We identify an important additional dynamical aspect, the aspect of *adversarial dynamics*, which adds an adversarial way of resolving the choices in the system dynamics. This dynamical aspect is important for understanding the core principles behind CPS, because multiple agents with possibly conflicting actions are featured frequently in CPS applications. Such conflicting actions might be chosen due to different goals or different perceptions of the world. It is helpful to learn under which circumstance adversarial dynamics is important for understanding a CPS and when it can be neglected without loss. CPSs in which all choices are resolved against you or all choices are resolved for you can already be described and analyzed in differential dynamic logic using its box and diamond modalities [7]. Adversarial dynamics is interesting in mixed cases, where some choices fall in your favor and others turn out against you. Another important goal of this chapter is to develop models and controls of CPS with adversarial dynamics corresponding to multiple agents.

Computational Thinking: This chapter follows fundamental principles from logic and computational thinking to capture the new phenomenon of adversarial dy-

namics in CPS models. We leverage core ideas from programming languages by extending syntax and semantics of program models and specification and verification logics with a new operator for duality to incorporate adversariality in a modular way into the realm of hybrid systems models. This leads to a compositional model of hybrid games with compositional operators. Modularity makes it possible to generalize our rigorous reasoning principles for CPS to hybrid games while simultaneously taming their complexity. This chapter introduces *differential game logic* dGL [9, 12] extending by adversarial dynamics the familiar differential dynamic logic, which has been used as the specification and verification language for CPS in Parts I and II. Computer science ultimately is about analysis such as worst-case analysis, expected-case analysis, or correctness analysis. Hybrid games enable analysis of CPSs at a more fine-grained level in between worst-case analysis and best-case analysis. In the dL formula $[\alpha]P$ all choices are resolved against us in the sense that $[\alpha]P$ is only true if P holds after all runs of α. In the dL formula $\langle\alpha\rangle P$ all choices are resolved in our favor in the sense that $\langle\alpha\rangle P$ is true if P holds after at least one run of α. Hybrid games can be used to attribute some but not all of the choices in a system to an opponent while leaving others to be resolved favorably. Finally, this chapter provides a perspective on advanced models of computation with alternating choices.

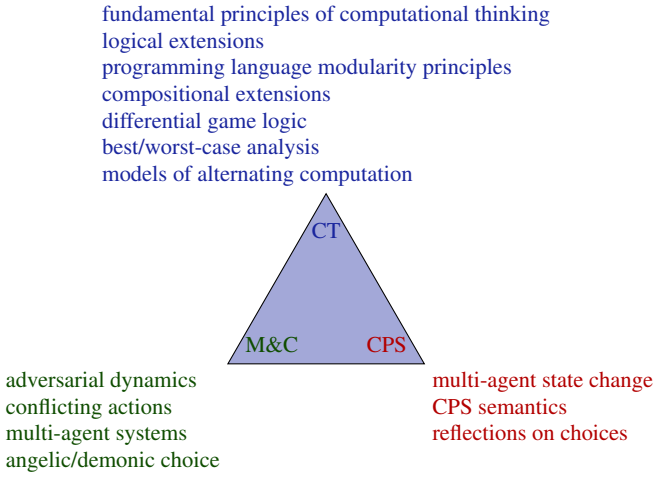
CPS Skills: We add a new dimension into our understanding of the semantics of a CPS model: the adversarial dimension corresponding to how a system changes state over time as multiple agents react to each other. This dimension is crucial for developing an intuition for the operational effects of multi-agent CPS. The presence of adversarial dynamics will cause us to reconsider the semantics of CPS models to incorporate the effects of multiple agents and their mutual reactions. This generalization, while crucial for understanding adversarial dynamics in CPS, also shines a helpful complementary light on the semantics of hybrid systems without adversariality by causing us to reflect on the rôle of choices.

14.2 A Gradual Introduction to Hybrid Games

This section gradually introduces the operations that hybrid games provide one step at a time. Its emphasis is on their motivation and an intuitive development starting from hybrid systems before subsequent sections provide a comprehensive view.

14.2.1 Choices & Nondeterminism

The first thing to remind ourselves about is that hybrid systems also already come with choices, and for good reasons, too.



Note 71 (Choices in hybrid systems) Hybrid systems involve choices. They manifest in hybrid programs as nondeterministic choices $\alpha \cup \beta$ whether to run HP α or HP β, in nondeterministic repetitions α^* where the choice is how often to repeat α, and in differential equations $x' = f(x) \,\&\, Q$ where the choice is how long to follow that differential equation. All those choices, however, have been resolved in one way, i.e., by the same entity or player: nondeterminism.

In which way the various choices are resolved depends on the context. In the box modality $[\alpha]$ of differential dynamic logic [1, 3, 7, 11], all nondeterminism is resolved in *all possible ways* so that the modal formula $[\alpha]P$ expresses that formula P holds for all ways in which the choices in HP α could be resolved. In the diamond modality $\langle\alpha\rangle$, instead, all nondeterminism is resolved in *some way* so that formula $\langle\alpha\rangle P$ expresses that formula P holds for at least one way of resolving the choices in HP α. The modality decides the mode of nondeterminism. The modal formula $[\alpha]P$ expresses that P holds necessarily after running α while $\langle\alpha\rangle P$ expresses that P is possible after α.

In particular, choices in α help $\langle\alpha\rangle P$, because what this formula calls for is *some* way of making P happen after α. If α has many possible behaviors, this is easier to satisfy. Choices in α hurt $[\alpha]P$, however, because this formula requires P to hold for *all* those choices. The more choices there are, the more difficult it is to make sure that P holds after every single combination of those choices.

In differential dynamic logic, choices in α help uniformly (when they occur in $\langle\alpha\rangle P$) or make matters more difficult uniformly (when they occur in $[\alpha]P$).

That is why these various forms of choices in hybrid programs have been called *nondeterministic*. They are "unbiased." All possible resolutions of the choices in α can happen nondeterministically when running α. Which possibilities we care about

(all or some) just depends on the modality around it. However, in each hybrid systems modality, all choices are uniformly resolved in one way, because we can only wrap one modality around the hybrid program. We cannot say that some choices within a modality are meant to help, others are meant to hinder.

By nesting other modalities in the postconditions, we can still express some limited form of alternation in how choices resolve:

$$[\alpha_1]\langle\alpha_2\rangle[\alpha_3]\langle\alpha_4\rangle P$$

This dL formula expresses that after all choices of HP α_1 there is a way of running HP α_2 such that for all ways of running HP α_3 there is a choice of running HP α_4 such that postcondition P is true. But that still only gives an opportunity for four rounds of alternation of choices in HPs and is not particularly concise even for that purpose. What we need is a more general way of ascribing actions to agents that allows an unbounded number of alternation of choices.

14.2.2 Control & Dual Control

Another way of looking at the choices that are to be resolved during the runs of a hybrid program α is that they can be resolved by one player. Let's call her *Angel*, because she helps us so much in making $\langle\alpha\rangle P$ formulas true. Whenever a choice is about to happen (by running the program statements $\alpha\cup\beta$, α^*, or $x' = f(x)\,\&\,Q$), Angel is called upon to see how the choice is supposed to be resolved this time. When playing $\alpha\cup\beta$, Angel chooses whether to play α or β. When playing α^*, Angel decides how often to play α. And when playing $x' = f(x)\,\&\,Q$, Angel decides how long to follow this differential equation within Q. Since Angel gets to choose, $\alpha\cup\beta$ is also called *angelic choice* and α^* is called *angelic repetition*.

From that perspective, it sounds easy enough to add a second player. Let's call him *Demon* as Angel's perpetual opponent.[1] Only so far, Demon will probably be quite bored after a while, when he realizes that he never actually gets to decide anything in the game, because Angel has all the fun in choosing how the game world unfolds and Demon just sits around idly and in apathy. So, to keep Demon engaged, we need to introduce some choices that fall under Demon's control.

One thing we could do to keep Demon interested in playing along in the hybrid game is to add a pair of shiny new controls especially for him. They might be called $\alpha\cap\beta$ for Demon's choice between α or β and $\alpha^\times$ for repetition of α under Demon's control. In fact, Demon might even demand an operation for continuous evolution under Demon's reign. But that would cause quite a lot of attention to Demon's controls, which might make him feel overly majestic. Let's not do that, because we don't want Demon to get any ideas.

[1] The names are quite arbitrary. But the responsibilities of such ontologically loaded names are easier to remember than those of neutral but boring player names such as player I and player II.

