15-424/15-624/15-824 Recitation 8
Game semantics

Important (Seriously, this is Actually the Best Explanation of Hybrid Game Semantics I've Found) Supplementary Material (Geri's Game, Pixar):`https://www.youtube.com/watch?v=9IYRC7g2ICg`

1 What Geri Teaches Us About Games

An important conceptual step to understanding games is understanding how the modalities $[\alpha]\phi$, $\langle\alpha\rangle\phi$ interact not only with each other, but how they interact with the new dual operator α^d and how they relate to their hybrid system equivalents.

To start with, it's essential to understand that the player Angel is always the current player, and Demon is always the opposite player. In Geri's game of chess, Angel is not the white player and Demon is not the black player. Angel is Geri and Demon is his imaginary friend. A mnemonic for this is that Angel is the Active player and Demon is the Dormant player.

But which one is box and which one is diamond? For this we can turn to a quote I misattributed:

> Diamond is easy, boxes are hard — "Clint Eastwood"

In hybrid systems, box proofs, intuitively, are "hard" because we have to make the postcondition true no matter what happened, while diamonds were "easy" because we got to pick our favorite, most cooperative execution of the program. This intuition works in games too. If we are the active player Angel, then we have it easy because we get to make all the choices. If we are the dormant player Demon, we have it hard because we have to put up with whatever Angel chooses to do, until maybe she runs a dual and gives us control. We can also remember this with the Red Bull Principle (courtesy André): Angel must be in the diamond modality, because diamond modalities give you wings.

Example: Catch-The-Bus Your TA lives in Point Breeze. Point Breeze is just far enough away that he can walk if he really has to (45 mins), but it's really a better idea to take the bus when he can (15 mins). However, he only has one bus route (71D) and, like all Pittsburgh buses, the 71D follows a truly random schedule. Sometimes they might come 5 minutes apart, sometimes it might be an hour. And sure you can track the bus on the PAT website, but that goes down all the time anyway.

Every day when he comes to campus, he plays a game against the bus. The TA's primary objective is to get to campus within some time bound T, with a secondary objective of getting to campus as fast as possible. As in every game we will consider, the bus' objective is the opposite of the TA's primary objective: prevent him from getting to campus within time T.

The rules of the game are as so:

- The TA starts at position $x_T = 0$ and the bus starts at position $x_B < 0$.
- If the bus has not reached the TA yet, it can accelerate or brake arbitrarily.
- Until the TA gets on the bus, he can always choose between walking ($v_T = v_W > 0$) or waiting ($v_T = 0$). (he can also do this after he boards the bus by getting off, but why would he do that?).
- If the bus stops next to the TA while he is waiting, he can board it.
- Once the TA has boarded the bus, it is guaranteed to be faster than walking. In a realistic model, the bus would accelerate up to its cruising speed. To simplify the math, we'll say it immediately jumps to $v_B = 3 \cdot v_W$, since the bus takes 15 minutes instead of 45.
- The TA wins if he gets to campus (position $x_C = 45 \cdot v_W$) within time T. The bus wins if the TA does not get to campus in time T.

We'll start by trying to model this game as a hybrid *program* instead of a hybrid *game*. Then we'll see where hybrid systems aren't enough and how hybrid games will help.

```
Pre_G ==
   x_T = 0 & x_B < 0 & v_W > 0 & t = 0 & T > 0 & x_C = 45*v_W

/* My control */
a_T ==
{   {?(xB = xT & vT = 0);  vT := vB;}
 ++ {vT := vW;}
 ++ {vT := 0;}}

/* Bus control*/
a_B ==
{   {?(xB != xT); aB :=*; ?(vB <= 0 -> aB >= 0); }
 ++ {?(xB = xT);  vB := vW * 3; aB := 0;}}

a_P ==
/* Physics */
{xB' = vB, vB' = aB, xT' = vT, t' = 1 & vB >= 0 & t <= T}
```

Given these components, we can define the overall hybrid program as

$$\alpha = (\alpha_T; \alpha_B; \alpha_P)^*$$

Strategizing and Preconditioning In any interesting game, we will only win some of the time, if we have the right starting conditions. The formula Pre_G already indicate some preconditions that are part of *the definition of the game.* If we want to prove that we win the game, we'll also want a precondition Pre_W that says "let's assume we're in one of the states where we can actually win, then we'll prove that we win in that state".

