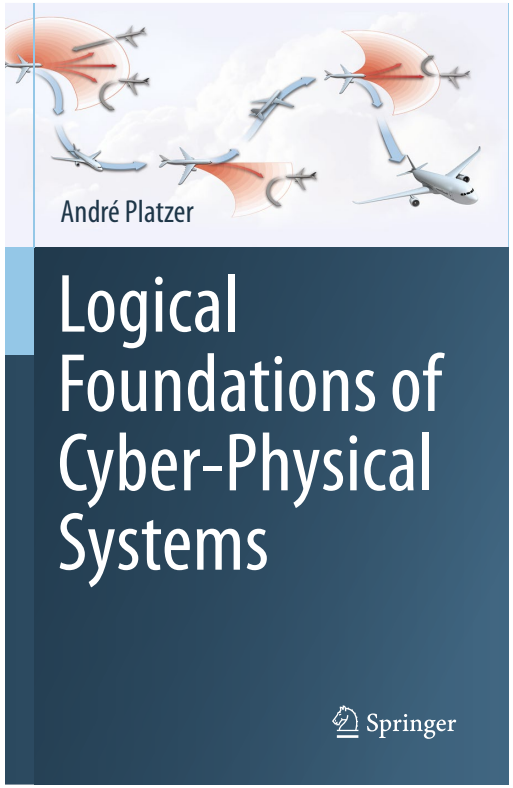
# 14: Hybrid Systems & Games

## Logical Foundations of Cyber-Physical Systems




André Platzer

Carnegie Mellon University
Computer Science Department

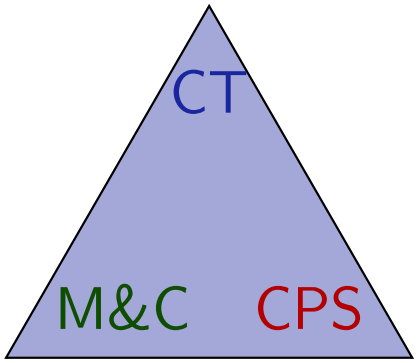
# Outline

# Outline

# 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

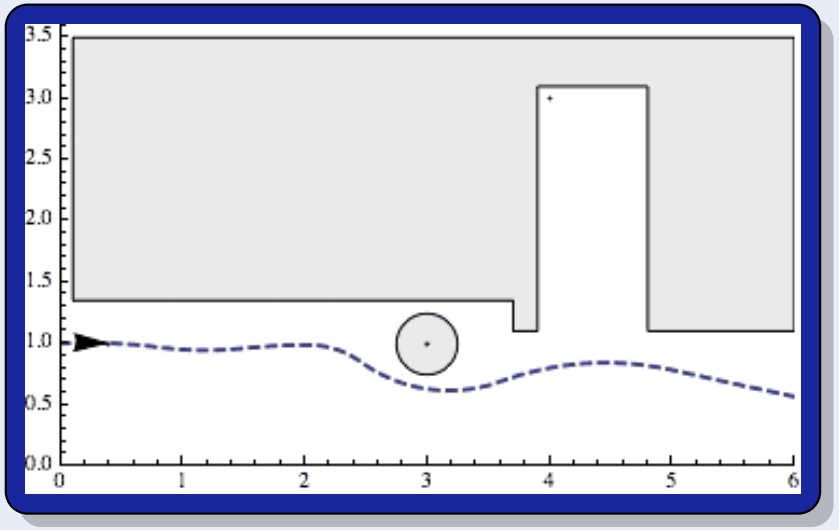
multi-agent state change
CPS semantics
reflections on choices

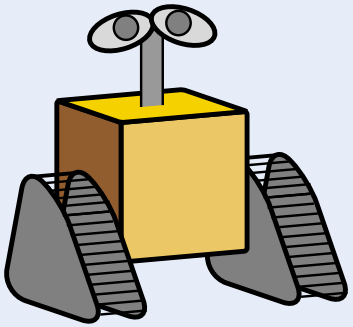
# Outline

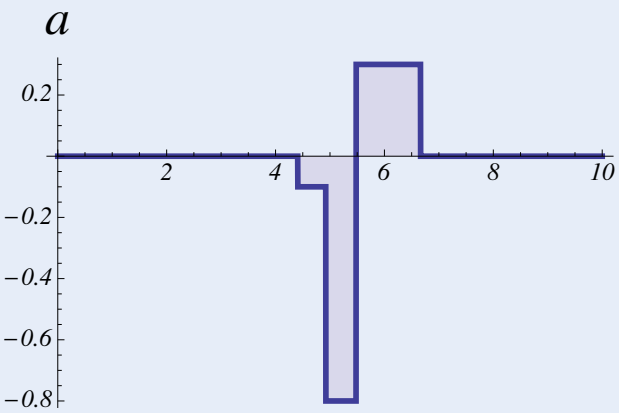
## Challenge (Hybrid Systems)

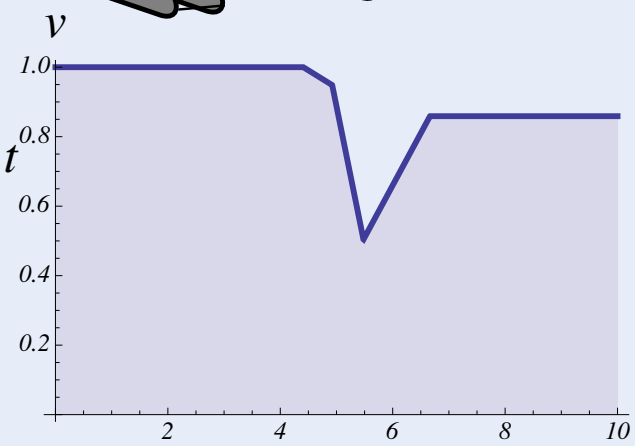
Fixed rule describing state evolution with both

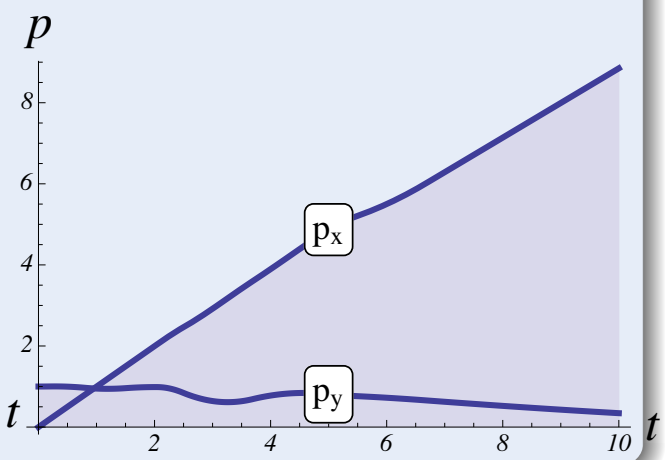
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)





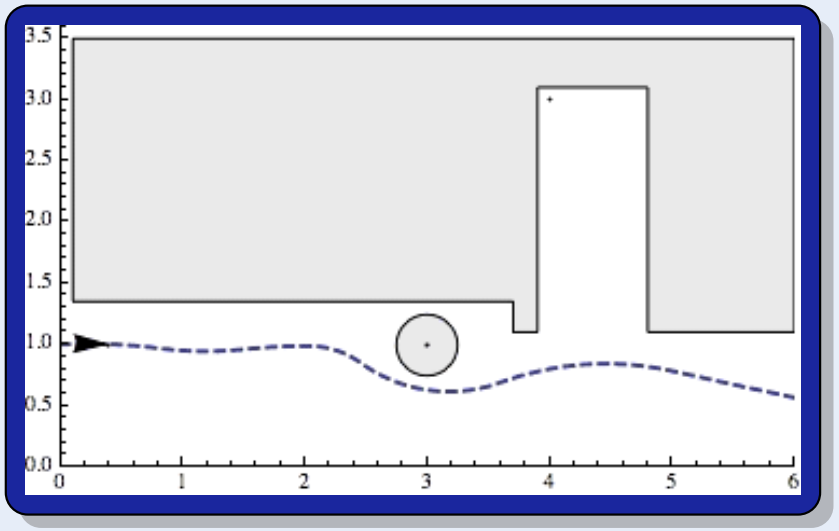



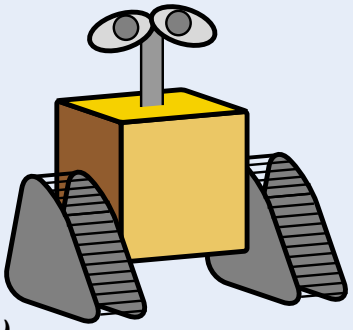



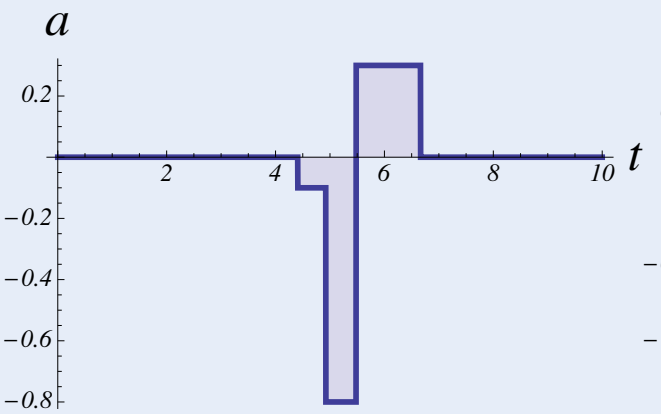
## Challenge (Hybrid Systems)

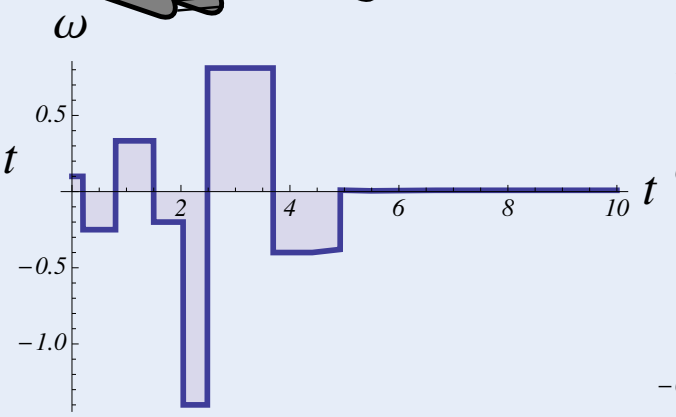
Fixed rule describing state evolution with both

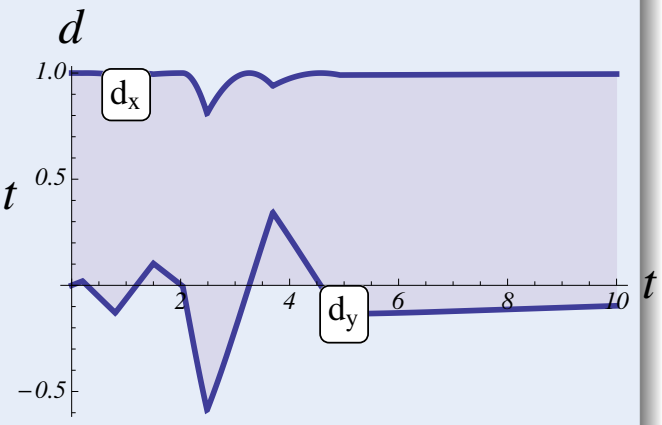
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)





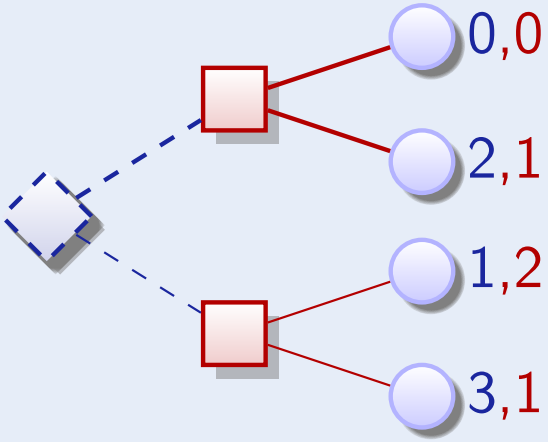



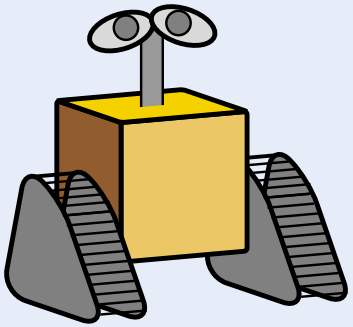



## Challenge (Games)

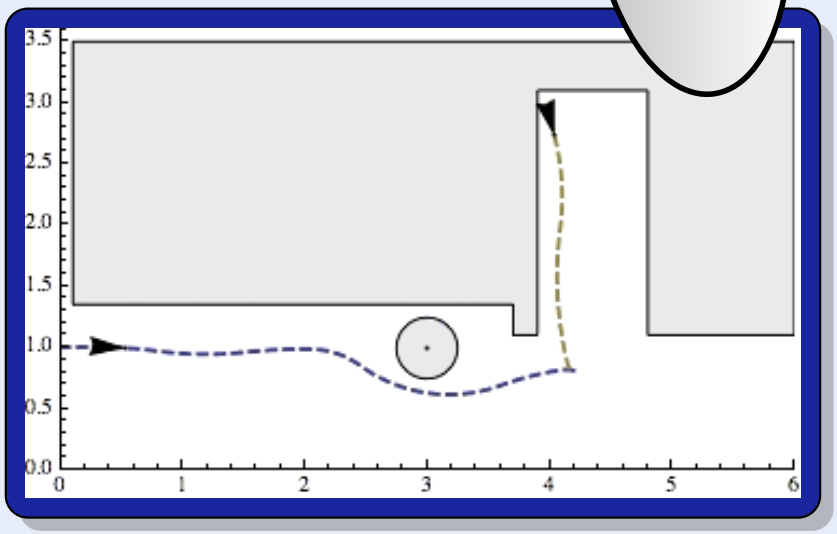
Game rules describing play evolution with both

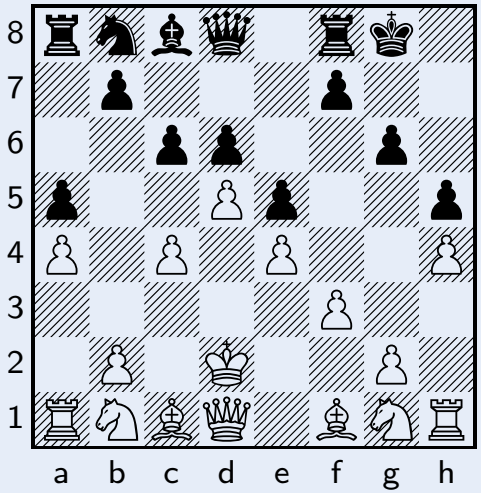
- Angelic choices (player ◇ Angel)
- Demonic choices (player □ Demon)



| ◇\□ | Tr | Pl |
|---|---|---|
| Trash | 1,2 | 0,0 |
| Plant | 0,0 | 2,1 |

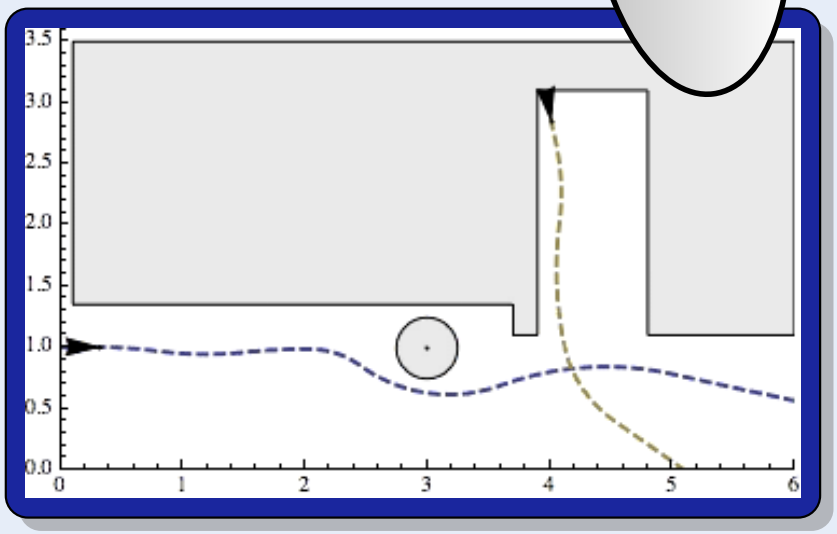


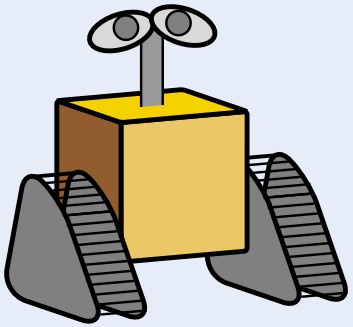



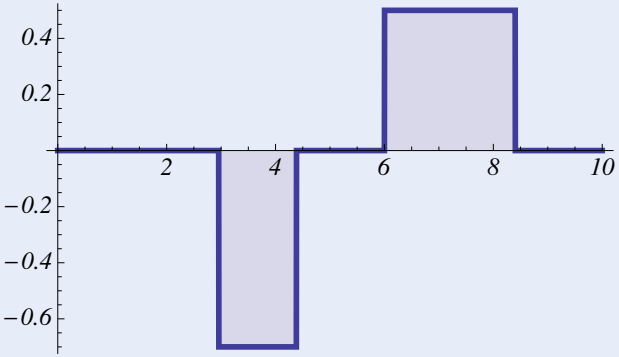
## Challenge (Hybrid Games)

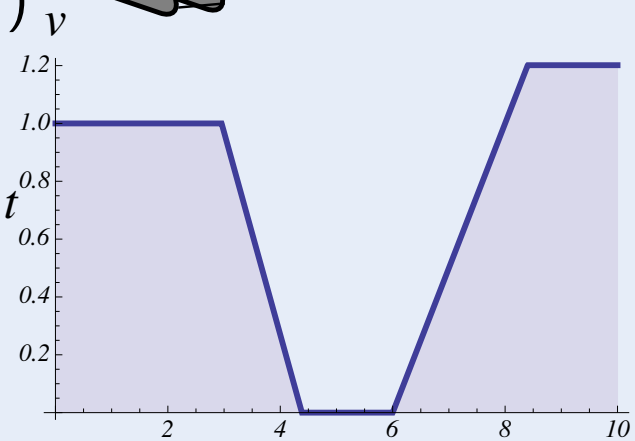
Game rules describing play evolution with

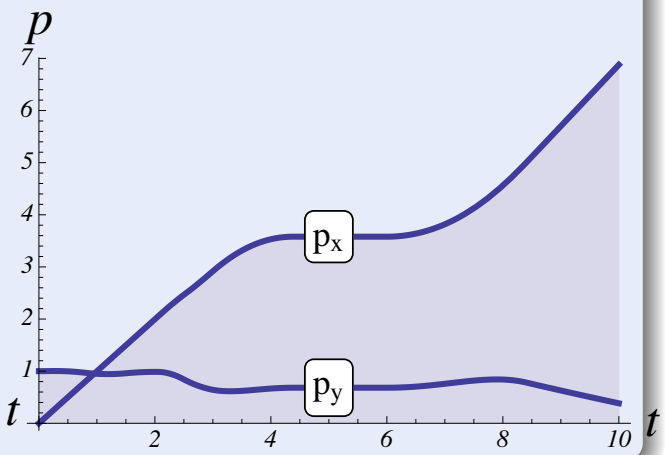
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ◇ vs. Demon □)





