Outline

1 Learning Objectives
2 Motivation
3 A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4 Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5 An Informal Operational Game Tree Semantics
6 Summary
Outline

1 Learning Objectives

2 Motivation

3 A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls

4 Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
     - Push-around Cart
     - Robot Dance
     - Example: Robot Soccer

5 An Informal Operational Game Tree Semantics

6 Summary
Learning Objectives
Hybrid Systems & Games

- fundamental principles of computational thinking
- logical extensions
- PL modularity principles
- compositional extensions
- differential game logic
- best/worst-case analysis
- models of alternating computation

- adversarial dynamics
- conflicting actions
- multi-agent systems
- angelic/demonic choice

- multi-agent state change
- CPS semantics
- reflections on choices

André Platzer (CMU)
Outline

1. Learning Objectives
2. Motivation
3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5. An Informal Operational Game Tree Semantics
6. Summary
CPS Analysis: Robot Control

**Challenge (Hybrid Systems)**

Fixed rule describing state evolution with both
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Fixed rule describing state evolution with both
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Challenge (Games)

Game rules describing play evolution with both

- Angelic choices (player \( \diamond \) Angel)
- Demonic choices (player \( \Box \) Demon)

<table>
<thead>
<tr>
<th></th>
<th>( \diamond ) ( \Box )</th>
<th>Tr</th>
<th>Pl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trash</td>
<td>1,2</td>
<td>0,0</td>
<td></td>
</tr>
<tr>
<td>Plant</td>
<td>0,0</td>
<td>2,1</td>
<td></td>
</tr>
</tbody>
</table>
CPS Analysis: Robot Control

Challenge (Hybrid Games)

Game rules describing play evolution with
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ♦ vs. Demon □)

André Platzer (CMU)
CPS Analysis: Robot Control

Challenge (Hybrid Games)

Game rules describing play evolution with

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel $\diamondsuit$ vs. Demon $\blacksquare$)
CPS Analysis: RoboCup Soccer

Challenge (Hybrid Games)

Game rules describing play evolution with

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ♦ vs. Demon □)

\[ a, \omega, d \]

\[ d_x, d_y \]
CPSs are Multi-Dynamical Systems

CPS Dynamics
CPS are characterized by multiple facets of dynamical systems.

CPS Compositions
CPS combines multiple simple dynamical effects.

Tame Parts
Exploiting compositionality tames CPS complexity.

Descriptive simplification

Analytic simplification
Dynamic Logics for Dynamical Systems

**differential dynamic logic**
\[ dL = DL + HP \]

**differential game logic**
\[ dGL = GL + HG \]

**stochastic differential DL**
\[ Sd\mathcal{L} = DL + SHP \]

**quantified differential DL**
\[ Qd\mathcal{L} = FOL + DL + QHP \]
1. Learning Objectives

2. Motivation

3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls

4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
     - Push-around Cart
     - Robot Dance
     - Example: Robot Soccer

5. An Informal Operational Game Tree Semantics

6. Summary
### Definition (Hybrid program $\alpha$)

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

### Definition (dL Formula $P$)

$e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P$
Differential Dynamic Logic dL: Syntax

Definition (Hybrid program $\alpha$)

\[
x := e \mid ?Q \mid x' = f(x) \land Q \mid \alpha \cup \beta \mid \alpha ; \beta \mid \alpha^*
\]

Definition (dL Formula $P$)

\[
e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P
\]

Discrete Assign \hspace{1cm} Test Condition \hspace{1cm} Differential Equation \hspace{1cm} Nondet. Choice \hspace{1cm} Seq. Compose \hspace{1cm} Nondet. Repeat

All Reals \hspace{1cm} Some Reals \hspace{1cm} All Runs \hspace{1cm} Some Runs
Differential Dynamic Logic dL: Nondeterminism

**Definition (Hybrid program \( \alpha \))**

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

**Definition (dL Formula \( P \))**

\[ e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P \]

Nondeterminism during HP runs
Differential Dynamic Logic $\text{dL}$: Nondeterminism

Definition (Hybrid program $\alpha$)

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

Definition (dL Formula $P$)

$e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x\ P \mid \exists x\ P \mid \lfloor \alpha \rfloor P \mid \langle \alpha \rangle P$

Nondeterminism during HP runs
### Differential Dynamic Logic dL: Nondeterminism

#### Definition (Hybrid program $\alpha$)

$$
\begin{align*}
    x & := e \mid ?Q \mid x' = f(x) \& Q \mid \alpha \cup \beta \mid \alpha ; \beta \mid \alpha^*
\end{align*}
$$