Let's think about how we would play this game and find out our winning strategy and under what circumstances we can win. The essential strategic observation is that the bus can do whatever it wants, and the bus might not help us at all. It could break down on the busway. It could chill across the street in the East Liberty Garage[1] for two hours. Heck, it could set its sign to say "East Liberty Garage" and then drive right past us even though it quite clearly *just left* the East Liberty Garage Thus, the winning strategy to achieve our *primary objective* of reaching campus in time T is to *always walk.* This means we can win whenever we'd get there in time T by walking. We know I can walk there in 45 minutes, so the winnable precondition is:

$$\text{Pre}_W \equiv T \geq 45$$

Before you get too bored by this strategy, remember we also have a secondary objective! That will make things more interesting, but one step at a time!

Victory Theorem, Attempt 1: Our theorem should say it's possible to get to campus on time. So far in the course, when we've been proving things about possibility (efficiency theorems), we phrase them in the following style:

$$\text{Pre} \rightarrow [\alpha](\text{Done} \rightarrow \text{Good})$$

Where Done gives some indication that the system is "finished" and Good says that some desirable state was reached when the system was finished. For example, if you're braking a 1-dimensional car like in Lab 1, your Done will say you've stopped moving and Good will say you reached the station by then. In the bus example, Done means $t = T$ and Good means $x_T \geq x_C$. So our initial attempt at a theorem is:

$$\text{Pre}_\text{G} \wedge \text{Pre}_\text{W} \rightarrow [\alpha](t = T \rightarrow x_T \geq x_C)$$

Victory Theorem, Attempt 2: But this theorem statement has a subtle flaw. We're saying that no matter what we do, any time we reach $t = T$ it will also be the case that $x_T \geq x_C$ (we have reached the objective). The flaw here, as with many modeling flaws we've

[1] which, incidentally, is actually in Larimer, not East Liberty

already seen, is one of *vacuity*: what if time $t = T$ never actually happens? If $t = T$ doesn't actually happen, then this theorem *says nothing.*

In a certain sense, we got lucky, because $t = T$ is a really simple property and in this case it's kind of obvious that if you let the physics happen for long enough, you'll always get to time $t = T$ eventually. But you can imagine lots of models where that wouldn't be the case. For a morbid example, a more realistic model might allow for the possibility that the 71D runs over the TA and the poor guy never makes it to time $t = T$. [2] Or for a more practical example, in the Lab 1 efficiency theorem, we never proved that we will *actually* ever stop moving. So if the controller is unsafe and keeps moving forever, then the efficiency theorem wouldn't say anything. That's not good!

Really what we want to say is (a) "Yes it is actually possible to get to time $t = T$" and also "when I do get to time $t = T$ it's possible to make it so I reach CMU." Recall that when we want to prove *possibility* about a program, we use the diamond modality instead of box:

$$\mathrm{Pre_G} \land \mathrm{Pre_W} \to \langle\alpha\rangle(t = T \land x_T \geq x_C)$$

Victory Theorem, Attempt 3: That's still not right, though. Remember that inside a diamond modality $\langle\alpha\rangle$ *we get to pick* what to do with all the nondeterministic choices. That means that I don't just get to pick whether I wait or walk, I also get to pick how fast the bus moves and how long the ODE runs (which determines whether I catch the bus). That's not fun at all! That would be a game with no competition! Who gets to make the decisions? I get to make the decision whether I walk or wait, but the bus gets to decide how fast it moves. It's up for debate who should get to control how long the ODE runs, but it will simplify the story a bit if I get to choose it. Really all that means is I get to pick how long I wait before making my next control choice, and that's pretty realistic: when I walk to campus I'm the one who chooses how often I look over my shoulder for the bus.