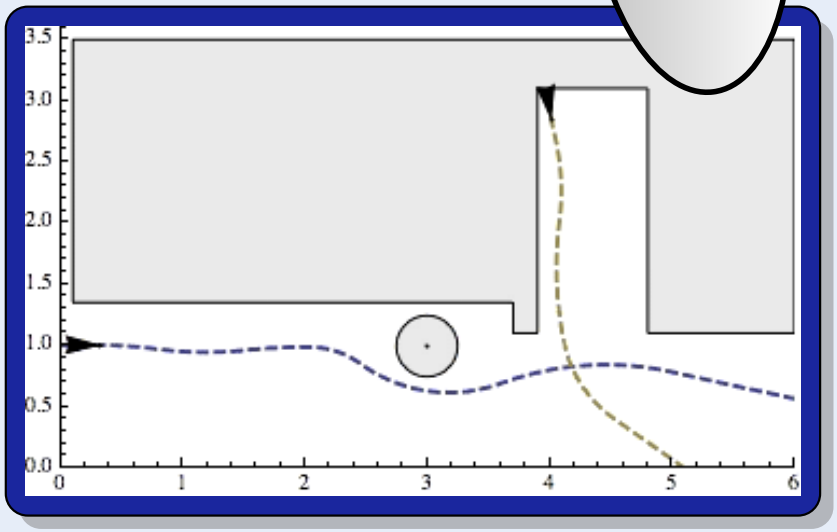



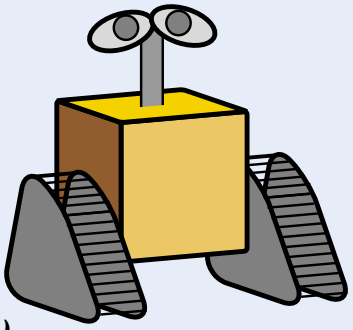



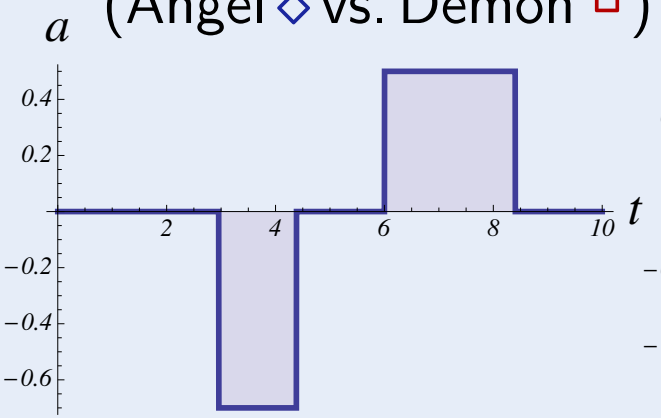
## Challenge (Hybrid Games)

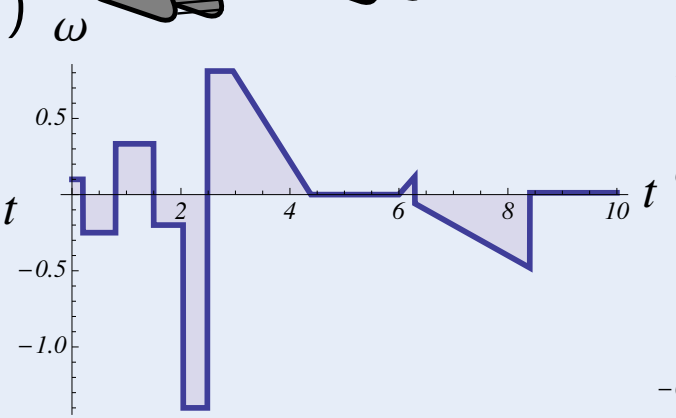
Game rules describing play evolution with

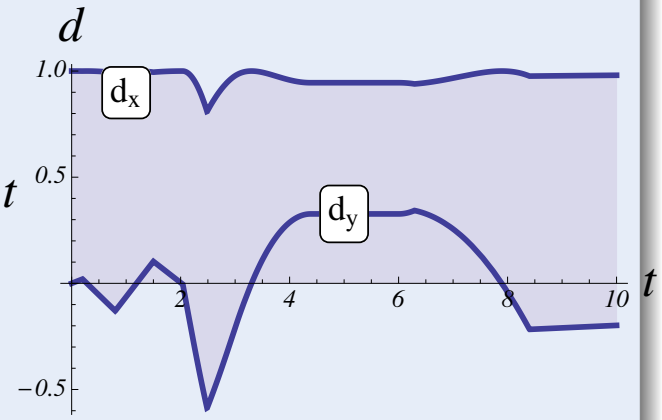
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ◇ vs. Demon □)





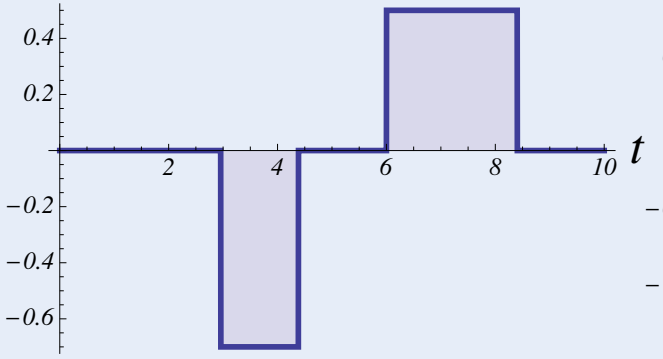



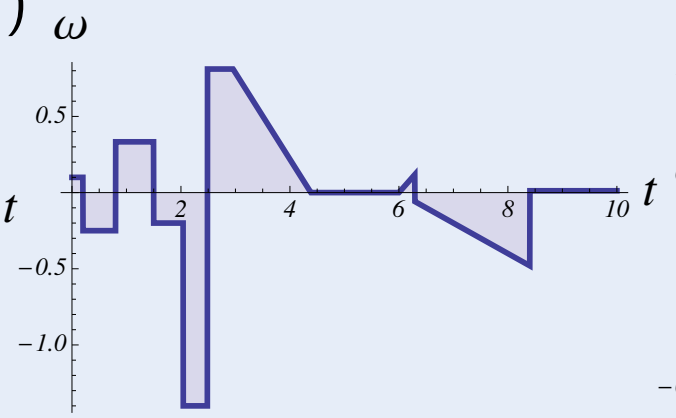



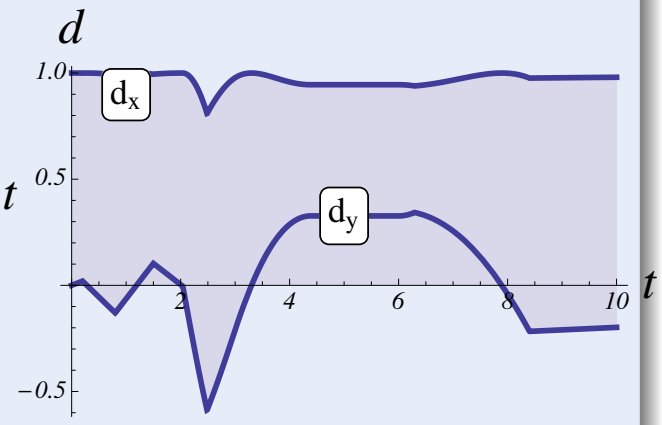
## Challenge (Hybrid Games)

Game rules describing play evolution with

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



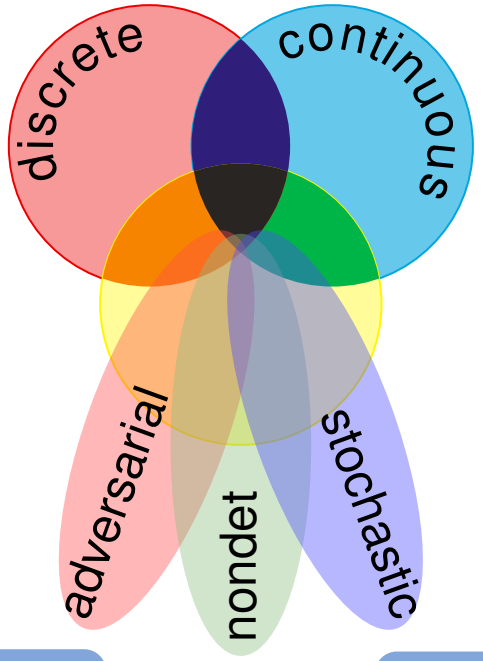



# CPSs are Multi-Dynamical Systems

## CPS Dynamics

CPS are characterized by multiple facets of dynamical systems.




## CPS Compositions

CPS combines multiple simple dynamical effects.

Descriptive simplification

## Tame Parts

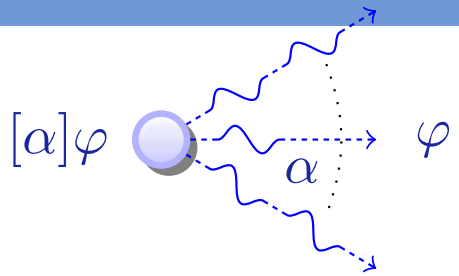
Exploiting compositionality tames CPS complexity.

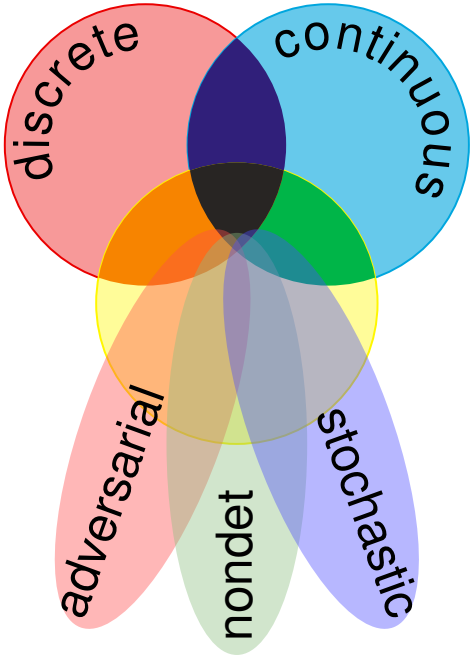
Analytic simplification

# Dynamic Logics for Dynamical Systems

**differential dynamic logic**

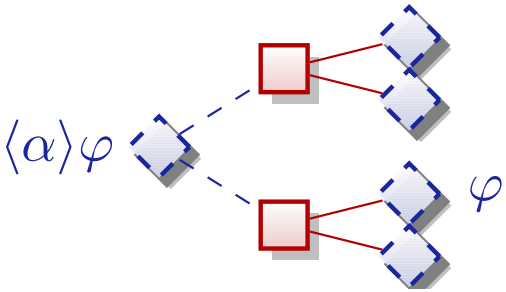
$\mathsf{dL} = \mathsf{DL} + \mathsf{HP}$



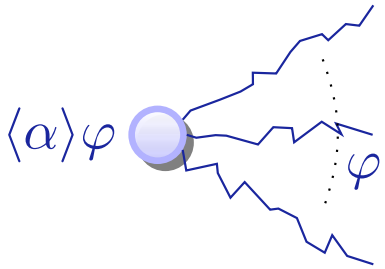

**differential game logic**

$\mathsf{dGL} = \mathsf{GL} + \mathsf{HG}$




**stochastic differential DL**

$\mathsf{Sd}\mathcal{L} = \mathsf{DL} + \mathsf{SHP}$




**quantified differential DL**

$\mathsf{Qd}\mathcal{L} = \mathsf{FOL} + \mathsf{DL} + \mathsf{QHP}$

# Outline

# Differential Dynamic Logic dL: Syntax

**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 $P$)**

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

# Differential Dynamic Logic dL: Syntax

Discrete Assign | Test Condition | Differential Equation | Nondet. Choice | Seq. Compose | Nondet. Repeat

**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 $P$)**

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

All Reals | Some Reals | All Runs | Some Runs

# Differential Dynamic Logic dL: Nondeterminism

Nondet. Choice

**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 $P$)**

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

Nondeterminism during HP runs

# Differential Dynamic Logic dL: Nondeterminism

Differential Equation

Nondet. Choice

Nondet. Repeat

**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 $P$)**

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

Nondeterminism during HP runs

# Differential Dynamic Logic dL: Nondeterminism

Differential Equation

Nondet. Choice

Nondet. Repeat

**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 $P$)**

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

All Choices

Some Choice

# Differential Dynamic Logic dL: Nondeterminism

All choices resolved in one way

Differential Equation

Nondet. Choice

Nondet. Repeat

**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 $P$)**

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

Modality decides the mode: help/hurt

All Choices

Some Choice

# Differential Dynamic Logic dL: Nondeterminism

All choices resolved in one way

Differential Equation

Nondet. Choice

Nondet. Repeat

**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 $P$)**

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

Modality decides the mode: help/hurt

All Choices

Some Choice

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

# Control & Dual Control Operators

**◇ Angel Ops**

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

Let Angel be one player

# Control & Dual Control Operators

**◇ Angel Ops**

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

**□ Demon Ops**

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

Let Angel be one player

Let Demon be another player

# Control & Dual Control Operators

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

$\xrightarrow{\mathsf{d}}$ $\xleftarrow{\mathsf{d}}$

| □ Demon Ops | |
|---|---|
| $\cap$ | choice |
| $\times$ | repeat |
| $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q^{\mathsf{d}}$ | challenge |

Duality operator $^{\mathsf{d}}$ passes control between players

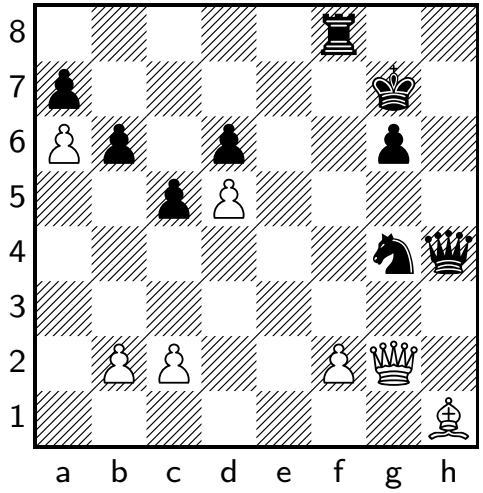
# Game Operators

| ◇ Angel Ops | | ◻ Demon Ops | |
|---|---|---|---|
| $\cup$ | choice | $\cap$ | choice |
| $*$ | repeat | $\times$ | repeat |
| $x' = f(x)$ | evolve | $x' = f(x)^{\mathrm{d}}$ | evolve |
| $?Q$ | challenge | $?Q^{\mathrm{d}}$ | challenge |

d

d

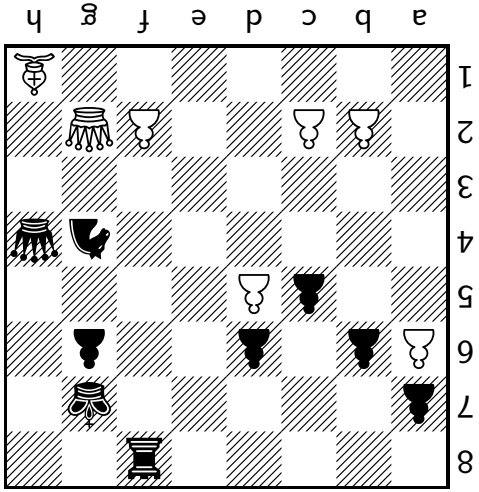
Duality operator $^{\mathrm{d}}$ passes control between players



# Game Operators

| ◇ Angel Ops | | d | □ Demon Ops | |
|---|---|---|---|---|
| $\cup$ | choice | → | $\cap$ | choice |
| $*$ | repeat | | $\times$ | repeat |
| $x' = f(x)$ | evolve | ← | $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q$ | challenge | d | $?Q^{\mathsf{d}}$ | challenge |

Duality operator $^{\mathsf{d}}$ passes control between players



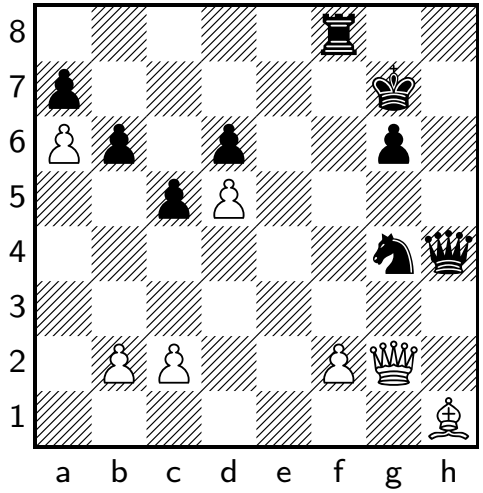
# Game Operators

| ◇ Angel Ops | | ◻ Demon Ops | |
|---|---|---|---|
| $\cup$ | choice | $\cap$ | choice |
| $*$ | repeat | $\times$ | repeat |
| $x' = f(x)$ | evolve | $x' = f(x)^{\mathrm{d}}$ | evolve |
| $?Q$ | challenge | $?Q^{\mathrm{d}}$ | challenge |

d

d

Duality operator $^{\mathrm{d}}$ passes control between players



# Definable Game Operators

| ◇ Angel Ops | | ◻ Demon Ops | |
|---|---|---|---|
| $\cup$ | choice | $\cap$ | choice |
| $*$ | repeat | $\times$ | repeat |
| $x' = f(x)$ | evolve | $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q$ | challenge | $?Q^{\mathsf{d}}$ | challenge |

$\mathsf{d}$ ⟶ ⟵ $\mathsf{d}$

$$\begin{aligned}
\text{if}(Q)\,\alpha\,\text{else}\,\beta &\equiv \\
\text{while}(Q)\,\alpha &\equiv \\
\alpha \cap \beta &\equiv \\
\alpha^{\times} &\equiv \\
(x' = f(x) \,\&\, Q)^{\mathsf{d}} &\quad x' = f(x) \,\&\, Q \\
(x := e)^{\mathsf{d}} &\quad x := e \\
?Q^{\mathsf{d}} &\quad ?Q
\end{aligned}$$

# Definable Game Operators

| ◇ Angel Ops | | □ Demon Ops | |
|---|---|---|---|
| $\cup$ | choice | $\cap$ | choice |
| $*$ | repeat | $\times$ | repeat |
| $x' = f(x)$ | evolve | $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q$ | challenge | $?Q^{\mathsf{d}}$ | challenge |

$\mathsf{d}$ →

← $\mathsf{d}$

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

$$\text{while}(Q)\,\alpha \equiv$$

$$\alpha \cap \beta \equiv$$

$$\alpha^{\times} \equiv$$

$$(x' = f(x)\,\&\,Q)^{\mathsf{d}} \quad x' = f(x)\,\&\,Q$$

$$(x := e)^{\mathsf{d}} \quad x := e$$

$$?Q^{\mathsf{d}} \quad ?Q$$

# Definable Game Operators

| ◇ Angel Ops | | ◻ Demon Ops | |
|---|---|---|---|
| $\cup$ | choice | $\cap$ | choice |
| $*$ | repeat | $\times$ | repeat |
| $x' = f(x)$ | evolve | $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q$ | challenge | $?Q^{\mathsf{d}}$ | challenge |