Instead, we will find it sufficient to add just a single operator to hybrid programs: the duality operator $\cdot^{\mathrm{d}}$ that can be used on any hybrid game α. What α^{d} does is to give all control that Angel had in game α to Demon, and, vice versa, all control that Demon had in α to Angel. The dual operator, thus, is a little bit like what happens when you turn a chessboard around by 180° in the middle of the game to play the game from the opponent's perspective. Whoever played the choices of player White previously will suddenly control Black, and whoever played Black now controls White (Fig. 14.1). Turning the game around twice as in $(\alpha^{\mathrm{d}})^{\mathrm{d}}$ restores the original game α. With just this single duality operator, Demon still gets his own set of controls ($\alpha \cap \beta$, $\alpha^{\times}$, and $\{x' = f(x)\,\&\,Q\}^{\mathrm{d}}$) by suitably nesting the operators, but we did not have to give him those controls specifically. Yet, now those extra controls are not special but simply an aspect of a more fundamental principle: duality.

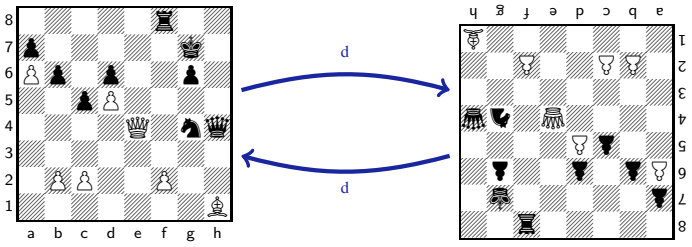


Fig. 14.1 Turning hybrid game α into the dual hybrid game α^{d} corresponds to turning a chessboard around by 180° so that the players control the choices in α^{d} that the opponent has in α

14.2.3 Demon's Derived Controls

Just as nondeterminism was in charge of all choices in a hybrid system, Angel has full control over all choices in each of the operators of hybrid games *except* when the operator $\cdot^{\mathrm{d}}$ comes into play. All choices within the scope of an odd number of $\cdot^{\mathrm{d}}$ belong to Demon, because $\cdot^{\mathrm{d}}$ makes the players switch sides. Demon's controls, i.e., direct controls for Demon, can be defined as derived operators with the duality operator $\cdot^{\mathrm{d}}$ from Angel's controls. Indeed, $(\alpha^{\mathrm{d}})^{\mathrm{d}}$, the dual of a dual, is the original game α, just like flipping a chessboard around twice results in the original chessboard. That is why it only matters whether a choice occurs within the scope of an odd number of $\cdot^{\mathrm{d}}$ (Demon's choice) or an even number of $\cdot^{\mathrm{d}}$ (Angel's choice).

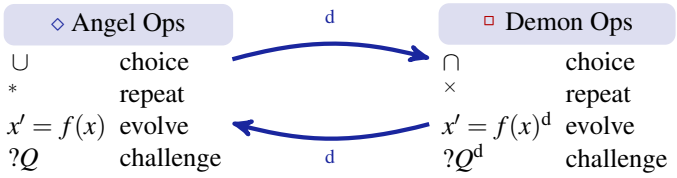
Demonic choice $\alpha \cap \beta$ will play either hybrid game α or hybrid game β by Demon's choice. It is defined by $(\alpha^{\mathrm{d}} \cup \beta^{\mathrm{d}})^{\mathrm{d}}$. The choice for the $\cup$ operator belongs to Angel, yet since it is nested within $\cdot^{\mathrm{d}}$, that choice goes to Demon, except that the $\cdot^{\mathrm{d}}$ operators around hybrid games α and β restore the original ownership of controls. The hybrid game $(\alpha^{\mathrm{d}} \cup \beta^{\mathrm{d}})^{\mathrm{d}}$ corresponds to turning the chessboard around, thus,

giving the choice between α^d and β^d that would have been Angel's to Demon, and then turning the chessboard in either α^d or β^d back again to either α or β.

Demonic repetition $\alpha^\times$ repeats hybrid game α as often as Demon chooses to. It is defined by $((\alpha^d)^*)^d$. The choice in the * operator belongs to Angel, but goes to Demon in a $\cdot^d$ context, while the choices in the α subgame underneath stay as they were originally thanks to the additional $\cdot^d$ operator that restores the game back to normal responsibilities. Again, $((\alpha^d)^*)^d$ corresponds to turning the chessboard around, thus giving the choice of repetition that would have been Angel's to Demon, yet turning the chessboard in α^d around again to play the original α.

The *dual differential equation* $\{x' = f(x)\,\&\,Q\}^d$ follows the same dynamics as $x' = f(x)\,\&\,Q$ except that, because of the duality operator, Demon now chooses the duration. He has to choose a duration during which Q holds all the time. Hence he loses when Q does not hold in the current state. Similarly, the *dual test* $?Q^d$ will make Demon lose the game immediately if the formula Q does not hold in the current state, just as the test $?Q$ will make Angel lose the game immediately if the formula Q does not hold currently. Dual assignment $(x := e)^d$ is equivalent to ordinary assignment $x := e$, because assignments never involve any choices to begin with, so it does not matter which player plays them.

Angel's control operators and Demon's control operators correspond to each other by duality:



Because the double dual $(\alpha^d)^d$ is the same as the game α, we never have to use the duality operator $\cdot^d$ except in Demon's choice $\cap$, Demon's repetition $^\times$, or around differential equations and tests. But it is more systematic to just allow $\cdot^d$ everywhere.

14.3 Syntax of Differential Game Logic

Differential game logic (dGL) is a logic for studying properties of hybrid games [9]. The idea is to describe the game form, i.e., rules, dynamics, and choices of the particular hybrid game of interest, using a program notation and to then study its properties by proving the validity of logical formulas that refer to the existence of winning strategies for objectives of those hybrid games. This is analogous to how a differential dynamic logic formula $[\alpha]P$ separately describes the dynamics of the hybrid system as a hybrid program α and the property of interest in the modality's postcondition P.

14.3.1 Hybrid Games

Even though hybrid game forms only describe the *form* of the game with its dynamics and rules and choices, not the actual objective, they are still simply called hybrid games just for simplicity of terminology. The objective for a hybrid game is defined in the postcondition of the modal logical formula that refers to that hybrid game form. During a hybrid game the players can only lose by violating the rules of the game, never win. The proper winning condition is specified in the dGL formula. Hybrid games (HGs in Definition 14.1) and differential game logic formulas (Definition 14.2) are defined subsequently.

Definition 14.1 (Hybrid games). The *hybrid games of differential game logic* dGL are defined by the following grammar (α, β are hybrid games, x is a vector of variables, $f(x)$ is a vector of (polynomial) terms of the same dimension, and Q is a dGL formula or just a formula of first-order real arithmetic):

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

The only syntactical difference of hybrid games compared to hybrid programs for hybrid systems as in Chap. 3 is that, unlike hybrid programs, hybrid games also allow the dual operator α^{d}. This minor syntactic change will require us to reinterpret the meaning of the other operators in a much more flexible way in order to make sense of the presence of subgames within the games in which the players already interact. The basic principle is that whenever there used to be nondeterminism in the hybrid program semantics, there will now be a choice that is up to Angel in the hybrid game semantics. But don't be fooled! The parts of such a hybrid game may still be hybrid games, in which players interact, rather than just a single system running. So *all* operators of hybrid games still need to be carefully understood as games, not just the *duality operator* $\cdot^{\mathrm{d}}$, because all operators can be applied to subgames that mention $\cdot^{\mathrm{d}}$ or be part of a context that mentions a $\cdot^{\mathrm{d}}$ duality.

The *atomic games* of dGL are assignments, continuous evolutions, and tests. In the *deterministic assignment game* (or discrete assignment game) $x := e$, the value of variable x changes instantly and deterministically to that of e by a discrete jump without any choices to resolve, just as it already was the case for the HP $x := e$. In the *continuous evolution game* (or continuous game) $x' = f(x) \,\&\, Q$, the system follows the differential equation $x' = f(x)$ where the duration is Angel's choice. But Angel is not allowed to choose a duration that would, at any time, take the state outside the region where *evolution domain constraint* formula Q holds. In particular, Angel is deadlocked and loses immediately if Q does not hold in the current state, because she cannot even evolve for duration 0 then without being outside Q.[2] The

[2] The most common case for Q is a formula of first-order real arithmetic, but any dGL formula will work (Definition 14.2). Evolution domain constraints turn out to be unnecessary, because they can be defined using hybrid games. In the ordinary differential equation $x' = f(x)$, the term x' denotes the time-derivative of x and $f(x)$ is a polynomial term that may mention x and other variables. More general forms of differential equations are possible [2, 3, 12], but will not be considered explicitly.

test game or *challenge* $?Q$ has no effect on the state, except that Angel loses the game immediately if dGL formula Q does not hold in the current state, because she failed the test she was supposed to pass. The test game $?Q$ challenges Angel and she loses immediately if she fails. Angel does not win just because she passes the challenge $?Q$, but at least the game continues. So passing challenges is a necessary condition to win games. Failing challenges, instead, immediately makes Angel lose. That makes tests $?Q$ in hybrid games the direct game counterpart of tests $?Q$ in hybrid programs. In order to properly track who won, we just need to get used to the notion of losing a game, instead of just aborting and discarding an HP execution.