So what I want to say is "**I can pick** a control, where **no matter** what the bus does, **I can pick an ODE duration** where I win". Everything that we get to pick turns into a diamond. When the bus gets to pick, we have to consider every possible bus behavior, so those turn into boxes. This gives us the safety theorem:

$$\mathrm{Pre_G} \land \mathrm{Pre_W} \to \langle\alpha_T\rangle[\alpha_B]\langle\alpha_P\rangle(t = T \land x_T \geq x_C)$$

Does this work? Well, yes and no. This does correctly capture the idea that I get to make some decisions and the bus gets to make the others. It does capture the rules of the game. It's even true. So what's the problem? Uhh… where did our loop go? Let's try to "add it back":

$$\mathrm{Pre_G} \land \mathrm{Pre_W} \to (\langle\alpha_T\rangle[\alpha_B]\langle\alpha_P\rangle)^*(t = T \land x_T \geq x_C)$$

[2] To be fair, I have never felt especially endangered by the Port Authority's driving as a pedestrian. The same cannot be said for *any* other Pittsburgh driver.

Sweet mother of logic, that's not even a syntactically valid formula! We have now reached the point where hybrid programs break down and where we will need hybrid games:

Summary: The essence of gameplay is switching turns back and forth between two players, represented by the modalities $\langle\alpha\rangle$ (player "Angel" a.k.a. me a.k.a. Player 1) and $[\alpha]$ (player "Demon" a.k.a. the bus a.k.a. Player 2). Hybrid systems are perfectly adequate for expressing a *finite, constant* number of turn switches. In the example above, I get two turns and the bus gets one: we could repeat the programs as often as we want to get several turns. The problem is that limiting ourselves to a finite number of turns is a fundamental limitation which alters gameplay drastically: As one of many examples: If I know I only get one turn, the *only* possible winning strategy is to run the ODE for time $t = T$ or I won't get to campus. All the interesting strategy happens when there's a possibility that I'll *catch the bus on the way to campus*, e.g. by running the controller again every 5 minutes. Morever, we're interested in *general strategies* that solve *general problems*, which we will never get if we limit our games to constant size.

A Solution, the Dual Operator α^d: Let's not be too sad, though. It seems like we already know what we want to say. We want to say "run $\alpha_T; \alpha_B; \alpha_P$ in a loop, but every time we get to α_B run that in a box (i.e. let Demon pick). Here we can draw some inspiration from the Geri's Game clip from the beginning of recitation. What we want is an operator that says "take this program and run it as the other player". In d$\mathcal{GL}$, that operator is α^d, the dual to α. Operationally, α^d tells Geri to switch seats and run α while sitting in the other seat (and then come back to his original seat at the end). Equipped with the dual operator, we can now write down exactly the game we wanted:

$$\mathrm{Pre_G} \land \mathrm{Pre_W} \rightarrow \langle(\alpha_T; (\alpha_B)^d; \alpha_P)^*\rangle(t = T \land x_T \geq x_C)$$

Hooray, that's syntactically valid and even means what we wanted! The dual operator lets us *represent modalities as a program.* Neat!

Even more dual operators The dual operator α^d is actually the only new operator in d$\mathcal{GL}$, but we can use it to define dual versions of all the other operators, which can make models a lot easier to write down. If I write $(\alpha \cup \beta)^d$, that means that Demon doesn't just get to pick whether we run α or β, he also gets to make *all* the decisions that come up in α or β. That was actually what we wanted in the bus example — great! But very often we also want the ability to say Demon **just** gets to pick between α or β, while we choose the rest. This is exactly what the *dual operators* like $\alpha \cap \beta$ do. We give the dual version of each operator below, along with a definition showing us how to implement it with just α^d:

