Playing Hybrid Games with KeYmaera

Jan-David Quesel[1] André Platzer[2]

1. University of Oldenburg, Department of Computing Science, Germany
2. Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, USA

The 6th International Joint Conference on Automated Reasoning
26th June 2012

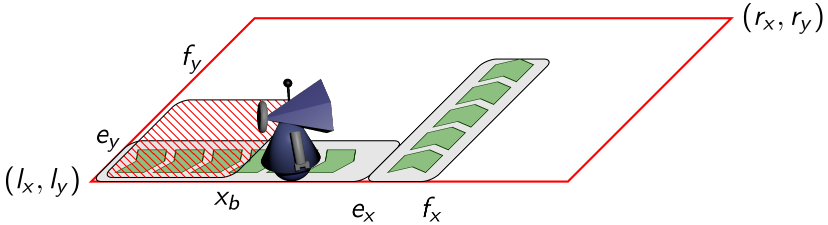
Outline

1. Motivation
2. Differential Dynamic Game Logic (dDG$\mathcal{L}$)
3. Proof Calculus
4. Case Study
5. Conclusion

Outline

1. Motivation
2. Differential Dynamic Game Logic (dDG$\mathcal{L}$)
3. Proof Calculus
4. Case Study
5. Conclusion

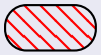
Automated Factory



Model

- (x, y): coordinates of the robot
- (v_x, v_y): velocities
- conveyor belts instantaneously increase the velocity of the robot

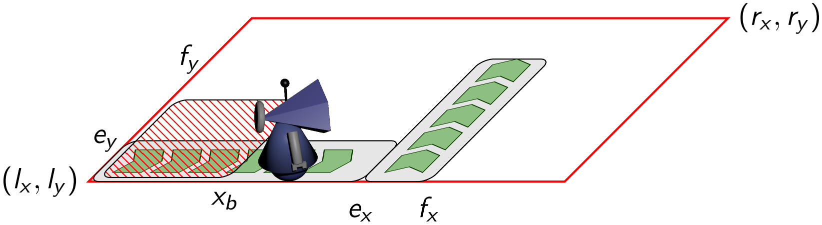
Primary objectives of the robot

- *Leave*  within ε time units.
- Do *not* leave .

Challenges

- Distributed, physical environment
- Possibly conflicting secondary objectives

Automated Factory

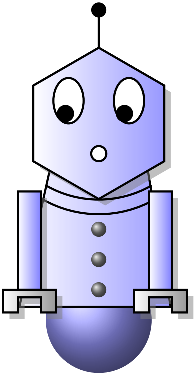


Is there a strategy for the robot to stay safe?

Modelling Language

Differential equations for robot movement

$$x' = v_x, \qquad v_x' = a_x,$$
$$y' = v_y, \qquad v_y' = a_y$$



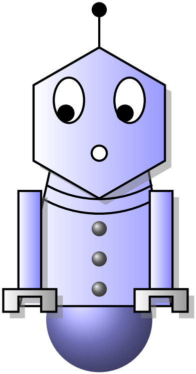
Modelling Language

Differential equations for robot movement

$$x' = v_x, \qquad v_x' = a_x,$$
$$y' = v_y, \qquad v_y' = a_y$$

Guards/Constraints

$$l_x \leq x \leq r_x, \qquad v_x^2 \leq 2A(r_x - f_x)$$



Modelling Language

Differential equations for robot movement

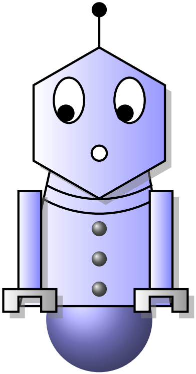
$$x' = v_x, \qquad v_x' = a_x,$$
$$y' = v_y, \qquad v_y' = a_y$$

Guards/Constraints

$$l_x \leq x \leq r_x, \qquad v_x^2 \leq 2A(r_x - f_x)$$

Discrete Assignments

$$a_x := -A, \qquad v_x := v_x + c_x, \qquad \left(s := \frac{v^2}{2b}\right)$$



Modelling Language

Hybrid Program	Effect
$\alpha;\ \beta$	sequential composition
$\alpha \cup \beta$	nondeterministic choice
α^*	nondeterministic repetition
$x := \theta$	discrete assignment (jump)
$x := *$	nondeterministic assignment
$(x_1' = \theta_1, \ldots, x_n' = \theta_n \& F)$	continuous evolution of x_i
$?F$	assert that formula F holds

Platzer, André
Differential dynamic logic for hybrid systems.
J. Autom. Reasoning **41**(2) (2008) 143–189