The *compound games* of dGL are sequential, choice, repetition, and duals. The *sequential game* $\alpha;\beta$ is the hybrid game that first plays hybrid game α and then, when hybrid game α terminates without a player having lost already (so no challenge in α failed), continues by playing game β. When playing the *choice game* $\alpha\cup\beta$, Angel chooses whether to play hybrid game α or play hybrid game β. Like all the other choices, this choice is dynamic, i.e., every time $\alpha\cup\beta$ is played, Angel gets to choose again whether she wants to play α or β this time. She is not bound by whatever Angel chose last time. The *repeated game* α^* plays hybrid game α repeatedly and Angel chooses, after each play of α that terminates without a player having lost already, whether to play the game again or not, although she cannot choose to play indefinitely but has to stop repeating ultimately. Angel is allowed to stop α^* right away after zero iterations of α. Most importantly, the *dual game* α^d is the same as playing the hybrid game α with the rôles of the players swapped. That is Demon decides all choices in α^d that Angel has in α, and Angel decides all choices in α^d that Demon has in α. Players who are supposed to move but deadlock lose. Thus, while the test game $?Q$ causes Angel to lose if formula Q does not hold, the *dual test game* (or *dual challenge*) $(?Q)^d$ instead causes Demon to lose if Q does not hold.

For example, if α describes the game of chess, then α^d is chess where the players switch sides. If α, instead, describes a hybrid game where you are controlling your robot and a test engineer controls the roguebot, then α^d describes the dual game where you take control of the roguebot and the test engineer is stuck with your robot controls. If your test engineer is out for lunch, you can also play both robots. You just have to remember to play them both faithfully according to their objectives and can't cheat to make the test engineer's roguebot run away in terror just because that would make the job of your own robot easier. The real world isn't likely to make robot control so easy for you later. In fact, this pretend-play is another good way of understanding the intuition behind the duality operator. When playing α^d, you pretend to play for the other player in game α, respecting his objectives.

The dual operator $\cdot^d$ is the only syntactic difference of hybrid games compared to hybrid systems [1, 8], but a fundamental one [9], because it is the only operator where control passes from Angel to Demon or back. Without $\cdot^d$ all choices are resolved uniformly by Angel without interaction. The presence of $\cdot^d$ requires a thorough semantic generalization throughout the logic to cope with such flexibility.

Example 14.1 (Push-around cart). Consider a cart at position x moving along a straight line with velocity v that both Angel and Demon are pushing around simultaneously. Depending on whether they push or pull the cart, the two players will

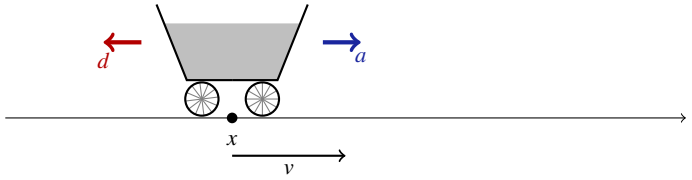


Fig. 14.2 Angel and Demon accelerating or braking by a and d, respectively, the cart at position x, which is moving with velocity x

each exert either an accelerating force or a braking force on x (Fig. 14.2):

$$\big((a:=1 \cup a:=-1);\, (d:=1 \cup d:=-1)^{\mathrm{d}};\, \{x' = v, v' = a+d\}\big)^* \tag{14.1}$$

First Angel chooses (by $\cup$) a positive or negative acceleration a, then Demon chooses a positive or negative acceleration d. This choice is Demon's because the choice $\cup$ occurs within the scope of a duality operator $\cdot^{\mathrm{d}}$, so what used to be Angel's choice becomes Demon's choice. Recall that it does not matter who controls the assignments, because they come without a choice. Finally, the game follows the differential equation system $x' = v, v' = a+d$ in which the sum of both accelerations a and d chosen by Angel and Demon, respectively, take effect, because the sum of all forces acts as acceleration on the cart x with unit mass. The cart is a point x, so Demon can't cheat and use his force to make it fall over. The duration of the differential equation is Angel's choice. Finally the game repeats $(^*)$, as often as Angel wants, because Demon is bored and walks away from the cart if Angel ever decides to stop playing. Each round of this repetition, Angel does not know which choice Demon will use for d, because she chooses a first before the sequential composition (;). This is unlike the following hybrid game where Demon chooses first:

$$\big((d:=1 \cup d:=-1)^{\mathrm{d}};\, (a:=1 \cup a:=-1);\, \{x' = v, v' = a+d\}\big)^* \tag{14.2}$$

But Angel controls the duration of the differential equation in both (14.1) and (14.2), so she can still choose duration 0 if she does not like Demon's choice of d. He just might choose the same inopportune value for d during the next repetition, so Angel will ultimately have to accept some decision by Demon and evolve for a positive duration or else the cart will never move anywhere, which would be permitted but incredibly boring for everyone. Whichever player decides on the acceleration last, so Demon in (14.1) and Angel in (14.2), can decide to keep the velocity unchanged by playing the opposite acceleration value such that $a+d=0$.

Which choices and decisions are particularly clever ones for Angel and Demon is a separate question and depends on the objective of the hybrid game, which is what dGL formulas will be used for. Hybrid systems lack the ability to express that the choice of a in (14.1) and (14.2) is Angel's while the choice of d is Demon's. A hybrid system could also have choices (without $\cdot^{\mathrm{d}}$ duals):

$$\big((d:=1 \cup d:=-1);\, (a:=1 \cup a:=-1);\, \{x' = v, v' = a+d\}\big)^*$$

But then all choices are nondeterministic, so resolved by the same player and either all help (if in a diamond modality) or all hurt (if in a box modality). In hybrid game (14.2), however, the choice of the acceleration d helps Demon while the choice of the acceleration a helps Angel, as do the choice of the duration of the differential equation and the number of repetitions.

Demon's controls such as $\alpha \cap \beta$ and $\alpha^\times$ can be defined with the help of the duality operator $\cdot^\mathrm{d}$ as in Sect. 14.2.3. In $\alpha^\times$, Demon chooses after each play of α whether to repeat the game, but cannot play indefinitely so he has to stop repeating ultimately. By duality, this follows from the fact that, in α^*, Angel also chooses after each play of α whether to repeat the game but she cannot play indefinitely.

Example 14.2 (Push-around cart). Demon's control operators rephrase (14.1) as

$$\big((a := 1 \cup a := -1); (d := 1 \cap d := -1); \{x' = v, v' = a + d\}\big)^*$$

Strictly speaking, $d := 1 \cap d := -1$ is $((d := 1)^\mathrm{d} \cup (d := -1)^\mathrm{d})^\mathrm{d}$. But that is equivalent to $(d := 1 \cup d := -1)^\mathrm{d}$, because deterministic assignments $x := e$ are equivalent to dual assignments $(x := e)^\mathrm{d}$, since both involve no choice. Similarly, (14.2) is

$$\big((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\big)^*$$

These were a lot of games but not a lot of purpose yet, which is where the dGL formulas come in. We consider them next.

14.3.2 Differential Game Logic Formulas

Hybrid games describe how the world can unfold when Angel and Demon interact according to their respective control choices. They explain the rules of the game, how Angel and Demon interact, and what the players can choose to do, but not who wins the game, nor what the respective objectives of the players are.[3] The actual winning conditions are specified by logical formulas of differential game logic.