Angel	Demon	Demon Definition
$\alpha \cup \beta$	$\alpha \cap \beta$	$(\alpha^d \cup \beta^d)^d$
α^*	$\alpha^\times$	$\left((\alpha^d)^*\right)^d$
$\alpha; \beta$	$\alpha; \beta$	$(\alpha^d; \beta^d)^d$
$x := \theta$	$x := \theta$	$(x := \theta)^d$
$x' = \theta$	$(x' = \theta)^d$	$(x' = \theta)^d$
$?H$	$(?H)^d$	$(?H)^d$

For example, we could have rewritten the bus controller like this (though the other way was arguably easier):

```
/* Bus control*/
a_B ==
{   {?(xB != xT)^d; aB :=*^d; ?(vB <= 0 -> aB >= 0)^d;}
 -- {?(xB = xT)^d;  vB := vW * 3; aB := 0;}
}
```

For example, $\alpha \cup \beta$'s Angel is matched by $\alpha \cap \beta$ for Demon, and can be rewritten equivalently as $(\alpha^d \cup \beta^d)^d$. The dual operators on α and β guarantee that after $\cup$ is resolved by Demon, the game control returns to Angel. This is what we mean by $\alpha \cap \beta$ - only the non-deterministic choice is resolved by Demon, and α and β play "as usual".

Neither $\alpha; \beta$ nor $x := \theta$ have non-determinism. Duals are all about switching who gets to pick the non-determinism, so dual doesn't change anything here. We can add as many ^{d} as we want, it won't change!

Both α^* and $x' = \theta$ have non-deterministic choices so they get dual operators. Unfortunately, there's no special notation for the ODE... but don't feel sorry for it, because it's demonically evil!

The rule for tests $?(P)$ is that the current player loses if they can't pass the test.[3] To prove $\langle ?(P) \rangle Q$ we need to $P \wedge Q$ because we need to first pass the test, then prove Q. When we have a dual test $\langle ?(P)^d \rangle Q$ that means we tossed the test to Demon — it's his problem now! If he fails the test (blows up the game), the game ends early and we don't even look at Q, we just win. If Demon passes the test $?(P)$ it means Angel "passed" the dual test, but because we survived to the end of the game, *we, Angel,* now have to prove Q because the diamond modality $\langle\rangle$ said that the initial player was Angel. Note this means $\langle ?(P)^d \rangle Q \iff (\neg P \vee Q) \iff (P \to Q) \iff [?(P)]Q$ and so this matches up precisely with the meaning of α^d as "switch modalities/switch players".

[3]This is where the analogy with exams breaks down: if you give your opponent your test and they fail it, you will BOTH fail the course!

2 Filibusters

Games have some subtle edge cases that can be instructive to help us understand the semantics more deeply. For example, should the following *filibuster formula* be true?:

$$\langle (x := 0 \cap x := 1)^* \rangle x = 1$$

First, how will this game play out? Demon gets the choice inside the loop, so he can always pick $x := 0$, which makes Angel's life hard. Angel can respond by repeating the loop, but there's no guarantee that this will help her if Demon keeps picking $x := 0$. So who should "win"? Let's consider a few properties we want hybrid games to have:

- All game plays are finite. Despite their immortal sounding names, Angel and Demon can't spend infinite time playing, and so for example you can't run the loop infinitely many times (running it a truly huge, non-constant but still finite number of times is fine). You might imagine that if the loop ran "infinitely" we could call Angel the winner, but infinite loops aren't what we want.
- We want games to be *consistent* and *determined*, meaning there is exactly one winner for each game, $[\alpha]\phi \leftrightarrow \neg\langle\alpha\rangle\neg\phi$. Not only is this a useful axiom next week when we do proofs, but it is also really key both to relating hybrid games with hybrid systems and relating the two players to each other. So even in awkward cases like the filibuster, we need to assign a winner.
- The "current player" Angel not only has the *power* to make all the non-deterministic choices, she also has *responsibilities*.