Outline

1 Motivation

2 Differential Dynamic Game Logic (dDG$\mathcal{L}$)

3 Proof Calculus

4 Case Study

5 Conclusion

Differential Dynamic Game Logic dDG$\mathcal{L}$

Hybrid Program	Effect
$\alpha;\ \beta$	sequential composition
$\alpha\ \cup\ \beta$	nondeterministic choice
α^*	nondeterministic repetition
$x := \theta$	discrete assignment (jump)
$x := *$	nondeterministic assignment
$\left(x_1' = \theta_1, \ldots, x_n' = \theta_n \& F\right)$	continuous evolution of x_i
$?F$	assert that formula F holds

Platzer, André
Differential dynamic logic for hybrid systems.
J. Autom. Reasoning **41**(2) (2008) 143–189

Differential Dynamic Game Logic dDG$\mathcal{L}$

Hybrid Program	Effect
$\alpha;\ \beta$	sequential composition
$\alpha\ \cup\ \beta$	nondeterministic choice
α^*	nondeterministic repetition
$x := \theta$	discrete assignment (jump)
$x := *$	nondeterministic assignment
$(x_1' = \theta_1, \dots, x_n' = \theta_n \& F)$	continuous evolution of x_i
$?F$	assert that formula F holds

Definition (Hybrid Game)

$$G ::= [\alpha] \mid \langle\alpha\rangle \mid (G_1 \cap G_2) \mid (G_1 \cup G_2) \mid (G_1 G_2) \mid (G)^{[*]} \mid (G)^{\langle*\rangle}$$

Falsifier vs. Verifier

Game Rules

Hybrid Game (informal) Rules

$[\alpha]$	Falsifier plays α
$(G_1 \cap G_2)$	Falsifier decides whether to play G_1 or G_2
$(G)^{[*]}$	Repeat G n times, where n is chosen in advance by Falsifier

Game Rules

Hybrid Game (informal) Rules

$[\alpha]$	Falsifier plays α
$\langle\alpha\rangle$	Verifier plays α
$(G_1 \cap G_2)$	Falsifier decides whether to play G_1 or G_2
$(G_1 \cup G_2)$	Verifier decides whether to play G_1 or G_2
$(G)^{[*]}$	Repeat G n times, where n is chosen in advance by Falsifier
$(G)^{\langle * \rangle}$	Repeat G n times, where n is chosen in advance by Verifier

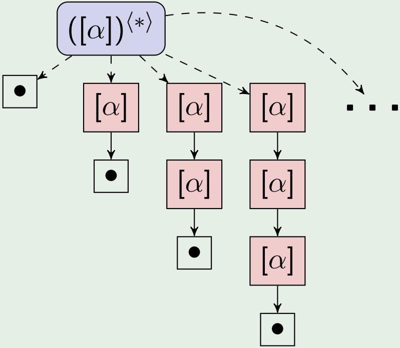
Game Rules

Hybrid Game (informal) Rules

$[\alpha]$	Falsifier plays α
$\langle\alpha\rangle$	Verifier plays α
$(G_1 \cap G_2)$	Falsifier decides whether to play G_1 or G_2
$(G_1 \cup G_2)$	Verifier decides whether to play G_1 or G_2
$(G)^{[*]}$	Repeat G n times, where n is chosen in advance by Falsifier
$(G)^{\langle * \rangle}$	Repeat G n times, where n is chosen in advance by Verifier
$(G_1 G_2)$	Play G_1 followed by G_2

Game Illustration

Example (Repetition with advance notice semantics)

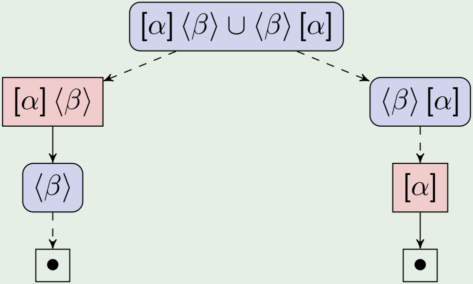


Observations

1. Countably infinite branching
2. Every path has finite depth

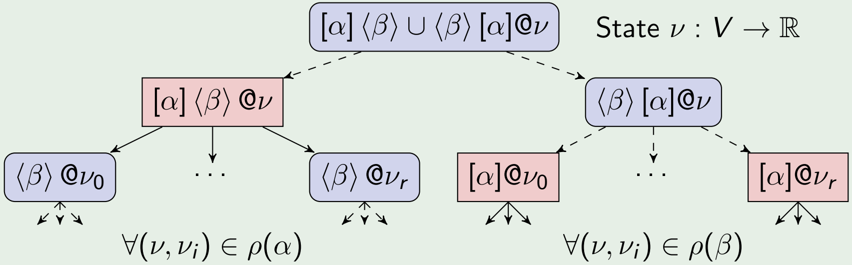
Game Illustration

Example (Explicit branching)



Game Illustration

Example (State)