$\mathsf{d}$ ⇄ $\mathsf{d}$

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

$$\text{while}(Q)\,\alpha \equiv (?Q;\alpha)^*;?\neg Q$$

$$\alpha \cap \beta \equiv$$

$$\alpha^\times \equiv$$

$$(x' = f(x) \,\&\, Q)^{\mathsf{d}} \quad x' = f(x) \,\&\, Q$$

$$(x := e)^{\mathsf{d}} \quad x := e$$

$$?Q^{\mathsf{d}} \quad ?Q$$

# Definable Game Operators

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

d

d

| □ Demon Ops | |
|---|---|
| $\cap$ | choice |
| $\times$ | repeat |
| $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q^{\mathsf{d}}$ | challenge |

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

$$\text{while}(Q)\,\alpha \equiv (?Q;\alpha)^*;?\neg Q$$

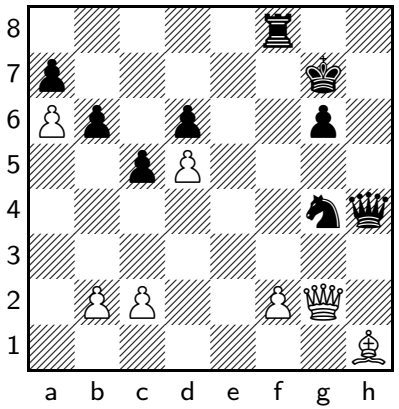
$$\alpha \cap \beta \equiv$$

$$\alpha^\times \equiv$$

$$(x' = f(x) \,\&\, Q)^{\mathsf{d}} \quad x' = f(x) \,\&\, Q$$

$$(x := e)^{\mathsf{d}} \quad x := e$$

$$?Q^{\mathsf{d}} \quad ?Q$$



# Definable Game Operators

| ◇ Angel Ops | | ◻ Demon Ops | |
|---|---|---|---|
| $\cup$ | choice | $\cap$ | choice |
| $*$ | repeat | $\times$ | repeat |
| $x' = f(x)$ | evolve | $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q$ | challenge | $?Q^{\mathsf{d}}$ | challenge |

$\mathsf{d}$ $\rightarrow$

$\leftarrow$ $\mathsf{d}$

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

$$\text{while}(Q)\,\alpha \equiv (?Q;\alpha)^*;?\neg Q$$

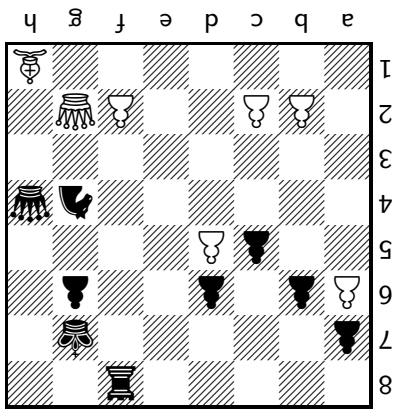
$$\alpha \cap \beta \equiv (\alpha^{\mathsf{d}} \cup \beta^{\mathsf{d}})^{\mathsf{d}}$$

$$\alpha^{\times} \equiv$$

$$(x' = f(x) \,\&\, Q)^{\mathsf{d}} \qquad x' = f(x) \,\&\, Q$$

$$(x := e)^{\mathsf{d}} \qquad x := e$$

$$?Q^{\mathsf{d}} \qquad ?Q$$



# Definable Game Operators

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

$\xrightarrow{\mathsf{d}}$ $\xleftarrow{\mathsf{d}}$

| □ Demon Ops | |
|---|---|
| $\cap$ | choice |
| $\times$ | repeat |
| $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q^{\mathsf{d}}$ | challenge |

$$\mathsf{if}(Q)\,\alpha\,\mathsf{else}\,\beta \equiv (?Q;\alpha) \cup (?\neg Q;\beta)$$

$$\mathsf{while}(Q)\,\alpha \equiv (?Q;\alpha)^*;?\neg Q$$

$$\alpha \cap \beta \equiv (\alpha^{\mathsf{d}} \cup \beta^{\mathsf{d}})^{\mathsf{d}}$$

$$\alpha^{\times} \equiv ((\alpha^{\mathsf{d}})^*)^{\mathsf{d}}$$

$$(x' = f(x) \,\&\, Q)^{\mathsf{d}} \quad x' = f(x) \,\&\, Q$$

$$(x := e)^{\mathsf{d}} \quad x := e$$

$$?Q^{\mathsf{d}} \quad ?Q$$

# Definable Game Operators

| ◇ Angel Ops | | ◻ Demon Ops | |
|---|---|---|---|
| $\cup$ | choice | $\cap$ | choice |
| $*$ | repeat | $\times$ | repeat |
| $x' = f(x)$ | evolve | $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q$ | challenge | $?Q^{\mathsf{d}}$ | challenge |

$\mathsf{d}$ →

← $\mathsf{d}$

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

$$\text{while}(Q)\,\alpha \equiv (?Q;\alpha)^*;?\neg Q$$

$$\alpha \cap \beta \equiv (\alpha^{\mathsf{d}} \cup \beta^{\mathsf{d}})^{\mathsf{d}}$$

$$\alpha^{\times} \equiv ((\alpha^{\mathsf{d}})^*)^{\mathsf{d}}$$

$$(x' = f(x)\,\&\,Q)^{\mathsf{d}} \not\equiv x' = f(x)\,\&\,Q$$

$$(x := e)^{\mathsf{d}} \quad x := e$$

$$?Q^{\mathsf{d}} \quad ?Q$$

# Definable Game Operators

| ◇ Angel Ops | | ◻ Demon Ops | |
|---|---|---|---|
| $\cup$ | choice | $\cap$ | choice |
| $*$ | repeat | $\times$ | repeat |
| $x' = f(x)$ | evolve | $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q$ | challenge | $?Q^{\mathsf{d}}$ | challenge |

$\mathsf{d}$ ⟶ ⟵ $\mathsf{d}$

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

$$\text{while}(Q)\,\alpha \equiv (?Q;\alpha)^*;?\neg Q$$

$$\alpha \cap \beta \equiv (\alpha^{\mathsf{d}} \cup \beta^{\mathsf{d}})^{\mathsf{d}}$$

$$\alpha^{\times} \equiv ((\alpha^{\mathsf{d}})^*)^{\mathsf{d}}$$

$$(x' = f(x) \,\&\, Q)^{\mathsf{d}} \not\equiv x' = f(x) \,\&\, Q$$

$$(x := e)^{\mathsf{d}} \equiv x := e$$

$$?Q^{\mathsf{d}} \quad ?Q$$

# Definable Game Operators

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

d

d

| □ Demon Ops | |
|---|---|
| $\cap$ | choice |
| $\times$ | repeat |
| $x' = f(x)^{\mathsf{d}}$ | evolve |
| $?Q^{\mathsf{d}}$ | challenge |

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

$$\text{while}(Q)\,\alpha \equiv (?Q;\alpha)^*;?\neg Q$$

$$\alpha \cap \beta \equiv (\alpha^{\mathsf{d}} \cup \beta^{\mathsf{d}})^{\mathsf{d}}$$

$$\alpha^{\times} \equiv ((\alpha^{\mathsf{d}})^*)^{\mathsf{d}}$$

$$(x' = f(x) \,\&\, Q)^{\mathsf{d}} \not\equiv x' = f(x) \,\&\, Q$$

$$(x := e)^{\mathsf{d}} \equiv x := e$$

$$?Q^{\mathsf{d}} \not\equiv ?Q$$

# Outline

# Hybrid Games: Syntax

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

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

# Hybrid Games: Syntax

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

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

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

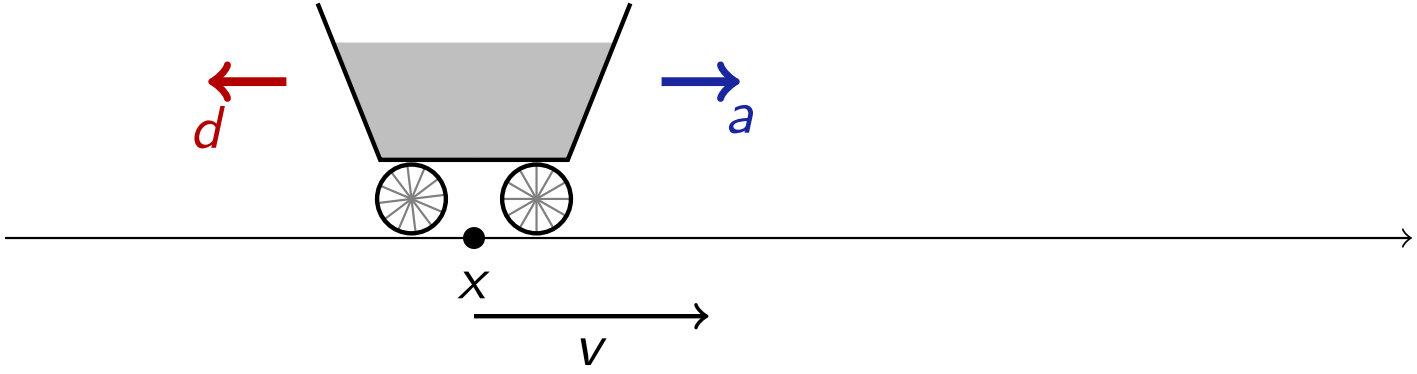
# Hybrid Games: Syntax

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

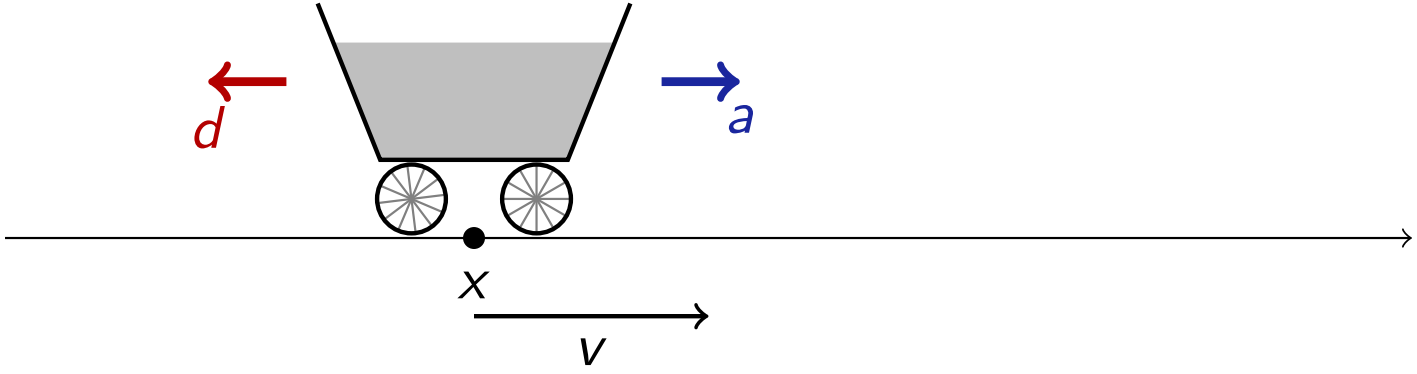
**Definition (Hybrid game $\alpha$)**

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

# Example: Push-around Cart

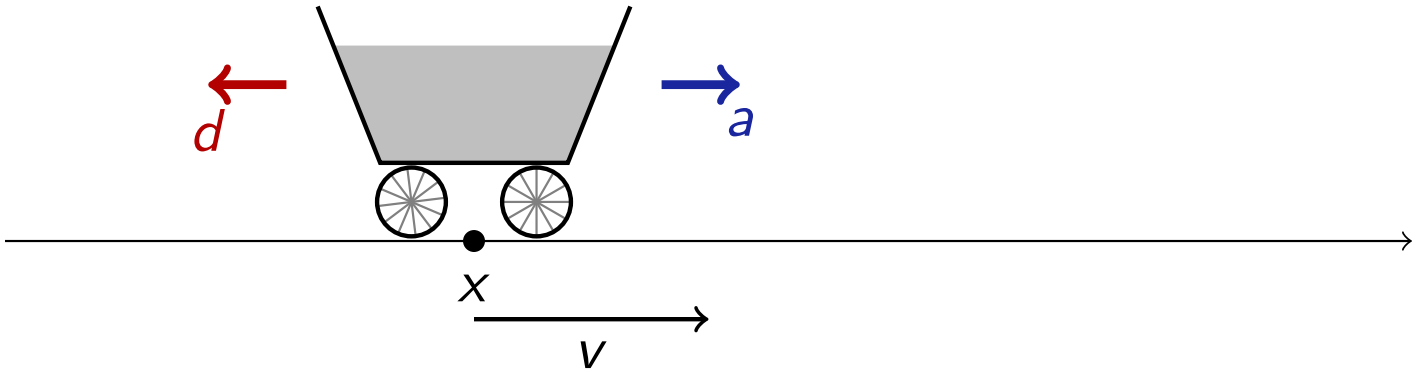


# Example: Push-around Cart




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

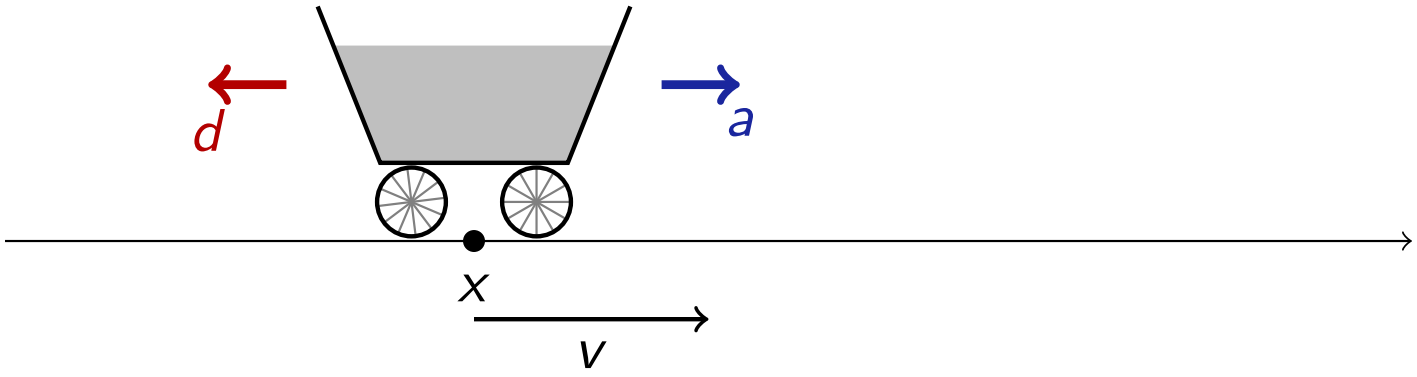
# Example: Push-around Cart




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

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

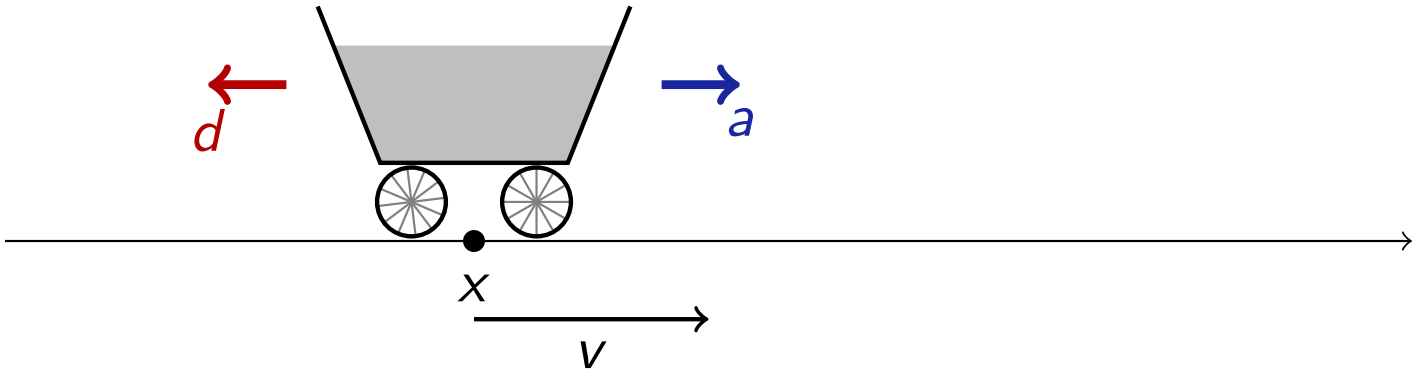
# Example: Push-around Cart




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

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

# Example: Push-around Cart

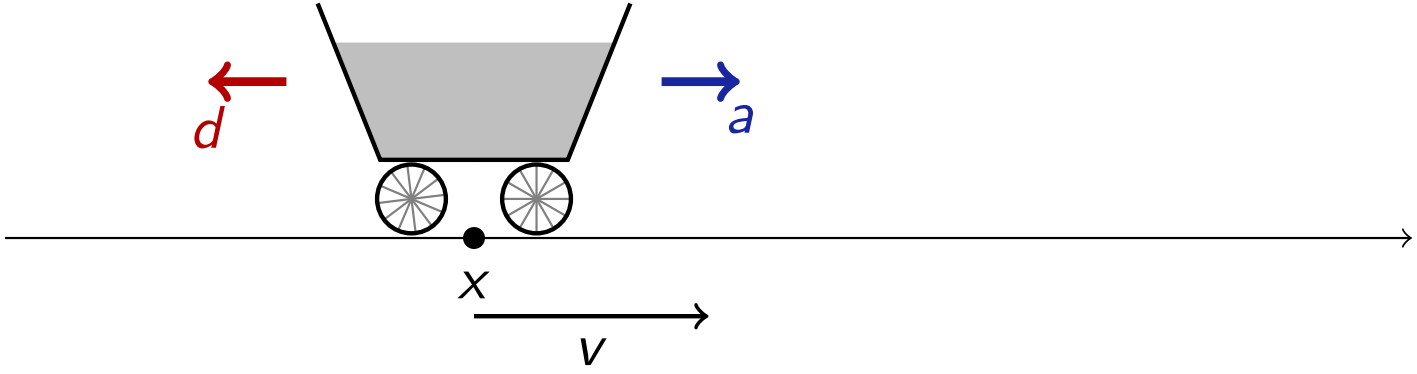



$$\big((a := 1 \cup a := -1); (d := 1 \cap d := -1); \{x' = v, v' = a + d\}\big)^*$$