#### Definition (dL Formula $P$)

$$
\begin{align*}
    e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P
\end{align*}
$$
Differential Dynamic Logic \( dL \): Nondeterminism

**Definition (Hybrid program \( \alpha \))**

\[
x := e | ?Q | x' = f(x) \& Q | \alpha \cup \beta | \alpha; \beta | \alpha^*
\]

**Definition (\( dL \) Formula \( P \))**

\[
e \geq \tilde{e} | \neg P | P \land Q | \forall x\ P | \exists x\ P | [\alpha]P | \langle \alpha \rangle P
\]

Modality decides the mode: help/hurt

- All choices resolved in one way
- Differential Equation
- Nondet. Choice
- Nondet. Repeat

- All Choices
- Some Choice
### Definition (Hybrid program $\alpha$)

Differential Dynamic Logic $dL$: Nondeterminism

<table>
<thead>
<tr>
<th>All choices resolved in one way</th>
<th>Differential Equation</th>
<th>Nondet. Choice</th>
<th>Nondet. Repeat</th>
</tr>
</thead>
</table>

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

### Definition (dL Formula $P$)

$e \geq \tilde{e} \mid \neg P \mid P \& Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha] P \mid \langle \alpha \rangle P$

### Modality decides the mode: help/hurt

$[\alpha_1]\langle\alpha_2\rangle[\alpha_3]\langle\alpha_4\rangle P$ only fixed interaction depth

- All Choices
- Some Choice
Let Angel be one player
Control & Dual Control Operators

- **Angel Ops**
  - $\cup$ choice
  - $\ast$ repeat
  - $x' = f(x)$ evolve
  - $?Q$ challenge

- **Demon Ops**
  - $\cap$ choice
  - $\times$ repeat
  - $x' = f(x)^d$ evolve
  - $?Q^d$ challenge

Let Angel be one player

Let Demon be another player
Control & Dual Control Operators

Duality operator $d$ passes control between players
Game Operators

**Angel Ops**
- $\cup$ choice
- $\ast$ repeat
- $x' = f(x)$ evolve
- $?Q$ challenge

**Demon Ops**
- $\cap$ choice
- $\times$ repeat
- $x' = f(x)^d$ evolve
- $?Q^d$ challenge

**Duality operator $^d$ passes control between players**
Game Operators

**Diamond Angel Ops**

- Union: \( \cup \)
- Choice: \( \times \)
- Repeat: \( * \)
- Evolve: \( x' = f(x) \)
- Challenge: \( ?Q \)

**Diamond Demon Ops**

- Intersection: \( \cap \)
- Choice: \( \times \)
- Repeat: \( * \)
- Evolve: \( x' = f(x)^d \)
- Challenge: \( ?Q^d \)

Duality operator \( d \) passes control between players.
Game Operators

Diamond Box: Angel Ops

- Union (U)
- Choice
- * (repeat)
- $x' = f(x)$ (evolve)
- ?Q (challenge)

Diamond Box: Demon Ops

- Intersection (∩)
- Choice
- × (repeat)
- $x' = f(x)^d$ (evolve)
- ?Q^d (challenge)

Duality operator $^d$ passes control between players

André Platzer (CMU)
LFCPS/14: Hybrid Systems & Games
Definable Game Operators

- Angel Ops
  - \( \bigcup \)
  - \( * \)
  - \( x' = f(x) \)
  - ?Q
  - choice
  - repeat
  - evolve
  - challenge

- Demon Ops
  - \( \cap \)
  - \( \times \)
  - \( x' = f(x)^d \)
  - ?Q^d
  - choice
  - repeat
  - evolve
  - challenge

\[
\text{if}(Q) \alpha \text{ else } \beta \equiv \\
\text{while}(Q) \alpha \equiv \\
\alpha \cap \beta \equiv \\
\alpha ^ \times \equiv \\
(x' = f(x) \& Q)^d \quad x' = f(x) \& Q \\
(x := e)^d \quad x := e \\
?Q^d \quad ?Q
\]
Definable Game Operators

**Angel Ops**
- \( \bigcup \)
- \( \ast \)
- \( x' = f(x) \)
- \( ?Q \)
- choice
- repeat
- evolve
- challenge

**Demon Ops**
- \( \bigcap \)
- \( \times \)
- \( x' = f(x)^d \)
- \( ?Q^d \)
- choice
- repeat
- evolve
- challenge

\[
\text{if}(Q) \alpha \text{ else } \beta \equiv (?Q ; \alpha) \cup (?\neg Q ; \beta)
\]

\[
\text{while}(Q) \alpha \equiv \\
\alpha \cap \beta \equiv \\
\alpha \times \equiv \\
(x' = f(x) \land Q)^d \quad x' = f(x) \land Q \\
(x := e)^d \quad x := e \\
?Q^d \quad ?Q
\]
Definable Game Operators