  With great power comes great responsibility — Uncle Ben; André Platzer

  The first example we saw was that if a challenge $?(\phi)$ shows up, it's her responsibility to make ϕ true. If she can't, she loses. Thus it's useful to think of "please stop running this loop after finite steps!" to be another responsibility for Angel. If she can't figure out a *finite* strategy for the loop, that's on her, so she loses.

But Angel gets to decide to repeat! But you can only win a game if it finishes, so Angel is actually not allowed to repeat to infinity. When that happens, Demon has won.

So that formula was pretty false. What happens if we flip the modality from diamond to box?

$$[(x := 0 \cap x := 1)^*]x = 1$$

In general, we don't know whether $[\alpha]\phi$ is true given just whether $\langle\alpha\rangle\phi$ is true. It might be that Demon knows how to make ϕ true but Angel doesn't, or it could be that neither of them have the ability to make it true. In this case, we don't know anything about the starting state, and Angel can choose to run the loop 0 times, so actually Angel wins in that case. But if we add the precondition $x = 1$:

$$[x = 1 \rightarrow (x := 0 \cap x := 1)^*]x = 1$$

Then it doesn't matter how long Angel runs the loop, Demon can always set $x = 1$. And since Demon has a winning strategy in every state, the formula is valid.

What if we give Demon control of an ODE? Do we get a "continuous filibuster"?

$$\left\langle \left((x' = -1)^d ; x := 0 \right)^* \right\rangle x \geq 0$$

Demon can choose to evolve the ODE for a *really long time* since angels and demons are immortal, he can *really* bore Angel... but eventually he has to stop evolving. Just like loops and just like in hybrid systems, ODEs still have to run for finite time. Once the ODE finishes, then x gets assigned to 0, and Angel can decide whether to repeat. She says "Aw *hell*, no!", chuckling at the pun, and then proceeds to promptly win the game.

Semantics Recall that in $\mathsf{d}\mathcal{L}$ we had two very equivalent ways to write formula semantics, $\nu \models \phi$ or $\nu \in [\![\phi]\!]$. Both of these are really saying the same thing: there's some set of states where ϕ is true. If that set is $\mathcal{S}$ then ϕ is valid. In $\mathsf{dG}\mathcal{L}$, it's the same, but specifically we'll write the semantics in the $\nu \in [\![\phi]\!]$ style. All the propositional connectives have the same semantics as they did in $\mathsf{d}\mathcal{L}$, so for example if we unpack the meaning of implication using set language, we get:

$$[\![\phi \rightarrow \psi]\!] = [\![\phi]\!]^{\complement} \cup [\![\psi]\!]$$

The semantics of a game is defined differently from the semantics of a hybrid program, though. To see why, first think about how you play games[4]. If you're playing a game where the rules are α and your objective is ϕ, you don't just first play the game α and then check whether ϕ was true or not (which is what we did in hybrid systems). Instead, you're going to build a *strategy*. Your strategy is going to be totally based on what your objective ϕ is, not just the rules α. You might play totally differently if you're trying to achieve different goals, and your opponent will play differently too! Truth of formulas $\langle\alpha\rangle\phi$ is now all about whether *you have a strategy* that can make ϕ true, and so to make strategy part of the semantics, we want to make our objectives part of the semantics too.

Recall that the semantics for games has two (very closely related!) flavors $\varsigma_\alpha(X)$ and $\delta_\alpha(X)$. They are used in the formula semantics as follows:

- $[\![\langle\alpha\rangle\phi]\!] = \varsigma_\alpha([\![\phi]\!])$. That is to say, $\varsigma_\alpha(X)$ means "here's all the starting conditions where Angel can achieve her objective of X." And here her objective is to make ϕ true.
- $[\![[\alpha]\phi]\!] = \delta_\alpha([\![\phi]\!])$. So $\delta_\alpha(X)$ means "all the start states where Demon could achieve X", very similar to the above.

[4]Hopefully you do play games sometimes!