We cannot continue the same understanding of modalities from Part I and Part II of this book, where the dL formula $[\alpha]P$ says that all runs of HP α satisfy P while the dL formula $\langle\alpha\rangle P$ says that at least one run of HP α satisfies P. It is not very meaningful to talk about all runs or some run of a hybrid game, because the whole point of games is that they provide a number of choices to the different players that may unfold differently in response to one another. Since the players have objectives, only some of those choices will manifest and be in their interest. What the players choose to do depends on what their opponent did before and vice versa. It is not particularly interesting if a player can lose a game by playing entirely stupidly. What is much more exciting is the question of whether the player can win if she plays in a clever way. And it is maximally compelling if a player even has a consistent way

[3] Except that players lose if they disobey the rules of the game by failing their respective challenges.

of always winning the game, no matter what the opponent is trying. Then the player has a *winning strategy*, i.e., a way to resolve her actions that will always win the game for all strategies that her opponent might try. This makes game play quite interactive; one has to find some choice for the player and consider all options for the opponent.

Modal formulas $\langle\alpha\rangle P$ and $[\alpha]P$ refer to hybrid games and the existence of winning strategies for Angel and Demon, respectively, in a hybrid game α with a winning condition specified by a logical formula P.

Definition 14.2 (dGL formulas). The *formulas of differential game logic* dGL are defined by the following grammar (P, Q are dGL formulas, $e, \tilde{e}$ are terms, x is a variable, and α is a hybrid game):

$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid \exists x P \mid \langle\alpha\rangle P \mid [\alpha]P$$

Other operators $>, =, \leq, <, \vee, \rightarrow, \leftrightarrow, \forall x$ can be defined, e.g., $\forall x P \equiv \neg\exists x \neg P$.

The modal formula $\langle\alpha\rangle P$ expresses that Angel[4] has a winning strategy to achieve P in hybrid game α, i.e., Angel has a strategy to reach any of the states satisfying dGL formula P when playing hybrid game α, no matter what strategy Demon chooses. The modal formula $[\alpha]P$ expresses that Demon has a winning strategy to achieve objective P in hybrid game α, i.e., a strategy to reach any of the states satisfying P, no matter what strategy Angel chooses. The same game is played in $[\alpha]P$ as in $\langle\alpha\rangle P$ with the same choices resolved by the same players. The difference between the two dGL formulas is the player whose winning strategy they refer to. Both use the set of states where dGL formula P is true as the set of winning states for that player. The winning condition is defined by the modal formula; α only defines the hybrid game form, not when the game is won, which is what P does. Hybrid game α defines the rules of the game, including conditions on state variables that, if violated, cause the present player to lose for violation of the rules of the game. The dGL formulas $\langle\alpha\rangle P$ and $[\alpha]\neg P$ consider complementary winning conditions for Angel and Demon. Of course, the propositional logical connectives $\neg, \wedge, \vee, \rightarrow$ still mean what they always do and the quantifiers $\exists x P$ and $\forall x P$ quantify over the reals.

14.3.3 Examples

This section discusses some examples of hybrid games and states differential game logic formulas expressing properties of winning strategies for these games.

Example 14.3 (Push-around cart). Continuing Example 14.1, consider a dGL formula for the cart-pushing hybrid game from (14.2):

[4] It is easy to remember which modal operator is which. The formula $\langle\alpha\rangle P$ clearly refers to Angel's winning strategies because the diamond operator $\langle\cdot\rangle$ has wings. This is consistent with the fact that Angel takes charge of what nondeterminism used to do in dL, so $\langle\alpha\rangle P$ is where Angel's control $\cup, ^*, x' = f(x)$ helps, just as nondeterminism helped in the diamond modality of dL.

$$v \geq 1 \rightarrow \left[\left((d := 1 \cup d := -1)^{\mathrm{d}}; (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\right)^*\right] v \geq 0$$

This dGL formula expresses that Demon has a winning strategy to ensure that the cart's velocity v is nonnegative if it initially started at $v \geq 1$. That would have been trivial if we had considered hybrid game (14.1), in which Demon chooses d after Angel chose a such that the choice $d := -a$ would trivially ensure $v' = 0$. But Demon's choice $d := 1$ still makes sure that the velocity will never decrease, whether Angel subsequently chooses to also push ($a := 1$) or to slow the cart down ($a := -1$). For the same reason, Demon also has a winning strategy to achieve $x \geq 0$ if the cart initially starts with $v \geq 0$ at $x \geq 0$. That is, the following formula is valid:

$$x \geq 0 \wedge v \geq 0 \rightarrow \left[\left((d := 1 \cup d := -1)^{\mathrm{d}}; (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\right)^*\right] x \geq 0$$

When replacing the box modality by a diamond modality, the formula

$$x \geq 0 \rightarrow \left\langle\left((d := 1 \cup d := -1)^{\mathrm{d}}; (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\right)^*\right\rangle x \geq 0$$

expresses that Angel also has a winning strategy to achieve $x \geq 0$ in the same hybrid game starting from just the initial condition $x \geq 0$. But even if that dGL is valid as well, it is trivially valid, because Angel controls repetition (*) and can simply decide on 0 iterations which makes the game stay in the initial state where $x \geq 0$ already holds. The same would happen if Demon were to control the repetition with Demon's repetition $^\times$ instead of Angel's repetition * as long as Angel still controls the differential equation, because she can simply go for duration 0 every time:

$$x \geq 0 \rightarrow \left\langle\left((d := 1 \cup d := -1)^{\mathrm{d}}; (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\right)^\times\right\rangle x \geq 0$$

Without that assumption $x \geq 0$ on the initial state, however,

$$\left\langle\left((d := 1 \cup d := -1)^{\mathrm{d}}; (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\right)^*\right\rangle x \geq 0$$

is not valid, because, unless v is already nonnegative initially, Demon can always play $d := -1$, which will make it impossible for Angel to give it a positive velocity $a + d$. If Angel is stronger than Demon, the corresponding dGL formula is valid:

$$\left\langle\left((d := 1 \cup d := -1)^{\mathrm{d}}; (a := 2 \cup a := -2); \{x' = v, v' = a + d\}\right)^*\right\rangle x \geq 0$$

All that Angel needs to do to achieve $x \geq 0$ is to push really hard with $a := 2$ and continuously evolve for long enough. More subtly, even if Demon has the same strength, Angel, nevertheless, has a winning strategy to achieve $x^2 \geq 100$:

$$\left\langle\left((d := 2 \cup d := -2)^{\mathrm{d}}; (a := 2 \cup a := -2); \right.\right.$$
$$\left.\left. t := 0; \{x' = v, v' = a + d, t' = 1 \,\&\, t \leq 1\}\right)^*\right\rangle x^2 \geq 100 \tag{14.3}$$

Angel has no influence on Demon's decision on d. But all it takes for Angel is to play $a := 2$ if $v > 0$ and play $a := -2$ if $v < 0$ to ensure that the sign of v never changes, whatever Demon plays, and, thus, x will eventually either grow above 10

or shrink below -10. If $v = 0$ initially, then Angel first plays $a := d$ to mimic Demon in the first round and make v nonzero, which she can since Demon decides first and Angel controls the differential equation's duration. Hence, (14.3) is valid, too. If (14.3) did not have a time bound on the duration of the evolution, it would be more obviously valid without repeating, because Angel could just mimic Demon once and then follow the differential equation for a long time. The differential equation is Angel's choice, but she has an evolution domain constraint $t \leq 1$ to worry about. Since clock $t' = 1$ was reset to $t := 0$ before, she cannot follow the differential equation for more than 1 time unit without losing for violation of the rules of the game. Thus, both players get to change their control variables at least once a second but Angel controls when exactly. Every time they get to change control variables d and a, Demon chooses first (before the sequential composition).

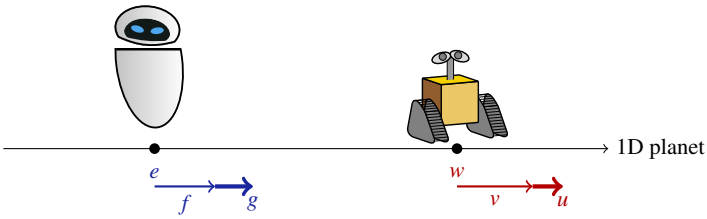


Fig. 14.3 Velocities v, f and accelerations u, g of two robots at w and e on a one-dimensional planet

*Example 14.4 (*WALL·E *and* EVE *robot dance).* Consider a game of the two robots WALL·E and EVE moving on a rather flat one-dimensional planet (Fig. 14.3):

$$\begin{aligned}(w-e)^2 \leq 1 \wedge v = f \rightarrow \big\langle \big((u := 1 \cap u := -1);\\ (g := 1 \cup g := -1);\\ t := 0; \{w' = v, v' = u, e' = f, f' = g, t' = 1 \,\&\, t \leq 1\}^{\mathrm{d}}\big)^{\times}\\ \big\rangle\, (w-e)^2 \leq 1\end{aligned} \tag{14.4}$$

Despite the dimensionally somewhat impoverished planet, this dGL formula provides a canonical use case for a hybrid game. Robot WALL·E is at position w with velocity v and acceleration u and plays the part of Demon. Robot EVE is at position e with velocity f and acceleration g and plays the part of Angel.