**Angel Ops**

- $\cup$
- $\ast$
- $x' = f(x)$
- $?Q$
- choice
- repeat
- evolve
- challenge

**Demon Ops**

- $\cap$
- $\times$
- $x' = f(x)^d$
- $?Q^d$
- choice
- repeat
- evolve
- challenge

\[
\text{if}(Q) \alpha \text{ else } \beta \equiv (?Q; \alpha) \cup (?\neg Q; \beta)
\]

\[
\text{while}(Q) \alpha \equiv (?Q; \alpha)^\ast; ?\neg Q
\]

\[
\alpha \cap \beta \equiv \alpha^\times \equiv
\]

\[
(x' = f(x) \& Q)^d \quad x' = f(x) \& Q
\]

\[
(x := e)^d \quad x := e
\]

\[
?Q^d \quad ?Q
\]
Definable Game Operators

**Angel Ops**
- $\cup$
- $\ast$
- $x' = f(x)$
- $?Q$
- choice
- repeat
- evolve
- challenge

**Demon Ops**
- $\cap$
- $\times$
- $x' = f(x)^d$
- $?Q^d$
- choice
- repeat
- evolve
- challenge

### If $Q$

if $(Q) \alpha$ else $\beta \equiv (\alpha; ?Q) \cup (?\neg Q; \beta)$

### While $Q$

while $(Q) \alpha \equiv (\alpha; ?Q)^*; ?\neg Q$

### Intersections

$\alpha \cap \beta \equiv \alpha^\times \equiv$

$(x' = f(x) \& Q)^d \quad x' = f(x) \& Q$

$(x := e)^d \quad x := e$

$?Q^d \quad ?Q$
Definable Game Operators

**Angel Ops**
- $\cup$
- $\ast$
- $x' = f(x)$
- $?Q$
- choice
- repeat
- evolve
- challenge

**Demon Ops**
- $\cap$
- $\times$
- $x' = f(x)^d$
- $?Q^d$
- choice
- repeat
- evolve
- challenge

**Game Operators**

\[
\text{if}(Q)\alpha \text{ else } \beta \equiv (\ ?Q; \alpha ) \cup (\neg Q; \beta )
\]

\[
\text{while}(Q)\alpha \equiv (\ ?Q; \alpha )^*; \neg Q
\]

\[
\alpha \cap \beta \equiv (\alpha^d \cup \beta^d)^d
\]

\[
\alpha^x \equiv (x' = f(x) \& Q)^d
\]

\[
(x := e)^d
\]

\[
?Q^d
\]

\[
?Q
\]
Definable Game Operators

Diamond Angel Ops

\[ \begin{align*}
\cup & \quad \text{choice} \\
* & \quad \text{repeat} \\
x' = f(x) & \quad \text{evolve} \\
?Q & \quad \text{challenge}
\end{align*} \]

Box Demon Ops

\[ \begin{align*}
\cap & \quad \text{choice} \\
\times & \quad \text{repeat} \\
x' = f(x)^d & \quad \text{evolve} \\
?Q^d & \quad \text{challenge}
\end{align*} \]

\[
\text{if}(Q) \; \alpha \; \text{else} \; \beta \equiv (?Q; \alpha) \cup (?\neg Q; \beta)
\]

\[
\text{while}(Q) \; \alpha \equiv (?Q; \alpha)^*; ?\neg Q
\]

\[
\alpha \cap \beta \equiv (\alpha^d \cup \beta^d)^d
\]

\[
\alpha^\times \equiv ((\alpha^d)^*)^d
\]

\[
(x' = f(x) \& Q)^d \quad x' = f(x) \& Q
\]

\[
(x := e)^d \quad x := e
\]

\[
?Q^d \quad ?Q
\]
Definable Game Operators

**Diamond Angel Ops**
- ∪
- ∗
- $x' = f(x)$
- ?$Q$
- choice
- repeat
- evolve
- challenge

**Box Demon Ops**
- ∩
- ∗
- $x' = f(x)^d$
- ?$Q^d$
- choice
- repeat
- evolve
- challenge

\[
\begin{align*}
\text{if}(Q) \alpha \text{ else } \beta \equiv (\ ?Q ; \alpha) \cup (\ ?\neg Q ; \beta) \\
\text{while}(Q) \alpha \equiv (\ ?Q ; \alpha)^* ; \ ?\neg Q \\
\alpha \cap \beta \equiv (\alpha^d \cup \beta^d)^d \\
\alpha^\times \equiv ((\alpha^d)^*)^d \\
(x' = f(x) & Q)^d \neq x' = f(x) & Q \\
(x := e)^d \quad x := e \\
?Q^d \quad ?Q
\end{align*}
\]
Definable Game Operators

