Differential Dynamic Logics

Automated Theorem Proving for Hybrid Systems

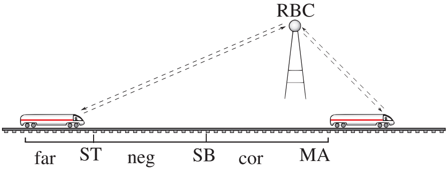
André Platzer

Department of Computing Science
Carl-von-Ossietzky University of Oldenburg, Germany

Disputation, 19.12.2008

Outline

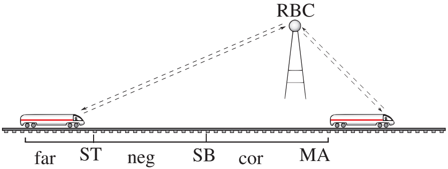
Verifying Parametric Hybrid Systems



ETCS objectives:

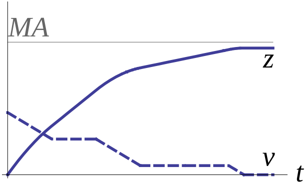
1. Collision free
2. Maximise throughput & velocity (300 km/h)
3. $2.1 * 10^6$ passengers/day

Verifying Parametric Hybrid Systems

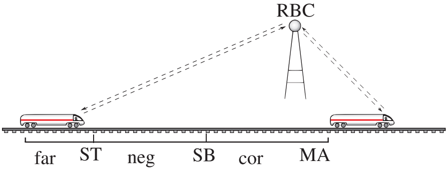


Parametric Hybrid Systems

continuous evolution along differential equations + discrete change

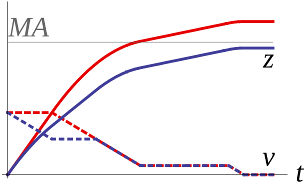


Verifying Parametric Hybrid Systems

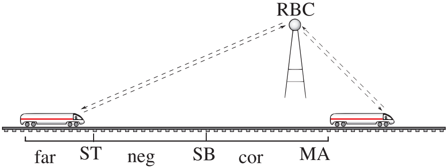


Parametric Hybrid Systems

continuous evolution along differential equations + discrete change

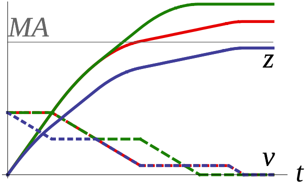


Verifying Parametric Hybrid Systems

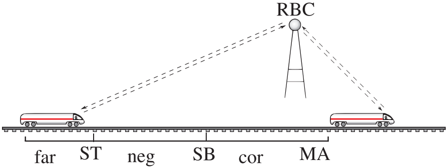


Parametric Hybrid Systems

continuous evolution along differential equations + discrete change



Verifying Parametric Hybrid Systems

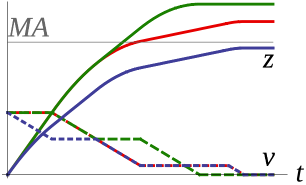


Parametric Hybrid Systems

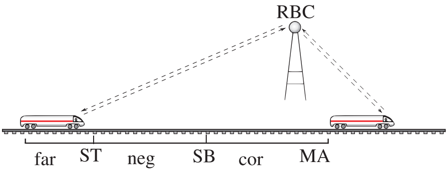
continuous evolution along differential equations + discrete change

- Parameters have nonlinear influence
- Handle SB as free symbolic parameter?
- Challenge: verification (falsifying is "easy")
- Which constraints for SB?

$\forall MA\, \exists SB$ "train always safe"

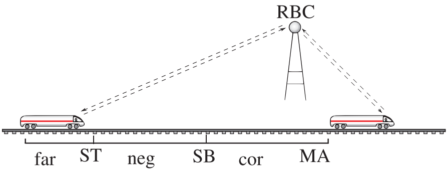


Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	CI	Aut
$ETCS \models z < MA$	TL-MC	✓	×	✓	×	✓

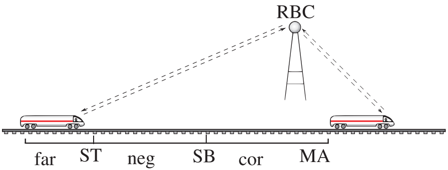
Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	CI	Aut
$ETCS \models z < MA$	TL-MC	✓	×	✓	×	✓

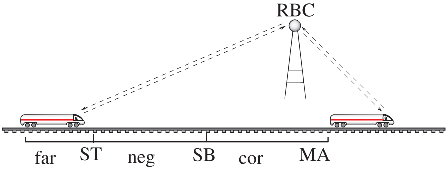
- × no finite-state bisimulation for HS
- × no general handling of free parameters
- × with parameters, everything gets nonlinear!

Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	CI	Aut
$ETCS \models z < MA$	TL-MC	✓	×	✓	×	✓
$\models (\mathrm{Ax}(ETCS) \rightarrow z < MA)$	TL-calculus	×	×	✓	..	×

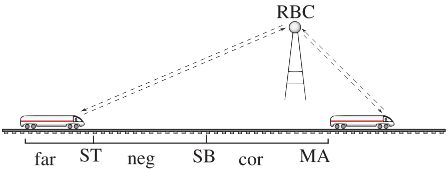
Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	×	✓	×	✓
$\models (\mathrm{Ax}(ETCS) \to z < MA)$	TL-calculus	×	×	✓	..	×

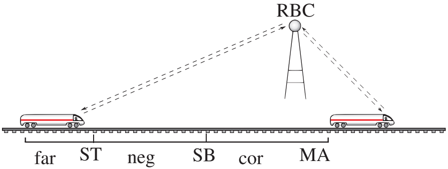
- × declaratively axiomatise operational model
- × expressiveness for characterisation?
- × automation

Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	CI	Aut
$ETCS \models z < MA$	TL-MC	✓	×	✓	×	✓
$\models (\mathrm{Ax}(ETCS) \to z < MA)$	TL-calculus	×	×	✓	..	×
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓	×

Verification Approaches for Hybrid Systems



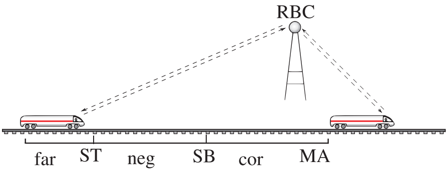
problem	technique	Op	Par	T	CI	Aut
$ETCS \models z < MA$	TL-MC	✓	×	✓	×	✓
$\models (\mathrm{Ax}(ETCS) \rightarrow z < MA)$	TL-calculus	×	×	✓	..	×
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓	×

✓ $[RBC]\mathrm{partitioned} \rightarrow \exists SB\, \langle \mathrm{Train} \rangle [RBC]\mathrm{safe}$

× intermediate states

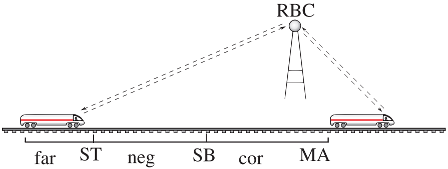
× automation

Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	CI	Aut
$ETCS \models z < MA$	TL-MC	✓	×	✓	×	✓
$\models (\mathrm{Ax}(ETCS) \to z < MA)$	TL-calculus	×	×	✓	..	×
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓	×

Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	CI	Aut
$ETCS \models z < MA$	TL-MC	✓	×	✓	×	✓
$\models (\mathrm{Ax}(ETCS) \rightarrow z < MA)$	TL-calculus	×	×	✓	..	×
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓	?

differential dynamic logic

$$d\mathcal{L} = \mathrm{DL} + \mathrm{HP}$$

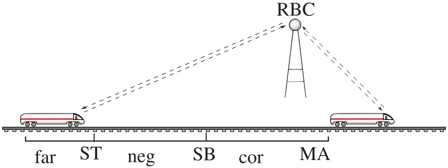
Outline

Outline (Conceptual Approach)

d$\mathcal{L}$ Motives: The Logic of Hybrid Systems

differential dynamic logic

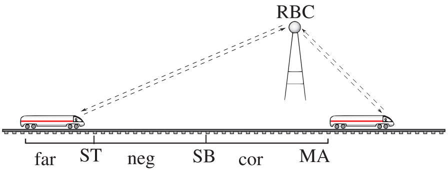
$\mathrm{d}\mathcal{L} = \mathrm{DL} + \mathrm{HP}$

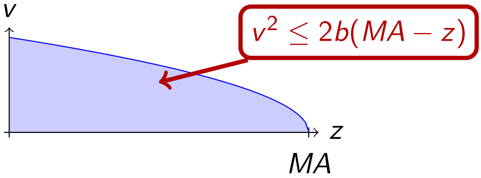


d$\mathcal{L}$ Motives: Regions in First-order Logic

differential dynamic logic

$d\mathcal{L} = \text{FOL}_{\mathbb{R}}$





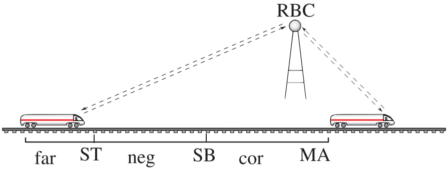
d$\mathcal{L}$ Motives: Regions in First-order Logic

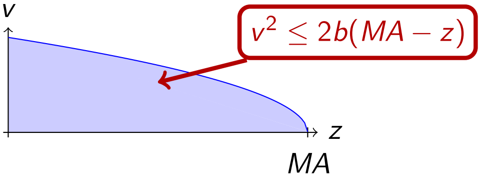
differential dynamic logic

$$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}}$$

$\forall MA \, \exists SB \; \ldots$

$\forall t \geq 0 \; \ldots$

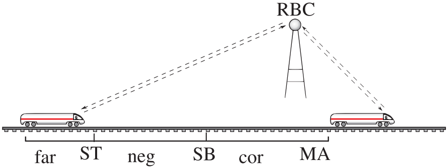




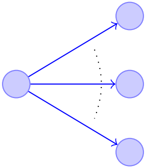
d$\mathcal{L}$ Motives: State Transitions in Dynamic Logic

differential dynamic logic

$\mathrm{d}\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} +$



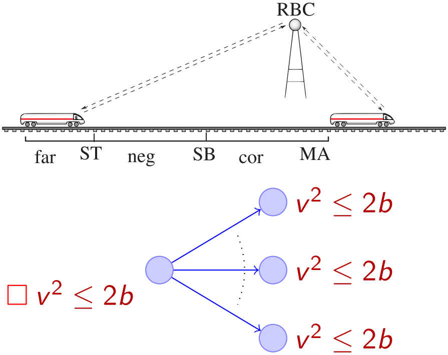
$v^2 \leq 2b$



d$\mathcal{L}$ Motives: State Transitions in Dynamic Logic

differential dynamic logic

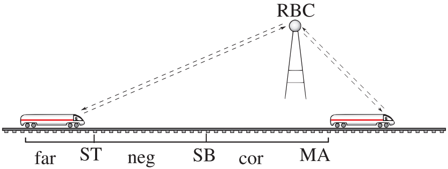
d$\mathcal{L}$ = $\text{FOL}_{\mathbb{R}}$ + ML

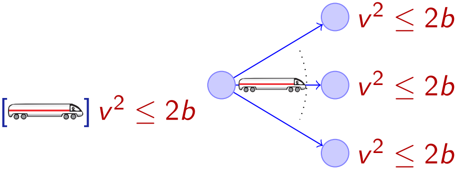


d$\mathcal{L}$ Motives: State Transitions in Dynamic Logic

differential dynamic logic

d$\mathcal{L}$ = $\text{FOL}_{\mathbb{R}}$ + DL

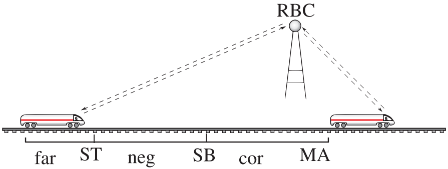




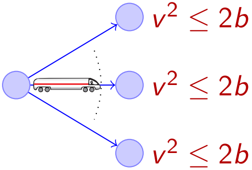
d$\mathcal{L}$ Motives: Hybrid Programs as Uniform Model

differential dynamic logic

$d\mathcal{L} = FOL_{\mathbb{R}} + DL + HP$



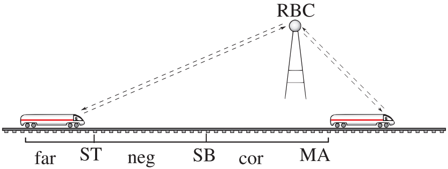
$[z'' = a]\, v^2 \leq 2b$



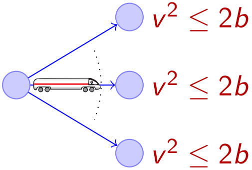
d$\mathcal{L}$ Motives: Hybrid Programs as Uniform Model

differential dynamic logic

d$\mathcal{L}$ = $\mathrm{FOL}_{\mathbb{R}}$ + DL + HP



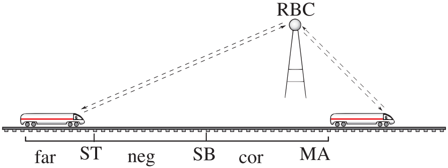
$[\mathtt{if}(z > SB)\, a := -b;\ z'' = a]\, v^2 \leq 2b$