Observations

1. Uncountably infinite branching
2. Every path has finite depth

Differential Dynamic Game Logic dDG$\mathcal{L}$

Hybrid Program	Effect
$\alpha;\ \beta$	sequential composition
$\alpha \cup \beta$	nondeterministic choice
α^*	nondeterministic repetition
$x := \theta$	discrete assignment (jump)
$x := *$	nondeterministic assignment
$(x_1' = \theta_1, \ldots, x_n' = \theta_n \& F)$	continuous evolution of x_i
$?F$	assert that formula F holds

Definition (Hybrid Game)

$$G ::= [\alpha] \mid \langle\alpha\rangle \mid (G_1 \cap G_2) \mid (G_1 \cup G_2) \mid (G_1 G_2) \mid (G)^{[*]} \mid (G)^{\langle * \rangle}$$

Falsifier vs. Verifier

Differential Dynamic Game Logic dDG$\mathcal{L}$

Hybrid Program	Effect
$\alpha;\ \beta$	sequential composition
$\alpha \cup \beta$	nondeterministic choice
α^*	nondeterministic repetition
$x := \theta$	discrete assignment (jump)
$x := *$	nondeterministic assignment
$(x_1' = \theta_1, \dots, x_n' = \theta_n \& F)$	continuous evolution of x_i
$?F$	assert that formula F holds

Definition (Hybrid Game)

$$G ::= [\alpha] \mid \langle\alpha\rangle \mid (G_1 \cap G_2) \mid (G_1 \cup G_2) \mid (G_1 G_2) \mid (G)^{[*]} \mid (G)^{\langle*\rangle}$$

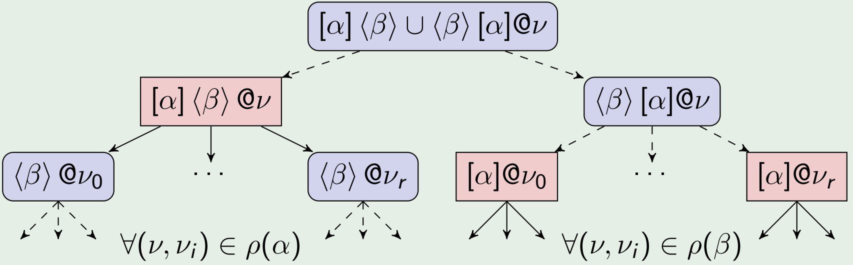
Definition (dDG$\mathcal{L}$ Formula)

$$\phi ::= \theta_1 \sim \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \forall x\, \phi \mid G\ \phi$$

$\rightsquigarrow FOL_{\mathbb{R}}$ + Hybrid Games

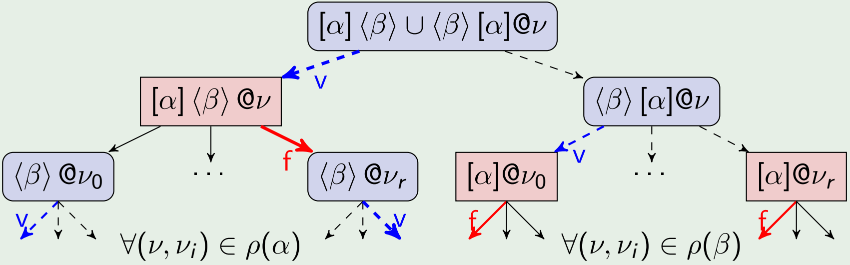
Strategy, Play, and Winning

Example (Strategy)



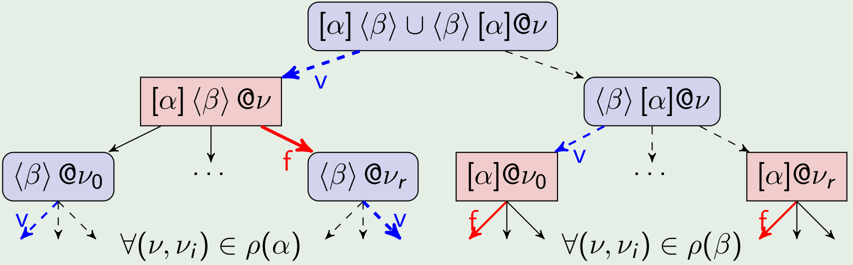
Strategy, Play, and Winning

Example (Strategy)



Strategy, Play, and Winning

Example (Strategy)



Definition (Strategy)

1. Strategy $s : \mathcal{G} \times Sta(V) \to (\mathcal{G} \cup \{\bullet, \bot, \top\}) \times Sta(V)$ maps game positions to followup positions.
2. s is compatible with G if $((g@\nu) \to s(g@\nu)) \in [\![G]\!]$ f.a. $g \in cl(G)$ and f.a. $\nu \in Sta(V)$.