**Angel Ops**

- $\cup$
- $\ast$
- $x' = f(x)$
- $?Q$
- choice
- repeat
- evolve
- challenge

**Demon Ops**

- $\cap$
- $\times$
- $x' = f(x)^d$
- $?Q^d$
- choice
- repeat
- evolve
- challenge

**Examples**

- if $(Q) \alpha$ else $\beta \equiv (\ ?Q; \alpha ) \cup ( \neg ?Q; \beta)$
- while $(Q) \alpha \equiv (\ ?Q; \alpha )^* ; ?\neg Q$
- $\alpha \cap \beta \equiv (\alpha^d \cup \beta^d)^d$
- $\alpha^\times \equiv ((\alpha^d)^*)^d$
- $(x' = f(x) \text{ & } Q)^d \not\equiv x' = f(x) \text{ & } Q$
- $(x := e)^d \equiv x := e$
- $?Q^d \not\equiv ?Q$
Definable Game Operators

\[ \begin{array}{c|c}
\text{Angel Ops} & \text{Demon Ops} \\
\hline
\bigcup & \bigcap \\
\ast & \times \\
\text{choice} & \text{choice} \\
\text{repeat} & \text{repeat} \\
\text{evolve} & \text{evolve} \\
\text{challenge} & \text{challenge} \\
\hline
x' = f(x) & x' = f(x)^d \\
\text{?}Q & \text{?}Q^d \\
\end{array} \]

\[
\begin{align*}
\text{if}(Q) \alpha \text{ else } \beta & \equiv (?Q; \alpha) \cup (?\neg Q; \beta) \\
\text{while}(Q) \alpha & \equiv (?Q; \alpha)^*; ?\neg Q \\
\alpha \cap \beta & \equiv (\alpha^d \cup \beta^d)^d \\
\alpha^\times & \equiv ((\alpha^d)^*)^d \\
(x' = f(x) & Q)^d & \not\equiv x' = f(x) & Q \\
(x := e)^d & \equiv x := e \\
?Q^d & \not\equiv ?Q
\end{align*}
\]
Outline

1 Learning Objectives

2 Motivation

3 A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls

4 Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
     - Push-around Cart
     - Robot Dance
     - Example: Robot Soccer

5 An Informal Operational Game Tree Semantics

6 Summary
Definition (Hybrid game $\alpha$)

\[
\alpha, \beta ::= x := e \mid ?Q \mid x' = f(x) \& Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \alpha^d
\]
Hybrid Games: Syntax

\[ \alpha, \beta ::= x := e \mid ?Q \mid x' = f(x) \land Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \alpha^d \]

Definition (Hybrid game \( \alpha \))

- Discrete Assign
- Test Game
- Differential Equation
- Choice Game
- Seq. Game
- Repeat Game
Hybrid Games: Syntax

Definition (Hybrid game $\alpha$)

$\alpha, \beta ::= x := e \mid ?Q \mid x' = f(x) & Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \alpha^d$
Example: Push-around Cart

Hybrid systems can't say that $a$ is Angel's choice and $d$ is Demon's.
Example: Push-around Cart

\[(a := 1 \cup a := -1); (d := 1 \cup d := -1)^d; \{x' = v, v' = a + d\}\]
Example: Push-around Cart

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

\[ ((d := 1 \cup d := -1)^d; (a := 1 \cup a := -1); \{x' = v, \nu' = a + d\})^* \]
Example: Push-around Cart

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

\[ ((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{ x' = v, v' = a + d \})^* \]
Example: Push-around Cart

\[
\begin{aligned}
(a := 1 \cup a := -1); (d := 1 \cap d := -1); \{x' = v, v' = a + d\}^* \\
(d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\}^* \\
HP \ (d := 1 \cup d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\}^*
\end{aligned}
\]
Example: Push-around Cart

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

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

\[ \text{HP } ((d := 1 \cup d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* \]

Hybrid systems can't say that \( a \) is Angel's choice and \( d \) is Demon's.
### Definition (Hybrid game $\alpha$)

\[
\alpha, \beta ::= x := e \mid ?Q \mid x' = f(x) & Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \alpha^d
\]
### Definition (Hybrid game $\alpha$)

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

### Definition (dGL Formula $P$)

\[
P, Q ::= e \geq \bar{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid \langle \alpha \rangle P \mid [\alpha]P
\]
Differential Game Logic: Syntax