One would expect δ and ς to be closely related because they talk about two players playing the same game. Indeed they are, for example switching between δ and ς means switching players, and α^d also means switching players, so it is in fact true that $\varsigma_{\alpha^d}(X) = \delta_\alpha(X)$ and $\delta_{\alpha^d}(X) = \varsigma_\alpha(X)$. Identities like this will help justify some of our axioms next week. Not only that, but it intuitively shows us that ς and δ are each, on their own, a perfectly legitimate and exhaustive definition of how to play games: we could define either one in terms of the other.

Let's look at some example cases for the ς semantics. For sequential composition, we have $\varsigma_{\alpha;\beta}(X) = \varsigma_\alpha(\varsigma_\beta(X))$. Here we notice a difference from hybrid systems: our hybrid systems semantics for composition was something like $R_{\alpha;\beta}(S) = R_\beta(T_\alpha(S))$ where S was our set of starting states. Notice that the α and β get swapped! This is a good example of how strategic games thinking is different from systems thinking: we want to *work backward* from our goal to come up with a strategy, so we look at β before α. Or in technical terms, the R semantics are forward-chaining, whereas the game ς/δ semantics are backward-chaining.

But again that just means that with the R semantics, you start with the initial conditions and try to get to your goal:

$$[\![\textit{initial conditions}]\!] = S_0 \Rightarrow S_1 \Rightarrow ... \Rightarrow S_n = [\![\textit{safe}]\!]$$

With the game ς/δ semantics you start from the end set of states, and go backwards to try to figure out *when* your goal is achievable:

$$[\![\textit{initial conditions}]\!] = S_0 \Leftarrow S_1 \Leftarrow ... \Leftarrow S_n = [\![\textit{safe}]\!]$$

Differential equations also have an interesting intuition. The definitions for Angel, $\varsigma_{x'=\theta}(S)$, and Demon, $\delta_{x'=\theta}(S)$, differ only in a quantifier: time is existentially quantified in φ, universally in δ. This is one instance of a very general and useful intuition for Demon proofs. Remember that **A**ngel always means the **a**ctive player and **D**emon always means the **d**ormant player. When you're the dormant player, your only possible strategy is to wait until it's your turn, so your strategy will have to work no matter what Angel is doing right now. If Angel wants to reach a goal X, she can pick the duration t, so there just has to exist one t that works and she'll win. But whene you're Demon you don't get to pick, so you'd better have a strategy that works no matter how long Angel runs that ODE. In fitting with our earlier intuition, this is hard, as are box properties in general.

3 Angel and Demon pull the ol' switcharoonies

So how is the dual operator handled?

$$\varsigma_{\alpha^d}(X) = (\varsigma_\alpha(X^\complement))^\complement$$

At first you might expect that we would have defined this case using the identity I mentioned earlier:

$$\varsigma_{\alpha^d}(X) = \delta_\alpha(X)$$

However this isn"t so easier as a definition because now ς and δ both depend on each other. The definition above is simpler in the sense that we can explain everything with only 1 player even though it's a two player game! It's best read aloud as "I win the dual game when I couldn't lose even if I tried". The intuitive reason this all works is due to what I'll call Geri's law, in reference to the Pixar short at the beginning of recitation:

Geri's Law: Taking turns works the same whether you have one player or two players. Instead of letting someone else play against you, you can get up and walk to their seat. So long as you *play like the opponent* when sitting their seat (i.e. you try to prevent your original objective X from happening) it's just like you have another person playing against you. Or mathematically:

$$(\varsigma_\alpha(X^{\complement}))^{\complement} = \delta_\alpha(X)$$

This style of reasoning about games can take some getting used to: it's important to pay close attention to who's playing right now, what their current objective is, and what *our* overall objective for the proof is. But to end on a positive note, we should be happy that we can relate two-player games to "one-player" games, because it means that even Demons can turn into Angels:

> Everything I've seen needs rearranging
> And for anyone who thinks it's strange
> Then you should be the first to want to make this change
> And for everyone who thinks that life is just a game
> Do you like the part you're playing?
> — Love, You Set The Scene