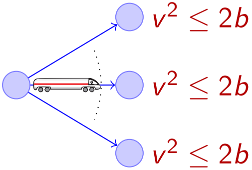
d$\mathcal{L}$ Motives: Hybrid Programs as Uniform Model

differential dynamic logic

$$d\mathcal{L} = \mathrm{FOL}_{\mathbb{R}} + \mathrm{DL} + \mathrm{HP}$$



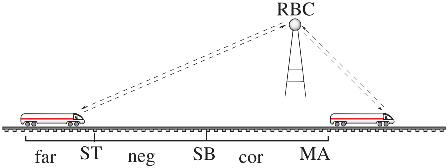
$$[\underbrace{\mathtt{if}(z > SB)\, a := -b;\ z'' = a}_{\text{hybrid program}}]\, v^2 \le 2b$$

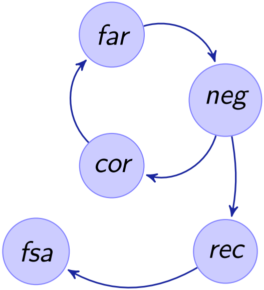


d$\mathcal{L}$ Motives: What about Hybrid Automata?

differential dynamic logic

$d\mathcal{L} = FOL_{\mathbb{R}} + DL + HP$



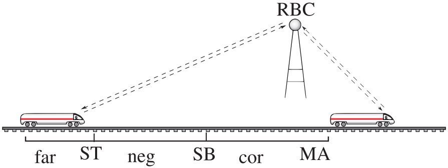


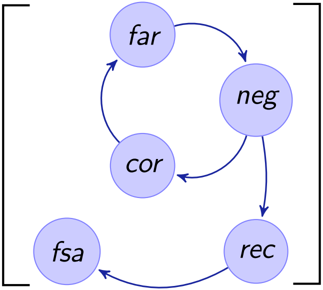
How about hybrid automata?

d$\mathcal{L}$ Motives: What about Hybrid Automata?

differential dynamic logic

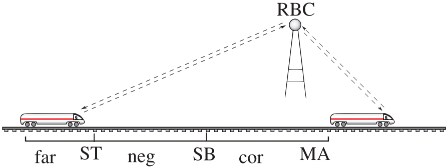
$$\text{d}\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$





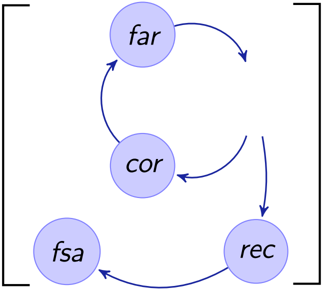
$v^2 \leq 2b\,..$

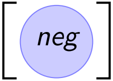
d$\mathcal{L}$ Motives: What about Hybrid Automata?



differential dynamic logic

$\text{d}\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$



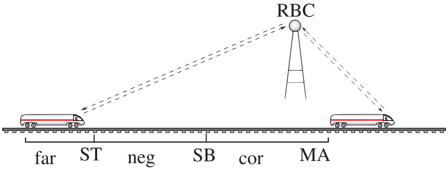


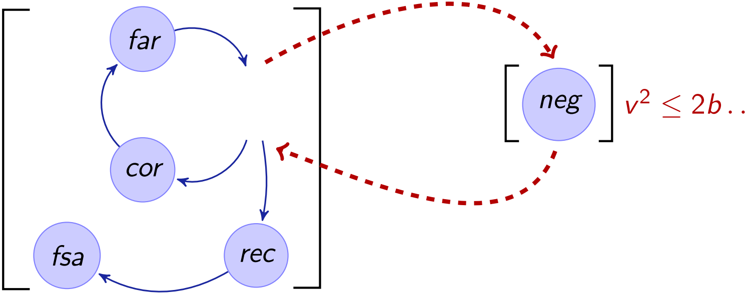
$v^2 \le 2b..$

d$\mathcal{L}$ Motives: What about Hybrid Automata?

differential dynamic logic

$$d\mathcal{L} = FOL_{\mathbb{R}} + DL + HP$$

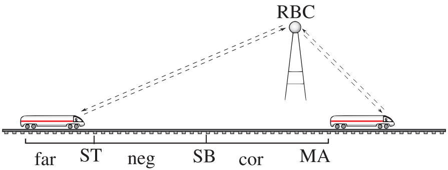


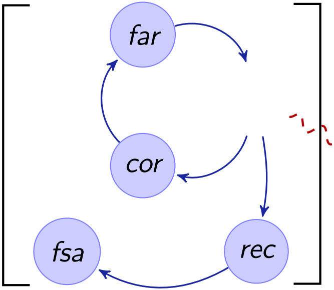


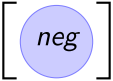
d$\mathcal{L}$ Motives: What about Hybrid Automata?

differential dynamic logic

d$\mathcal{L}$ = $\text{FOL}_\mathbb{R}$ + DL + HP







$v^2 \leq 2b\,..$

not compositional

Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Hybrid program α)

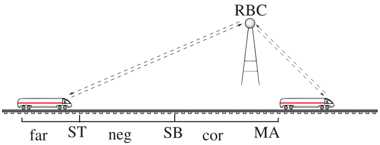
$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$
\begin{aligned}
ETCS &\equiv (ctrl\,; drive)^* \\
ctrl &\equiv (?MA - z \leq SB; a := -b) \\
&\quad \cup (?MA - z \geq SB; a := \ldots) \\
drive &\equiv \quad z'' = a \\
&\quad \wedge v \geq 0 \wedge \tau \leq \varepsilon
\end{aligned}
$$

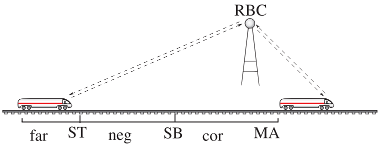


Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$ETCS \equiv (ctrl\,;drive)^*$$
$$ctrl \equiv (?MA - z \leq SB; a := -b)$$
$$\cup\,(?MA - z \geq SB; a := \dots)$$
$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$
$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon$$

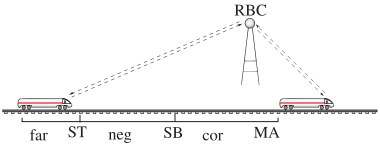


Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Hybrid program α)

$x' = f(x) \land \chi$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$
\begin{aligned}
ETCS &\equiv (ctrl\,; drive)^* \\
ctrl &\equiv (?MA - z \leq SB; a := -b) \\
&\quad \cup (?MA - z \geq SB; a := \ldots) \\
drive &\equiv \tau := 0; z' = v, v' = a, \tau' = 1 \\
&\quad \land v \geq 0 \land \tau \leq \varepsilon
\end{aligned}
$$



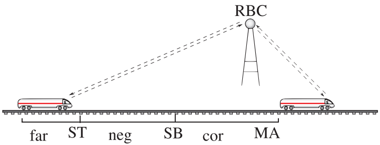
Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Formulas ϕ)

$\neg, \wedge, \vee, \rightarrow, \ \forall x, \exists x, \ =, \leq, \ +, \cdot$ ($\mathbb{R}$-first-order part)

$[\alpha]\phi, \quad \langle\alpha\rangle\phi$ (dynamic part)

$$SB \geq \ldots \ \rightarrow \ [(ctrl; drive)^*]\, z \leq MA$$



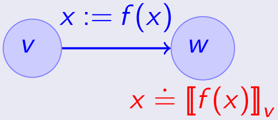
All trains respect *MA*

RBC partitions *MA*

$\Rightarrow$ system collision free

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

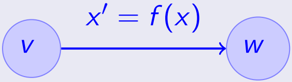
Definition (Hybrid programs α: transition semantics)



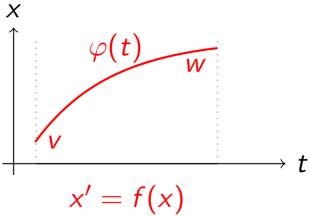
▸ Details

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

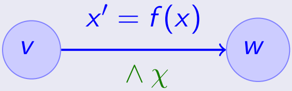


▸ Details

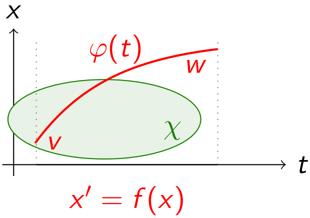


Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

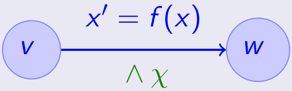


Details

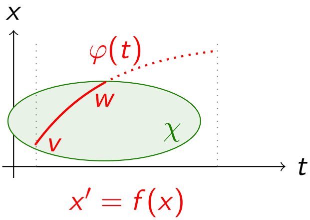


Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

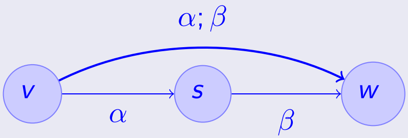


▸ Details



Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

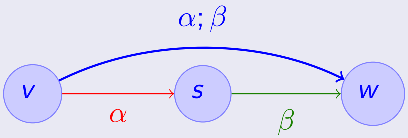
Definition (Hybrid programs α: transition semantics)



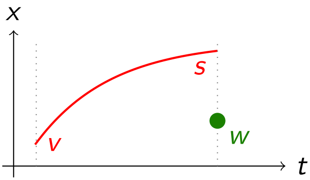
Details

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

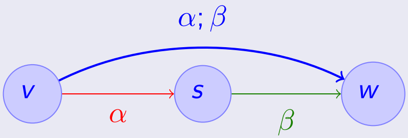


Details

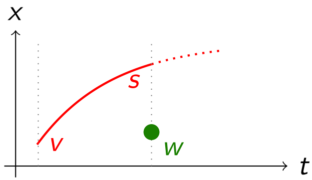


Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

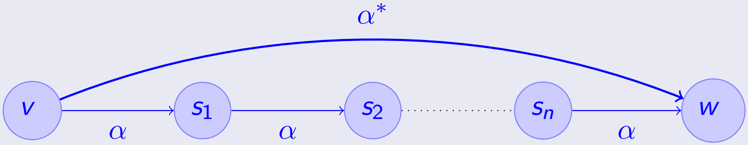


Details



Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

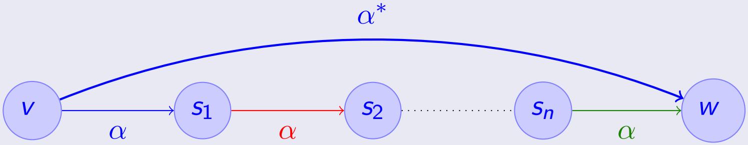
Definition (Hybrid programs α: transition semantics)



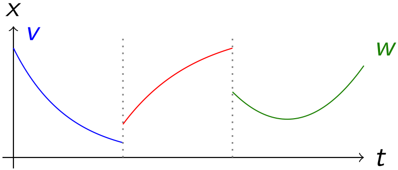
▸ Details

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

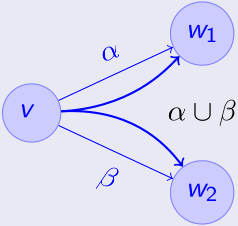


▸ Details



Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

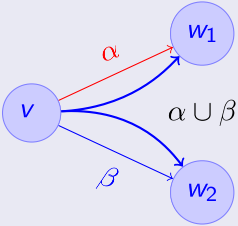
Definition (Hybrid programs α: transition semantics)



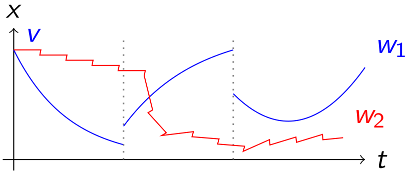
▸ Details

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)



Details



Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

if $v \models \chi$

Details

Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

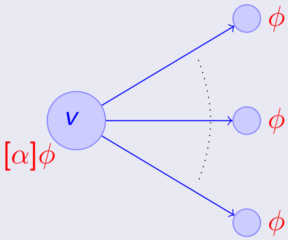
Definition (Hybrid programs α: transition semantics)

if $v \not\models \chi$

Details

Differential Dynamic Logic d$\mathcal{L}$: Semantics

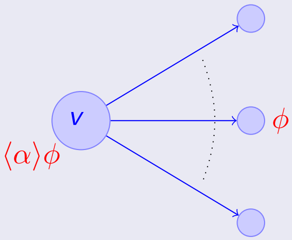
Definition (Formulas ϕ)



▸ Details

Differential Dynamic Logic d$\mathcal{L}$: Semantics

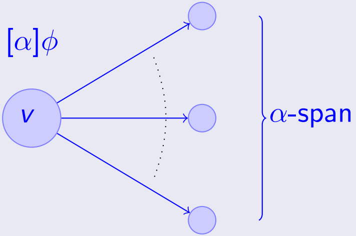
Definition (Formulas ϕ)