$cl(G)$: closure under subgame

Strategy, Play, and Winning

Definition (Play)

$G \in \mathcal{G}$, $\nu \in Sta(V)$, two compatible strategies (**f** for Falsifier and **v** for Verifier), a play $p_{\mathbf{f},\mathbf{v}}(G@\nu)$ is defined by:

while $G \notin \{\bullet, \bot, \top\}$ do
Match form of G:

od
return $G@\nu$

Strategy, Play, and Winning

Definition (Play)

$G \in \mathcal{G}$, $\nu \in Sta(V)$, two ▸ compatible strategies ($\mathbf{f}$ for Falsifier and $\mathbf{v}$ for Verifier), a play $p_{\mathbf{f},\mathbf{v}}(G@\nu)$ is defined by:

while $G \notin \{\bullet, \perp, \top\}$ do

Match form of G:

Case $[\alpha]$, $G_1 \cap G_2$, or $(G_1)^{[*]} \Rightarrow G@\nu := \mathbf{f}(G@\nu)$ //*Falsifier chooses*

od

return $G@\nu$

Strategy, Play, and Winning

Definition (Play)

$G \in \mathcal{G}$, $\nu \in Sta(V)$, two compatible strategies (**f** for Falsifier and **v** for Verifier), a play $p_{\mathbf{f},\mathbf{v}}(G@\nu)$ is defined by:

while $G \notin \{\bullet, \bot, \top\}$ do
Match form of G:
Case $[\alpha]$, $G_1 \cap G_2$, or $(G_1)^{[*]} \Rightarrow G@\nu := \mathbf{f}(G@\nu)$ //*Falsifier chooses*
Case $\langle\alpha\rangle$, $G_1 \cup G_2$, or $(G_1)^{\langle*\rangle} \Rightarrow G@\nu := \mathbf{v}(G@\nu)$ //*Verifier chooses*

od
return $G@\nu$

Strategy, Play, and Winning

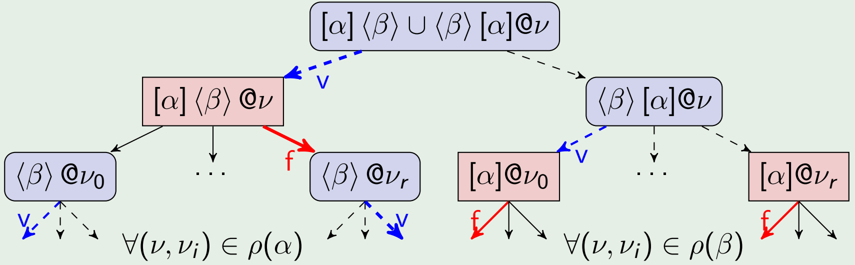
Definition (Play)

$G \in \mathcal{G}$, $\nu \in Sta(V)$, two compatible strategies ($\mathbf{f}$ for Falsifier and $\mathbf{v}$ for Verifier), a play $p_{\mathbf{f},\mathbf{v}}(G@\nu)$ is defined by:

while $G \notin \{\bullet, \bot, \top\}$ do
    Match form of G:
        Case $[\alpha]$, $G_1 \cap G_2$, or $(G_1)^{[*]} \Rightarrow G@\nu := \mathbf{f}(G@\nu)$ //*Falsifier chooses*
        Case $\langle\alpha\rangle$, $G_1 \cup G_2$, or $(G_1)^{\langle*\rangle} \Rightarrow G@\nu := \mathbf{v}(G@\nu)$ //*Verifier chooses*
        Case $G_1 G_2 \Rightarrow$ do
            $G@\nu := p_{\mathbf{f},\mathbf{v}}(G_1@\nu)$ //*play* G_1
            If $G = \bullet$ then $G := G_2$ fi //*if* G_1 *terminated with* $\bullet$ *move to* G_2
        od
od
return $G@\nu$

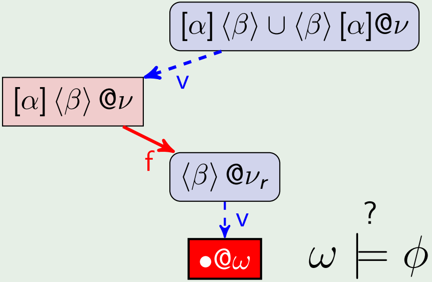
Strategy, Play, and Winning

Example (Winning)



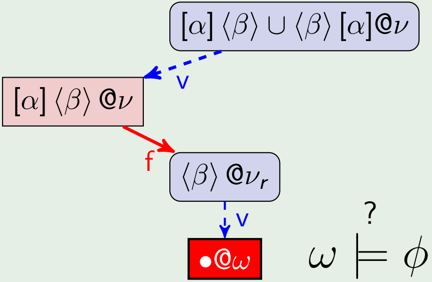
Strategy, Play, and Winning

Example (Winning)



Strategy, Play, and Winning

Example (Winning)



Definition (Winning)

- Winning condition: dDG$\mathcal{L}$ formula ϕ
- Initial state ν
- G is won by Verifier iff G ends in a position $H@\omega$ where
  - $H = \bullet$ and $\omega \models \phi$
  - or $H = \top$.