**Definition (Hybrid game \( \alpha \))**

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

**Definition (dGL Formula \( P \))**

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

Discrete Assign Test Game Differential Equation Choice Game Seq. Game Repeat Game

All Reals

Some Reals
Definition (Hybrid game $\alpha$)

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

Definition (dGL Formula $P$)

$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid \langle \alpha \rangle P \mid [\alpha] P$
# Differential Game Logic: Syntax

**Definition (Hybrid game \( \alpha \))**

\[
\begin{align*}
\alpha, \beta & ::= \ x := e \ | \ ?Q \ | \ x' = f(x) \& Q \ | \ \alpha \cup \beta \ | \ \alpha; \beta \ | \ \alpha^* \ | \ \alpha^d
\end{align*}
\]

**Definition (dGL Formula \( P \))**

\[
\begin{align*}
P, Q & ::= \ e \geq \tilde{\epsilon} \ | \ \neg P \ | \ P \& Q \ | \ \forall x \ P \ | \ \exists x \ P \ | \ \langle \alpha \rangle P \ | \ [\alpha]P
\end{align*}
\]
Definition (Hybrid game $\alpha$)

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

Definition (dGL Formula $P$)

\[
P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x P \mid \exists x P \mid \langle \alpha \rangle P \mid [\alpha] P
\]
Simple Examples

\[ \langle (x := x + 1; (x' = 1)^d \cup x := x - 1)^* \rangle (0 \leq x < 1) \]

\[ \langle (x := x + 1; (x' = 1)^d \cup (x := x - 1 \cap x := x - 2))^* \rangle (0 \leq x < 1) \]
Simple Examples

\[ \models \langle (x := x + 1; \ (x' = 1)^d \cup x := x - 1)^* \rangle (0 \leq x < 1) \]

\[ \langle (x := x + 1; \ (x' = 1)^d \cup (x := x - 1 \land x := x - 2))^* \rangle (0 \leq x < 1) \]
Simple Examples

\[ \models (x := x + 1; (x' = 1)^d \cup x := x - 1)^* (0 \leq x < 1) \]

\[ \not\models (x := x + 1; (x' = 1)^d \cup (x := x - 1 \cap x := x - 2))^* (0 \leq x < 1) \]
\( v \geq 1 \rightarrow \\
\left[ ((d := 1 \cup d := -1)d; (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* \right] v \geq 0 \)
Example: Push-around Cart

\[ v \geq 1 \rightarrow \]
\[ \left[ (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\} \right]^* \] \[ v \geq 0 \]
Example: Push-around Cart

\[ d \text{ before } a \text{ can compensate} \]

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

\[ x \geq 0 \land v \geq 0 \rightarrow 
[(d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\}]^* x \geq 0 \]
Example: Push-around Cart

$\vdash v \geq 1 \rightarrow$ \hspace{1cm} $\exists$ before $a$ can compensate

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

$\vdash x \geq 0 \land v \geq 0 \rightarrow$ \hspace{1cm} $\exists$ before $a$ can compensate

$[((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^*] x \geq 0$
Example: Push-around Cart

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

\( \models v \geq 1 \quad \rightarrow \quad d \) before \( a \) can compensate

\[ [((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^*] v \geq 0 \]
Example: Push-around Cart

\[ x \geq 0 \rightarrow v \geq 0 \]

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

\[ x \geq 0 \rightarrow \text{d before a can compensate} \]

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

boring by skip
Example: Push-around Cart

\[ \vdash v \geq 1 \rightarrow \]

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

\[ \langle ((d := 1 \land d := -1); (a := 1 \lor a := -1); \{x' = v, v' = a + d\})^* \rangle x \geq 0 \]

\( d \) before \( a \) can compensate
Example: Push-around Cart

\[ d \] before \( a \) can compensate

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

\[ \not\vdash \]

\[ \langle ((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* \rangle x \geq 0 \]
Example: Push-around Cart

\[ x \xrightarrow{v} \text{d before a can compensate} \]

\[ \begin{align*}
\models v \geq 1 & \rightarrow \quad \text{d before a can compensate} \\
\left[ ((d := 1 \land d := -1); (a := 1 \lor a := -1); \{x' = v, v' = a + d\})^* \right] v \geq 0 \\
\not\models \quad \text{counterstrategy} \quad d := -1 \\
\langle ((d := 1 \land d := -1); (a := 1 \lor a := -1); \{x' = v, v' = a + d\})^* \rangle x \geq 0 \\
\langle ((d := 1 \land d := -1); (a := 2 \lor a := -2); \{x' = v, v' = a + d\})^* \rangle x \geq 0
\end{align*} \]
Example: Push-around Cart

\[ x \geq 0 \quad \vdash \quad v \geq 1 \rightarrow \quad d \quad \text{before} \quad a \quad \text{can compensate} \]