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

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

# Example: Push-around Cart




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

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

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

Hybrid systems can't say that $a$ is Angel's choice and $d$ is Demon's

# Differential Game Logic: Syntax

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

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

# Differential Game Logic: Syntax

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

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

**Definition (dGL Formula $P$)**

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

# Differential Game Logic: Syntax

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

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

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

**Definition (dGL Formula $P$)**

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

All Reals | Some Reals

# Differential Game Logic: Syntax

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

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

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

**Definition (dGL Formula $P$)**

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

All Reals | Some Reals

# Differential Game Logic: Syntax

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

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

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

**Definition (dGL Formula $P$)**

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

All Reals | Some Reals | Angel Wins

# Differential Game Logic: Syntax

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

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

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

**Definition (dGL Formula $P$)**

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

All Reals | Some Reals | Angel Wins | Demon Wins

# Simple Examples

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

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

# Simple Examples

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

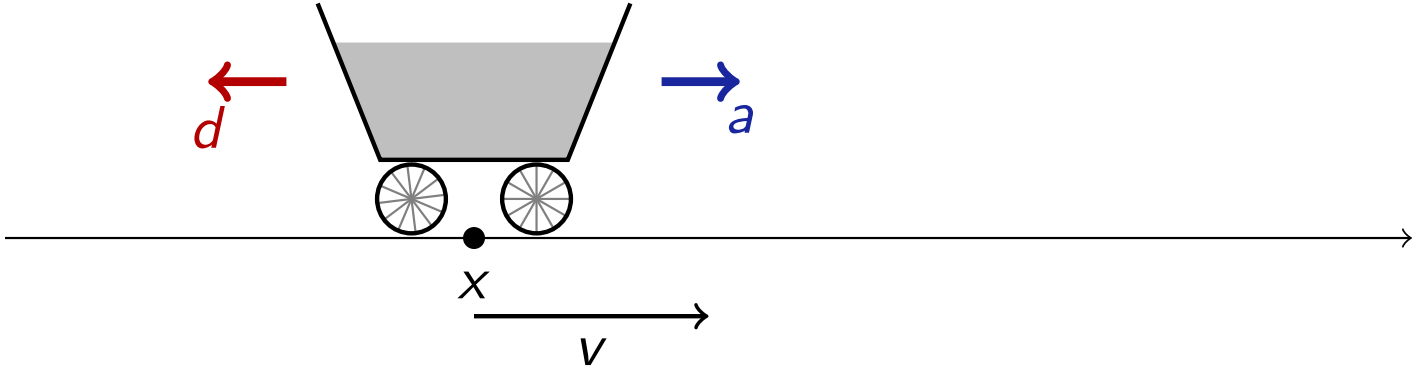
$\langle (x := x + 1; (x' = 1)^{\mathsf{d}} \cup (x := x - 1 \cap x := x - 2))^* \rangle (0 \leq x < 1)$

# Simple Examples

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

$$\not\models \langle (x := x + 1; (x' = 1)^{\mathsf{d}} \cup (x := x - 1 \cap x := x - 2))^* \rangle (0 \leq x < 1)$$

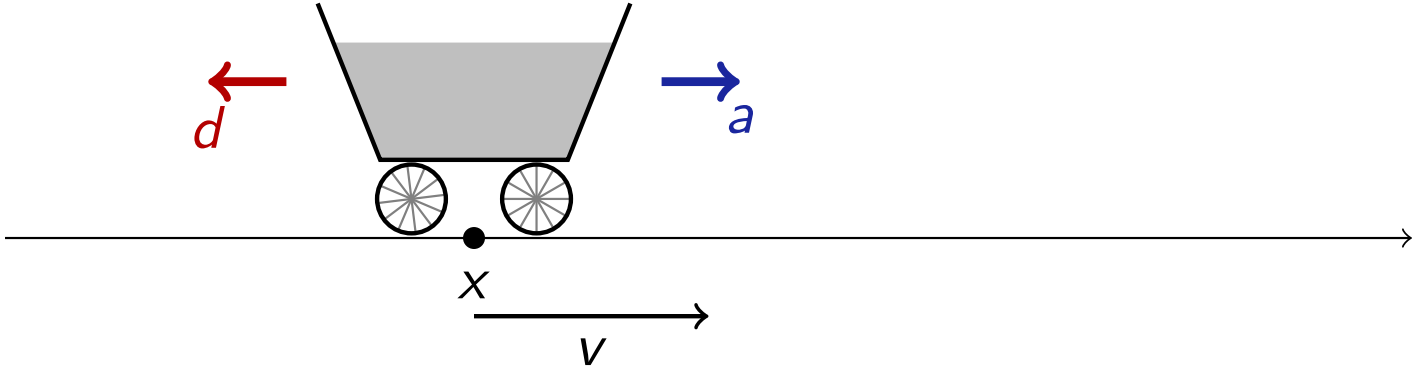
# Example: Push-around Cart




$$v \geq 1 \rightarrow$$

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

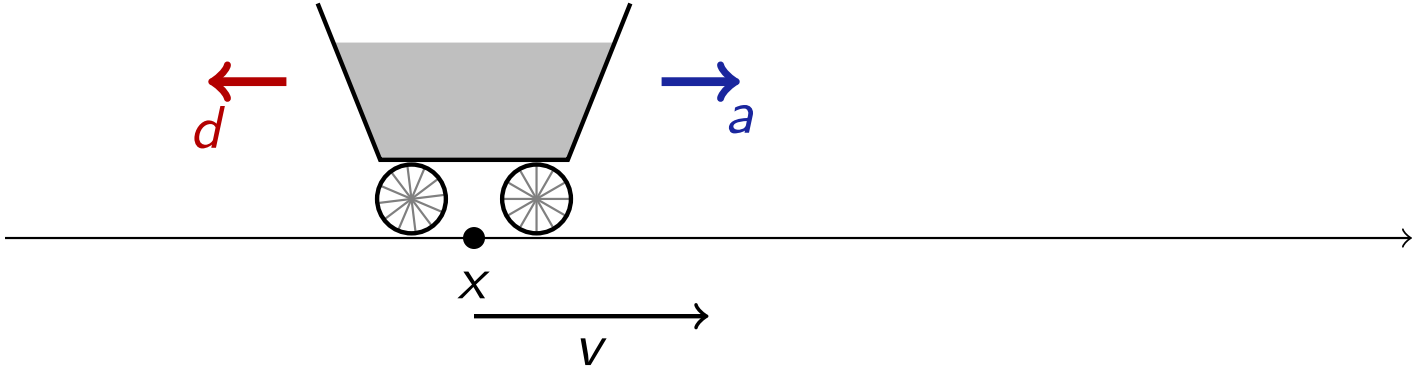
# Example: Push-around Cart




$$\models v \geq 1 \rightarrow$$

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

# Example: Push-around Cart



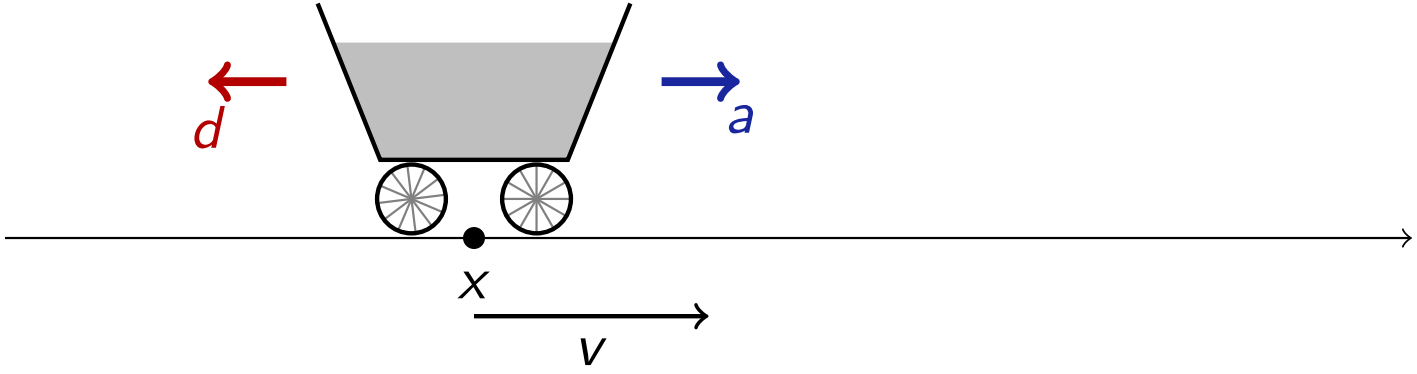

$\models v \geq 1 \rightarrow$ $\quad$ $d$ before $a$ can compensate

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

$x \geq 0 \wedge v \geq 0 \rightarrow$

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

# Example: Push-around Cart



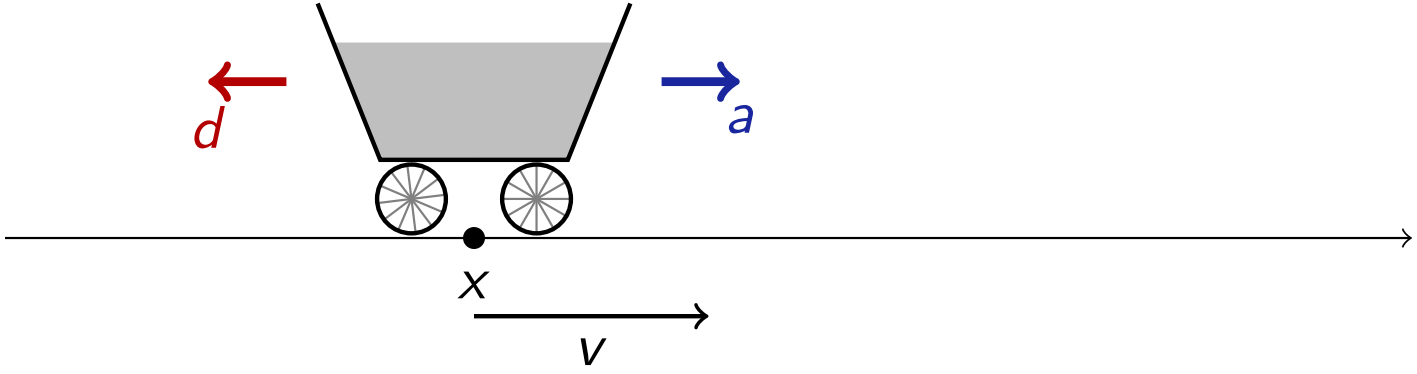

$\models v \geq 1 \rightarrow$ $d$ before $a$ can compensate

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

$\models x \geq 0 \wedge v \geq 0 \rightarrow$ $d$ before $a$ can compensate

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

# Example: Push-around Cart



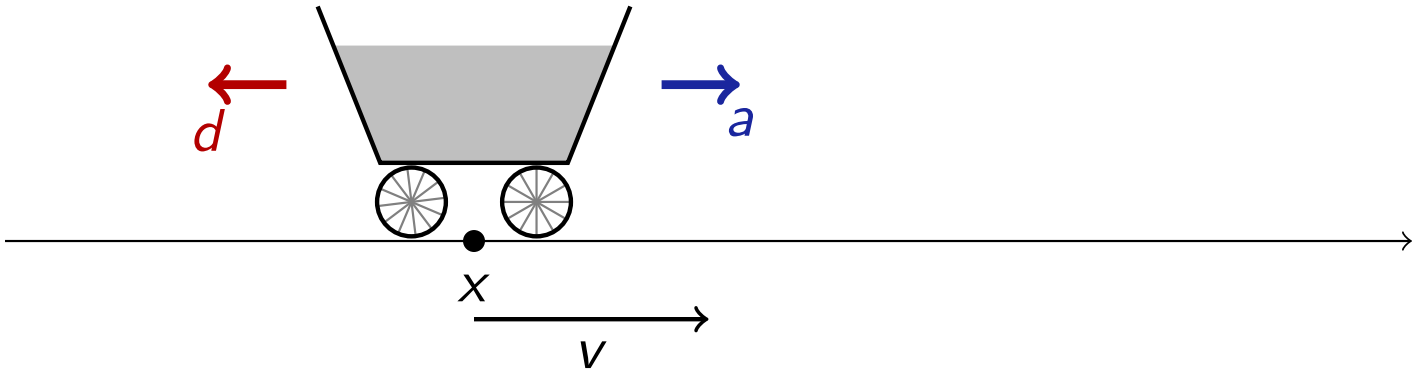

$\models v \geq 1 \rightarrow$ $\qquad$ $d$ before $a$ can compensate

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

$x \geq 0 \qquad \rightarrow$

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

# Example: Push-around Cart



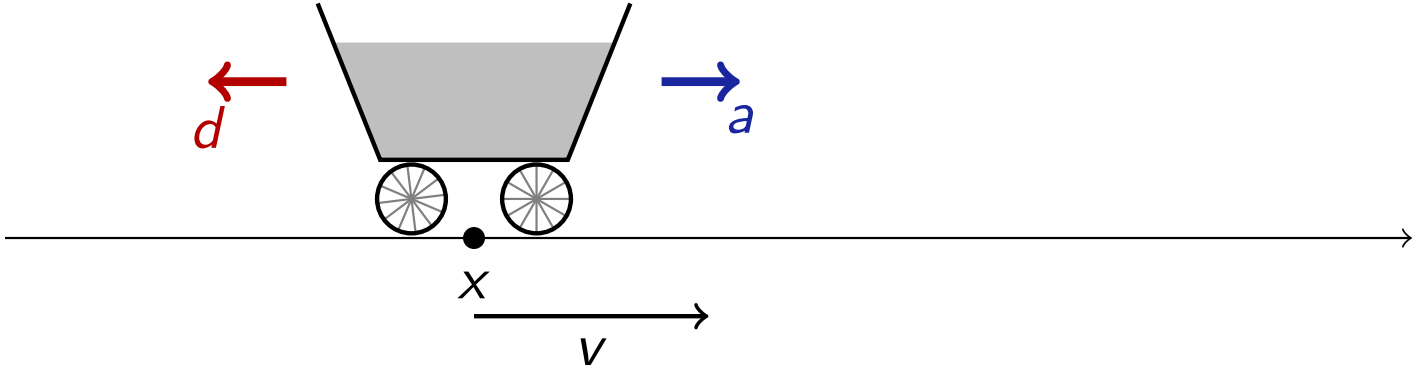

$\models v \geq 1 \rightarrow$ $\quad$ $d$ before $a$ can compensate

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

$\models x \geq 0 \quad \rightarrow$ $\quad$ boring by skip

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

# Example: Push-around Cart

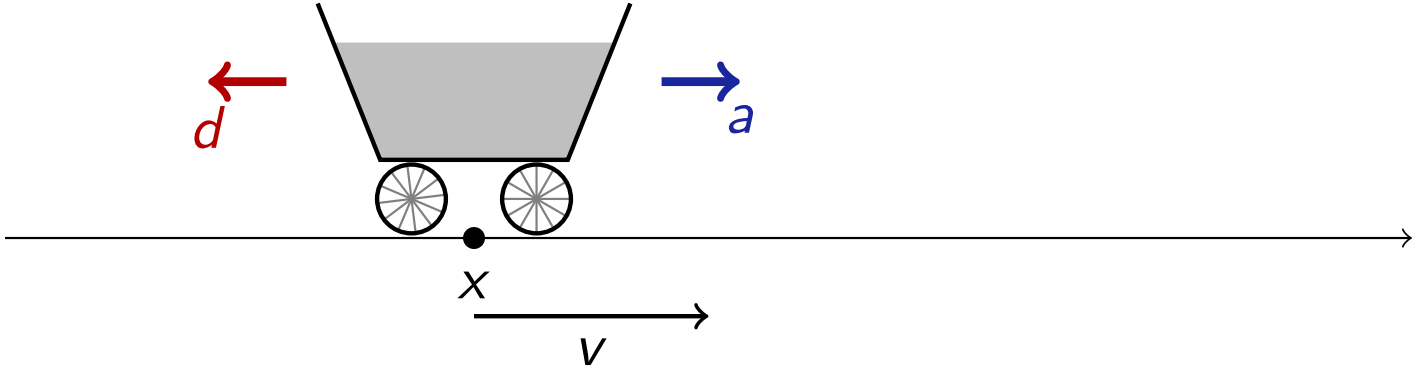



$\models v \geq 1 \rightarrow$ $d$ before $a$ can compensate

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

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

# Example: Push-around Cart



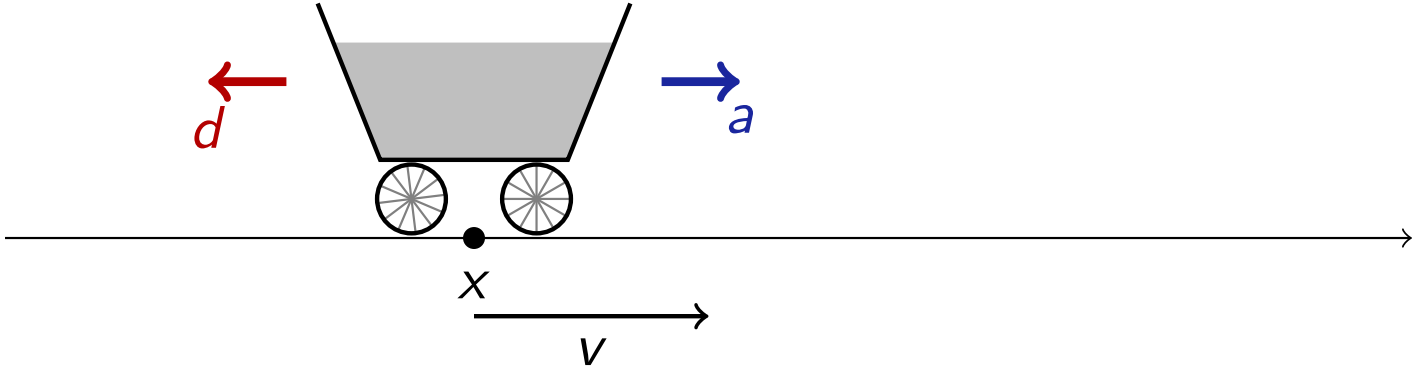

$\models v \geq 1 \rightarrow$ $\quad$ $d$ before $a$ can compensate

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

$\not\models$ $\quad$ counterstrategy $d := -1$

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

# Example: Push-around Cart




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

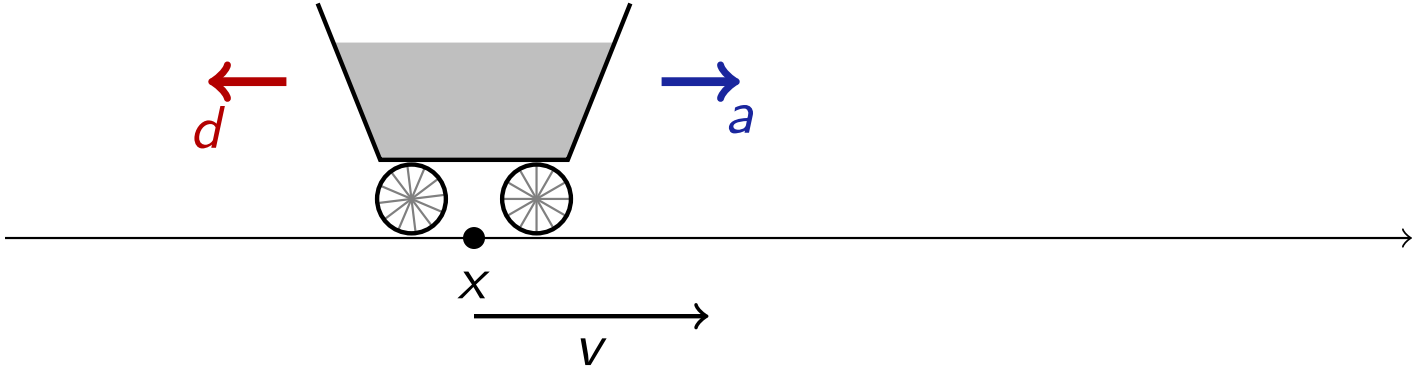
$$\left[\left((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\right)^*\right] v \geq 0$$