The antecedent of (14.4) before the implication assumes that WALL·E and EVE start close to one another (distance at most 1) and with identical velocities. The objective of EVE, who plays Angel's part in (14.4), is to be close to WALL·E (i.e., $(w-e)^2 \leq 1$) as specified after the $\langle\cdot\rangle$ modality in the succedent. The hybrid game proceeds as follows. Demon WALL·E controls how often the hybrid game repeats by operator $^{\times}$. In each iteration, Demon WALL·E first chooses (with Demon's choice operator $\cap$) to accelerate ($u := 1$) or brake ($u := -1$), then Angel EVE chooses (with

Angel's choice operator $\cup$) whether to accelerate ($g := 1$) or brake ($g := -1$). Every time that the $^\times$ loop repeats, the players get to make that choice again. They are not bound by what they chose in the previous iterations. Yet, depending on the previous choices, the state will have evolved differently, which influences indirectly what moves a player needs to choose to win. After this sequence of choices of u and g by Demon and Angel, respectively, a clock variable t is reset to $t := 0$. Then the hybrid game follows a differential equation system such that the time-derivative of WALL·E's position w is his velocity v and the time-derivative of v is his acceleration u; simultaneously, the time-derivative of EVE's position e is her velocity f and the time-derivative of f is her acceleration g. The time-derivative of clock variable t is 1, yet the differential equation is restricted to the evolution domain $t \leq 1$ so it can at most be followed for 1 time unit. Angel controls the duration of differential equations. Yet, this differential equation is within a dual game due to the operator $\cdot^d$ around it, so Demon actually controls the duration of the continuous evolution. Here, both WALL·E and EVE evolve continuously but Demon WALL·E decides how long. He cannot choose durations > 1, because that would make him violate the evolution domain constraint $t \leq 1$ and lose. So both players can change their control after at most one time unit, but Demon decides when exactly. Similar games can be studied for robot motion in higher dimensions using dGL.

The dGL formula (14.4) is valid, because Angel EVE has a winning strategy to get close to WALL·E by mimicking Demon's choices. Recall that Demon WALL·E controls the repetition $^\times$, so the fact that the hybrid game starts EVE off close to WALL·E is not sufficient for EVE to win the game. Mimicking by $g := u$ will also only work so easily because both start with the same initial velocity $v = f$. The hybrid game in (14.4) would be trivial if Angel were to control the repetition (because she would then win just by choosing not to repeat) or if Angel were to control the differential equation (because she would then win by always just evolving for duration 0). Hybrid games are most interesting when the choices are not already stacked in one player's favor. The analysis of (14.4) is more difficult if the first two lines in the hybrid game are swapped so that Angel EVE chooses g before Demon WALL·E chooses u, because she cannot play the copy strategy if Angel has to choose first.

Example 14.1 had a single differential equation system in which the controls of Angel and Demon mix via $x'' = a + d$, while Example 14.4 had a bigger differential equation system consisting of differential equations $w' = v, v' = u$ that belong to WALL·E and other differential equations $e' = f, f' = g$ that belong to EVE, which are joined together with time $t' = 1$. Both players evolve their respective variables together. The question of whether the resulting combined differential equation system is under Angel's or Demon's control is separate, and just depends on who gets to decide on the duration. This is in direct analogy to a loop body in which multiple operations by Angel and Demon might occur but still one of the two players needs to be responsible for deciding how often to repeat the loop itself, because the players might never come to an agreement if both were in charge of the same operator.

*Example 14.5 (*WALL·E *and* EVE *and the world).* The game in (14.4) accurately reflects the situation when WALL·E, who plays the part of Demon, is in control of

time since the differential equation occurs within an odd number of $\cdot^{\mathrm{d}}$ operators. But this is not the only circumstance in which (14.4) is the right game to look at for EVE. Suppose there really is a third player, the external environment, which controls time. So, neither WALL·E nor EVE really gets to decide on how long differential equations are followed, nor on how often the loop repeats.

EVE can use a common modeling device to conservatively attribute the control of the differential equation to WALL·E, even if time is really under the control of a third player, the external environment. EVE's reason for this model would be that she is certainly not in control of time, so there is no reason to believe that time would help her. EVE, thus, conservatively cedes control of time to Demon, which corresponds to assuming that the third player of the external environment is allowed to collaborate with WALL·E to form an aggregate Demon player consisting of WALL·E and the environment. If, as the validity of the resulting formula (14.4) indicates, Angel EVE wins against the Demon team consisting of WALL·E and the world, then she wins no matter what WALL·E and the external world decide to do.

This answers what hybrid game EVE needs to analyze to find out when she has a winning strategy for all actions of WALL·E and the world. When WALL·E wants to analyze his winning strategies he cannot just use the $[\cdot]$ modality for the same hybrid game as in (14.4) anymore, because that hybrid game was formed by conservatively attributing the external world's control of time to Demon. However, Demon WALL·E can use the same modeling device to flip the world's differential equation control over to Angel's control by removing the $\cdot^{\mathrm{d}}$ to conservatively associate the environment with his opponent (and negate the postcondition to consider the opposite goal):

$$\begin{aligned}(w-e)^2 \leq 1 \wedge v = f \rightarrow \big[\big((u:=1 \cap u:=-1);\\ (g:=1 \cup g:=-1);\\ t:=0;\{w'=v,v'=u,e'=f,f'=g,t'=1\,\&\,t\leq 1\}\big)^{\times}\\ \big]\,(w-e)^2 > 1\end{aligned} \tag{14.5}$$

Observe how a three-player game of WALL·E, EVE, and the environment can be analyzed by combining the dGL formulas (14.4) and (14.5) propositionally, which then analyze the same game from different perspectives of different possible collaborations. The dGL formula expressing that neither (14.4) nor (14.5) is true, is true in exactly the states where WALL·E and EVE draw, because the external environment can choose the winner by helping either WALL·E or EVE. Here, (14.5) is unsatisfiable, because Demon needs to move first, so Angel can always mimic him to stay close.

The WALL·E and EVE examples were *games for analytic purposes*. WALL·E and EVE are not actually in adversarial competition with opposing objectives. They just did not know each other any better yet when they first met. And they still suffer some amount of uncertainty about each other's decisions, which can lead to a game situation for lack of better knowledge. The next example is one of *true adversarial competition* where the two players seriously complete.

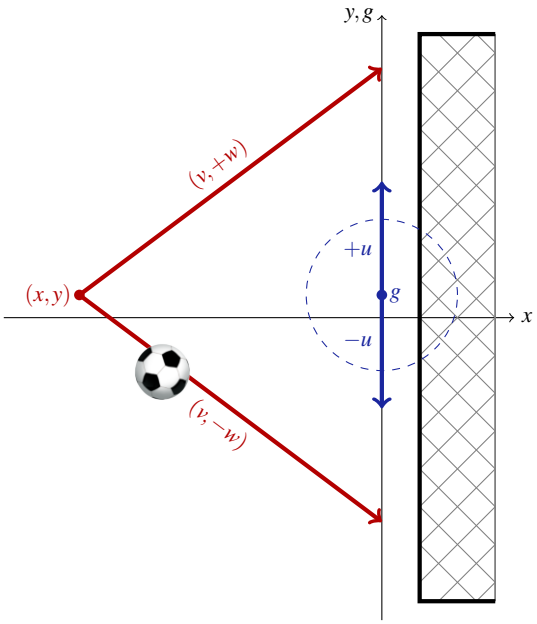


Fig. 14.4 Goalie g in robot soccer moves with velocity $\pm u$ up or down and, if within radius 1, can capture the ball (x,y) moving with velocity $(v,\pm w)$ sloped up or downards.