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

\[ x \geq 0 \quad \not\vdash \quad \langle (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* \rangle x \geq 0
\]

\[ x \geq 0 \quad \vdash \quad \langle (d := 1 \cap d := -1); (a := 2 \cup a := -2); \{x' = v, v' = a + d\})^* \rangle x \geq 0
\]
Example: Push-around Cart

\[ d \] before \( a \) can compensate

\[ \{x' = v, v' = a + d\} \]

\[ v \geq 0 \]

counterstrategy \( d := -1 \)

\[ x \geq 0 \]

\[ t := 0; \{x' = v, v' = a + d, t' = 1 \land t \leq 1\} \]

\[ x^2 \geq 100 \]
Example: Push-around Cart

\[ \vdash v \geq 1 \rightarrow \]
\[ [((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^*]v \geq 0 \]
\[ \not\vdash \langle (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\} \rangle^* x \geq 0 \]
\[ \vdash \langle (d := 1 \cap d := -1); (a := 2 \cup a := -2); \{x' = v, v' = a + d\} \rangle^* x \geq 0 \]
\[ \vdash \langle (d := 2 \cap d := -2); (a := 2 \cup a := -2); a := d \text{ then } a := 2 \text{ sign } v \]
\[ t := 0; \{x' = v, v' = a + d, t' = 1 \& t \leq 1\} \rangle^* x^2 \geq 100 \]
Example: WALL•E and EVE Robot Dance

\[(w - e)^2 \leq 1 \land v = f \rightarrow \]
\[\langle (u := 1 \cap u := -1); \]
\[(g := 1 \cup g := -1); \]
\[t := 0; \]
\[\{ w' = v, v' = u, e' = f, f' = g, t' = 1 \land t \leq 1 \}^d \]
\[\times \rangle (w - e)^2 \leq 1 \]

EVE at \(e\) plays Angel’s part controlling \(g\)

WALL•E at \(w\) plays Demon’s part controlling \(u\)
Example: WALL·E and EVE Robot Dance and the World

\[(w - e)^2 \leq 1 \land v = f \rightarrow \langle ((u := 1 \cap u := -1); (g := 1 \cup g := -1)); t := 0; \{w' = v, v' = u, e' = f, f' = g, t' = 1 \& t \leq 1\}^d \rangle\] (w - e)^2 \leq 1

EVE at \(e\) plays Angel’s part controlling \(g\)

WALL·E at \(w\) plays Demon’s part controlling \(u\) and world time
Example: WALL·E and EVE

\[(w - e)^2 \leq 1 \land v = f \rightarrow \]
\[
\left[ ((u := 1 \cap u := -1); \\
(g := 1 \cup g := -1); \\
t := 0; \\
\{ w' = v, v' = u, e' = f, f' = g, t' = 1 \land t \leq 1 \} \right) \times \]
\[(w - e)^2 > 1 \]

WALL·E at \( w \) plays Demon’s part controlling \( u \) and world time

EVE at \( e \) plays Angel’s part controlling \( g \)
Example: Goalie in Robot Soccer

\[ x < 0 \land v > 0 \land y = g \rightarrow \]

\[ \left< \left< w := +w \cap w := -w \right); \right. \]

\[ \left. \left( u := +u \cup u := -u \right); \{ x' = v, y' = w, g' = u \} \right)^* \right> x^2 + (y - g)^2 \leq 1 \]
Example: Goalie in Robot Soccer

\[ x < 0 \land v > 0 \land y = g \rightarrow \]
\[ \langle (w := +w \cap w := -w); ((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \rangle x^2 + (y - g)^2 \leq 1 \]
Example: Goalie in Robot Soccer

\[
x < 0 \land v > 0 \land y = g \rightarrow \\
\left( (w := +w \cap w := -w); \right. \\
\left. ((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\}^*) \right) x^2 + (y - g)^2 \leq 1
\]
Example: Goalie in Robot Soccer

\[
x < 0 \land v > 0 \land y = g \rightarrow \\
\langle (w := +w \cap w := -w); \\
((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^{*} \rangle x^2 + (y - g)^2 \leq 1
\]
Example: Goalie in Robot Soccer

\[
x < 0 \land v > 0 \land y = g \rightarrow \\
\left( (w := +w \cap w := -w); \\
((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \right) x^2 + (y - g)^2 \leq 1
\]
Example: Goalie in Robot Soccer