$\not\models$ $\quad$ counterstrategy $d := -1$

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

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

# Example: Push-around Cart




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

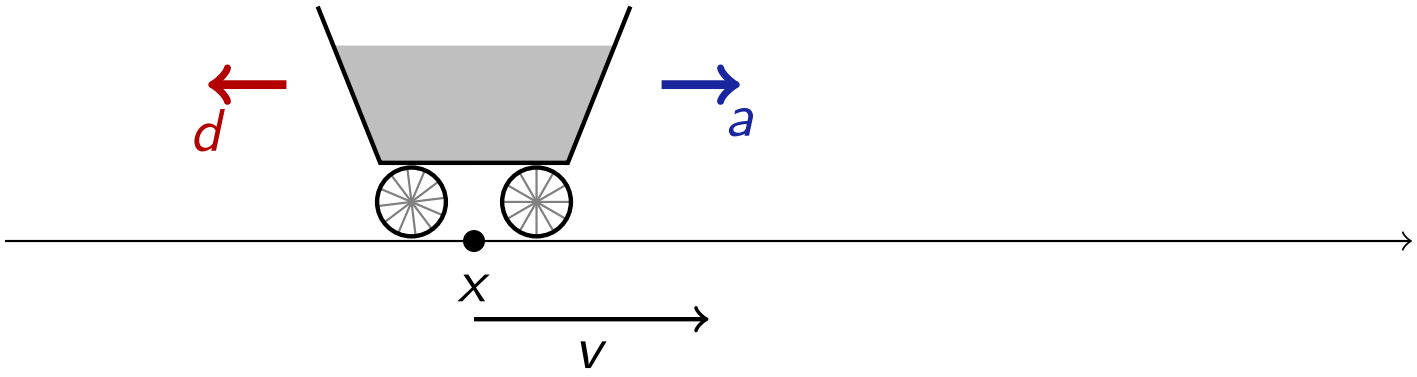
$$\left[\left((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\right)^*\right] v \geq 0$$

$\not\models$ $\quad$ counterstrategy $d := -1$

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

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

# Example: Push-around Cart




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

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

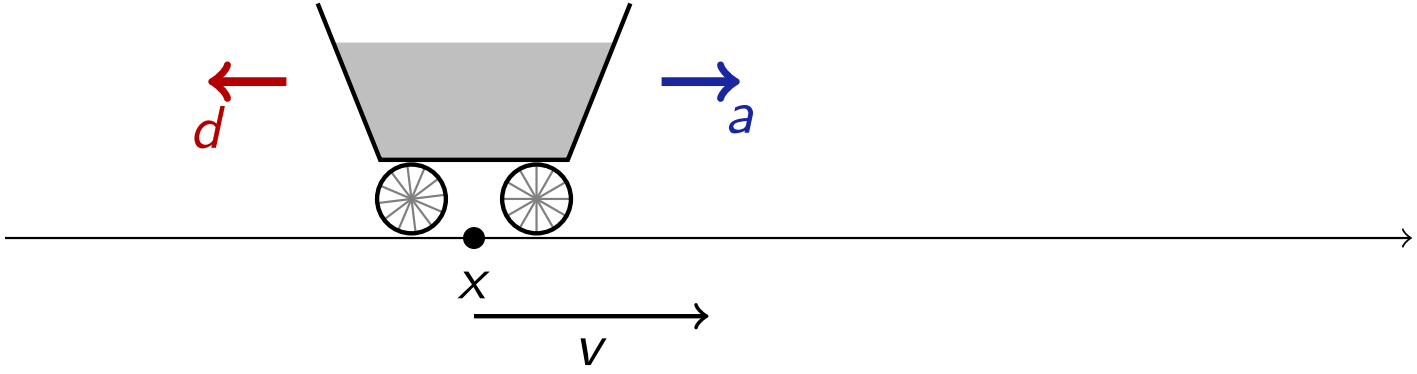
$\not\models$ $\quad$ counterstrategy $d := -1$

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

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

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

# Example: Push-around Cart




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

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

$\not\models$ $\quad$ counterstrategy $d := -1$

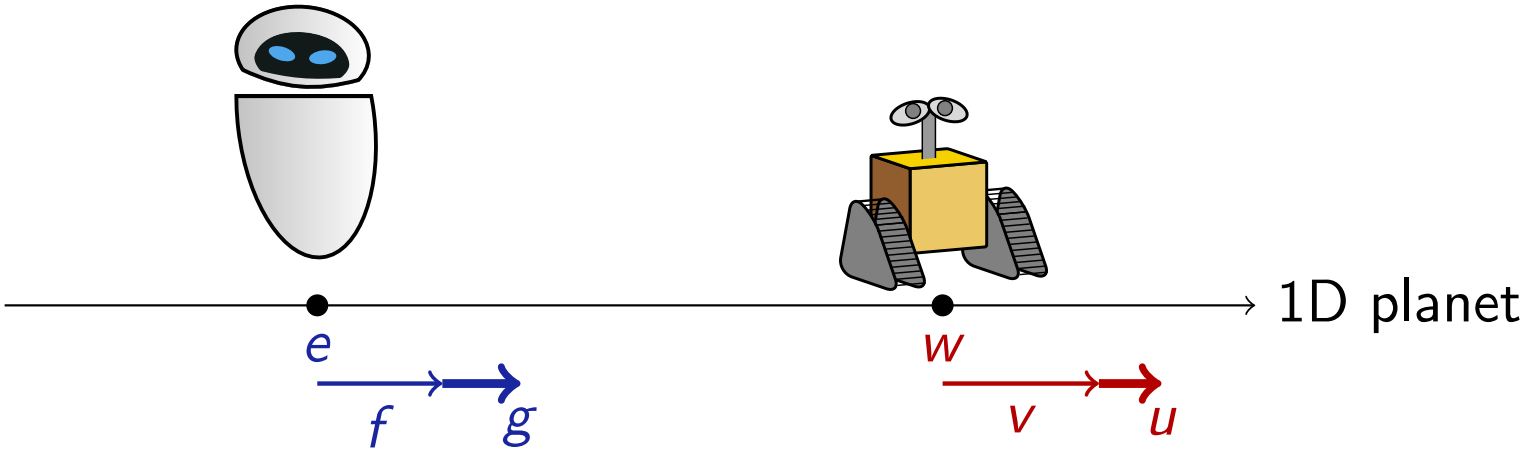
$$\left\langle\left((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\}\right)^*\right\rangle x \geq 0$$

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

$\models \left\langle\left((d := 2 \cap d := -2);\ (a := 2 \cup a := -2);\right.\right.$ $\quad$ $a := d$ then $a := 2 \operatorname{sign} v$

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

# Example: WALL•E and EVE Robot Dance

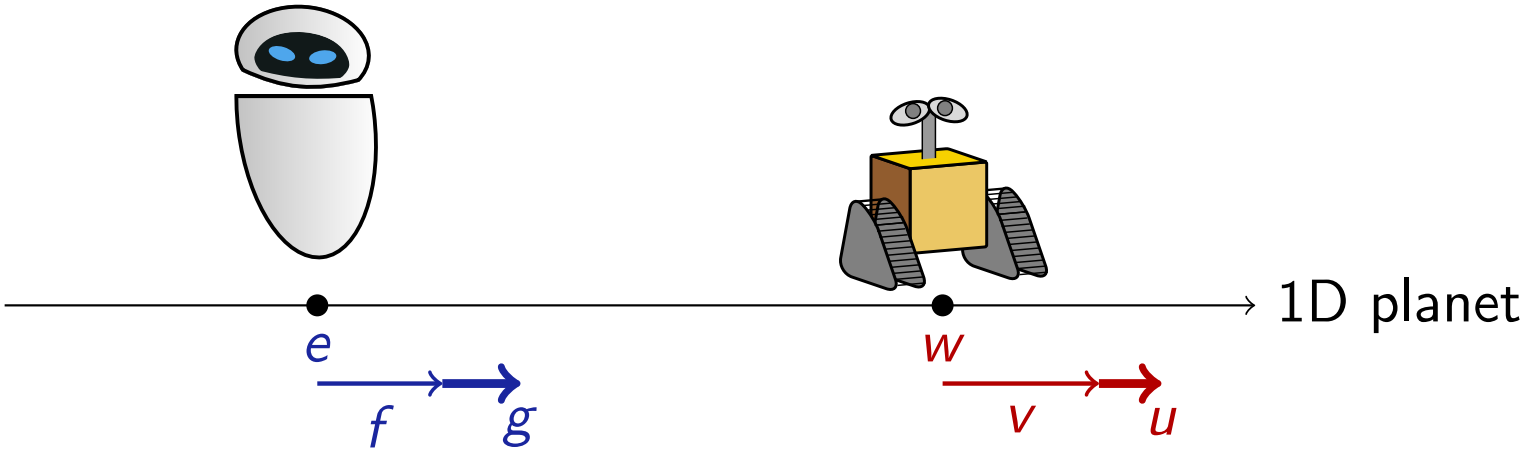



$$(w-e)^2 \leq 1 \wedge v = f \rightarrow$$
$$\langle ((u := 1 \cap u := -1);$$
$$(g := 1 \cup g := -1);$$
$$t := 0;$$
$$\{w' = v, v' = u, e' = f, f' = g, t' = 1 \,\&\, t \leq 1\}^{\mathrm{d}}$$
$$)^{\times} \rangle\, (w-e)^2 \leq 1$$

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

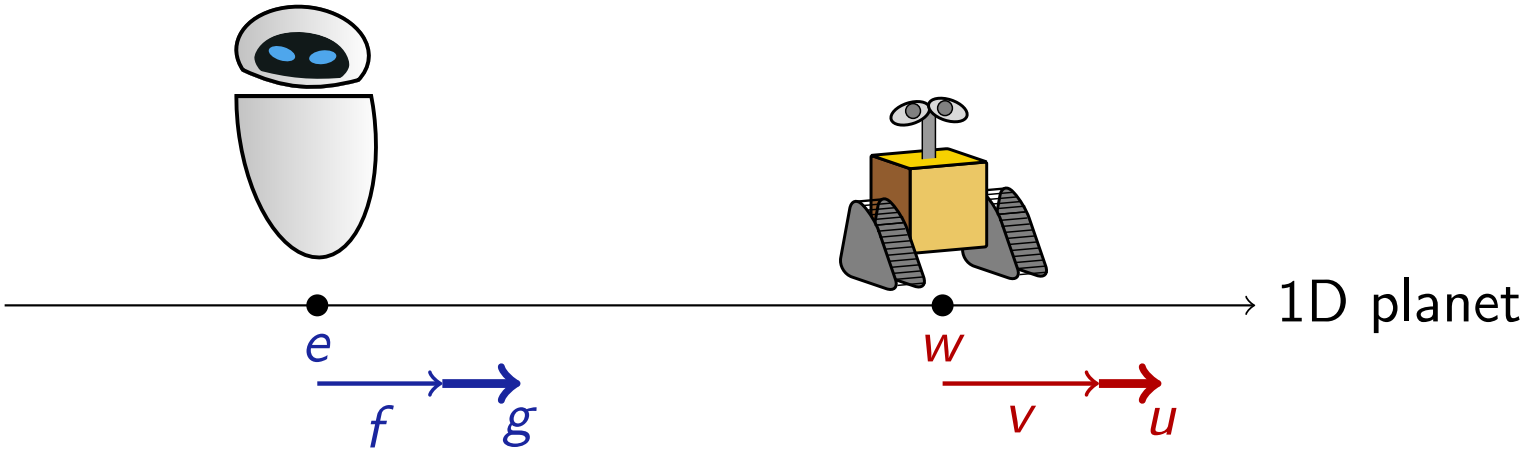



$$(w-e)^2 \leq 1 \wedge v = f \rightarrow$$
$$\langle ((u := 1 \cap u := -1);$$
$$(g := 1 \cup g := -1);$$
$$t := 0;$$
$$\{w' = v, v' = u, e' = f, f' = g, t' = 1 \,\&\, t \leq 1\}^{\mathrm{d}}$$
$$)^{\times}\rangle\ (w-e)^2 \leq 1$$

EVE at $e$ plays Angel's part controlling $g$

WALL•E at $w$ plays Demon's part controlling $u$ and world time

# Example: WALL•E and EVE

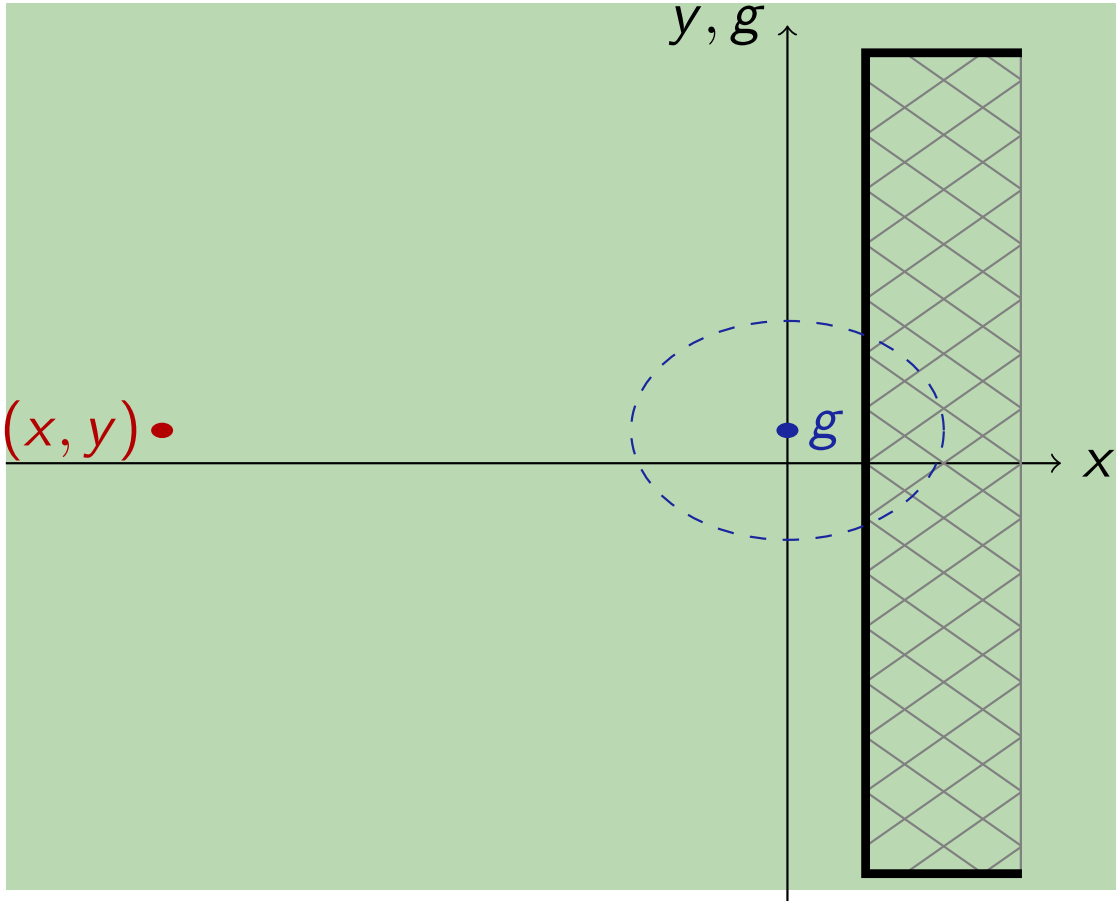



$$(w-e)^2 \leq 1 \wedge v = f \rightarrow$$
$$[((u := 1 \cap u := -1);$$
$$(g := 1 \cup g := -1);$$
$$t := 0;$$
$$\{w' = v, v' = u, e' = f, f' = g, t' = 1 \,\&\, t \leq 1\}$$
$$)^{\times}]\ (w-e)^2 > 1$$

WALL•E at $w$ plays Demon's part controlling $u$ and world time

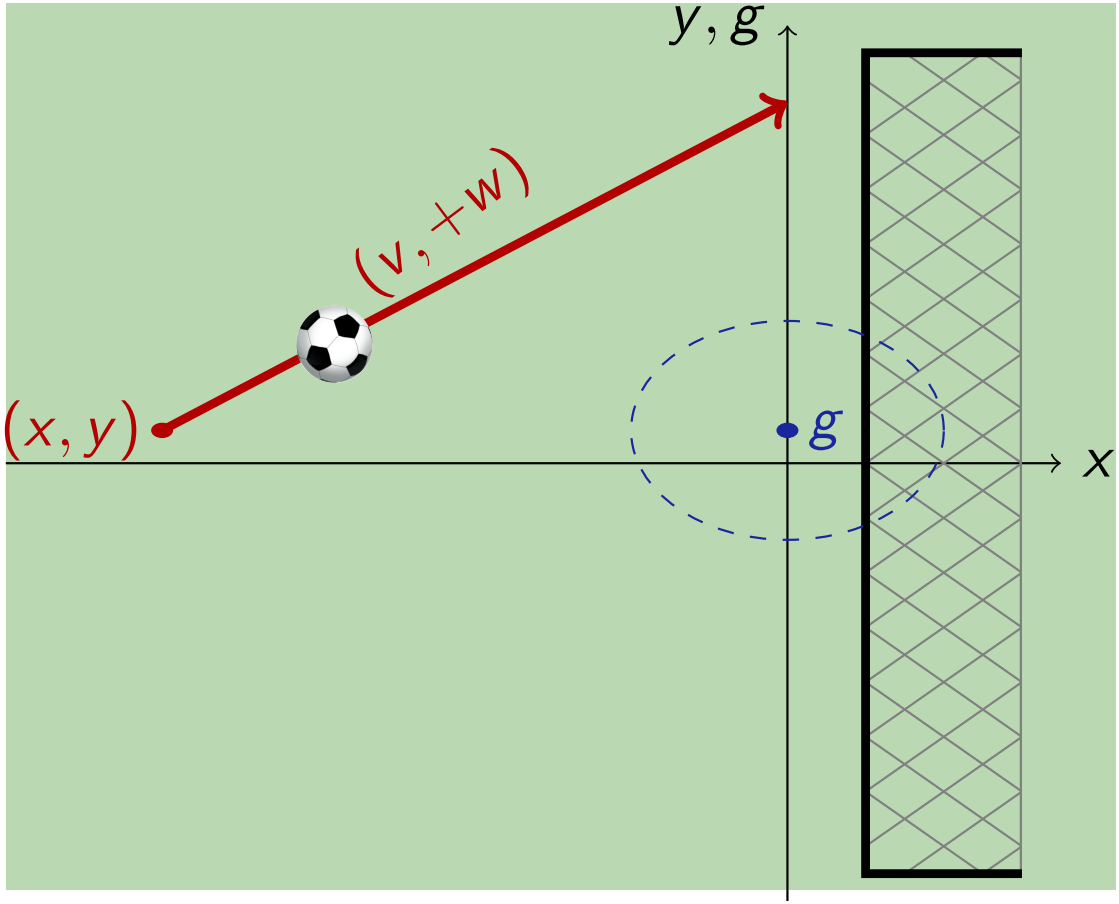
EVE at $e$ plays Angel's part controlling $g$