Details

Differential Dynamic Logic d$\mathcal{L}$: Semantics

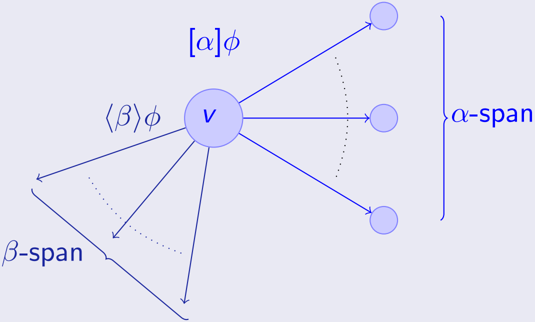
Definition (Formulas ϕ)



Details

Differential Dynamic Logic d$\mathcal{L}$: Semantics

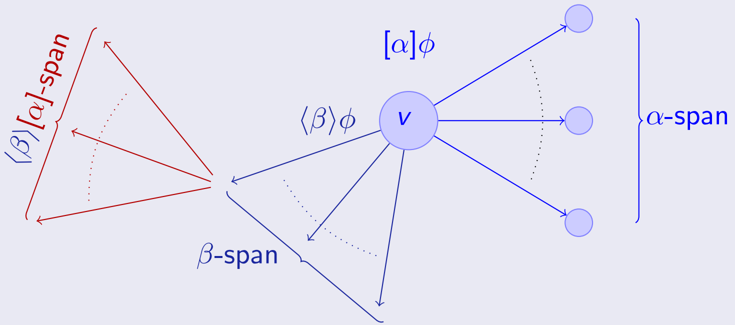
Definition (Formulas ϕ)



Details

Differential Dynamic Logic d$\mathcal{L}$: Semantics

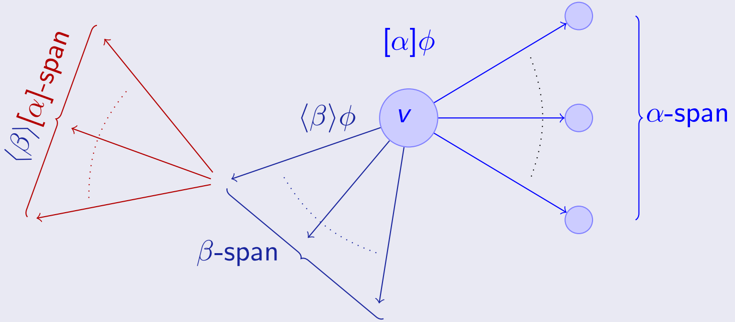
Definition (Formulas ϕ)



Details

Differential Dynamic Logic d$\mathcal{L}$: Semantics

Definition (Formulas ϕ)



Details

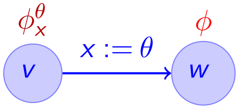
compositional semantics $\Rightarrow$ compositional calculus!

Outline (Verification Approach)

1. Motivation
2. Differential Dynamic Logic $d\mathcal{L}$
   - Design Motives
   - Syntax
   - Semantics
3. Verification Calculus for Differential Dynamic Logic $d\mathcal{L}$
   - Compositional Verification Calculus
   - Deduction Modulo by Side Deduction
   - Deduction Modulo with Free Variables & Skolemization
   - Soundness and Completeness
4. Survey
5. Conclusions & Future Work

Verification Calculus for Differential Dynamic Logic

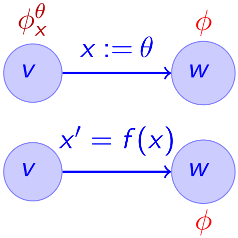
$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$



Verification Calculus for Differential Dynamic Logic

$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$

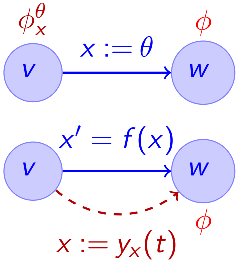
$$\frac{\exists t \geq 0\, \langle x := y_x(t) \rangle \phi}{\langle x' = f(x) \rangle \phi}$$



Verification Calculus for Differential Dynamic Logic

$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$

$$\frac{\exists t{\geq}0\, \langle x := y_x(t)\rangle\phi}{\langle x' = f(x)\rangle\phi}$$

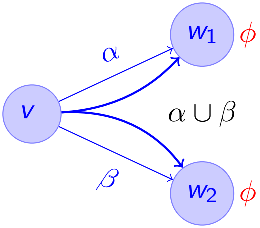


Verification Calculus for Differential Dynamic Logic

compositional semantics $\Rightarrow$ compositional rules!

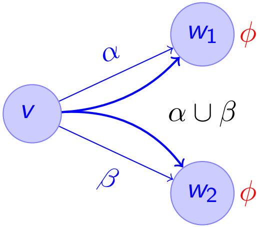
Verification Calculus for Differential Dynamic Logic

$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$

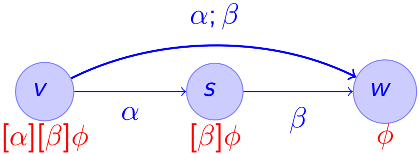


Verification Calculus for Differential Dynamic Logic

$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$

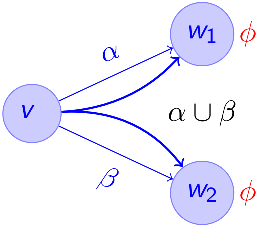


$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$

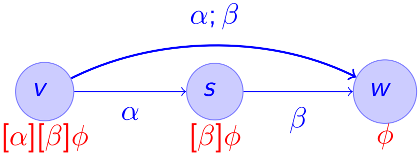


Verification Calculus for Differential Dynamic Logic

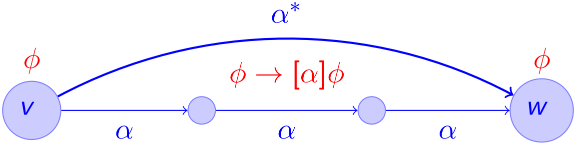
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



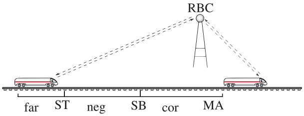
$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$



$$\frac{\vdash \phi \quad \vdash (\phi \to [\alpha]\phi)}{\vdash [\alpha^*]\phi}$$

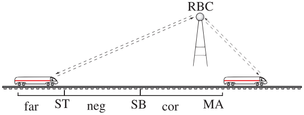


Deduction Modulo Real Arithmetic



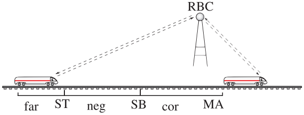
$$\vdash v \geq 0 \land z < MA \to \langle z' = v, v' = -b \rangle\ z > MA$$

Deduction Modulo Real Arithmetic



$$\dfrac{\dfrac{v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA}}{\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle\ z > MA}$$

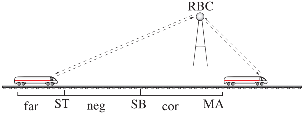
Deduction Modulo Real Arithmetic



Collins/Tarski QE not applicable!

$$\frac{\dfrac{v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA}}{\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle\ z > MA}$$

Deduction Modulo (Side Deduction)

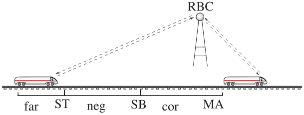


$$\frac{}{v \geq 0, z < MA \vdash t \geq 0 \land \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

start side

$$\frac{\frac{}{v \geq 0, z < MA \vdash \exists t {\geq} 0\, \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}}{\frac{v \geq 0, z < MA \vdash \langle z' = v, v' = -b \rangle z > MA}{\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b \rangle\, z > MA}}$$

Deduction Modulo (Side Deduction)



$$v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA$$

$$v \geq 0, z < MA \vdash t{\geq}0 \qquad v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z\rangle z{>}M$$

$$v \geq 0, z < MA \vdash t \geq 0 \land \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA$$

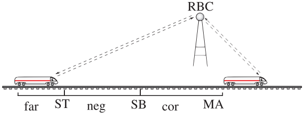
start side

$$v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA$$

$$v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA$$

$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle\ z > MA$$

Deduction Modulo (Side Deduction)



$$v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA$$

$$v \geq 0, z < MA \vdash t \geq 0 \qquad v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > M\!A$$

$$v \geq 0, z < MA \vdash t \geq 0 \land \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA$$

QE

start side

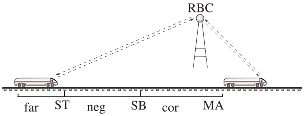
$$v \geq 0, z < MA \vdash \mathrm{QE}(\exists t\,(\ldots t \geq 0 \land -\frac{b}{2}t^2 + vt + z > MA))$$

$$v \geq 0, z < MA \vdash \exists t \geq 0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA$$

$$v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA$$

$$\vdash v \geq 0 \land z < MA \to \langle z' = v, v' = -b\rangle\, z > MA$$

Deduction Modulo (Side Deduction)



$$v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA$$

$$\dfrac{v \geq 0, z < MA \vdash t \geq 0 \qquad v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > M}{v \geq 0, z < MA \vdash t \geq 0 \land \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA}$$

QE

start side

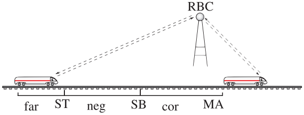
$$v \geq 0, z < MA \vdash v^2 > 2b(MA - z)$$

$$v \geq 0, z < MA \vdash \exists t \geq 0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z > MA$$

$$v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA$$

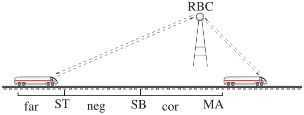
$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle\, z > MA$$

Deduction Modulo (Free Variables for Automation)



$$\frac{\dfrac{v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z{>}MA}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA}}{\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA}$$

Deduction Modulo (Free Variables for Automation)



$$v \geq 0, z < MA \vdash -\frac{b}{2}T^2 + vT + z > MA$$

$$v \geq 0, z < MA \vdash T \geq 0 \qquad v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > M$$

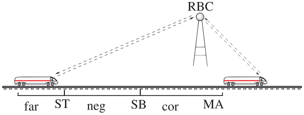
$$v \geq 0, z < MA \vdash T \geq 0 \land \langle z := -\frac{b}{2}T^2 + vT + z\rangle z > MA$$

$$v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z{>}MA$$

$$v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA$$

$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



$$v \geq 0, z < MA \vdash \quad \exists T\,(\dots T{\geq}0 \land -\tfrac{b}{2}T^2 + vT + z > MA)$$

$$v \geq 0, z < MA \vdash -\tfrac{b}{2}T^2 + vT + z > MA$$

$$v \geq 0, z < MA \vdash T{\geq}0 \qquad v \geq 0, z < MA \vdash \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > M$$

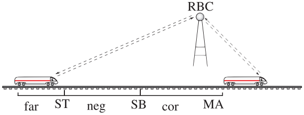
$$v \geq 0, z < MA \vdash T \geq 0 \land \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > MA$$

$$v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\tfrac{b}{2}t^2 + vt + z\rangle z{>}MA$$

$$v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA$$

$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



$$v \geq 0, z < MA \vdash \mathrm{QE}\big(\exists T\,(\ldots T{\geq}0 \land -\tfrac{b}{2}T^2 + vT + z > MA)\big)$$

$$v \geq 0, z < MA \vdash -\tfrac{b}{2}T^2 + vT + z > MA$$

$$v \geq 0, z < MA \vdash T{\geq}0 \qquad v \geq 0, z < MA \vdash \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > M$$

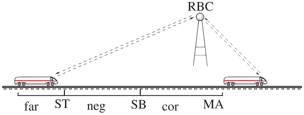
$$v \geq 0, z < MA \vdash T \geq 0 \land \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > MA$$

$$v \geq 0, z < MA \vdash \exists t{\geq}0\,\langle z := -\tfrac{b}{2}t^2 + vt + z\rangle z{>}MA$$

$$v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA$$

$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



$$v \geq 0, z < MA \vdash {\color{red} v^2 > 2b(MA - z)}$$

$$v \geq 0, z < MA \vdash -\frac{b}{2}{\color{red}T}^2 + v{\color{red}T} + z > MA$$

$$v \geq 0, z < MA \vdash {\color{red}T}{\geq}0 \qquad v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}{\color{red}T}^2 + v{\color{red}T} + z\rangle z > M$$

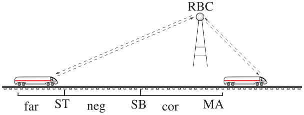
$$v \geq 0, z < MA \vdash {\color{red}T} \geq 0 \land \langle z := -\frac{b}{2}{\color{red}T}^2 + v{\color{red}T} + z\rangle z > MA$$

$$v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\frac{b}{2}t^2 + vt + z\rangle z{>}MA$$

$$v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA$$

$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)



- For requantification, not for unification

$$v \geq 0, z < MA \vdash \mathrm{QE}\big(\exists T\,(\ldots T{\geq}0 \land -\tfrac{b}{2}T^2 + vT + z > MA)\big)$$

$$v \geq 0, z < MA \vdash -\tfrac{b}{2}T^2 + vT + z > MA$$

$$v \geq 0, z < MA \vdash T{\geq}0 \qquad v \geq 0, z < MA \vdash \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > M$$

$$v \geq 0, z < MA \vdash T \geq 0 \land \langle z := -\tfrac{b}{2}T^2 + vT + z\rangle z > MA$$

$$v \geq 0, z < MA \vdash \exists t{\geq}0\, \langle z := -\tfrac{b}{2}t^2 + vt + z\rangle z{>}MA$$

$$v \geq 0, z < MA \vdash \langle z' = v, v' = -b\rangle z > MA$$

$$\vdash v \geq 0 \land z < MA \rightarrow \langle z' = v, v' = -b\rangle z > MA$$