Example 14.6 (Goalie in robot soccer). Consider two robots engaged in a robot soccer match. Demon's robot is in possession of the ball and has a free kick toward the goal. Angel's robot is a goalie at position g who is trying hard to prevent Demon's robot from scoring a goal (Fig. 14.4). The ball is at position (x,y). Demon can either kick to roll the ball with vectorial velocity (v,w) into the left corner of the goal or with velocity $(v,-w)$ into its right corner. Angel's goalie robot can repeatedly move up or down near the goal line with linear velocity u or $-u$. She will capture the ball if the ball (v,w) is within radius 1 of the goalie $(0,g)$, i.e., if $x^2+(y-g)^2 \leq 1$. The two robots (and thus also the ball) are initially assumed to be at different x coordinates but the same y coordinate, with the ball being kicked toward the goal ($v>0$):

$$\begin{aligned} &x<0 \wedge v>0 \wedge y=g \rightarrow \\ &\quad \langle (w:=+w \cap w:=-w); \\ &\quad \big((u:=+u \cup u:=-u); \{x'=v, y'=w, g'=u\}\big)^* \rangle x^2+(y-g)^2 \leq 1 \end{aligned} \tag{14.6}$$

Demon's robot only has one control decision, in line 2, which is the direction in which he kicks the ball. Once the ball is rolling, there's no turning back. Angel subsequently has a series of control decisions, both how often to repeat the subsequent control loop but also whether to move the goalie up or down (in line 3) and how

long to follow the differential equation where the ball at position (x,y) rolls with velocity (v,w) in that direction and the goalie at position g moves with velocity u.

Whether dGL formula (14.6) is true depends on the relationship of the initial ball position x to the respective velocities v,w,u. The easiest case where it is true is $w=u$, in which case the vertical velocity w of the ball is identical to the goalie's velocity u, so that a mimic strategy $u:=w$ will make Angel win and capture the ball. More generally,

$$\left(\frac{x}{v}\right)^2 (u-w)^2 \leq 1 \tag{14.7}$$

implies that (14.6) is true. The time it takes for the ball to reach the goal line when moving with horizontal velocity v is $-x/v$. During that time the ball moves a distance of $-\frac{x}{v}w$ laterally in the y-direction, while the goalie moves a distance $-\frac{x}{v}u$. Since $y=g$ initially, the two positions will then be within capture distance 1 if (14.7) holds initially.

14.4 An Informal Operational Game Tree Semantics

Due to the subtleties and shift of perspective that hybrid games provide, the treatment of a proper semantics for differential game logic will be deferred to the next chapter. A graphical illustration of the choices that arise when playing hybrid games is depicted in Fig. 14.5. The nodes where Angel gets to decide are shown as diamonds $\diamond$, the nodes where Demon decides are shown as boxes $\diamond$. Circle nodes $\circ$ are shown when it depends on the remaining hybrid game which player gets to decide. Dashed edges - - - indicate Angel's actions to choose from, solid edges —— indicates Demon's actions, while zigzag edges ⌐¬ indicate that a hybrid game is played and the respective players move as specified by that game.

The actions are the choice of real duration for $x'=f(x)\&Q$, the choice of playing the left or the right subgame for a choice game $\alpha\cup\beta$, and, after each round of a repetition, the choice of whether to stop or repeat in a repeated game α^*. The sequential game $\alpha;\beta$ has no actions to decide, except that game β starts after game α is done. There are no particular actions for the duality operator $\cdot^d$, which, however, flips the rôles of the players by flipping box nodes $\boxdot$ that are under Demon's control with diamond nodes $\diamond$ that are under Angel's control. Mnemonically, $\cdot^d$ makes all nodes roll over by 45° so that boxes $\boxdot$ turn into diamonds $\diamond$ and diamonds $\diamond$ turn into boxes $\boxdot$. Assignments and tests also have no particularly interesting actions, except that Angel loses game $?Q$ unless Q is true in the current state.

The game tree action principles can be made rigorous in an operational game semantics [9], which conveys the intuition of interactive game play for hybrid games, relates to classical game theory and descriptive set theory, but is beyond the scope of this textbook, because Chap. 15 will investigate a significantly simpler denotational semantics. Observe how all choices involve at most two possibilities except differential equations, which have uncountably infinitely many choices, one option

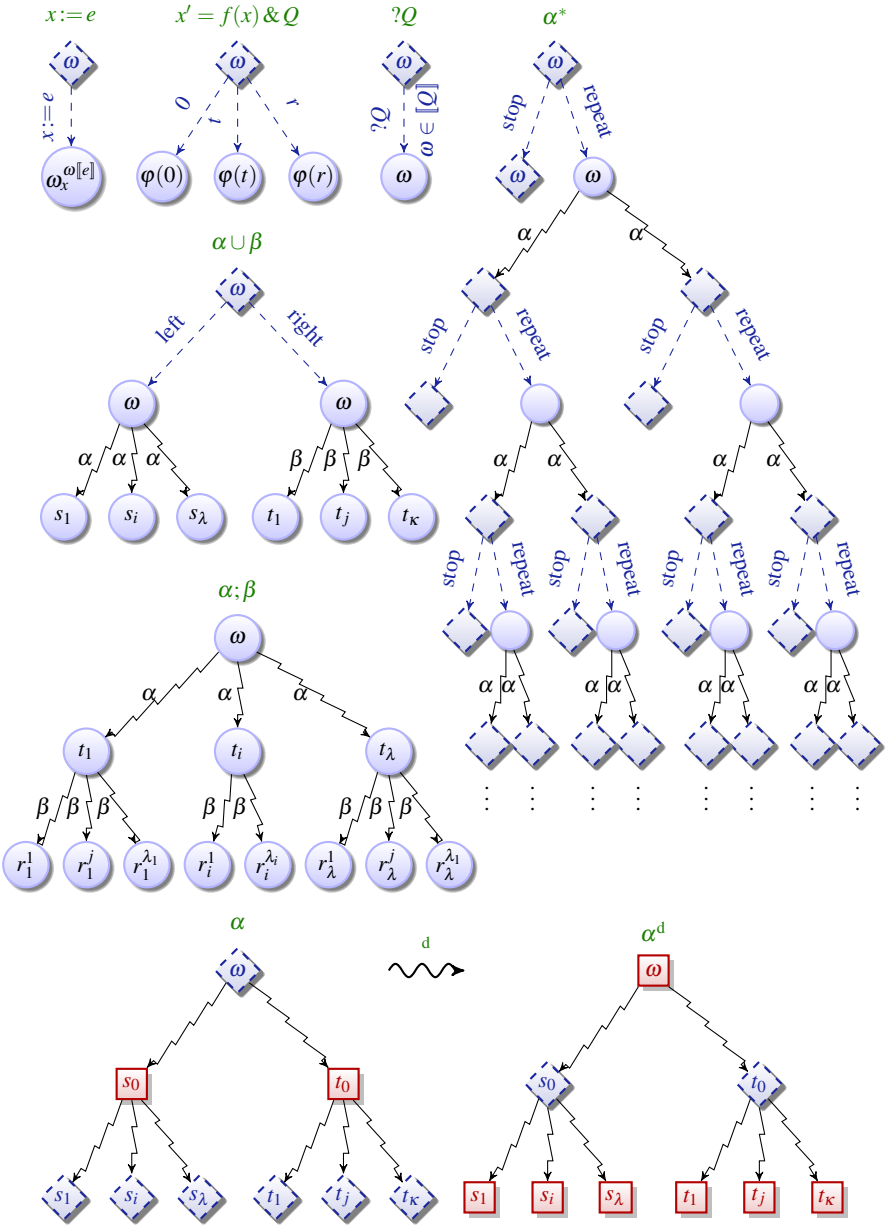


Fig. 14.5 Operational game semantics for hybrid games of dGL

for each nonnegative duration $r \in \mathbb{R}$. Of course, some of those durations may be a pretty bad idea if they would fail the evolution domain constraint, but that is up to the respective player to decide.

A *strategy* for a player in a hybrid game can be understood as a way of selecting a (state-dependent) action at each of the nodes of the game tree where that player has the choice, so an action at each diamond node for a strategy for Angel or an action at each box node for a strategy for Demon. A *winning strategy* is a strategy that leads to a winning state for all of the opponent's strategies.

As an example to illustrate some of the subtle nuances in defining an appropriate semantics for hybrid games, consider the discrete *filibuster formula*:

$$\langle (x := 0 \cap x := 1)^* \rangle x = 0 \tag{14.8}$$

It is Angel's choice whether to repeat (*), but every time Angel repeats, it is Demon's choice ($\cap$) whether to play $x := 0$ or $x := 1$. What is the truth-value of the dGL formula (14.8)?