# Example: Goalie in Robot Soccer



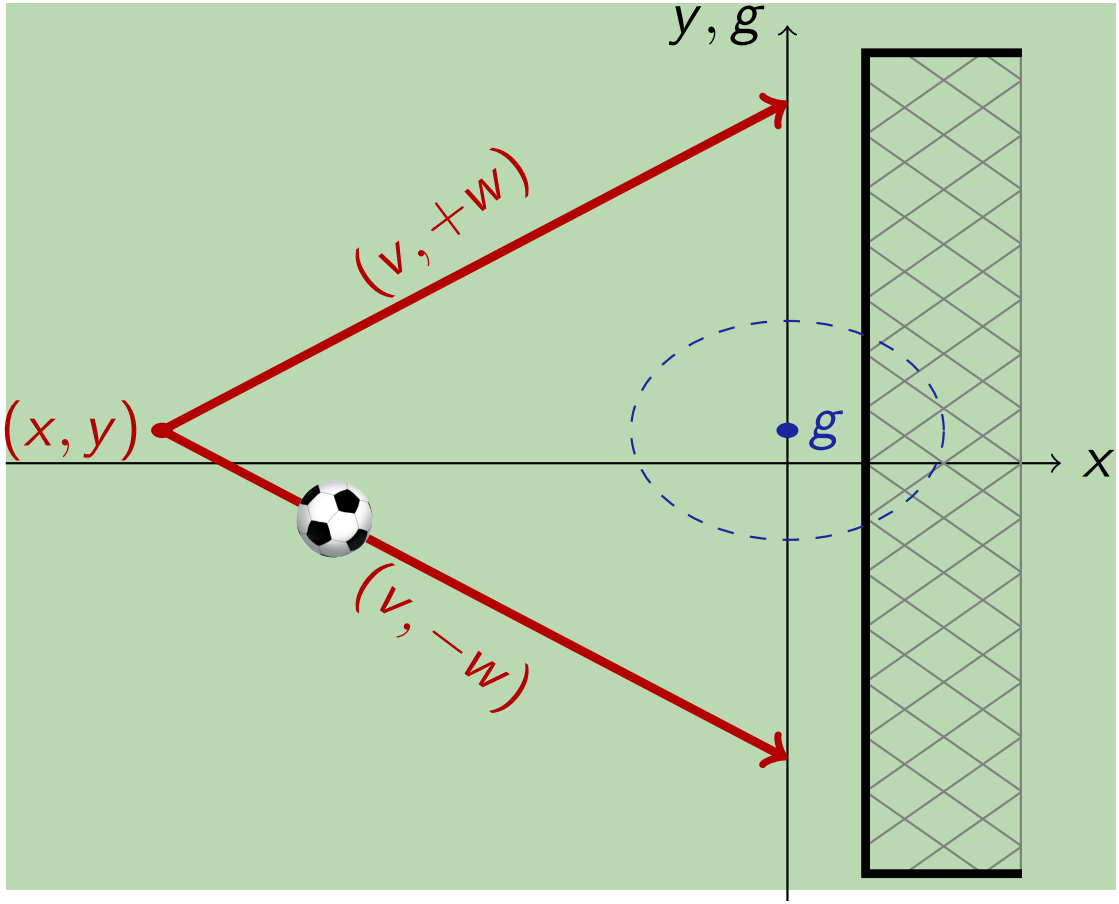

$$x < 0 \wedge v > 0 \wedge y = g \rightarrow$$
$$\langle (w := +w \cap w := -w);$$
$$((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \rangle\, x^2 + (y - g)^2 \leq 1$$

# Example: Goalie in Robot Soccer



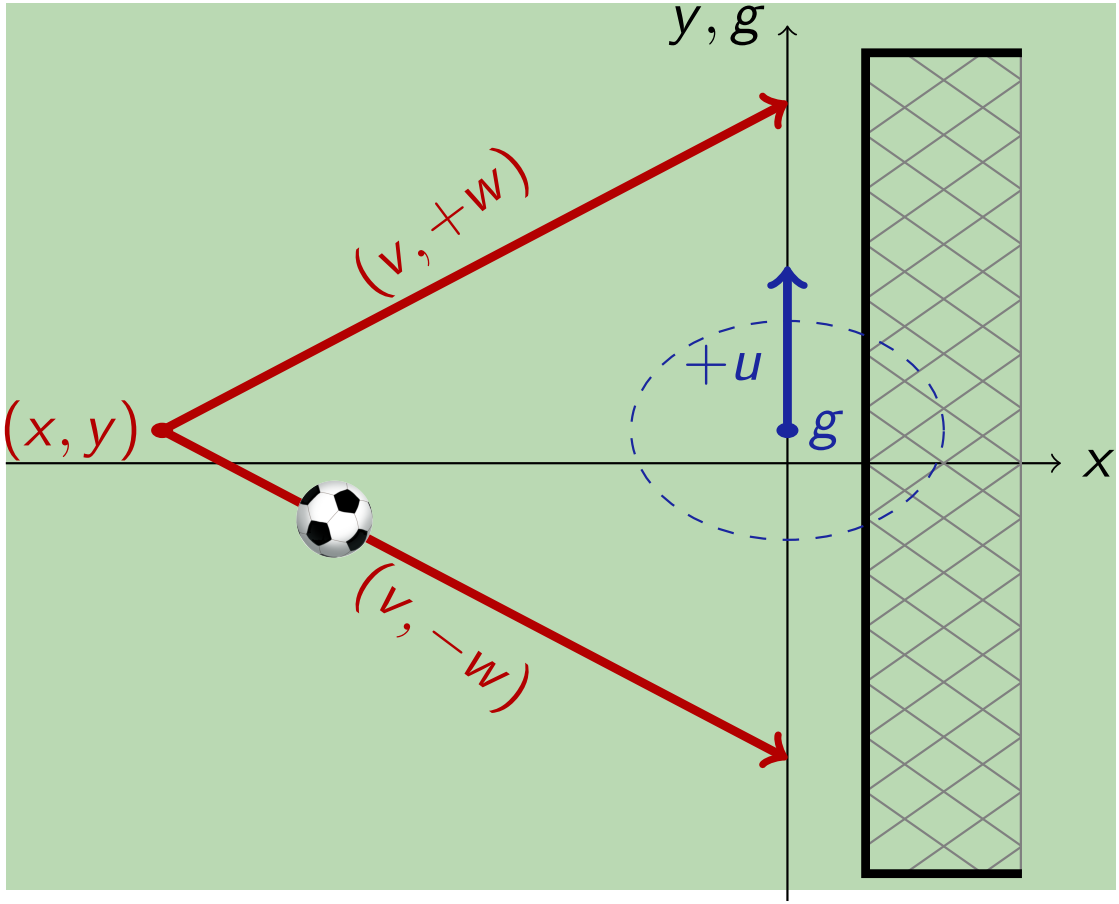

$$x < 0 \wedge v > 0 \wedge y = g \rightarrow$$
$$\langle (w := +w \cap w := -w);$$
$$((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \rangle\, x^2 + (y - g)^2 \leq 1$$

# Example: Goalie in Robot Soccer



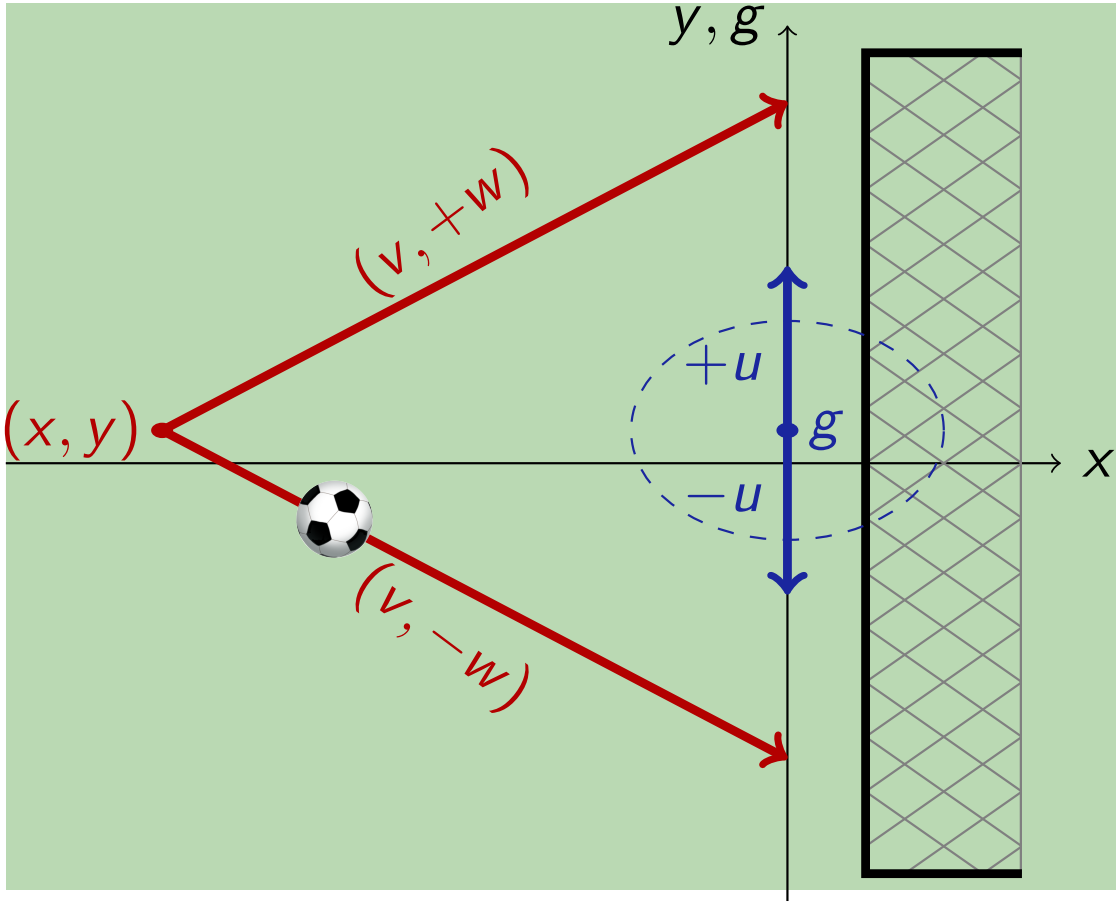

$$x < 0 \wedge v > 0 \wedge y = g \rightarrow$$
$$\langle (w := +w \cap w := -w);$$
$$((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \rangle\, x^2 + (y - g)^2 \leq 1$$

# Example: Goalie in Robot Soccer



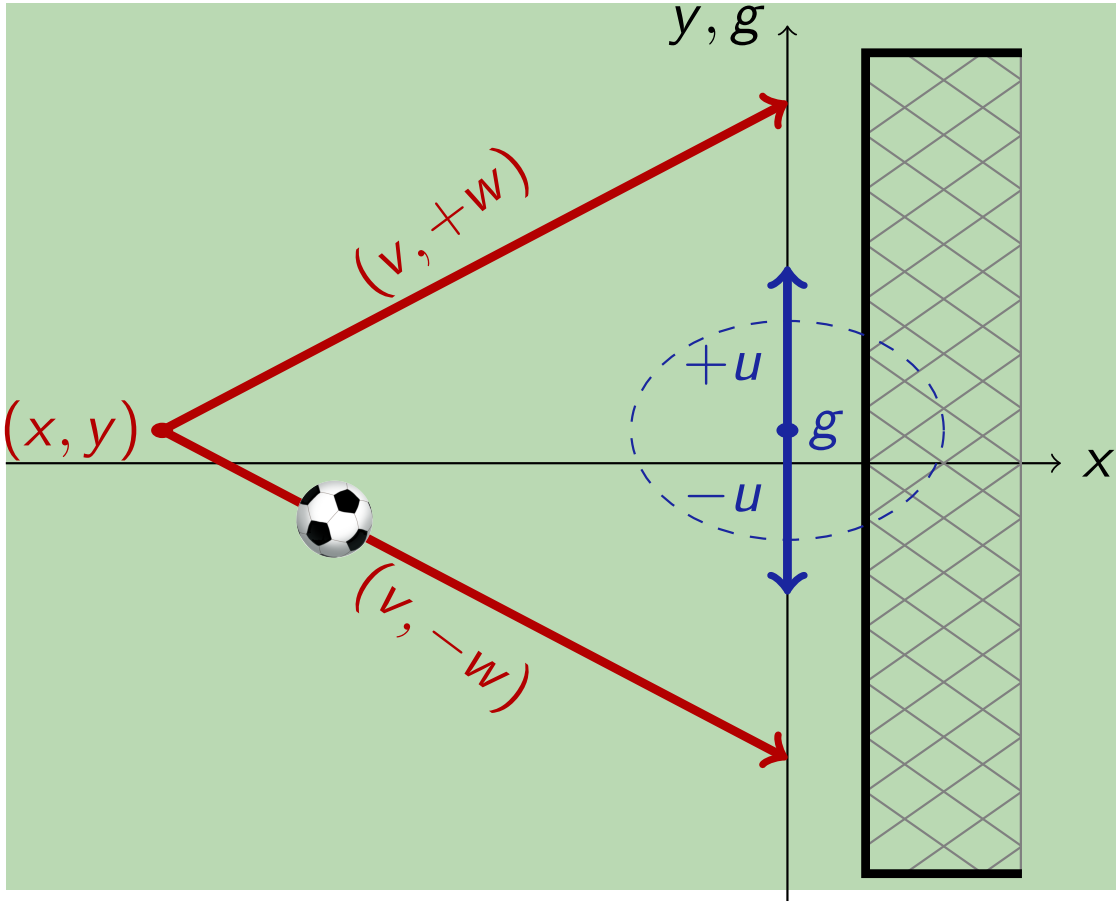

$$x < 0 \wedge v > 0 \wedge y = g \rightarrow$$
$$\langle (w := +w \cap w := -w);$$
$$((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \rangle\, x^2 + (y - g)^2 \leq 1$$

# Example: Goalie in Robot Soccer




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

# Example: Goalie in Robot Soccer




Goalie's Secret

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

$$x < 0 \wedge v > 0 \wedge y = g \rightarrow$$

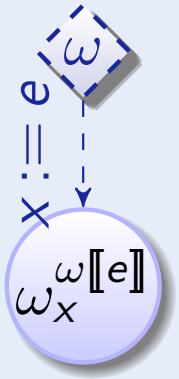
$$\langle (w := +w \cap w := -w);$$

$$((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \rangle\, x^2 + (y-g)^2 \leq 1$$

# Outline

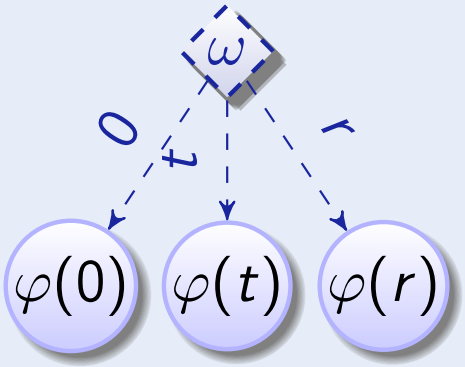
## Definition (Hybrid game $\alpha$: operational semantics)

$x := e$

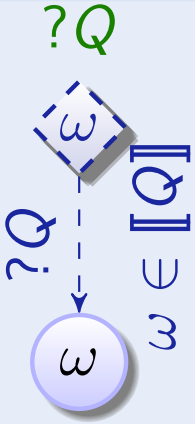


## Definition (Hybrid game $\alpha$: operational semantics)

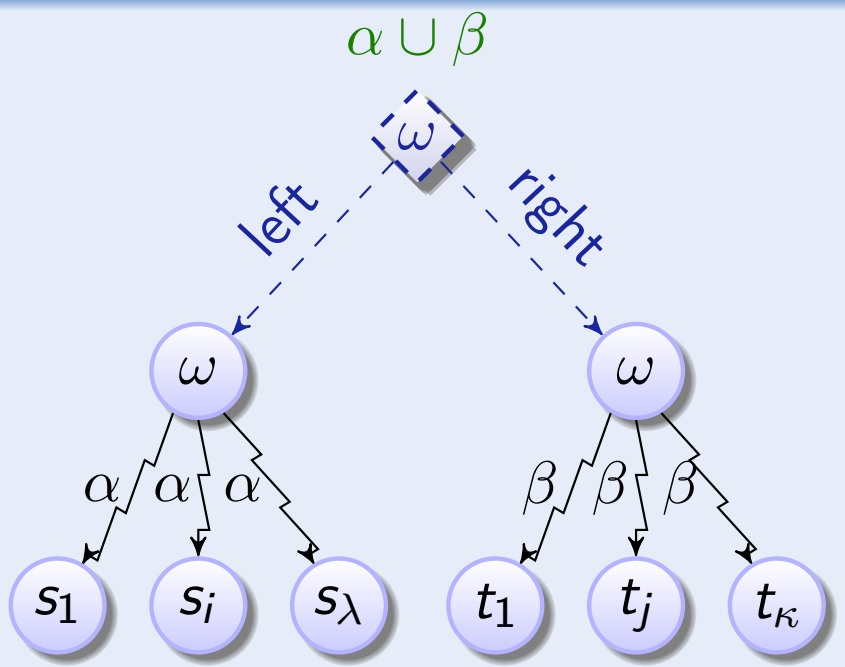
$$x' = f(x) \,\&\, Q$$



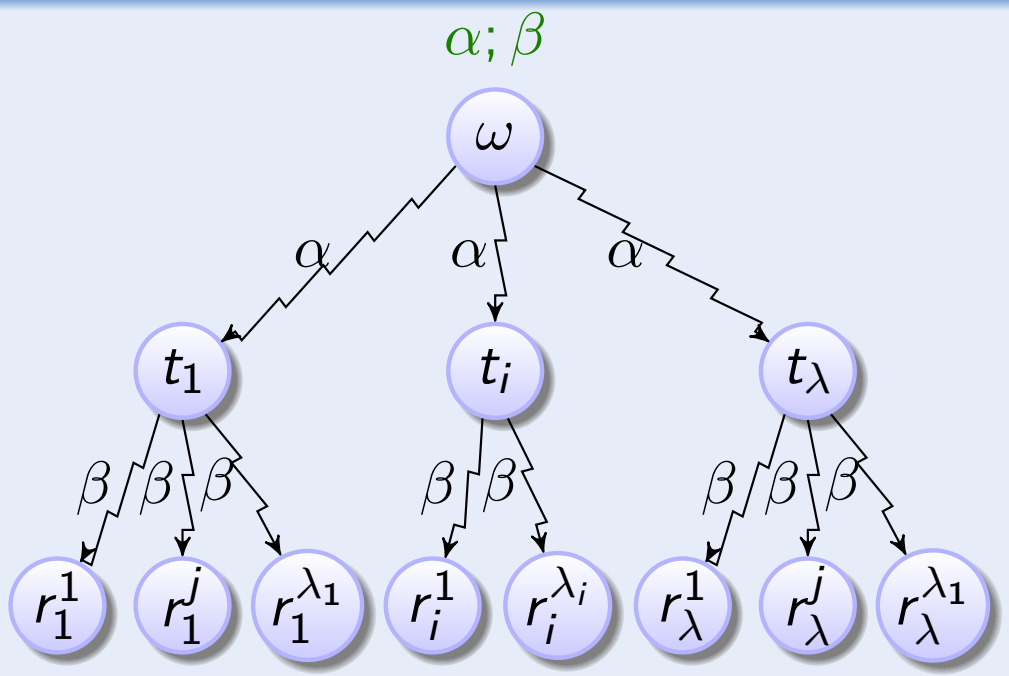
## Definition (Hybrid game $\alpha$: operational semantics)