Outline

1 Motivation

2 Differential Dynamic Game Logic (d$\mathcal{DGL}$)

3 Proof Calculus

4 Case Study

5 Conclusion

Extension

Theorem

dD$\mathcal{G}\mathcal{L}$ *is a conservative extension of* d$\mathcal{L}$*, i.e. for a* d$\mathcal{L}$ *formula* ϕ *holds:*

$$\models_{\mathsf{dDG}\mathcal{L}} \phi \textit{ iff } \models_{\mathsf{d}\mathcal{L}} \phi$$

Proof Calculus for d$\mathcal{L}$

10 propositional rules

(P1) $\dfrac{\vdash \phi}{\neg\phi \vdash}$

(P4) $\dfrac{\phi, \psi \vdash}{\phi \land \psi \vdash}$

(P7) $\dfrac{\phi \vdash \quad \psi \vdash}{\phi \lor \psi \vdash}$

(P2) $\dfrac{\phi \vdash}{\vdash \neg\phi}$

(P5) $\dfrac{\vdash \phi \quad \vdash \psi}{\vdash \phi \land \psi}$

(P8) $\dfrac{\vdash \phi, \psi}{\vdash \phi \lor \psi}$

(P3) $\dfrac{\phi \vdash \psi}{\vdash \phi \to \psi}$

(P6) $\dfrac{\vdash \phi \quad \psi \vdash}{\phi \to \psi \vdash}$

(P9) $\dfrac{}{\phi \vdash \phi}$

(P10) $\dfrac{\vdash \phi \quad \phi \vdash}{\vdash}$

Proof Calculus for d$\mathcal{L}$

13 dynamic rules

(D1) $\dfrac{\phi \land \psi}{\langle ?\phi \rangle \psi}$

(D2) $\dfrac{\phi \rightarrow \psi}{[?\phi]\psi}$

(D3) $\dfrac{\langle \alpha \rangle \phi \lor \langle \beta \rangle \phi}{\langle \alpha \cup \beta \rangle \phi}$

(D4) $\dfrac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$

(D5) $\dfrac{\phi \lor \langle \alpha; \alpha^* \rangle \phi}{\langle \alpha^* \rangle \phi}$

(D6) $\dfrac{\phi \land [\alpha; \alpha^*]\phi}{[\alpha^*]\phi}$

(D7) $\dfrac{\langle\![\alpha]\!\rangle \langle\![\beta]\!\rangle \phi}{\langle\![\alpha; \beta]\!\rangle \phi}$

(D8) $\dfrac{\phi_x^\theta}{\langle\![x := \theta]\!\rangle \phi}$

(D9) $\dfrac{\exists t {\geq} 0\, (\chi t \land \langle x := y_x(t) \rangle \phi)}{\langle x' = \theta \,\&\, \chi \rangle \phi}$

(D10) $\dfrac{\forall t {\geq} 0\, (\chi t \rightarrow [x := y_x(t)]\phi)}{[x' = \theta \,\&\, \chi]\phi}$

(D11) $\dfrac{\phi \vdash \psi}{\langle\![\alpha]\!\rangle \phi \vdash \langle\![\alpha]\!\rangle \psi}$

(D12) $\dfrac{\phi \vdash [\alpha]\phi}{\phi \vdash [\alpha^*]\phi}$

(D13) $\dfrac{\phi(x) \vdash \langle \alpha \rangle \phi(x-1)}{\exists v\, \phi(v) \vdash \langle \alpha^* \rangle \exists v {\leq} 0\, \phi(v)}$

Proof Calculus for d$\mathcal{L}$

6 quantifier rules

$$\text{(F1)}\ \frac{\vdash \phi(s(X_1,\ldots,X_n))}{\vdash \forall x\, \phi(x)} \qquad \text{(F4)}\ \frac{\vdash \phi(X)}{\vdash \exists x\, \phi(x)}$$

$$\text{(F2)}\ \frac{\phi(s(X_1,\ldots,X_n)) \vdash}{\exists x\, \phi(x) \vdash} \qquad \text{(F5)}\ \frac{\phi(X) \vdash}{\forall x\, \phi(x) \vdash}$$

$$\text{(F3)}\ \frac{\vdash \mathrm{QE}(\forall X\, (\Phi(X) \rightarrow \Psi(X)))}{\Phi(s(X_1,\ldots,X_n)) \vdash \Psi(s(X_1,\ldots,X_n))} \qquad \text{(F6)}\ \frac{\vdash \mathrm{QE}(\exists X\, \bigwedge_i (\Phi_i \rightarrow \Psi_i))}{\Phi_1 \vdash \Psi_1 \quad \ldots \quad \Phi_n \vdash \Psi_n}$$

Proof Calculus for dDG$\mathcal{L}$

Calculus (dDG$\mathcal{L}$ specific rules)