Deduction Modulo (Free Variables for Automation)

$$\dfrac{\dfrac{\dfrac{}{\vdash (X < S)}}{\vdash \forall s\,(X < s)}}{\vdash \exists x\,\forall s\,(x < s)}$$

Deduction Modulo (Free Variables for Automation)

$$\cfrac{\cfrac{\cfrac{\cfrac{\vdash \mathrm{QE}(\forall S \exists X (X < S))}{\vdash (X < S)}}{\vdash \forall s\, (X < s)}}{\vdash \exists x\, \forall s\, (x < s)}}{}$$

Deduction Modulo (Free Variables for Automation)

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathrm{QE}(\forall S \exists X (X < S))} \qquad \dfrac{}{\vdash \mathrm{QE}(\exists X \forall S (X < S))}}{\vdash (X < S)}}{\vdash \forall s\, (X < s)}}{\vdash \exists x\, \forall s\, (x < s)}$$

Deduction Modulo (Free Variables for Automation)

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textit{true}}{\vdash \mathrm{QE}(\forall S \exists X (X < S))} \qquad \dfrac{\textit{false}}{\vdash \mathrm{QE}(\exists X \forall S (X < S))}}{\vdash (X < S)}}{\vdash \forall s\, (X < s)}}{\vdash \exists x\, \forall s\, (x < s)}}{\textit{false}!}
$$

Deduction Modulo (Free Variables for Automation)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textit{true}}{\vdash \mathrm{QE}(\forall S\exists X(X<S))} \qquad \dfrac{\textit{false}}{\vdash \mathrm{QE}(\exists X\forall S(X<S))}}{\vdash (X<S)}}{\vdash \forall s\,(X<s)}}{\vdash \exists x\,\forall s\,(x<s)}}{\textit{false}!}$$

Deduction Modulo (Free Variables & Skolemisation)

Skolemisation $S(X)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textit{false}}{\vdash \mathrm{QE}(\exists X \forall S\, (X < S))}}{\vdash (X < S(X))}}{\vdash \forall s\, (X < s)}}{\vdash \exists x\, \forall s\, (x < s)}}{\textit{false}!}$$

Soundness and Completeness

Theorem (Relative Completeness)

d$\mathcal{L}$ *calculus is a sound & complete axiomatisation of hybrid systems relative to differential equations.*

Proof Outline 15p

Soundness and Completeness

Theorem (Relative Completeness)

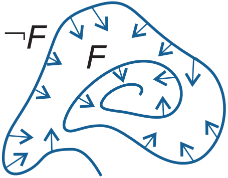
d$\mathcal{L}$ *calculus is a sound & complete axiomatisation of hybrid systems relative to differential equations.*

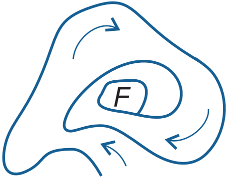
▸ Proof Outline 15p

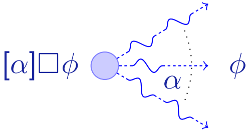
Corollary (Proof-theoretical Alignment)

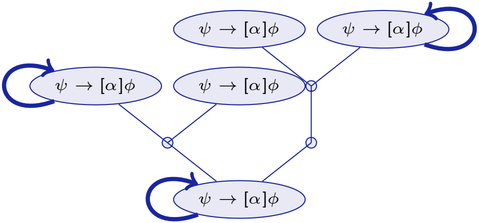
verification of hybrid systems = verification of dynamical systems!

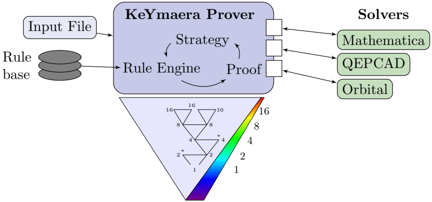
Outline

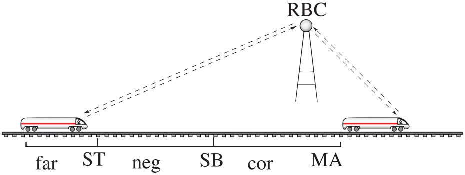

¬F
F

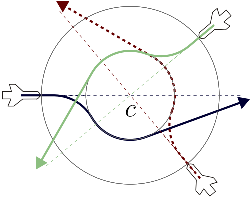

F


$[\alpha]\Box\phi$
α
ϕ


$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$
$\psi \to [\alpha]\phi$


KeYmaera Prover
Input File
Rule base
Strategy
Rule Engine
Proof
Solvers
Mathematica
QEPCAD
Orbital
16
8
4
2
1


RBC
far
ST
neg
SB
cor
MA


c

Experimental Results

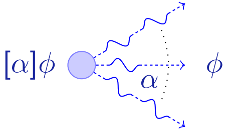
Case Study	Interact	Time(s)	Mem(Mb)	Steps	Dim
ETCS-kernel	0	10.5	24.2	58	9
	1	2.8	14.2	61	9
ETCS-binary safety	0	18.6	12.4	204	14
	1	7.2	15.8	235	14
ETCS controllability	0	0.6	6.9	14	5
SB reactivity	0	103.9	61.7	47	14
ETCS liveness	4	35.2	92.2	62	10
Roundabout(2)	0	9.9	6.8	197	13
	3	1.9	6.7	139	13
Roundabout(3)	0	636.2	15.1	342	18
Roundabout(4)	0	884.9	31.4	520	23
Roundabout(5)	0	3552.6	46.9	735	28
	3	108.9	43.6	503	28
flyable roundabout entry*	0	10.1	9.6	132	8

Outline

Conclusions

differential dynamic logic

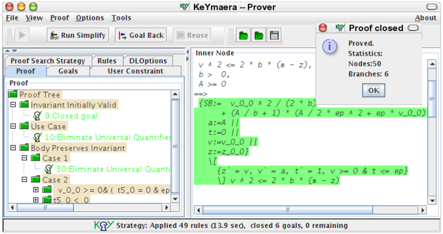
$d\mathcal{L} = \mathrm{DL} + \mathrm{HP}$



Verifying parametric hybrid systems:

- Logics for hybrid systems
- Compositional calculi
- $\mathbb{R}$-Skolem for automation
- Sound & complete / ODE
- Differential invariants
- Verification algorithms
- Challenging case studies

KeYmaera



Landscape


Logic-based Verification of Hybrid Systems
Logic
Proof Calculus
Theorem Proving
Model Checking
Algebra
Computer Algebra
Algebraic Geometry
Differential Algebra
Algorithms
Fixedpoint Loops
Proof Strategies
Decision Procedures
Calculus
Differential Equations
Dynamical Systems
Differen-tiation

Outline

Outline

	Op	Par	T	Cl	Tec	Aut	Cex	Dim	
HenzingerH94, HyTech	✓	×	✓	×	✓	✓	✓		LHA
LafferrierePY99	✓	×	✓	×	✓		✓		forgetful reset
Fränzle99	✓	×	✓	×	✓		✓	×	robust systems
CKrogh03, CheckMate	✓	×	✓	×	✓	✓	✓		polyhedral
Frehse05, PHAVer	✓	×	✓	×	✓	✓	✓	8	LHA (+affine)
MysorePM05	✓	×	✓	×	✓	•	✓	4	bounded prefix
TomlinPS98,MBT05	○	×	×	×	○	○	•	4	HJB numPDE
RatschanS07, HSolver	✓	×		×	✓	✓	×	4	interval
MannaS98, STeP	✓			×	✓	○	×	7	inv↦VCG, flat
ÁbrahámSH01, PVS	•			×	•	○	×	≈9	HA↪PVS, -“-
ZhouRH92, EDC	×	•	✓	..	×	×	×	×	no maths
DavorenN00, Lμ	×	×		✓	○	×	×	×	prop. H-semantics
RönkköRS03, HGC	✓	×	×	×	×	×	×	×	HGC↪HOL
SSManna04	•	○		×	✓		×	4/1	equational system
CTiwari05	•	○		×	✓		×	6/0	linear, -”-
PrajnaJP07, barrier	•	×		×	•		×	3	needs 10000-dim
d$\mathcal{L}$ & dTL	✓	✓	✓	✓	✓	•	×	28	expr., compos.

	Dom	Op	Base	Modal	Quant	Cmpl	Aut
DL	$\mathbb{N}$		$\mathrm{FOL}_{(\mathbb{N})}$		FV+unify	$/\mathbb{N}$	
d$\mathcal{L}$	$\mathbb{R}$	x'	$\mathrm{FOL}_{\mathbb{R}}$	ODE	FV+requant+QE	/ODE	IBC

Differential Dynamic Logic d$\mathcal{L}$: Formal Semantics

Definition (Kripke state)

$v : V \to \mathbb{R}$ with set of variables V

Return

Differential Dynamic Logic d$\mathcal{L}$: Formal Semantics

Definition (Formulas ϕ)

$$v \models [\alpha]\phi \quad :\iff \quad w \models \phi \text{ for all } w \text{ with } (v,w) \in \rho(\alpha)$$
$$v \models \langle\alpha\rangle\phi \quad :\iff \quad w \models \phi \text{ for some } w \text{ with } (v,w) \in \rho(\alpha)$$

Definition (Hybrid programs α)

$$\begin{aligned}
\rho(x' = f(x)) &= \{(\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \text{ for duration } r\} \\
(v,w) \in \rho(x := \theta) &:\iff w = v[x \mapsto [\![\theta]\!]_v] \\
\rho(?\chi) &= \{(v,v) : v \models \chi\} \\
\rho(\alpha \cup \gamma) &= \rho(\alpha) \cup \rho(\gamma) \\
\rho(\alpha;\gamma) &= \rho(\alpha) \circ \rho(\gamma) \\
(v,w) \in \rho(\alpha^*) &:\iff \text{there is } v \xrightarrow{\rho(\alpha)} v_1 \xrightarrow{\rho(\alpha)} v_2 \cdots\cdots\cdots \xrightarrow{\rho(\alpha)} w
\end{aligned}$$

Return

Differential Dynamic Logic $d\mathcal{L}$: Formal Semantics

Definition (Formulas ϕ)

$$v \models [\alpha]\phi \quad :\iff \quad w \models \phi \quad \text{for all } w \text{ with } (v,w) \in \rho(\alpha)$$

$$v \models \langle\alpha\rangle\phi \quad :\iff \quad w \models \phi \quad \text{for some } w \text{ with } (v,w) \in \rho(\alpha)$$

Definition (Hybrid programs α)

$$\rho(x' = f(x)) \quad = \quad \{(\varphi(0), \varphi(r)) \;:\; \varphi \models x' = f(x) \text{ for duration } r\}$$

$$(v,w) \in \rho(x := \theta) \quad :\iff \quad w = v[x \mapsto [\![\theta]\!]_v]$$

$$\rho(?\chi) \quad = \quad \{(v,v) \;:\; v \models \chi\}$$

$$\rho(\alpha \cup \gamma) \quad = \quad \rho(\alpha) \cup \rho(\gamma)$$

$$\rho(\alpha;\gamma) \quad = \quad \rho(\alpha) \circ \rho(\gamma)$$

$$(v,w) \in \rho(\alpha^*) \quad :\iff \quad \text{there is} \quad v \xrightarrow{\rho(\alpha)} v_1 \xrightarrow{\rho(\alpha)} v_2 \cdots\cdots \xrightarrow{\rho(\alpha)} w$$

◂ Return

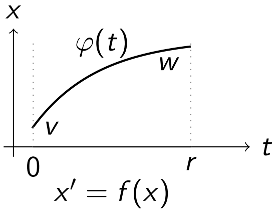
Differential Dynamic Logic d$\mathcal{L}$: Formal Semantics

Definition (Hybrid programs α)

$$\rho(x' = f(x)) = \{(\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \text{ for duration } r\}$$

with $[\![x']\!]_{\varphi(\zeta)} = \frac{d\varphi(t)(x)}{dt}(\zeta)$

- there is $\varphi : [0, r] \to \text{States}$ with $\varphi(0) = v, \varphi(r) = w$
- $[\![x]\!]_{\varphi(\zeta)}$ is continuous in ζ on $[0, r]$
- $\frac{d\,[\![x]\!]_{\varphi(t)}}{dt}(\zeta) = [\![f(x)]\!]_{\varphi(\zeta)}$ for $\zeta \in (0, r)$
- $[\![y]\!]_{\varphi(\zeta)} = [\![y]\!]_v$ otherwise



Return

Soundness

Proof (Soundness).