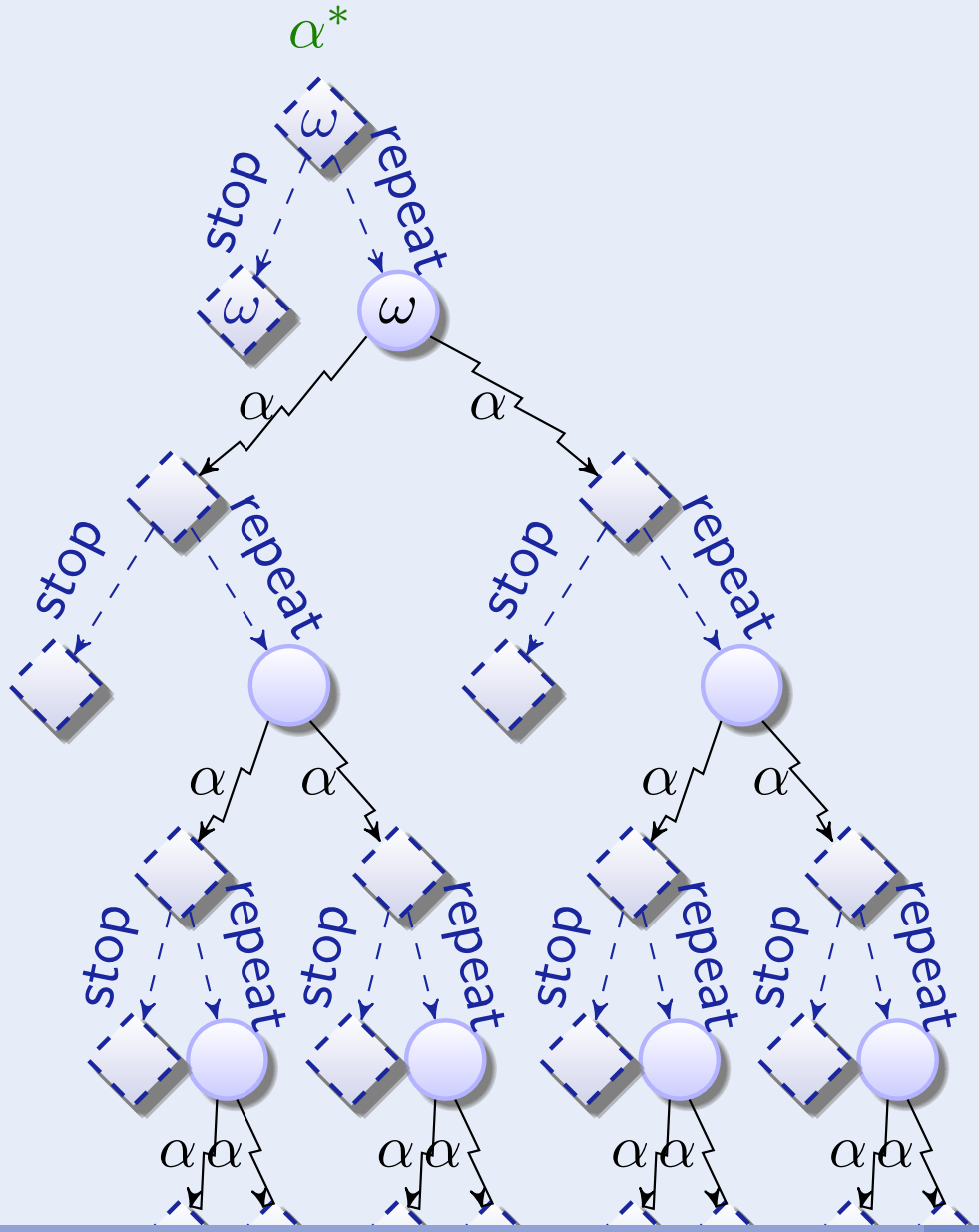
## Definition (Hybrid game $\alpha$: operational semantics)



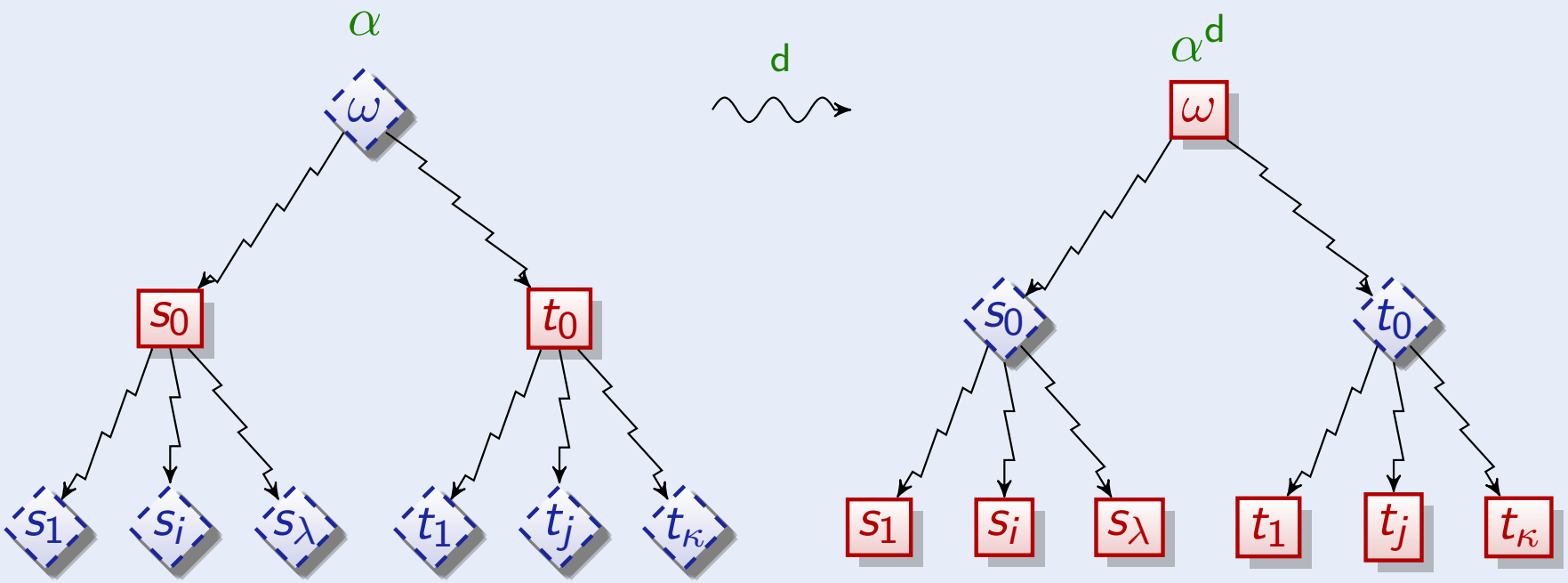
## Definition (Hybrid game $\alpha$: operational semantics)



## Definition (Hybrid game $\alpha$: operational semantics)

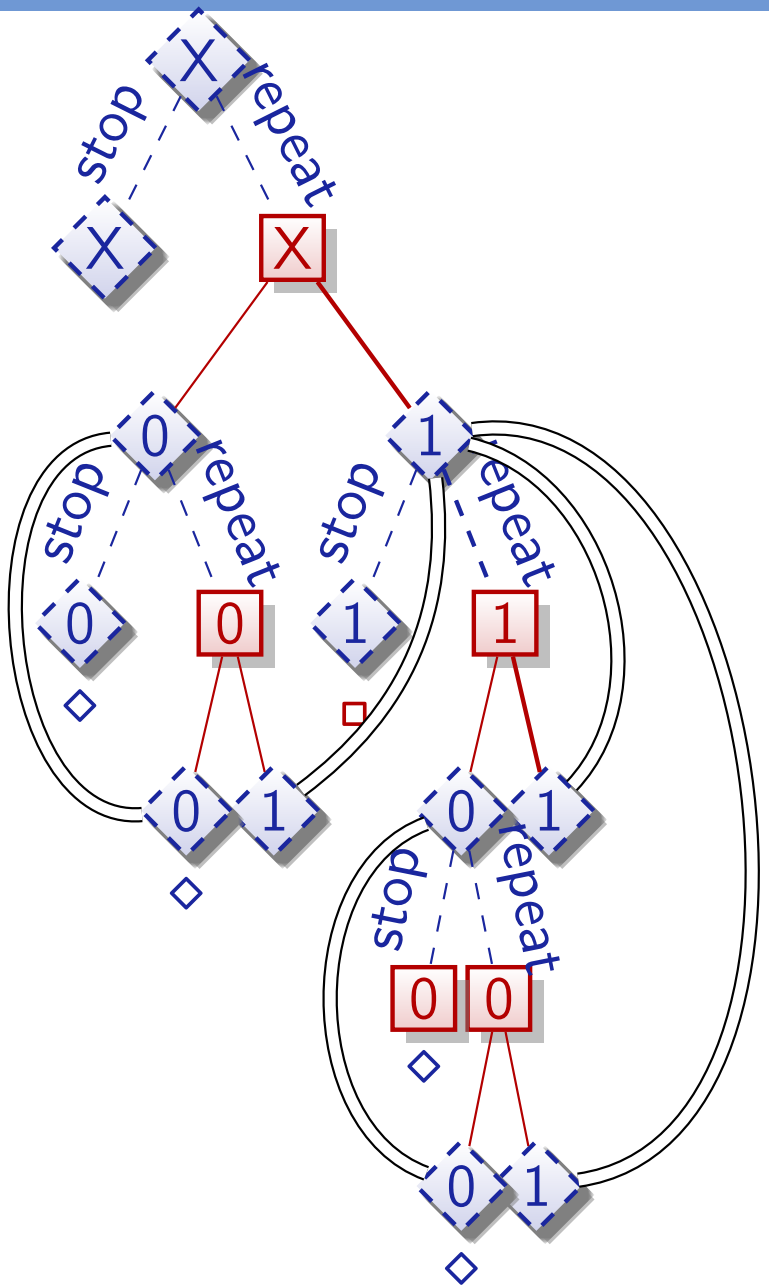


## Definition (Hybrid game $\alpha$: operational semantics)



# Filibusters

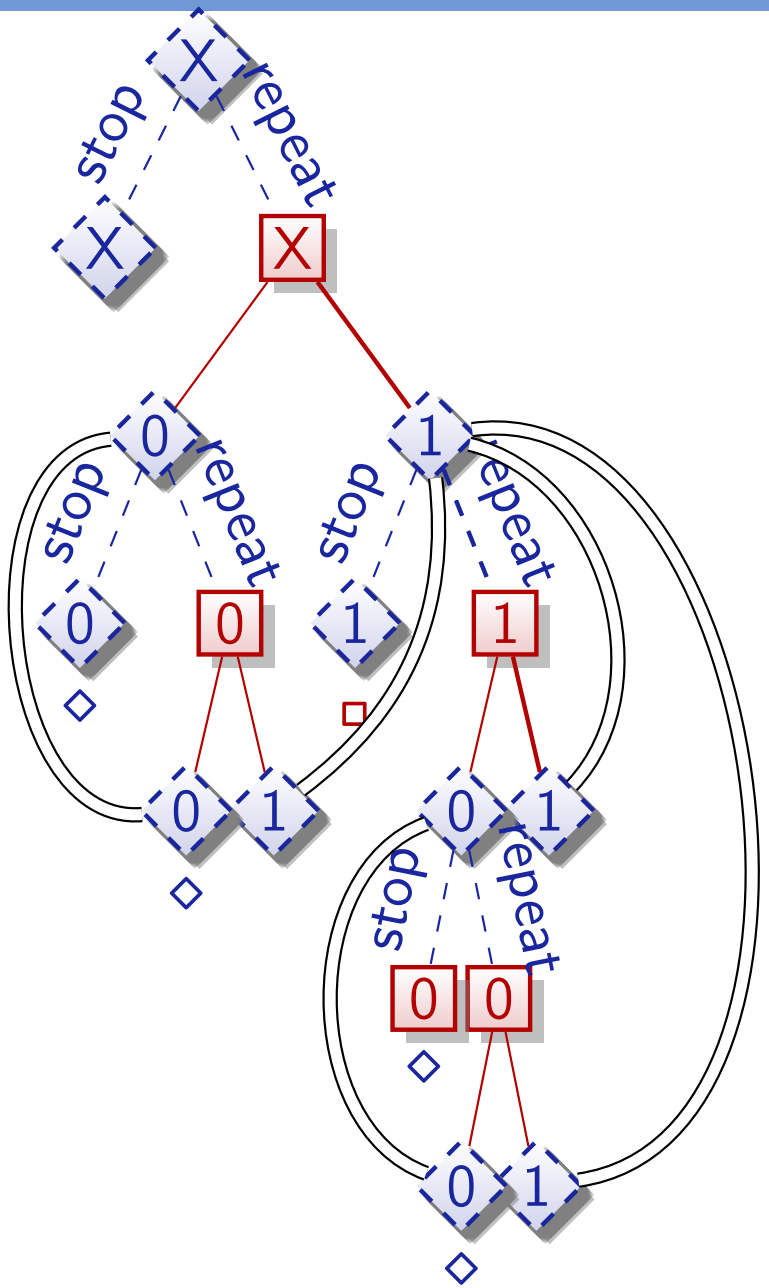
$$\langle (x := 0 \cap x := 1)^* \rangle x = 0$$



# Filibusters & The Significance of Finitude

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

$\overset{\text{wfd}}{\rightsquigarrow}$ false unless $x = 0$



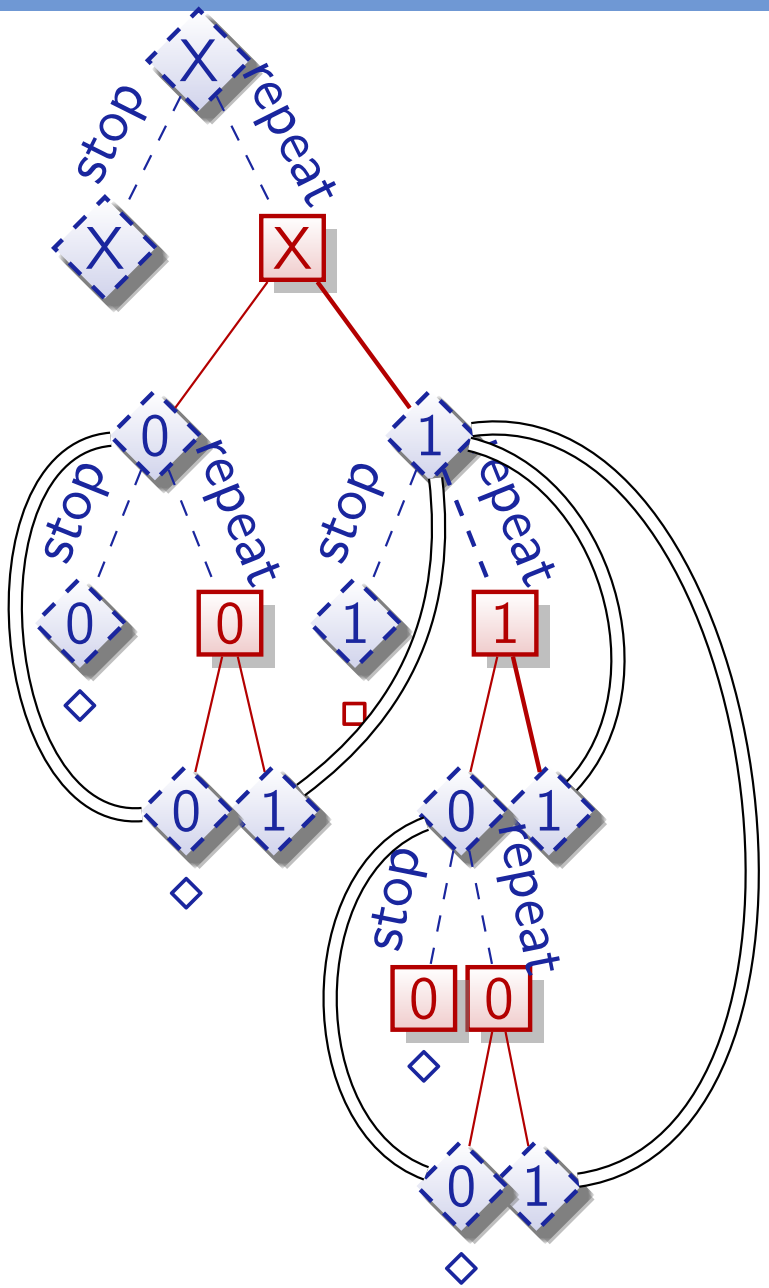
# Filibusters & The Significance of Finitude

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

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

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

$\overset{\text{wfd}}{\rightsquigarrow}$ false unless $x = 0$



# Filibusters & The Significance of Finitude

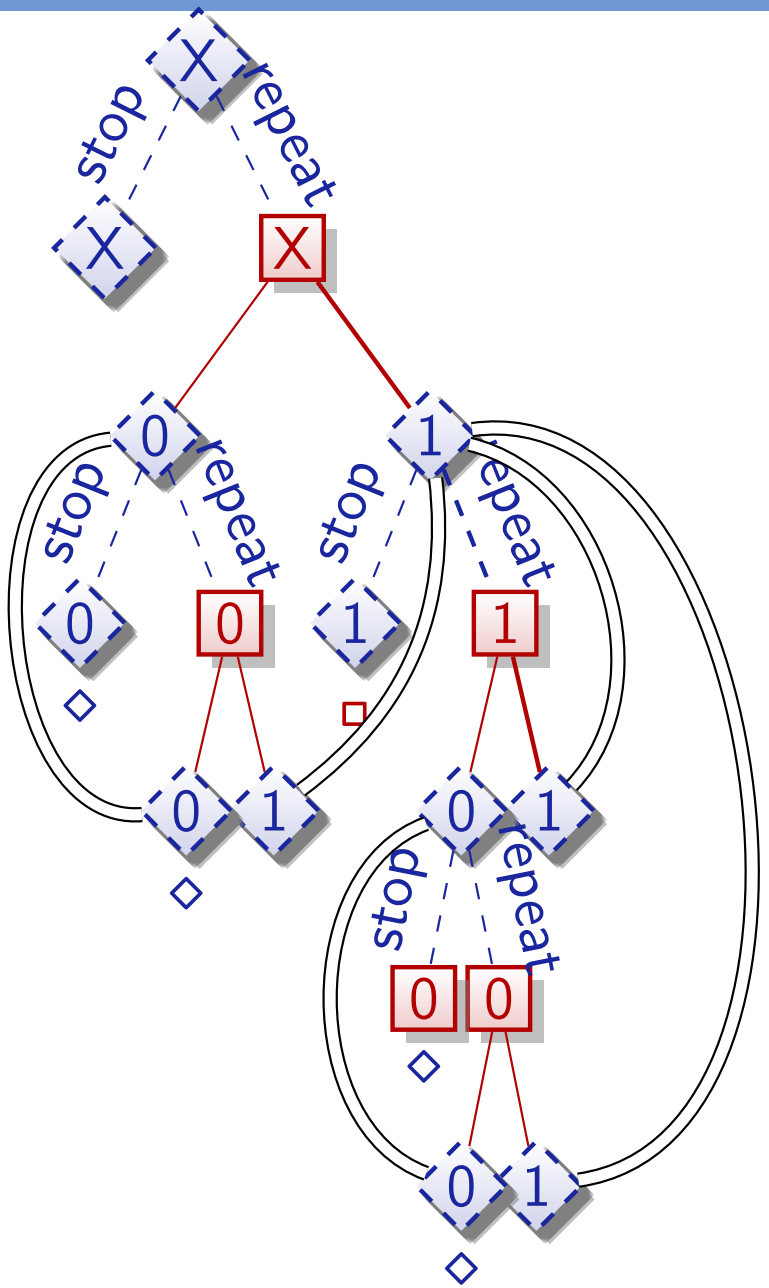
$\overset{<\infty}{\rightsquigarrow}$ true