(G1) $\dfrac{G_1\phi \vee G_2\phi}{(G_1 \cup G_2)\phi}$ (G2) $\dfrac{G_1\phi \wedge G_2\phi}{(G_1 \cap G_2)\phi}$ (G3) $\dfrac{G_1(G_2\phi)}{(G_1 G_2)\phi}$

(G4) $\dfrac{\vdash \forall^G(\phi \rightarrow \psi)}{G\phi \vdash G\psi}$

(G5) $\dfrac{\vdash \forall^G(\phi \rightarrow G\phi)}{\phi \vdash (G)^{[*]}\phi}$ (G6) $\dfrac{\vdash \forall^G \forall n > 0(\phi(n) \rightarrow G\ (\phi(n-1)))}{\exists n \phi(n) \vdash (G)^{\langle * \rangle} \exists n (n \leq 0 \wedge \phi(n))}$

$\forall^G$: universal closure over variables in G

Proof Calculus for dDG$\mathcal{L}$

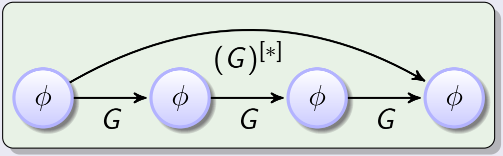
Calculus (dDG$\mathcal{L}$ specific rules)

(G1) $$\frac{G_1\phi \lor G_2\phi}{(G_1 \cup G_2)\phi}$$

(G2) $$\frac{G_1\phi \land G_2\phi}{(G_1 \cap G_2)\phi}$$

(G3) $$\frac{G_1(G_2\phi)}{(G_1 G_2)\phi}$$

(G4) $$\frac{\vdash \forall^G(\phi \rightarrow \psi)}{G\phi \vdash G\psi}$$



(G5) $$\frac{\vdash \forall^G(\phi \rightarrow G\phi)}{\phi \vdash (G)^{[*]}\phi}$$

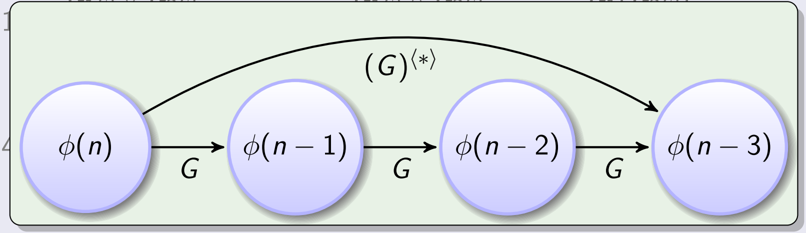
(G6) $$\frac{\vdash \forall^G \forall n > 0(\phi(n) \rightarrow G\ (\phi(n-1)))}{\exists n \phi(n) \vdash (G)^{\langle * \rangle} \exists n (n \leq 0 \land \phi(n))}$$

$\forall^G$: universal closure over variables in G

Proof Calculus for dDG$\mathcal{L}$

Calculus (dDG$\mathcal{L}$ specific rules)

(G1



(G4

(G5) $\dfrac{\vdash \forall^G(\phi \rightarrow G\phi)}{\phi \vdash (G)^{[*]}\phi}$ (G6) $\dfrac{\vdash \forall^G \forall n > 0(\phi(n) \rightarrow G\ (\phi(n-1)))}{\exists n \phi(n) \vdash (G)^{\langle * \rangle} \exists n(n \leq 0 \wedge \phi(n))}$

$\forall^G$: universal closure over variables in G

Sound but Incomplete

Theorem (Soundness)

The sequent calculus for dDG$\mathcal{L}$ *is sound.*

Theorem (Incompleteness)

The sequent calculus for dDG$\mathcal{L}$ *is incomplete.*

Proof Sketch (incompleteness)

1. x is a natural number iff

$$\langle y := 0; (y := y + 1)^* \rangle y = x$$

2. $FOL_{\mathbb{R}}$ + natural numbers: incompletness of the calculus follows by Gödel's incompletness theorem

Relative Completeness

Propositional Dynamic Logic (PDL)

- Game Logic: Game extension of PDL
- Game Logic is strictly more express than PDL:
  PDL cannot express the absence of an infinite g-branch
  ($\langle (g^d)^* \rangle$ *false*).

Parikh, R.:
The logic of games and its applications.
In: Annals of Discrete Mathematics. pp. 111–140. Elsevier (1985)

Relative Completeness

Propositional Dynamic Logic (PDL)

- Game Logic: Game extension of PDL
- Game Logic is strictly more express than PDL:
  PDL cannot express the absence of an infinite g-branch
  ($\langle (g^d)^* \rangle$ *false*).

d$\mathcal{L}$ encoding of $([\alpha])^{\langle * \rangle}$ *false*

$$\exists n \in \mathbb{N} : \forall Z : \exists 0 \leq i < n \in \mathbb{N} : \left[\vec{x} := Z^{(i)}; \alpha\right] \vec{x} \neq Z^{(i+1)}$$

where Z is interpreted as a sequence of real numbers.