- $x' = f(x)$
- Side deductions
- Free variables & Skolemisation

$\square$

Return

Incompleteness

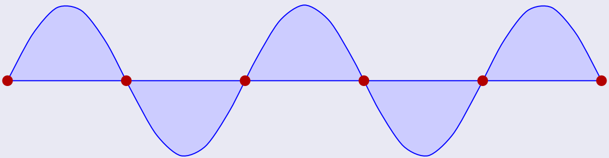
Proof (Incompleteness).

Discrete fragment:

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

Continuous fragment:

$$\langle s'' = -s, \tau' = 1 \rangle (s = 0 \wedge \tau = n) \qquad \rightsquigarrow s = \sin$$



□

Return

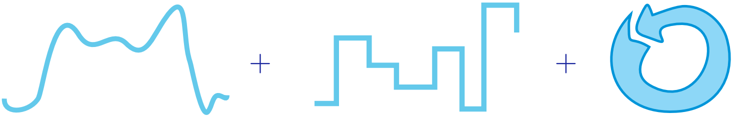
Incomplete! But are we missing proof rules?

Incomplete! But are we missing proof rules?

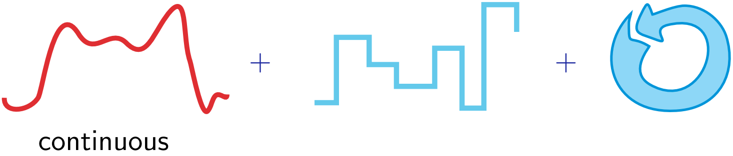
Relativity

Cook,Harel: discrete-DL/data$_{\mathbb{N}}$ hybrid-d$\mathcal{L}$/data$_{\mathbb{R}}$??

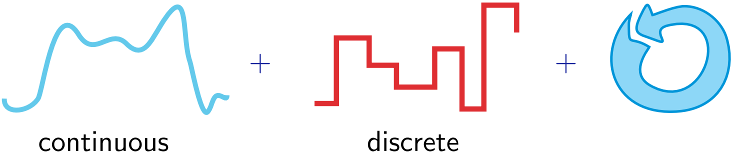
Sources of Incompleteness



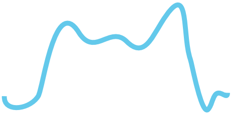
Sources of Incompleteness



Sources of Incompleteness

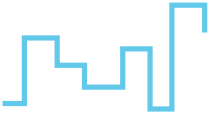


Sources of Incompleteness



continuous

+

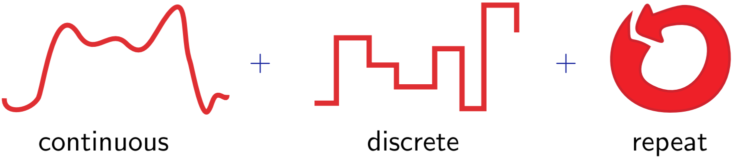


discrete

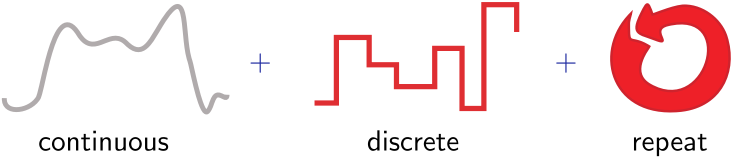
+

repeat

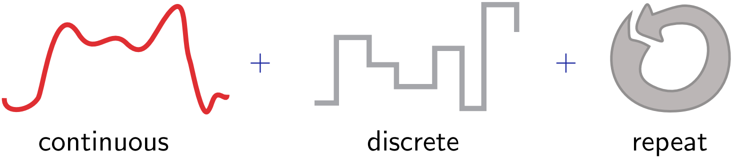
Sources of Incompleteness



Relative Completeness



Relative Completeness



Relative Completeness

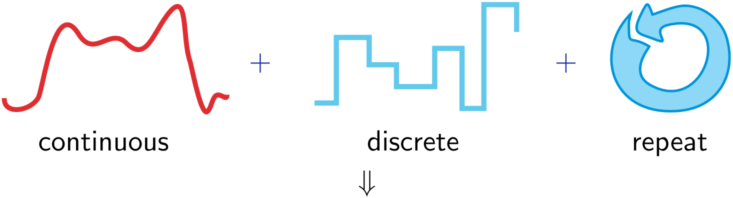
Theorem (Relative Completeness)

d$\mathcal{L}$ calculus is complete relative to first-order logic of differential equations.

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Proof Outline 15p



Relative Completeness

Theorem (Relative Completeness)

d$\mathcal{L}$ calculus is complete relative to first-order logic of differential equations.

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathsf{FOD}} \vdash \phi$$

where $\mathsf{FOD} = \mathsf{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

▸ Proof Outline 15p

$\Downarrow$

Relativity

Cook,Harel: discrete-DL/data P.: hybrid-d$\mathcal{L}$/differential equations

Relative Completeness

Theorem (Relative Completeness)

d$\mathcal{L}$ calculus is complete relative to first-order logic of differential equations.

$$\vDash \phi \quad \text{iff} \quad Taut_{\text{FOD}} \vdash \phi$$

where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$ ▸ Proof Outline 15p

$\Downarrow$

Corollary (Proof-theoretical Alignment)

verification of hybrid systems = verification of dynamical systems!

Relative Completeness

Theorem (Relative Completeness)

d$\mathcal{L}$ calculus is complete relative to first-order logic of differential equations.

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$ ▸ Proof Outline 15p

$\Downarrow$

Corollary (Deductive Power)

d$\mathcal{L}$ calculus is *supremal hybrid* verification technique

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\quad \mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Return

Proof (Relative Completeness, 10 pages).

1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition:
   for each ϕ there is $F \in \mathrm{FOD}$ $\vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \rightarrow [\alpha]G$
9. Relative complete for first-order liveness $F \rightarrow \langle\alpha\rangle G$

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

◂ Return

Proof (Relative Completeness, 10 pages).

1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition:
   for each ϕ there is $F \in \mathrm{FOD}$ $\vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \rightarrow [\alpha]G$
9. Relative complete for first-order liveness $F \rightarrow \langle\alpha\rangle G$

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\quad \mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$ ◂ Return

Proof (Relative Completeness, 10 pages).

1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition: for each ϕ there is $F \in \mathrm{FOD}$ $\vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \to [\alpha]G$
9. Relative complete for first-order liveness $F \to \langle\alpha\rangle G$

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \dots, x_n' = \theta_n]F$

Return

Proof (Relative Completeness, 10 pages).

1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition:
   for each ϕ there is $F \in \mathrm{FOD}$ $\vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \to [\alpha]G$
9. Relative complete for first-order liveness $F \to \langle\alpha\rangle G$

Relative Completeness Proof

$$\vDash \phi \quad \text{iff} \quad Taut_{\mathrm{FOD}} \vdash \phi$$

where $\mathrm{FOD} = \mathrm{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Return

Proof (Relative Completeness, 10 pages).

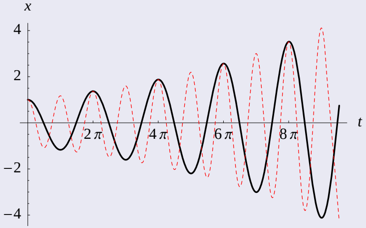
1. Strong invariants and variants expressible in d$\mathcal{L}$
2. d$\mathcal{L}$ expressible in FOD
3. valid d$\mathcal{L}$ formulas d$\mathcal{L}$-derivable from corresponding FOD axioms
4. finite FOD formula characterising unbounded hybrid repetition
5. FOD characterises $\mathbb{R}$-Gödel encoding
6. First-order expressible & program rendition:
   for each ϕ there is $F \in \mathrm{FOD}$ $\vDash \phi \leftrightarrow F$
7. Propositionally & first-order complete
8. Relative complete for first-order safety $F \rightarrow [\alpha]G$
9. Relative complete for first-order liveness $F \rightarrow \langle\alpha\rangle G$

Relative Completeness Proof

where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$ ◂ Return

Proof ($\mathbb{R}$-Gödel encoding).

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$



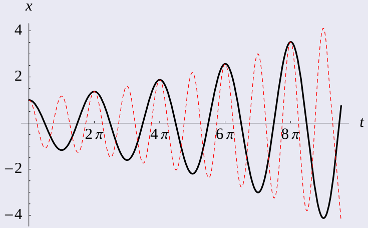
□

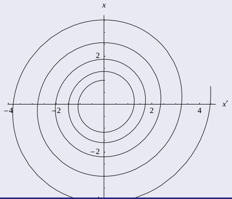
Relative Completeness Proof

where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$ ◂ Return

Proof ($\mathbb{R}$-Gödel encoding).

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$





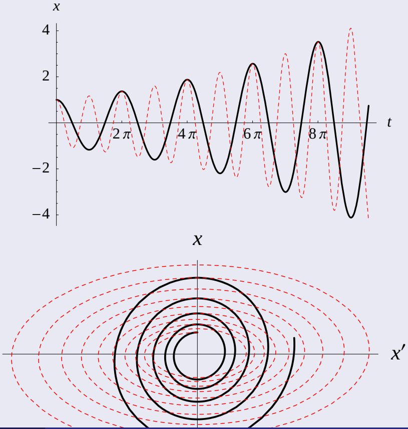
Relative Completeness Proof

where $\quad \text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Return

Proof ($\mathbb{R}$-Gödel encoding).

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$



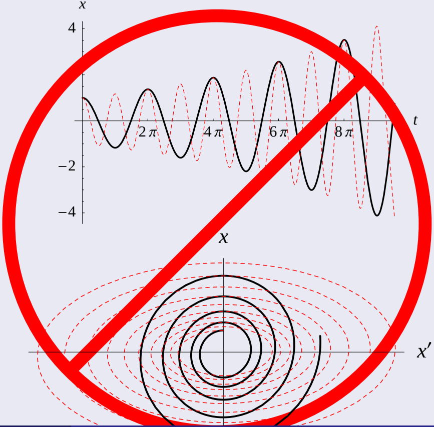
Relative Completeness Proof

where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$

Return

Proof ($\mathbb{R}$-Gödel encoding).

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$ not differentiable!



Relative Completeness Proof

where $\quad \text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \ldots, x_n' = \theta_n]F$ ◂ Return

Proof ($\mathbb{R}$-Gödel encoding).

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$

$$\sum_{i=1}^{\infty} \frac{a_i}{2^i} = 0.a_1 a_2 \ldots$$

$$\sum_{i=1}^{\infty} \frac{b_i}{2^i} = 0.b_1 b_2 \ldots$$

$$\sum_{i=0}^{\infty} \left(\frac{a_i}{2^{2i+1}} + \frac{b_i}{2^{2i+2}} \right) = 0.a_1 b_1 a_2 b_2 \ldots$$

Relative Completeness Proof

where $\text{FOD} = \text{FOL}_{\mathbb{R}} + [x_1' = \theta_1, \dots, x_n' = \theta_n]F$

Return

Proof ($\mathbb{R}$-Gödel encoding).

FOD characterises constructive bijection $\mathbb{R} \to \mathbb{R}^2$

$$\sum_{i=1}^{\infty} \frac{a_i}{2^i} = 0.a_1 a_2 \dots$$

$$\sum_{i=1}^{\infty} \frac{b_i}{2^i} = 0.b_1 b_2 \dots$$

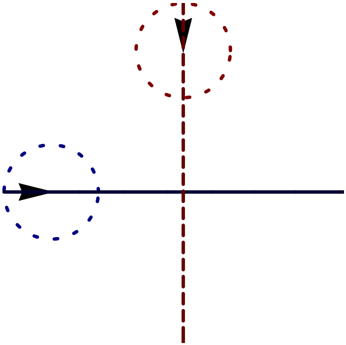
$$\sum_{i=0}^{\infty} \left(\frac{a_i}{2^{2i+1}} + \frac{b_i}{2^{2i+2}} \right) = 0.a_1 b_1 a_2 b_2 \dots$$

$$2^n = z \quad \leftrightarrow \quad \langle x := 1; \tau := 0; x' = x \ln 2 \land \tau' = 1 \rangle (\tau = n \land x = z)$$

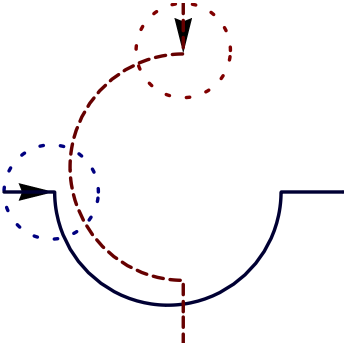
$$\ln 2 = z \quad \leftrightarrow \quad \langle x := 1; \tau := 0; x' = x \land \tau' = 1 \rangle (x = 2 \land \tau = z)$$

Outline

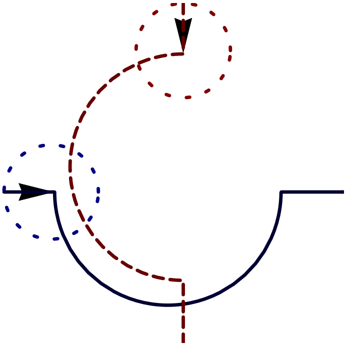
Air Traffic Control



Air Traffic Control



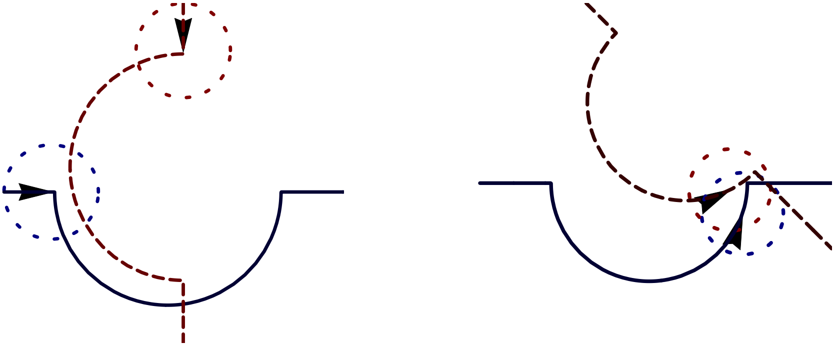
Air Traffic Control



Verification?

looks correct

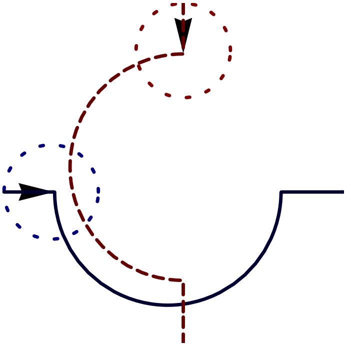
Air Traffic Control

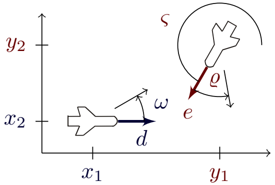


Verification?

looks correct NO!