Goalie’s Secret

\[
\left(\frac{x}{v}\right)^2 (u - w)^2 \leq 1 \land \\
x < 0 \land v > 0 \land y = g \\
\langle (w := +w \land w := -w); \\
\langle (u := +u \cup u := -u); \{x' = v, y' = w, g' = u\}^* \rangle x^2 + (y - g)^2 \leq 1
\]

André Platzer (CMU)

LFCPS/14: Hybrid Systems & Games

LFCPS/14  20 / 24
Outline

1 Learning Objectives
2 Motivation
3 A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4 Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5 An Informal Operational Game Tree Semantics
6 Summary
Definition (Hybrid game $\alpha$: operational semantics)

\[ x \ ::= \ e \]

\[ \omega_x [e] \]

\[ \omega \]

\[ e \]

\[ \parallel \]

\[ x \]

\[ \omega \]

\[ \omega_x \]

\[ \omega \]

\[ \parallel \]

\[ x \]

\[ \omega \]

\[ \omega_x \]

\[ \omega \]

\[ \parallel \]

\[ x \]

\[ \omega \]
Definition (Hybrid game $\alpha$: operational semantics)

$$x' = f(x) \& Q$$
Definition (Hybrid game $\alpha$: operational semantics)
Definition (Hybrid game $\alpha$: operational semantics)

$$\omega \xrightarrow{\alpha} o \xrightarrow{\omega}$$

$$\omega \xrightarrow{\alpha} s_1 \xrightarrow{\alpha} s_j \xrightarrow{\alpha} s_\lambda$$

$$\omega \xrightarrow{\beta} t_1 \xrightarrow{\beta} t_j \xrightarrow{\beta} t_\kappa$$

$$\omega \xrightarrow{\alpha} \alpha \cup \beta$$

The diagram illustrates the operational semantics of a hybrid game $\alpha$. The states $s_1$, $s_j$, and $s_\lambda$ are connected to the state $\omega$ via the transitions $\alpha$, indicating that the game can move from $\omega$ to $s_1$, $s_j$, and $s_\lambda$. Similarly, the states $t_1$, $t_j$, and $t_\kappa$ are also connected to $\omega$ via $\beta$. The union $\alpha \cup \beta$ represents the combined operational semantics of the game, showing the possible transitions from $\omega$. The states are labeled with indices and the transitions are labeled with the corresponding actions $\alpha$ and $\beta$. The diagram also includes a left and right direction indicating the flow of the operational semantics.
Definition (Hybrid game $\alpha$: operational semantics)

$$\omega \equiv e$$

$$\omega^x \equiv e$$

$$\omega^x \equiv f(x) \wedge Q$$

$$r \phi(t)$$

$$t \phi(0)$$

$$\omega ? Q \omega \in \{Q\}$$

$$\alpha; \beta$$

André Platzer (CMU)
Definition (Hybrid game $\alpha$: operational semantics)
Definition (Hybrid game $\alpha$: operational semantics)

\[
\omega := \begin{cases} 
\begin{cases} e \omega \omega [ [ e ] ] 
\end{cases} & x := e \\
\omega \omega' = f(x) & Q \phi(r) \\
\phi(t) & 0 \\
\omega ? Q \omega \in [ [ Q ] ] & \omega \in \alpha \cup \beta
\end{cases}
\]

$\alpha$ and $\alpha^d$
Filibusters

\[ \langle x := 0 \cap x := 1 \rangle^* \] \[ x = 0 \]
\[(x := 0 \land x := 1)^* x = 0\]

\[\text{wfd} \Rightarrow \text{false unless } x = 0\]
\[(x' = 1^d; x := 0)^* \] \(x = 0\)

\[(x := 0; x' = 1^d)^* \] \(x = 0\)

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

\(\xrightarrow{\text{wfd}} \) false unless \(x = 0\)
Filibusters & The Significance of Finitude

\[ \langle (x' = 1^d; x := 0)^* \rangle x = 0 \]
\[ \langle (x := 0; x' = 1^d)^* \rangle x = 0 \]
\[ \langle (x := 0 \cap x := 1)^* \rangle x = 0 \]

\( wfd \) \( \sim \) false unless \( x = 0 \)
Filibusters & The Significance of Finitude

\[ \langle x' = 1^d; x := 0 \rangle^* x = 0 \]
\[ \langle x := 0; x' = 1^d \rangle^* x = 0 \]
\[ \langle x := 0 \cap x := 1 \rangle^* x = 0 \]

\[ \text{wfd} \Rightarrow \text{false unless } x = 0 \]

Well-defined games can't be postponed forever
Outline

1. Learning Objectives
2. Motivation
3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5. An Informal Operational Game Tree Semantics
6. Summary
Definition (Hybrid game $\alpha$)

\[ \alpha, \beta ::= x := e \mid ?Q \mid x' = f(x) \land Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \alpha^d \]