Observation

Implicit quantification over states in games
$\leadsto$ completeness modulo d$\mathcal{L}$ unclear.

Outline

1 Motivation

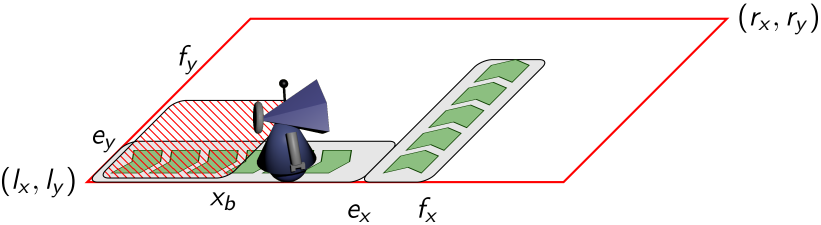
2 Differential Dynamic Game Logic (dDG$\mathcal{L}$)

3 Proof Calculus

4 Case Study

5 Conclusion

Automated Factory



Model

- (x, y): coordinates of the robot
- (v_x, v_y): velocities
- conveyor belts instantaneously increase the velocity of the robot

Primary objectives of the robot

- *Leave* [hatched region] within ε time units.
- Do *not* leave [red box].

Challenges

- Distributed, physical environment
- Possibly conflicting secondary objectives

Robotic Factory Automation (*RF*)

Example (Environment vs. Robot)

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

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

$$\Big)^{[*]}$$

Robotic Factory Automation (RF)

Example (Environment vs. Robot)

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

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

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

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

$$t_s := 0 \,\rangle$$

$$\Big)^{[*]}$$

Robotic Factory Automation (RF)

Example (Environment vs. Robot)

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

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

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

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

$$t_s := 0\ \rangle$$

$$[\, x' = v_x, y' = v_y, v_x' = a_x, v_y' = a_y, t' = 1, t_s' = 1 \& t_s \leq \varepsilon\ ]$$

$$\Big)^{[*]}$$

Robotic Factory Automation (RF)

Example (Environment vs. Robot)

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

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

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

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

$$t_s := 0\ \rangle$$

$$(\,[x' = v_x, y' = v_y, v_x' = a_x, v_y' = a_y, t' = 1, t_s' = 1 \& t_s \le \varepsilon\ ]$$

$$\cup(\langle\, ?a_x v_x \le 0 \wedge a_y v_y \le 0;$$

$$\text{if } v_x = 0 \text{ then } a_x := 0 \text{ fi};$$

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

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

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

Results

Proposition (Robot stays in □)

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

Note: KeYmaera proof has 2471 proof steps on 742 branches (159 interactive steps)

Proposition (Stays in □ + leaves shaded region in time)

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

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

Note: KeYmaera proof has 375079 proof steps on 10641 branches (1673 interactive steps)

Outline

1 Motivation

2 Differential Dynamic Game Logic (dDG$\mathcal{L}$)

3 Proof Calculus

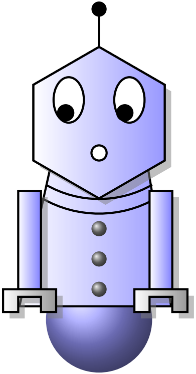
4 Case Study

5 Conclusion

Summary

We ...

- defined a logic for hybrid games (dDG$\mathcal{L}$).
- proved that dDG$\mathcal{L}$ is a conservative extension of d$\mathcal{L}$.
- presented a proof calculus for the logic.
- implemented the calculus in KeYmaera.
- showed a factory automation case study.
- proved the existence of a survival strategy for an robot in an hostile environment.



Outline

6 Related Work

7 Hybrid Games

8 Details

Outline

6 Related Work

7 Hybrid Games

8 Details

Structural Operational Semantics

Falsifier rules

(F1) $\dfrac{(\nu,\omega) \in \rho(\alpha)}{[\alpha]@\nu \to \bullet@\omega}$ (F2) $\dfrac{\rho(\alpha) = \emptyset}{[\alpha]@\nu \to \top@\nu}$ (F3) $\dfrac{G@\nu \to G'@\omega}{G \cap H@\nu \to G'@\omega}$

(F4) $\dfrac{G \cap H@\nu \to G'@\omega}{H \cap G@\nu \to G'@\omega}$ (F5) $\dfrac{n \in \mathbb{N}}{(G)^{[*]}@\nu \to G^n@\nu}$

Back

Structural Operational Semantics

Falsifier rules

(F1) $\dfrac{(\nu,\omega) \in \rho(\alpha)}{[\alpha]@\nu \to \bullet@\omega}$ (F2) $\dfrac{\rho(\alpha) = \emptyset}{[\alpha]@\nu \to \top@\nu}$ (F3) $\dfrac{G@\nu \to G'@\omega}{G \cap H@\nu \to G'@\omega}$