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

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

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

$\overset{\text{wfd}}{\rightsquigarrow}$ false unless $x = 0$



# Filibusters & The Significance of Finitude

$\overset{<\infty}{\rightsquigarrow}$ true

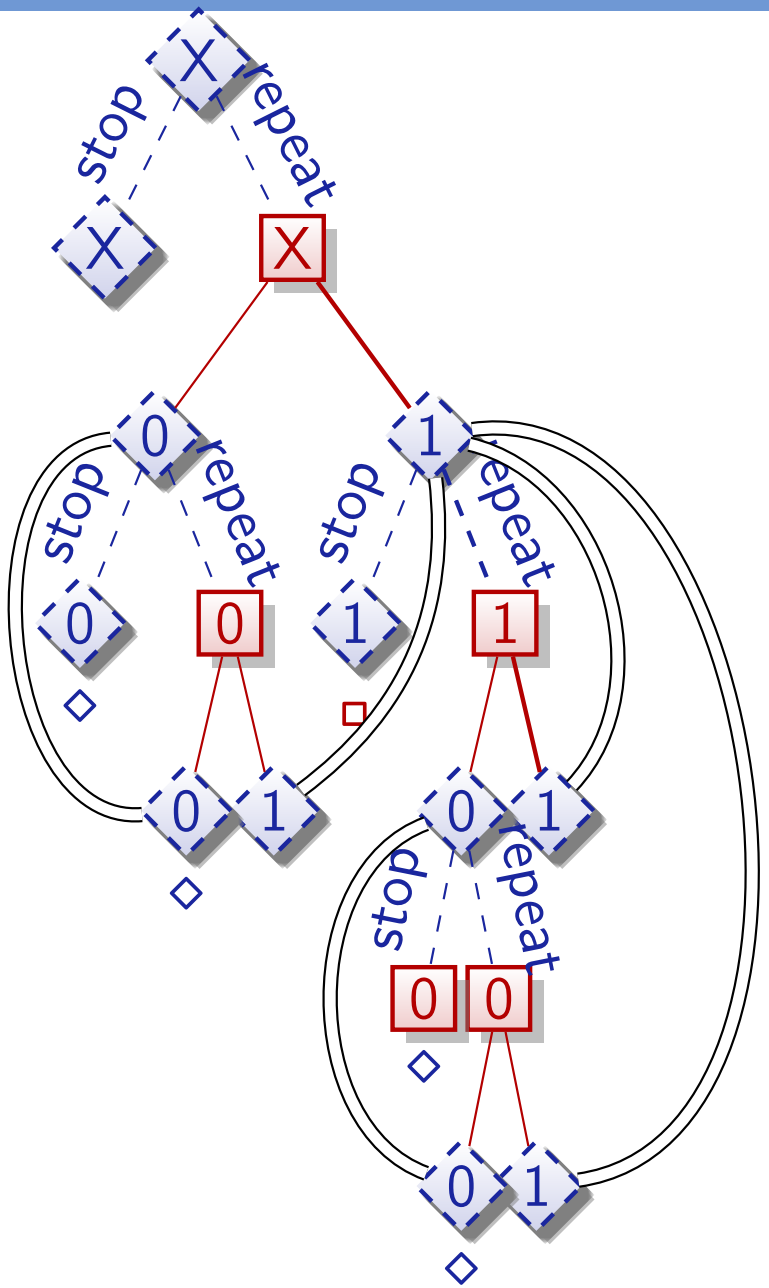
$\langle (x' = 1^{\mathsf{d}}; x := 0)^* \rangle x = 0$

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

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

$\overset{\text{wfd}}{\rightsquigarrow}$ false unless $x = 0$

Well-defined games
can't be postponed forever



# Outline

# Differential Game Logic: Syntax

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

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

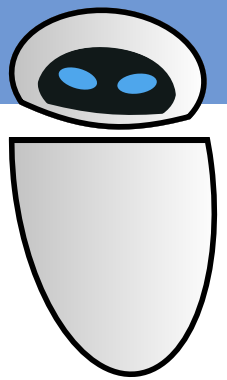
$$\alpha, \beta \ ::= \ x := e \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \alpha^{\mathsf{d}}$$

**Definition (dGL Formula $P$)**

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

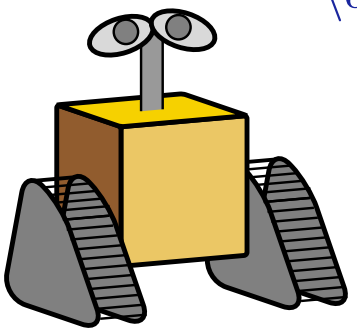
All Reals | Some Reals | Angel Wins | Demon Wins

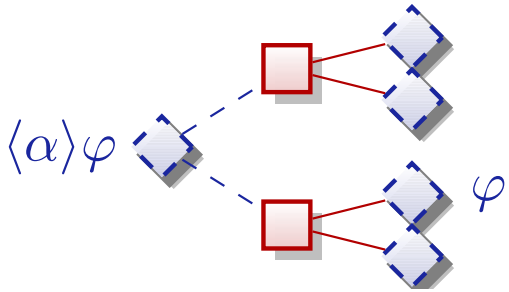
# Summary



**differential game logic**

$\mathsf{dGL} = \mathsf{GL} + \mathsf{HG} = \mathsf{dL} + {}^{\mathsf{d}}$

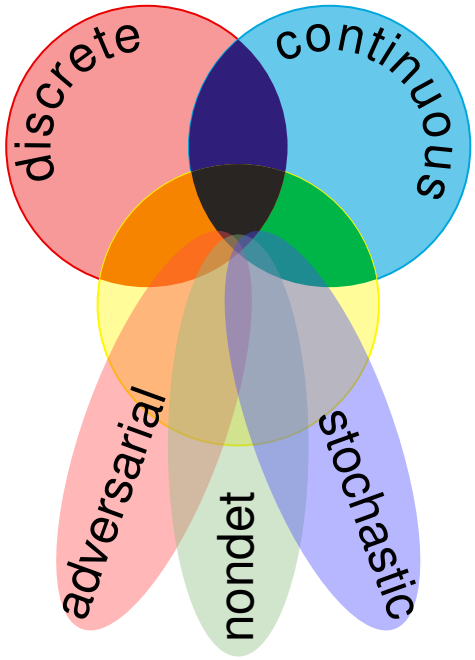





- Differential game logic
- Logic for hybrid games
- Compositional PL + logic
- Discrete + continuous + adversarial
- Operational semantics (informally)

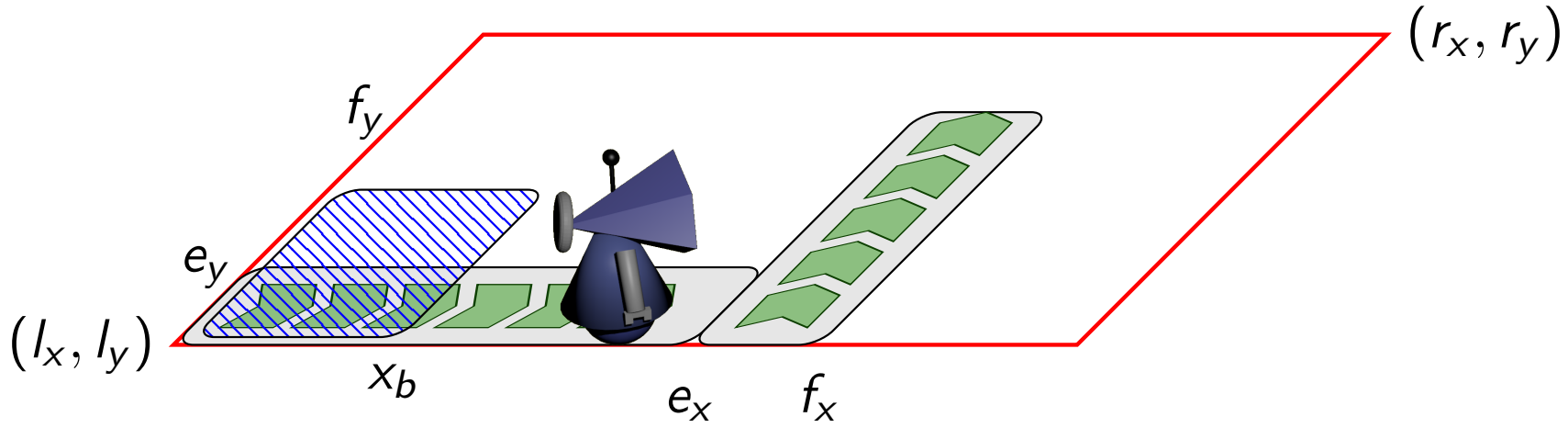
Next chapter

1. Formal semantics



# Outline

# Example: Robot Factory Decentralized Automation




## 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

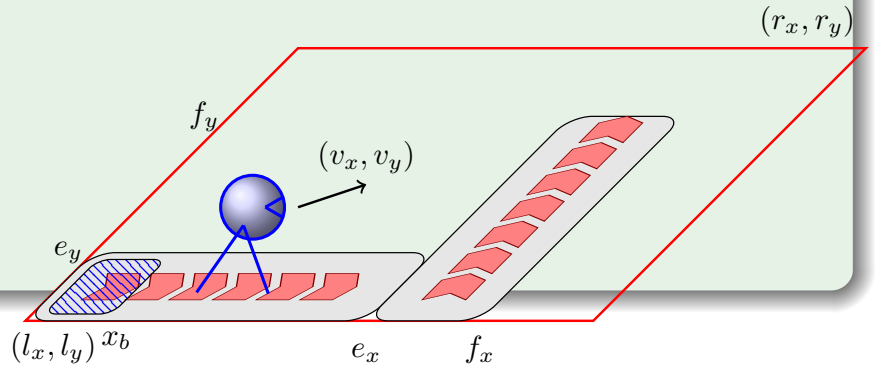
- Distributed, physical environment
- Possibly conflicting secondary objectives

## Example (Robot-Demon vs. Angel-Factory Environment)

$$\Big( ( ?true \cup (?(x < e_x \wedge y < e_y \wedge \mathrm{eff}_1 = 1);\ v_x := v_x + c_x;\ \mathrm{eff}_1 := 0) \quad // \text{ belt}$$

$$\cup (?(e_x \leq x \wedge y \leq f_y \wedge \mathrm{eff}_2 = 1);\ v_y := v_y + c_y;\ \mathrm{eff}_2 := 0)\ )\ ;$$

$$\Big)^*$$



## Example (Robot-Demon vs. Angel-Factory Environment)

$$\Big( ( ?true \cup (?(x < e_x \wedge y < e_y \wedge \mathrm{eff}_1 = 1);\ v_x := v_x + c_x;\ \mathrm{eff}_1 := 0) \quad \text{// belt}$$

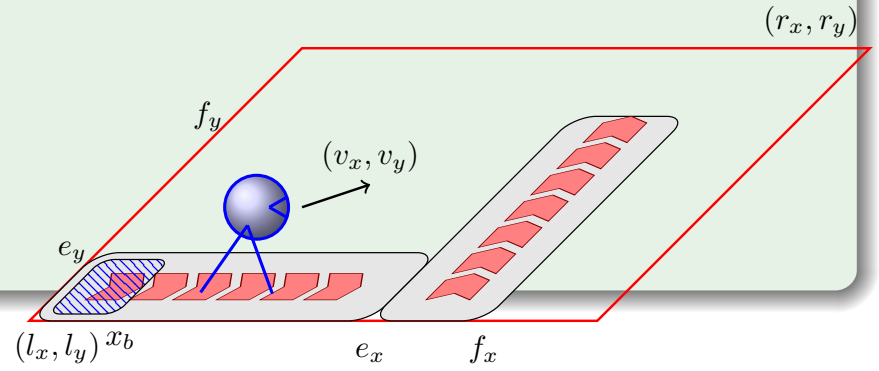
$$\cup (?(e_x \leq x \wedge y \leq f_y \wedge \mathrm{eff}_2 = 1);\ v_y := v_y + c_y;\ \mathrm{eff}_2 := 0)\ );$$

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

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

$$t_s := 0\ )^{\mathrm{d}};$$

$$)^*$$



## Example (Robot-Demon vs. Angel-Factory Environment)

$$\Big(\big(?true \cup (?(x < e_x \wedge y < e_y \wedge \mathrm{eff}_1 = 1);\ v_x := v_x + c_x;\ \mathrm{eff}_1 := 0) \qquad \text{// belt}$$

$$\cup\, (?(e_x \leq x \wedge y \leq f_y \wedge \mathrm{eff}_2 = 1);\ v_y := v_y + c_y;\ \mathrm{eff}_2 := 0)\ \big);$$

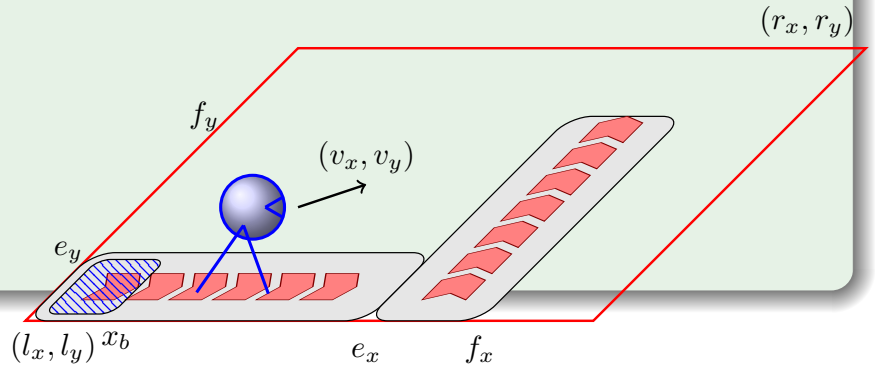
$$\big(\, a_x := *;\ ?(-A \leq a_x \leq A);$$

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

$$t_s := 0\ \big)^{\mathrm{d}};$$

$$(x' = v_x, y' = v_y, v_x' = a_x, v_y' = a_y, t' = 1, t_s' = 1\ \&\ t_s \leq \varepsilon\ \big);$$

$$\Big)^*$$



# Robot Factory Automation ($RF$)

**Example (Robot-Demon vs. Angel-Factory Environment)**

$$
\Big(\big(?true \cup (?(x < e_x \wedge y < e_y \wedge \mathrm{eff}_1 = 1);\ v_x := v_x + c_x;\ \mathrm{eff}_1 := 0) \qquad \text{// belt}
$$

$$
\cup\, (?(e_x \le x \wedge y \le f_y \wedge \mathrm{eff}_2 = 1);\ v_y := v_y + c_y;\ \mathrm{eff}_2 := 0)\ \big);
$$

$$
(\, a_x := *;\ ?(-A \le a_x \le A);
$$

$$
a_y := *;\ ?(-A \le a_y \le A); \qquad \text{// “independent” robot acceleration}
$$

$$
t_s := 0\ )^{\mathrm{d}};
$$

$$
\big(\big(x' = v_x, y' = v_y, v_x' = a_x, v_y' = a_y, t' = 1, t_s' = 1\ \&\ t_s \le \varepsilon\ \big);
$$

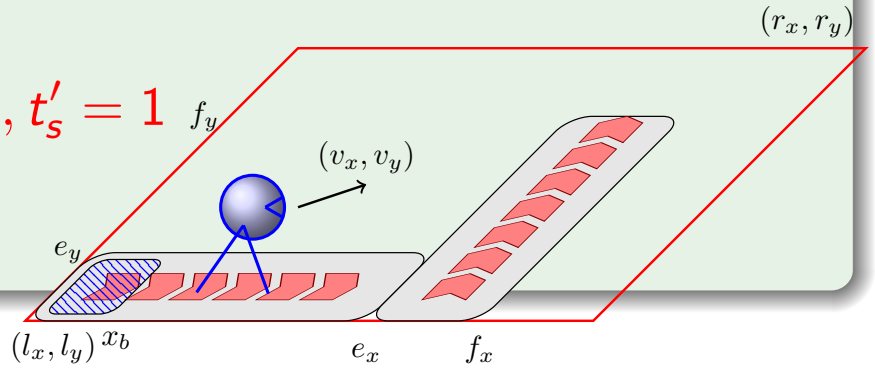
$$
\cap\, (?(a_x v_x \le 0 \wedge a_y v_y \le 0)^{\mathrm{d}}; \qquad \text{// brake}
$$

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

$$
\text{if } v_y = 0 \text{ then } a_y := 0 \text{ fi};
$$

$$
(x' = v_x, y' = v_y, v_x' = a_x, v_y' = a_y, t' = 1, t_s' = 1
$$

$$
\&\ t_s \le \varepsilon \wedge a_x v_x \le 0 \wedge a_y v_y \le 0)))\Big)^*
$$



# Robot Factory Automation ($RF$)

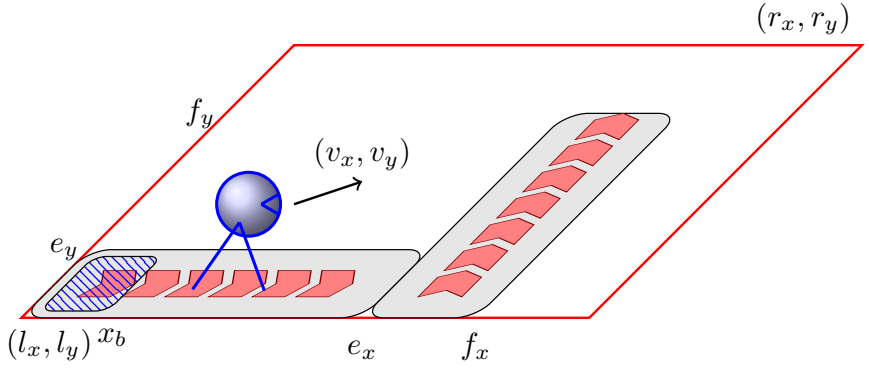
## Proposition (Robot stays in □)

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

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

*$RF|_x$: RF projected to the x-axis*

$$\models (x = 0 \land v_x = 0 \land \text{Controllability Assumptions}) \to [RF|_x](x \in [l_x, r_x] \land (t \geq \varepsilon \to x \geq x_b))$$



André Platzer.
*Logical Foundations of Cyber-Physical Systems*.
Springer, Switzerland, 2018.
URL: http://www.springer.com/978-3-319-63587-3,
doi:10.1007/978-3-319-63588-0.

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

André Platzer.
Logics of dynamical systems.
In LICS [12], pages 13–24.
doi:10.1109/LICS.2012.13.

André Platzer.
Logic & proofs for cyber-physical systems.
In Nicola Olivetti and Ashish Tiwari, editors, *IJCAR*, volume 9706 of *LNCS*, pages 15–21, Berlin, 2016. Springer.

doi:10.1007/978-3-319-40229-1_3.

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

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

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

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.
*Log. Meth. Comput. Sci.*, 8(4:17):1–44, 2012.
Special issue for selected papers from CSL'10.
doi:10.2168/LMCS-8(4:17)2012.

André Platzer.
Stochastic differential dynamic logic for stochastic hybrid programs.
In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *LNCS*, pages 446–460, Berlin, 2011. Springer.
doi:10.1007/978-3-642-22438-6_34.

Jan-David Quesel and André Platzer.
Playing hybrid games with KeYmaera.
In Bernhard Gramlich, Dale Miller, and Ulrike Sattler, editors, *IJCAR*, volume 7364 of *LNCS*, pages 439–453, Berlin, 2012. Springer.
doi:10.1007/978-3-642-31365-3_34.

*Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on*, Los Alamitos, 2012. IEEE.