Definition (dGL Formula $P$)

\[ P, Q ::= e \geq \bar{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid \langle \alpha \rangle P \mid [\alpha] P \]
differential game logic

\[ d\text{GL} = GL + HG = d\text{L} + ^d \]

- Differential game logic
- Logic for hybrid games
- Compositional PL + logic
- Discrete + continuous + adversarial
- Operational semantics (informally)

Next chapter

1. Formal semantics
Example: Robot Factory
Example: Robot Factory Decentralized Automation

((l_x, l_y), (r_x, r_y))

Model
- (x, y) robot coordinates
- (v_x, v_y) velocities
- conveyor belts may instantaneously increase robot’s velocity by (c_x, c_y)

Primary objectives of the robot
- Leave within time \( \varepsilon \)
- Never leave outer

Challenges
- Distributed, physical environment
- Possibly conflicting secondary objectives
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( (\text{true} \cup (? (x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0) \right) \right)

// belt

\cup (? (e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0); \right);
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\begin{align*}
&\left(\ ?true \cup (\ ?(x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0) \ \ // \ \text{belt} \\
&\quad \cup (\ ?(e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0) \right); \\
&\ (a_x := \ast; \ ?(-A \leq a_x \leq A); \\
&\ a_y := \ast; \ ?(-A \leq a_y \leq A); \quad // \ \text{“independent” robot acceleration} \\
&\ t_s := 0 \right) ^d ; \\
\end{align*}
\]
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( \text{true} \cup \left( (x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0 \right) \right) \quad \text{// belt}
\]

\[
\cup \left( (e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0 \right) \right) ;
\]

\[
( a_x := \ast; \ ?(-A \leq a_x \leq A); \quad a_y := \ast; \ ?(-A \leq a_y \leq A); \quad \text{// “independent” robot acceleration}
\]

\[
t_s := 0 \right) \quad \text{d};
\]

\[
( x' = v_x, y' = v_y, v'_x = a_x, v'_y = a_y, t' = 1, t'_s = 1 \land t_s \leq \varepsilon ) ;
\]

\[
\right) \ast
\]
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( (\text{true} \cup (? (x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0) \right. \\
\left. \quad \cup (? (e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0) \right) ;
\]

\[
(a_x := *; ?(-A \leq a_x \leq A); \\

a_y := *; ?(-A \leq a_y \leq A); \quad \text{// “independent” robot acceleration}
\]

\[
t_s := 0 \right)^d ;
\]

\[
\left( (x' = v_x, y' = v_y, v'_x = a_x, v'_y = a_y, t' = 1, t'_s = 1 \& t_s \leq \varepsilon ) \right)
\]

\[
\cap (? (a_x v_x \leq 0 \land a_y v_y \leq 0)^d; \quad \text{// brake}
\]

if \ v_x = 0 then \ a_x := 0 fi; \quad \text{// per direction: no time lock}

if \ v_y = 0 then \ a_y := 0 fi;

\[
\left( (x' = v_x, y' = v_y, v'_x = a_x, v'_y = a_y, t' = 1, t'_s = 1 \right)
\]

\[
\left. \quad \land t_s \leq \varepsilon \land a_x v_x \leq 0 \land a_y v_y \leq 0)\right) \right) \right)^*
\]
Robot Factory Automation (RF)

Proposition (Robot stays in □)

\[ \models (x = y = 0 \land v_x = v_y = 0 \land \text{Controllability Assumptions}) \]
\[ \rightarrow [RF](x \in [l_x, r_x] \land y \in [l_y, r_y]) \]

Proposition (Stays in □ and leaves □□ on time)

\[ RF|_x: RF \text{ projected to the } x\text{-axis} \]
\[ \models (x = 0 \land v_x = 0 \land \text{Controllability Assumptions}) \]
\[ \rightarrow [RF|_x](x \in [l_x, r_x] \land (t \geq \varepsilon \rightarrow x \geq x_b)) \]
André Platzer.

Logical Foundations of Cyber-Physical Systems.
URL: http://www.springer.com/978-3-319-63587-3,
doi:10.1007/978-3-319-63588-0.

André Platzer.
Differential game logic.

André Platzer.
Logics of dynamical systems.
In LICS [12], pages 13–24.

André Platzer.
Logic & proofs for cyber-physical systems.
André Platzer.
Differential dynamic logic for hybrid systems.

André Platzer.
A complete uniform substitution calculus for differential dynamic logic.

André Platzer.
Differential hybrid games.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [12], pages 541–550.
doi:10.1109/LICS.2012.64.
André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.

André Platzer.

Jan-David Quesel and André Platzer.