Air Traffic Control



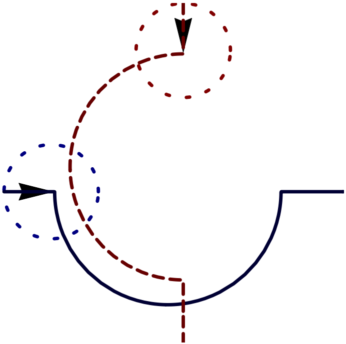


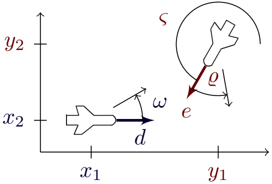
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!

Air Traffic Control



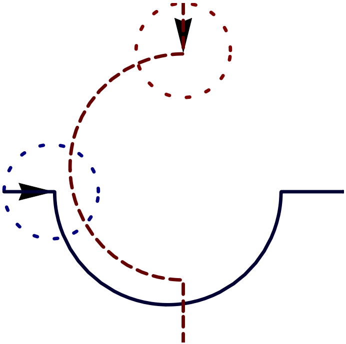


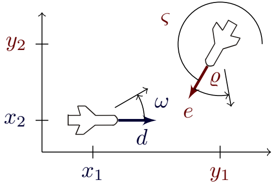
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

$$x_1(t) = \frac{1}{\omega\varpi}\big(x_1\omega\varpi\cos t\omega - v_2\omega\cos t\omega\sin\vartheta + v_2\omega\cos t\omega\cos t\varpi\sin\vartheta - v_1\varpi\sin t\omega$$
$$+ x_2\omega\varpi\sin t\omega - v_2\omega\cos\vartheta\cos t\varpi\sin t\omega - v_2\omega\sqrt{1-\sin\vartheta^2}\sin t\omega$$
$$+ v_2\omega\cos\vartheta\cos t\omega\sin t\varpi + v_2\omega\sin\vartheta\sin t\omega\sin t\varpi\big)\ldots$$

Air Traffic Control





$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

$$\forall t \geq 0 \quad \frac{1}{\omega\varpi}\big(x_1\omega\varpi \cos t\omega - v_2\omega \cos t\omega \sin\vartheta + v_2\omega \cos t\omega \cos t\varpi \sin\vartheta - v_1\varpi \sin t\omega$$

$$+ x_2\omega\varpi \sin t\omega - v_2\omega \cos\vartheta \cos t\varpi \sin t\omega - v_2\omega\sqrt{1 - \sin\vartheta^2} \sin t\omega$$

$$+ v_2\omega \cos\vartheta \cos t\omega \sin t\varpi + v_2\omega \sin\vartheta \sin t\omega \sin t\varpi\big) \ldots$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

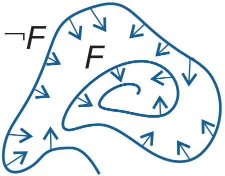
F closed under total differentiation with respect to differential constraints

▸ Details

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints



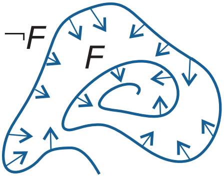
$$\frac{\vdash (\chi \to F')}{\chi \to F \vdash [x' = \theta \land \chi]F}$$

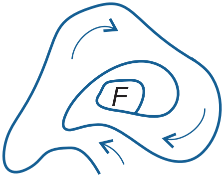
Details

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





Details

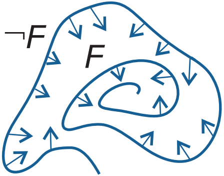
$$\frac{\vdash (\chi \rightarrow F')}{\chi \rightarrow F \vdash [x' = \theta \land \chi]F}$$

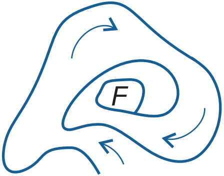
$$\frac{\vdash (\neg F \land \chi \rightarrow F'_{\gg})}{[x' = \theta \land \neg F]\chi \vdash \langle x' = \theta \land \chi \rangle F}$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





▸ Details

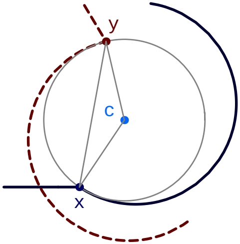
$$\frac{\vdash (\chi \to F')}{\chi \to F \vdash [x' = \theta \land \chi]F}$$

$$\frac{\vdash (\neg F \land \chi \to F'_{\gg})}{[x' = \theta \land \neg F]\chi \vdash \langle x' = \theta \land \chi \rangle F}$$

Total differential F' of *formulas*?

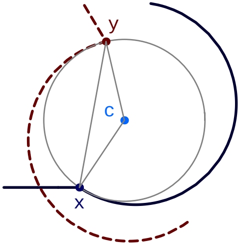
Differential Induction for Aircraft Roundabouts

$$\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$



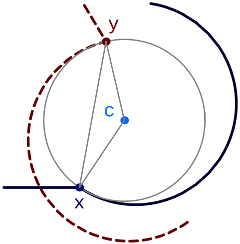
Differential Induction for Aircraft Roundabouts

$$\frac{\vdash \frac{\partial\|x-y\|^2}{\partial x_1}x_1' + \frac{\partial\|x-y\|^2}{\partial y_1}y_1' + \frac{\partial\|x-y\|^2}{\partial x_2}x_2' + \frac{\partial\|x-y\|^2}{\partial y_2}y_2' \geq \frac{\partial p^2}{\partial x_1}x_1' \dots}{\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}$$



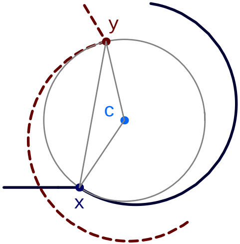
Differential Induction for Aircraft Roundabouts

$$\frac{\vdash \frac{\partial\|x-y\|^2}{\partial x_1} x_1' + \frac{\partial\|x-y\|^2}{\partial y_1} y_1' + \frac{\partial\|x-y\|^2}{\partial x_2} x_2' + \frac{\partial\|x-y\|^2}{\partial y_2} y_2' \geq \frac{\partial p^2}{\partial x_1} x_1' \dots}{\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}$$



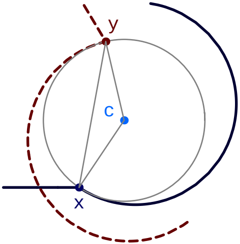
Differential Induction for Aircraft Roundabouts

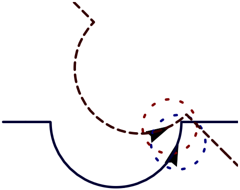
$$\frac{\vdash \frac{\partial\|x-y\|^2}{\partial x_1} d_1 + \frac{\partial\|x-y\|^2}{\partial y_1} e_1 + \frac{\partial\|x-y\|^2}{\partial x_2} d_2 + \frac{\partial\|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots}{\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}$$



Differential Induction for Aircraft Roundabouts

$$\frac{\dfrac{\dfrac{\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0}{\vdash \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots}}{\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}}{}$$



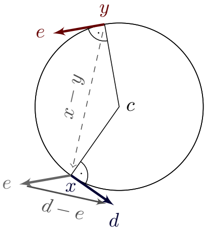


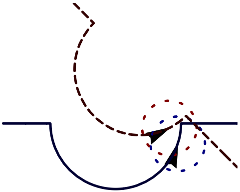
Differential Induction for Aircraft Roundabouts

$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

$$\vdash \frac{\partial\|x-y\|^2}{\partial x_1} d_1 + \frac{\partial\|x-y\|^2}{\partial y_1} e_1 + \frac{\partial\|x-y\|^2}{\partial x_2} d_2 + \frac{\partial\|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots$$

$$\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$



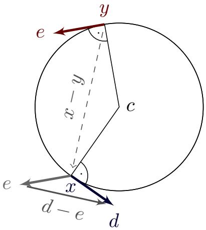


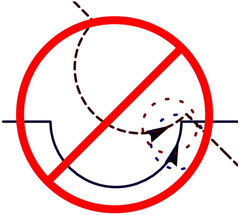
Differential Induction for Aircraft Roundabouts

$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

$$\vdash \frac{\partial\|x-y\|^2}{\partial x_1} d_1 + \frac{\partial\|x-y\|^2}{\partial y_1} e_1 + \frac{\partial\|x-y\|^2}{\partial x_2} d_2 + \frac{\partial\|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots$$

$$\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$

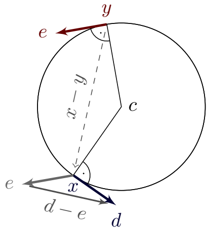




$$.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..] d_1 - e_1 = -\omega(x_2 - y_2)$$

Differential Induction for Aircraft Roundabouts

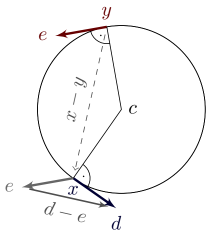
$$\frac{\dfrac{\dfrac{\vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0}{\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0}}{\vdash \frac{\partial\|x-y\|^2}{\partial x_1} d_1 + \frac{\partial\|x-y\|^2}{\partial y_1} e_1 + \frac{\partial\|x-y\|^2}{\partial x_2} d_2 + \frac{\partial\|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots}}{\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}$$



$$\overline{.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..] d_1 - e_1 = -\omega(x_2 - y_2)}$$

Differential Induction for Aircraft Roundabouts

$$\frac{\frac{\frac{\vdash 2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0}{\vdash 2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}}{\vdash \frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\dots}}{\vdash [x_1'=d_1, d_1'=-\omega d_2, x_2'=d_2, d_2'=\omega d_1, ..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2}$$



$$\frac{\vdash \frac{\partial(d_1-e_1)}{\partial d_1}d_1'+\frac{\partial(d_1-e_1)}{\partial e_1}e_1'=-\frac{\partial\omega(x_2-y_2)}{\partial x_2}x_2'-\frac{\partial\omega(x_2-y_2)}{\partial y_2}y_2'}{..\vdash [d_1'=-\omega d_2, e_1'=-\omega e_2, x_2'=d_2, d_2'=\omega d_1, ..]\,d_1-e_1=-\omega(x_2-y_2)}$$

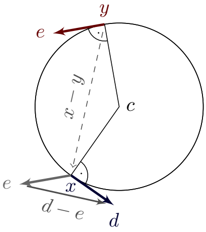
Differential Induction for Aircraft Roundabouts

$$\vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0$$

$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

$$\vdash \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots$$

$$\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$

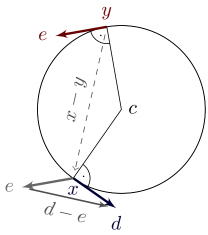


$$\vdash \frac{\partial(d_1 - e_1)}{\partial d_1} d_1' + \frac{\partial(d_1 - e_1)}{\partial e_1} e_1' = -\frac{\partial \omega(x_2 - y_2)}{\partial x_2} x_2' - \frac{\partial \omega(x_2 - y_2)}{\partial y_2} y_2'$$

$$.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..]d_1 - e_1 = -\omega(x_2 - y_2)$$