(F4) $\dfrac{G \cap H@\nu \to G'@\omega}{H \cap G@\nu \to G'@\omega}$ (F5) $\dfrac{n \in \mathbb{N}}{(G)^{[*]}@\nu \to G^n@\nu}$

Verifier rules

(V1) $\dfrac{(\nu,\omega) \in \rho(\alpha)}{\langle\alpha\rangle@\nu \to \bullet@\omega}$ (V2) $\dfrac{\rho(\alpha) = \emptyset}{\langle\alpha\rangle@\nu \to \bot@\nu}$ (V3) $\dfrac{G@\nu \to G'@\omega}{G \cup H@\nu \to G'@\omega}$

(V4) $\dfrac{G \cup H@\nu \to G'@\omega}{H \cup G@\nu \to G'@\omega}$ (V5) $\dfrac{n \in \mathbb{N}}{(G)^{\langle*\rangle}@\nu \to G^n@\nu}$

Back

Structural Operational Semantics

Falsifier rules

(F1) $\dfrac{(\nu, \omega) \in \rho(\alpha)}{[\alpha]@\nu \to \bullet@\omega}$ (F2) $\dfrac{\rho(\alpha) = \emptyset}{[\alpha]@\nu \to \top@\nu}$ (F3) $\dfrac{G@\nu \to G'@\omega}{G \cap H@\nu \to G'@\omega}$

(F4) $\dfrac{G \cap H@\nu \to G'@\omega}{H \cap G@\nu \to G'@\omega}$ (F5) $\dfrac{n \in \mathbb{N}}{(G)^{[*]}@\nu \to G^n@\nu}$

Verifier rules

(V1) $\dfrac{(\nu, \omega) \in \rho(\alpha)}{\langle\alpha\rangle@\nu \to \bullet@\omega}$ (V2) $\dfrac{\rho(\alpha) = \emptyset}{\langle\alpha\rangle@\nu \to \bot@\nu}$ (V3) $\dfrac{G@\nu \to G'@\omega}{G \cup H@\nu \to G'@\omega}$

(V4) $\dfrac{G \cup H@\nu \to G'@\omega}{H \cup G@\nu \to G'@\omega}$ (V5) $\dfrac{n \in \mathbb{N}}{(G)^{\langle*\rangle}@\nu \to G^n@\nu}$

Sequential rules

(S1) $\dfrac{G@\nu \to \bullet@\omega}{(G\ H)@\nu \to H@\omega}$ (S2) $\dfrac{G@\nu \to \bot@\omega}{(G\ H)@\nu \to \bot@\omega}$ (S3) $\dfrac{G@\nu \to \top@\omega}{(G\ H)@\nu \to \top@\omega}$

Back

Outline

6 Related Work

7 Hybrid Games

8 Details

Results (detailed)

Assumptions

$$x_b < \frac{1}{2}A\varepsilon^2 \wedge c_x > 0 \wedge (c_x + 4A\varepsilon)^2 \le 2A(r_x - f_x) \quad (1)$$

$$c_y > 0 \wedge c_y^2 \le 2A(r_y - l_y) \quad (2)$$

$$l_x = l_y = 0 \wedge r_x = r_y = 10 \wedge e_x = 2 \wedge e_y = 1 \wedge f_x = 3 \wedge f_y = 10 \wedge A = 2 \quad (3)$$

Proposition

$$\models (x = y = 0 \wedge v_x = v_y = 0 \wedge (1) \wedge (2) \wedge (3)) \rightarrow (RF)(x \in [l_x, r_x] \wedge y \in [l_y, r_y])$$

Proposition

$$\models (x = 0 \wedge v_y = 0 \wedge (1) \wedge (3)) \rightarrow (RF|_x)(x \in [l_x, r_x] \wedge (t \ge \varepsilon \rightarrow (x \ge x_b)))$$

RF projected to the x-axis (denoted $RF|_x$)

▸ Invariant ◂ Return

Invariant

Invariant

$$
\begin{aligned}
\mathrm{eff}_1 \in \{0,1\} \land x \geq l_x \land v_x \geq 0 \land (t \geq \varepsilon \rightarrow x \geq x_b) \\
\land (v_x + c_x \mathrm{eff}_1)^2 \leq 2A(r_x - x) \\
\land \Big(x < x_b \rightarrow t \leq \varepsilon \land \big(x_b - x \leq \frac{1}{2}A\varepsilon^2 - \frac{1}{2}At^2 \\
\land (\mathrm{eff}_1 = 1 \rightarrow v_x = At) \land (\mathrm{eff}_1 = 0 \rightarrow v_x = At + c_x) \\
\land\, r_x - x \geq \frac{(v_x + \mathrm{eff}_1 c_x)^2}{2A} + A(2\varepsilon - t)^2 + 2(2\varepsilon - t)(v_x + \mathrm{eff}_1 c_x) \big) \Big)
\end{aligned}
$$

Return