The game in this formula never deadlocks, because each player always has at least one remaining move (here even two because Angel can stop or repeat and Demon can assign 0 or 1 to x). But it may appear that the game has perpetual checks, because no strategy helps either player win the game; see Fig. 14.6. Every time Angel chooses to repeat, hoping for an outcome of $x = 0$, Demon can stubbornly choose to play the right subgame $x := 1$ to make x one. That will not make Demon win either, because Angel is still in charge of deciding about repeating, which she will want to do to avoid the catastrophic outcome $x = 1$ that would make her lose. But next time around the loop, the situation is essentially unchanged, because Demon will still not want to give in and will, thus, cleverly play $x := 1$ again. How can that happen in this game and what can be done about it?

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

The mystery of the filibuster game can be solved when we remember that the game still ultimately has to stop in order that we may inspect who finally won the game. Angel is in charge of the * repetition and she can decide whether to stop or repeat. The filibuster game has no tests, so the winner only depends on the final state of the game, because both players are allowed to play arbitrarily without having to pass any tests in between. Angel wins a game play if $x = 0$ holds in the final state and Demon wins if $x \neq 0$ holds in the final state. What do the strategies indicated in Fig. 14.6 suggest? They postpone the end of the game, but if they did so indefinitely, there would never be a final state in which it could be evaluated who won. That is, indeed, not a way for anybody to win anything. Yet, Angel is in charge of the repetition *, so it is her responsibility to stop repeating eventually to evaluate who won. Consequently, the semantics of hybrid games allows the player in charge of a repetition to repeat as often as she wants, but she cannot repeat indefinitely. This will become apparent in the denotational semantics of hybrid games we will investigate in Chap. 15. Thus, (14.8) is *false* unless the winning condition $x = 0$ already holds initially, which allows Angel to just never repeat anything at all.

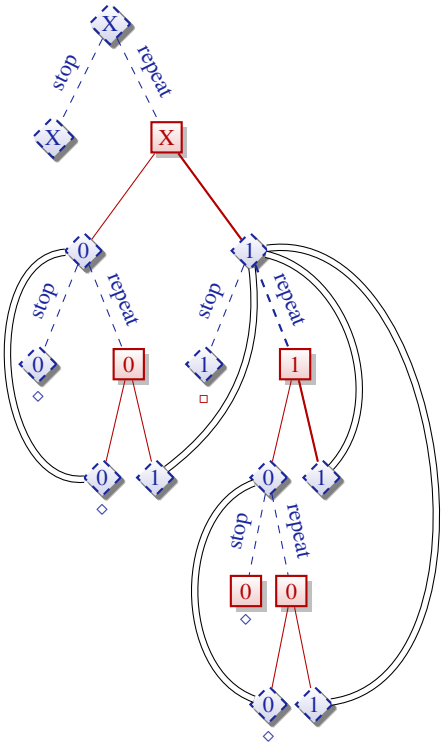


Fig. 14.6 The filibuster game formula $\langle(x:=0 \cap x:=1)^*\rangle x=0$ looks as though it might be nondetermined and not have a truth-value (unless $x=0$ initially) when the strategies follow the thick actions. Angel's action choices are illustrated by dashed edges from dashed diamonds, Demon's action choices by solid edges from solid squares, and double lines indicate identical states with the same continuous state and a subgame of the same structure of subsequent choices. States where Angel wins are marked by ◇ and states where Demon wins by □

The same phenomenon happens in the hybrid filibuster game:

$$\langle(x:=0;x'=1^{\mathrm{d}})^*\rangle x=0 \tag{14.9}$$

Both players can let the other one win. Demon can let Angel win by choosing to evolve his differential equation $x'=1^{\mathrm{d}}$ for duration 0. And Angel can let Demon win by choosing to stop the repetition even if $x\neq 0$. Only because Angel will ultimately have to stop repeating does the formula in (14.9) have a proper truth-value and the formula is *false* unless $x=0$ already holds initially.

It is of similar importance that the players cannot decide to follow a differential equation forever (duration ∞), because that would make this game nondetermined:

$$\langle(x'=1^{\mathrm{d}};x:=0)^*\rangle x=0 \tag{14.10}$$

If players were allowed to evolve along a differential equation forever (duration ∞), then Demon would have an incentive to evolve along $x' = 1^{\mathrm{d}}$ forever in the continuous filibuster (14.10). As soon as he stops the ODE, Angel can stop the loop and wins because of the subsequent assignment $x := 0$. But Angel cannot win without Demon stopping. Since Demon can evolve along $x' = 1^{\mathrm{d}}$ for *any finite real amount of time* he wants, he will ultimately have to stop so that Angel wins and (14.10) is valid.

Table 14.1 Operators and (informal) meaning in differential game logic (dGL)

dGL	Operator	Meaning
$e = \tilde{e}$	equals	true iff values of e and $\tilde{e}$ are equal
$e \geq \tilde{e}$	greater or equal	true iff value of e greater-or-equal to $\tilde{e}$
$\neg P$	negation / not	true iff P is false
$P \wedge Q$	conjunction / and	true iff both P and Q are true
$P \vee Q$	disjunction / or	true iff P is true or if Q is true
$P \rightarrow Q$	implication / implies	true iff P is false or Q is true
$P \leftrightarrow Q$	bi-implication / equiv.	true iff P and Q are both true or both false
$\forall x P$	universal quantifier	true iff P is true for all values of variable x
$\exists x P$	existential quantifier	true iff P is true for some value of variable x
$[\alpha]P$	$[\cdot]$ modality / box	true iff Demon has winning strategy to achieve P in HG α
$\langle\alpha\rangle P$	$\langle\cdot\rangle$ modality / diamond	true iff Angel has winning strategy to achieve P in HG α

14.5 Summary

This chapter saw the introduction of differential game logic, summarized in Table 14.1, which extends the familiar differential dynamic logic with capabilities for modeling and understanding hybrid games. Hybrid games combine discrete dynamics, continuous dynamics, and adversarial dynamics, summarized in Table 14.2. Compared to hybrid systems, the new dynamical aspect of adversarial dynamics is captured entirely by the duality operator $\cdot^{\mathrm{d}}$. Without it, hybrid games are single-player hybrid games, which are equivalent to hybrid systems. But the adversarial dynamics caused by the presence of the duality operator $\cdot^{\mathrm{d}}$ also made us reflect on the semantics of all other composition operators for hybrid games.

After this chapter showed an informal and intuitive discussion of the actions that hybrid games allow, the next chapter gives a rigorous semantics to differential game logic and its hybrid games.

Table 14.2 Statements and effects of hybrid games (HGs)

HG Notation	Operation	Effect
$x:=e$	assignment game	deterministically assigns value of e to variable x
$x'=f(x)\,\&\,Q$	continuous game	differential equation for x with term $f(x)$ within first-order constraint Q (evolution domain)
$?Q$	test / challenge	Angel loses unless formula Q holds at current state
$\alpha;\beta$	sequential game	HG β starts after HG α finishes
$\alpha\cup\beta$	choice game	Angel chooses between alternatives HG α or HG β
α^*	repeated game	Angel repeats HG α any finite number of times
α^d	dual game	swaps rôles of Angel and Demon in HG α

Exercises

14.1 (One-player games). Single-player hybrid games, i.e., ^{d}-free hybrid games, are just hybrid programs. For each of the following formulas, convince yourself that it has the same meaning whether you understand it as a differential dynamic logic formula with a hybrid system or as a differential game logic formula with a hybrid game (that happens to have only a single player):

$$\langle x:=0\cup x:=1\rangle x=0$$
$$[x:=0\cup x:=1]x=0$$
$$\langle (x:=0\cup x:=1);?x=1\rangle x=0$$
$$[(x:=0\cup x:=1);?x=1]x=0$$
$$\langle (x:=0\cup x:=1);?x=0\rangle x=0$$
$$[(x:=0\cup x:=1);?x=0]x=0$$
$$\langle (x:=0\cup x:=1)^*\rangle x=0$$
$$[(x:=0\cup x:=1)^*]x=0$$
$$\langle (x:=0\cup x:=x+1)^*\rangle x=0$$
$$[(x:=0\cup x:=x+1)^*]x=0$$

14.2 (Single-player push-around carts). Hybrid game (14.2) was a single-player formulation in which all choices go to player Angel and Demon has nothing to do. Is the following dGL formula about it valid?

$$v\geq 1\rightarrow\langle\big((d:=1\cup d:=-1);(a:=1\cup a:=-1);\{x'=v,v'=a+d\}\big)^*\rangle v\geq 0$$

Is the following dGL formula with a box modality valid, too?

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

What does this imply about the required cleverness for appropriate control choices in similar hybrid games that we considered in Example 14.3? Even if we have not even fully considered a semantics let alone a proof calculus for hybrid games yet,

can you still find a proof justifying the validity or a counterexample for the above two single-player hybrid game formulas?