Differential Induction for Aircraft Roundabouts

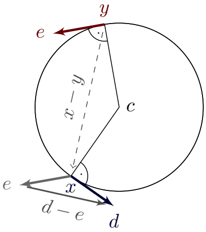
$$\dfrac{\dfrac{\dfrac{\vdash 2(x_1-y_1)(-\omega(x_2-y_2))+2(x_2-y_2)\omega(x_1-y_1)\geq 0}{\vdash 2(x_1-y_1)(d_1-e_1)+2(x_2-y_2)(d_2-e_2)\geq 0}}{\vdash \frac{\partial\|x-y\|^2}{\partial x_1}d_1+\frac{\partial\|x-y\|^2}{\partial y_1}e_1+\frac{\partial\|x-y\|^2}{\partial x_2}d_2+\frac{\partial\|x-y\|^2}{\partial y_2}e_2\geq\frac{\partial p^2}{\partial x_1}d_1\dots}}{\vdash [x_1'=d_1,d_1'=-\omega d_2,x_2'=d_2,d_2'=\omega d_1,..](x_1-y_1)^2+(x_2-y_2)^2\geq p^2}$$



$$\dfrac{\dfrac{}{\vdash \frac{\partial(d_1-e_1)}{\partial d_1}(-\omega d_2)+\frac{\partial(d_1-e_1)}{\partial e_1}(-\omega e_2)=-\frac{\partial\omega(x_2-y_2)}{\partial x_2}d_2-\frac{\partial\omega(x_2-y_2)}{\partial y_2}e_2}}{..\vdash [d_1'=-\omega d_2,e_1'=-\omega e_2,x_2'=d_2,d_2'=\omega d_1,..]d_1-e_1=-\omega(x_2-y_2)}$$

Differential Induction for Aircraft Roundabouts

$$\dfrac{\dfrac{\dfrac{\vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0}{\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0}}{\vdash \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots}}{\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}$$



$$\dfrac{\dfrac{\vdash -\omega d_2 + \omega e_2 = -\omega(d_2 - e_2)}{\vdash \frac{\partial(d_1 - e_1)}{\partial d_1}(-\omega d_2) + \frac{\partial(d_1 - e_1)}{\partial e_1}(-\omega e_2) = -\frac{\partial \omega(x_2 - y_2)}{\partial x_2} d_2 - \frac{\partial \omega(x_2 - y_2)}{\partial y_2} e_2}}{.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..] d_1 - e_1 = -\omega(x_2 - y_2)}$$

Differential Induction & Differential Saturation

$$\dfrac{\dfrac{\dfrac{\vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0}{\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0}}{\vdash \frac{\partial\|x-y\|^2}{\partial x_1} d_1 + \frac{\partial\|x-y\|^2}{\partial y_1} e_1 + \frac{\partial\|x-y\|^2}{\partial x_2} d_2 + \frac{\partial\|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots}}{\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2}$$

Proposition (Differential saturation)

F differential invariant of $[x' = \theta \wedge H]\phi$, then

$$[x' = \theta \wedge H]\phi \quad \text{iff} \quad [x' = \theta \wedge H \wedge F]\phi$$

$$\dfrac{\dfrac{\vdash -\omega d_2 + \omega e_2 = -\omega(d_2 - e_2)}{\vdash \frac{\partial(d_1-e_1)}{\partial d_1}(-\omega d_2) + \frac{\partial(d_1-e_1)}{\partial e_1}(-\omega e_2) = -\frac{\partial\omega(x_2-y_2)}{\partial x_2} d_2 - \frac{\partial\omega(x_2-y_2)}{\partial y_2} e_2}}{.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..]d_1 - e_1 = -\omega(x_2 - y_2)}$$

Differential Induction & Differential Saturation

$$\vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0$$

$$\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0$$

$$\vdash \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots$$

$$\vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$$

refine dynamics

by differential saturation

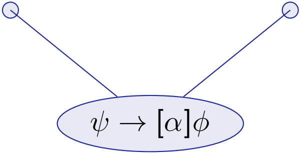
$$\vdash -\omega d_2 + \omega e_2 = -\omega(d_2 - e_2)$$

$$\vdash \frac{\partial (d_1 - e_1)}{\partial d_1}(-\omega d_2) + \frac{\partial (d_1 - e_1)}{\partial e_1}(-\omega e_2) = -\frac{\partial \omega(x_2 - y_2)}{\partial x_2} d_2 - \frac{\partial \omega(x_2 - y_2)}{\partial y_2} e_2$$

$$.. \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, ..] d_1 - e_1 = -\omega(x_2 - y_2)$$

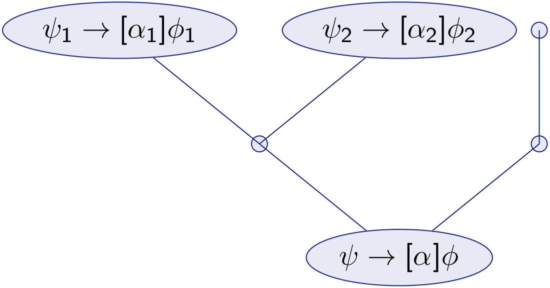
Outline

Differential Invariants as Fixedpoints



Details

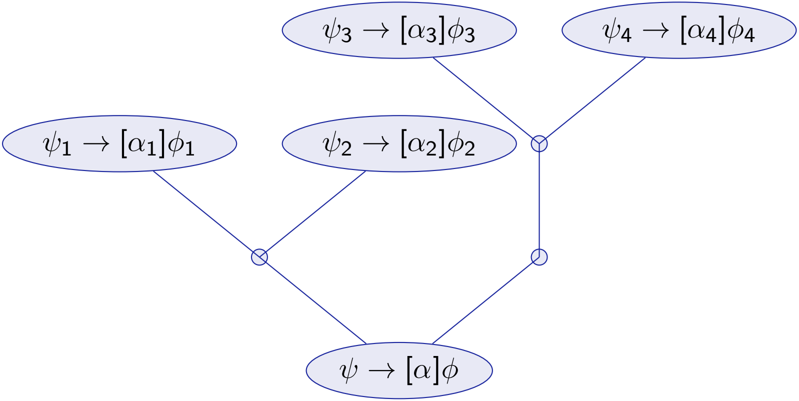
Differential Invariants as Fixedpoints



for $\cup\,,;\,,:=$ do decompose

▸ Details

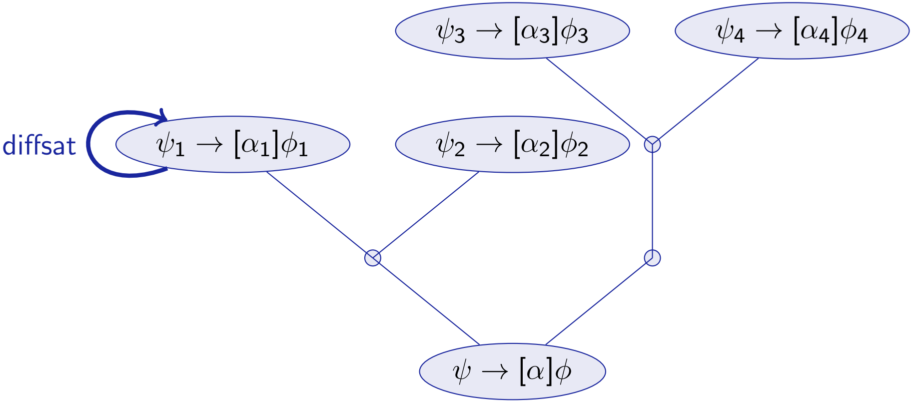
Differential Invariants as Fixedpoints



for $\cup$, ; , := do decompose

▸ Details

Differential Invariants as Fixedpoints

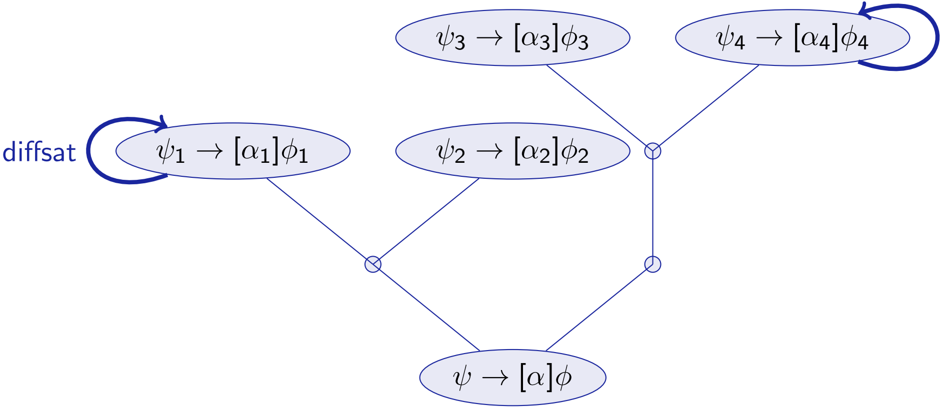


for $\cup\, ,\, ;\, ,:=$ do decompose

for $x' = \ldots$ do diffsat

Details

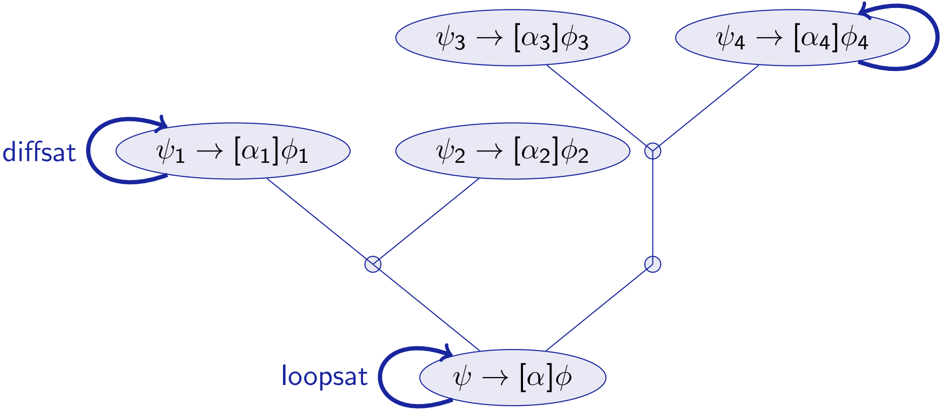
Differential Invariants as Fixedpoints



for $\cup\, , ; , :=$	do decompose
for $x' = \ldots$	do diffsat

▸ Details

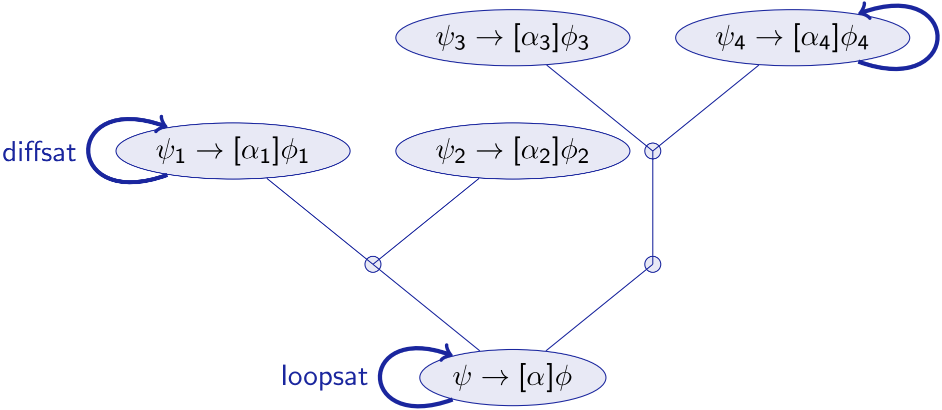
Differential Invariants as Fixedpoints



for $\cup\,,;\,,:=$	do decompose
for $x' = \ldots$	do diffsat
for α^*	do loopsat

Details

Differential Invariants as Fixedpoints



for $\cup$, ; , :=	do decompose	
for $x' = \ldots$	do diffsat	repeat until fixedpoint
for α^*	do loopsat	

▸ Details

Differential Induction Principle

$$\sigma_1 \quad \mapsto \quad [\![F]\!]_{\sigma_1}$$

Return

Differential Induction Principle

$$\begin{aligned} \sigma_1 &\mapsto [\![F]\!]_{\sigma_1} \\ \sigma_2 &\mapsto [\![F]\!]_{\sigma_2} \end{aligned}$$

◂ Return

Differential Induction Principle

$$\begin{array}{rcl} \sigma_1 & \mapsto & [\![F]\!]_{\sigma_1} \\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2} \end{array}$$

In the limit:

$$\frac{\mathrm{d}\,[\![F]\!]_\sigma}{\mathrm{d}\sigma}$$

◂ Return

Differential Induction Principle

$$\begin{array}{rcl} \sigma_1 & \mapsto & [\![F]\!]_{\sigma_1} \\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2} \end{array}$$

In the limit:

$$\frac{\mathrm{d}\,[\![F]\!]_{\sigma(t)}}{\mathrm{d}t}$$

where $\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t}$ according to ODE

Return

Differential Induction Principle

$$\begin{array}{rcl} \sigma_1 & \mapsto & [\![F]\!]_{\sigma_1} \\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2} \end{array}$$

In the limit:

$$\frac{\mathrm{d}\,[\![F]\!]_{\sigma(t)}}{\mathrm{d}t}(\zeta) = [\![F']\!]_{\bar{\sigma}(\zeta)}$$

where $\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t}$ according to ODE