14.3. In which states is the following dGL formula true and what is Demon's winning strategy in those states

$$[(((\{x' = 1\} \cup \{x' = -1\}); (\{y' = 1\}^{\mathrm{d}} \cap \{y' = -1\}^{\mathrm{d}}))^*] x < y$$

In which states is this variation true and what is Demon's winning strategy?

$$[(((\{x' = 1\} \cup \{x' = -1\}); (\{y' = 1\}^{\mathrm{d}} \cap \{y' = -1\}^{\mathrm{d}}))^*] (x-y)^2 < 5$$

These dGL formulas have disconnected physics, where the duration of evolution of Angel's differential equation may have nothing to do with the duration of evolution of Demon's differential equation. Most games synchronize in time, however. The following dGL formula has different control choices for the different players but the differential equations are combined into a single differential equation system under Angel's control of time. In which states is the following formula true and what is Demon's winning strategy?

$$[((v := 1 \cup v := -1); (w := 1 \cap w := -1)\{x' = v, y' = w\})^*] (x-y)^2 < 5$$

14.4. Consider the following dGL formulas and identify under which circumstance they are true:

$$\langle (x := x+1; \{x' = x^2\}^{\mathrm{d}} \cup x := x-1)^* \rangle (0 \leq x < 1)$$
$$\langle (x := x+1; \{x' = x^2\}^{\mathrm{d}} \cup (x := x-1 \cap x := x-2))^* \rangle (0 \leq x < 1)$$

14.5. Write down a valid formula that characterizes an interesting game between two robots and convince yourself whether it is valid or not.

14.6 (Robot simple chase game). The following dGL formula characterizes a one-dimensional game of chase between a robot at position x and another robot at position y, each with instant control of the velocity v among $a, -a, 0$ for x (Angel's choice) and velocity w among $b, -b, 0$ for y (Demon's subsequent choice). The game repeats any number of control rounds following Angel's choice (*). Angel is trying to get her robot x close to Demon's robot y. Under which circumstances is the formula true?

$$\langle (\ (v := a \cup v := -a \cup v := 0);$$
$$(w := b \cap w := -b \cap w := 0);$$
$$\{x' = v, y' = w\}\ )^* \rangle (x-y)^2 \leq 1$$

14.7 (Say when). For each of the following dGL formulas identify the set of states in which it is true and characterize this set by a formula of real arithmetic. For each case, briefly sketch the player's winning strategy when it is true and explain why the dGL formula is false in all other states:

$$\langle x := -1 \cup (x := 0 \cap x := y)\rangle x \geq 0$$
$$\langle \big(x := x+2 \cup (x := x-1; \{x' = -1\}^{\mathrm{d}})\big)^*\rangle 0 < x \leq 2$$
$$\langle x := x+2; x := x-1\rangle x \geq 0$$
$$\langle x := x-1 \cup (x := 0 \cap x := -y^2+1)\rangle x \geq 0$$
$$\langle x := y-1 \cup (\{x' = 1\}^{\mathrm{d}}; x := x+2)\rangle x \geq 0$$
$$\langle x := -y \cup (x' = 2; \{x' = -1\}^{\mathrm{d}}; x := x+2)\rangle x \geq 0$$
$$[(x := x \cap x' = -2)^*] x \geq 0$$
$$\langle (v := v \cap v := -v); (w := w \cup w := -w)\rangle v = w$$
$$\langle (v := v \cap v := -v); \{x' = v, y' = w\}\rangle x = y$$
$$\langle (v := v \cap v := -v); (w := w \cup w := -w); \{x' = v, y' = w\}\rangle x = y$$
$$\langle (x := x-1 \cap n := n-1; ?(n \geq 0)^{\mathrm{d}}; x := x^2)^*\rangle x < 0$$
$$[(x := -x \cap x' = -x^2)^*] x \geq 0$$
$$\langle \big(x := 0 \cup \big((x := x+1; \{x' = 1\}^{\mathrm{d}}) \cup x := x-1\big)\big)^*\rangle 0 < x \leq 1$$
$$\langle \big(x := x^2 \cup (x := x+1 \cap x' = 2)\big)^*\rangle x > 0$$
$$\langle \big((x := x+1; \{x' = x^2\}^{\mathrm{d}}) \cup (x := x-1; \{x' = -1\}^{\mathrm{d}})\big)^*\rangle 0 \leq x \leq 2$$
$$\langle \big((x := x+1; \{x' = -1\}^{\mathrm{d}}) \cup (x := x-1; \{x' = 1\}^{\mathrm{d}})\big)^*\rangle 0 \leq x \leq 2$$
$$\langle \big((x := x+1; \{x' = 1\}^{\mathrm{d}}) \cup (x := x-1; \{x' = -1\}^{\mathrm{d}})\big)^*\rangle 0 \leq x \leq 2$$

14.8 (* Robot chase). The following dGL formula characterizes a two-dimensional game of chase between a robot at position (x_1, x_2) facing in direction (d_1, d_2) and a robot at position (y_1, y_2) facing in direction (e_1, e_2). Angel has direct control over the angular velocity ω among $1, -1, 0$ for robot (x_1, x_2) and, subsequently, Demon has direct control over the angular velocity ρ among $1, -1, 0$ for robot (y_1, y_2). The game repeats any number of control rounds following Angel's choice $(^*)$. Angel is trying to get her robot close to Demon's robot. Is the following dGL formula valid? Can you identify some circumstances under which it is true? Or some circumstances under which it is false?

$$\Big\langle \Big((\omega := 1 \cup \omega := -1 \cup \omega := 0);$$
$$(\rho := 1 \cap \rho := -1 \cap \rho := 0);$$
$$\{x_1' = d_1, x_2' = d_2, d_1' = -\omega d_2, d_2' = \omega d_1, y_1' = e_1, y_2' = e_2, e_1' = -\rho e_2, e_2' = \rho e_1\}^{\mathrm{d}}$$
$$\Big)^* \Big\rangle\ (x_1 - y_1)^2 + (x_2 - y_2)^2 \leq 1$$

14.9 (Goalies with resistance). Robot soccer balls have the irritating tendency to slow down after they have been kicked. Extend Example 14.6 with a model that takes the slowdown due to roll resistance and/or air resistance into account. Can you identify a condition under which the resulting formula is true? On a straight line, a

point x of mass m with velocity v on flat terrain follows the differential equation

$$x' = v, v' = -av^2 - cgm$$

with gravity $g = 9.81\ldots$, a small roll resistance coefficient c and an even smaller aerodynamic coefficient a. What changes to say that no goal is scored?

References

[1] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.* **41**(2) (2008), 143–189. DOI: 10.1007/s10817-008-9103-8.

[2] André Platzer. Differential-algebraic dynamic logic for differential-algebraic programs. *J. Log. Comput.* **20**(1) (2010), 309–352. DOI: 10.1093/logcom/exn070.

[3] André Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Heidelberg: Springer, 2010. DOI: 10.1007/978-3-642-14509-4.

[4] André Platzer. Stochastic differential dynamic logic for stochastic hybrid programs. In: *CADE*. Ed. by Nikolaj Bjørner and Viorica Sofronie-Stokkermans. Vol. 6803. LNCS. Berlin: Springer, 2011, 446–460. DOI: 10.1007/978-3-642-22438-6_34.

[5] André Platzer. A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems. *Log. Meth. Comput. Sci.* **8**(4:17) (2012). Special issue for selected papers from CSL'10, 1–44. DOI: 10.2168/LMCS-8(4:17)2012.

[6] André Platzer. Dynamic logics of dynamical systems. *CoRR* **abs/1205.4788** (2012).

[7] André Platzer. Logics of dynamical systems. In: *LICS*. Los Alamitos: IEEE, 2012, 13–24. DOI: 10.1109/LICS.2012.13.

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

[9] André Platzer. Differential game logic. *ACM Trans. Comput. Log.* **17**(1) (2015), 1:1–1:51. DOI: 10.1145/2817824.

[10] André Platzer. Logic & proofs for cyber-physical systems. In: *IJCAR*. Ed. by Nicola Olivetti and Ashish Tiwari. Vol. 9706. LNCS. Berlin: Springer, 2016, 15–21. DOI: 10.1007/978-3-319-40229-1_3.

[11] 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.

[12] André Platzer. Differential hybrid games. *ACM Trans. Comput. Log.* **18**(3) (2017), 19:1–19:44. DOI: 10.1145/3091123.