Return

Differential Induction Principle

$$\begin{array}{rcl}\sigma_1 & \mapsto & [\![F]\!]_{\sigma_1}\\ \sigma_2 & \mapsto & [\![F]\!]_{\sigma_2}\end{array}$$

In the limit:

$$\frac{\mathrm{d}\,[\![F]\!]_{\sigma(t)}}{\mathrm{d}t}(\zeta) = [\![F']\!]_{\bar{\sigma}(\zeta)}$$

where $\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t}$ according to ODE

Lemma (Derivation lemma)

Valuation is a differential homomorphism

◂ Return

Derivations and Differentiation

Definition (Syntactic total derivation $D : \mathrm{Trm}(\Sigma \cup \Sigma') \to \mathrm{Trm}(\Sigma \cup \Sigma')$)

$$D(r) = 0 \quad \text{if } r \text{ is a (rigid) number symbol}$$

$$D(x^{(n)}) = x^{(n+1)} \quad \text{if } x \in \Sigma \text{ is non-rigid, } n \geq 0$$

$$D(a + b) = D(a) + D(b)$$

$$D(a \cdot b) = D(a) \cdot b + a \cdot D(b)$$

$$D(a/b) = (D(a) \cdot b - a \cdot D(b))/b^2$$

$$D(F) \equiv \bigwedge_{i=1}^{m} D(F_i) \quad \{F_1, \dots, F_m\} \text{ all literals of } F$$

$$D(a \geq b) \equiv D(a) \geq D(b) \quad \text{accordingly for } <, >, \leq, =$$

◂ Return

Derivations and Differentiation

Lemma (Derivation lemma)

Valuation is a differential homomorphism: for all flows φ *all* $\zeta \in [0, r]$

$$\frac{\mathrm{d}\,[\![\theta]\!]_{\varphi(t)}}{\mathrm{d}t}(\zeta) = [\![D(\theta)]\!]_{\bar{\varphi}(\zeta)}$$

Lemma (Differential substitution principle)

If $\varphi \models x_i' = \theta_i \land \chi$, *then* $\varphi \models \mathcal{D} \leftrightarrow (\chi \rightarrow \mathcal{D}_{x_i'}^{\theta_i})$ *for all* $\mathcal{D}$.

Definition (Differential Invariant)

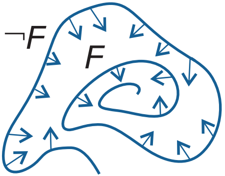
$$(\chi \rightarrow F') \;\equiv\; \chi \rightarrow D(F)_{x_i'}^{\theta_i} \qquad \text{for } [x_i' = \theta_i \land \chi]F$$

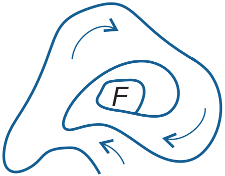
Return

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





Details

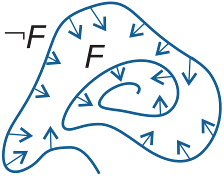
$$\frac{\vdash (\chi \rightarrow F')}{\chi \rightarrow F \vdash [x' = \theta \land \chi]F} \qquad \frac{\vdash (\neg F \land \chi \rightarrow F'_{\gg})}{[x' = \theta \land \neg F]\chi \vdash \langle x' = \theta \land \chi \rangle F}$$

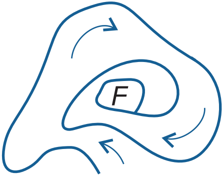
$$\begin{aligned}(d_1^2 + d_2^2 \geq a^2)' &\equiv \frac{\partial(d_1^2 + d_2^2)}{\partial d_1} d_1' + \frac{\partial(d_1^2 + d_2^2)}{\partial d_2} d_2' \geq \frac{\partial a^2}{\partial d_1} d_1' + \frac{\partial a^2}{\partial d_2} d_2' \\ &\equiv 2d_1(-\omega d_2) + 2d_2(\omega d_1) \geq 0 \\ \text{for} \quad & d_1' = -\omega d_2 \quad d_2' = \omega d_1\end{aligned}$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





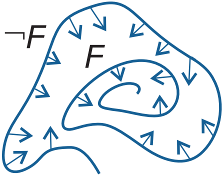
▸ Details

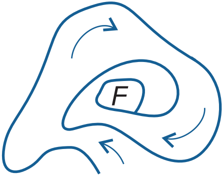
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1; \quad {\color{red} d_1' = -\omega d_2, d_2' = \omega d_1} \quad]\, d_1 \geq d_2$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





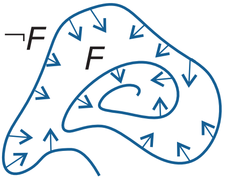
▸ Details

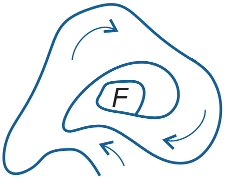
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$(d_1' = -\omega d_2 \wedge d_2' = \omega d_1) \vee (d_1' \leq 2d_1)$$
$$]\, d_1 \geq d_2$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





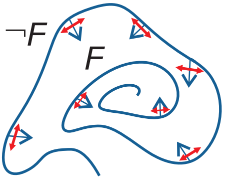
▸ Details

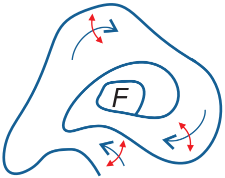
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$\exists \omega\, (\omega \leq 1 \wedge d_1' = -\omega d_2 \wedge d_2' = \omega d_1) \vee (d_1' \leq 2 d_1)$$
$$]\, d_1 \geq d_2$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





▸ Details

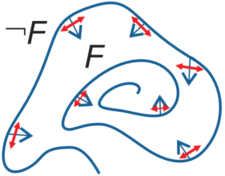
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;$$
$$\exists \omega \, (\omega \leq 1 \land d_1' = -\omega d_2 \land d_2' = \omega d_1) \lor (d_1' \leq 2 d_1)$$
$$]\, d_1 \geq d_2$$

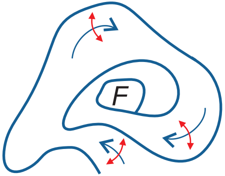
- quantified nondeterminism/disturbance

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





▸ Details

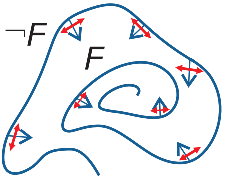
$$d_1 \geq d_2 \rightarrow [x := a^2 + 1;\ \exists \omega\, (\omega \leq 1 \land d_1' = -\omega d_2 \land d_2' = \omega d_1) \lor (d_1' \leq 2d_1)\ ]\, d_1 \geq d_2$$

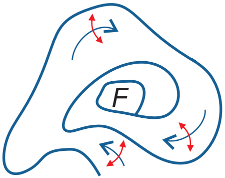
- quantified nondeterminism/disturbance

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints





▸ Details

$$d_1 \geq d_2 \rightarrow [x > 0 \rightarrow \exists a\,(a < 5 \land x := a^2 + 1);\ \exists \omega\,(\omega \leq 1 \land d_1' = -\omega d_2 \land d_2' = \omega d_1) \lor (d_1' \leq 2 d_1)\ ]\, d_1 \geq d_2$$

- discrete quantified nondeterminism/disturbance

Differential Invariants and Variants

Counterexample

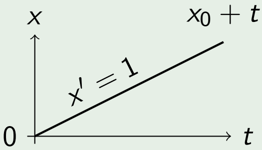
$$\frac{\vdash \forall x\,(x^2 \leq 0 \rightarrow 2x \cdot 1 \leq 0)}{x^2 \leq 0 \vdash [x' = 1]x^2 \leq 0}$$

$$\frac{\vdash \forall x\,(x > 0 \rightarrow -x < 0)}{\vdash \langle x' = -x \rangle x \leq 0}$$

Differential Invariants and Variants

Counterexample

$$\frac{\vdash \forall x\,(x^2 \leq 0 \rightarrow 2x \cdot 1 \leq 0)}{x^2 \leq 0 \vdash [x' = 1]x^2 \leq 0}$$

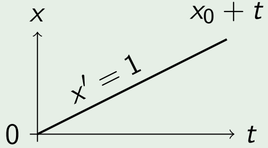


$$\frac{\vdash \forall x\,(x > 0 \rightarrow -x < 0)}{\vdash \langle x' = -x\rangle x \leq 0}$$

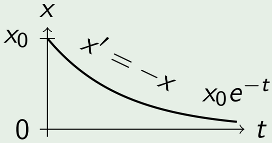
Differential Invariants and Variants

Counterexample

$$\frac{\vdash \forall x\,(x^2 \leq 0 \rightarrow 2x \cdot 1 \leq 0)}{x^2 \leq 0 \vdash [x' = 1]x^2 \leq 0}$$



$$\frac{\vdash \forall x\,(x > 0 \rightarrow -x < 0)}{\vdash \langle x' = -x\rangle x \leq 0}$$



Differential Saturation Procedure

refine d$\mathcal{L}$ verification calculus to automatic verification fixedpoint algorithm

$\leadsto$

```
function prove(ψ ⊢ [D ∧ H]φ):
2: if prove((H → φ)) then
     return true /* property proven */
   for each F ∈ Candidates(ψ ⊢ [D ∧ H]φ, H) do
     if prove(ψ ∧ H ⊢ F) and prove((H → F')) then
       H := H ∧ F /* refine by differential invariant */
       goto 2;    /* repeat fixedpoint loop */
   end for
   return "not provable using candidates"
```

Return

Outline

Temporal Modalities + Dynamic Modalities

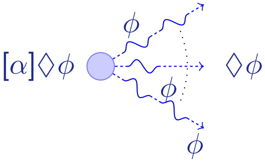
problem	technique	Op	Par	T	closed
$ETCS \models z < MA$	TL-MC	✓	×	✓	×
$\models (\mathrm{Ax}(ETCS) \rightarrow z < MA)$	TL-calculus	×	...	✓	...
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓
$\models [ETCS]\Box\, z < MA$	dTL-calculus	✓	✓	✓	✓

Temporal Modalities + Dynamic Modalities

problem	technique	Op	Par	T	closed
$ETCS \models z < MA$	TL-MC	✓	×	✓	×
$\models (\mathrm{Ax}(ETCS) \to z < MA)$	TL-calculus	×	...	✓	...
$\models [ETCS]\, z < MA$	DL-calculus	✓	✓	×	✓
$\models [ETCS]\Box\, z < MA$	dTL-calculus	✓	✓	✓	✓

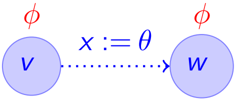
differential temporal dynamic logic

$$\mathrm{dTL} = \mathsf{TL} + \mathsf{DL} + \mathsf{HP}$$



Modular Verification Calculus for Temporal dTL

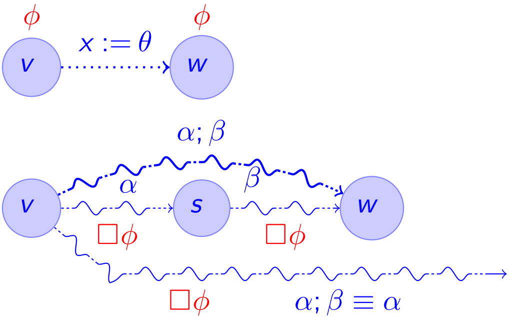
$$\frac{\phi \land [x := \theta]\phi}{[x := \theta]\Box\phi}$$



Modular Verification Calculus for Temporal dTL

$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$

$$\frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha; \beta]\Box\phi}$$

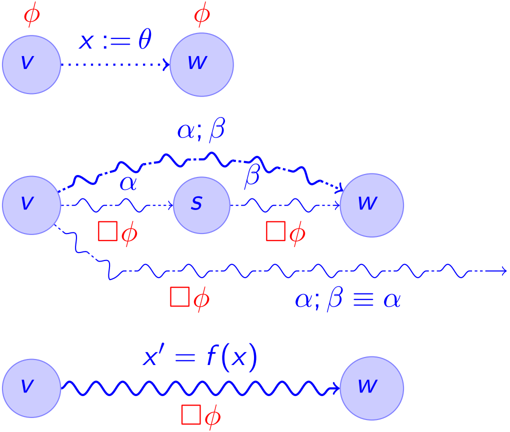


Modular Verification Calculus for Temporal dTL

$$\frac{\phi \land [x := \theta]\phi}{[x := \theta]\Box\phi}$$

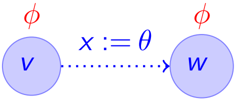
$$\frac{[\alpha]\Box\phi \land [\alpha][\beta]\Box\phi}{[\alpha; \beta]\Box\phi}$$

$$\frac{[x' = \theta]\phi}{[x' = \theta]\Box\phi}$$

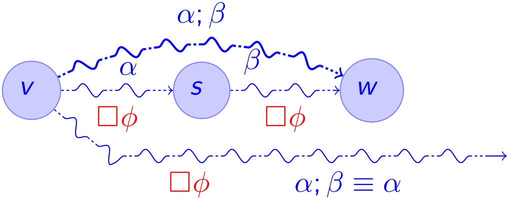


Modular Verification Calculus for Temporal dTL

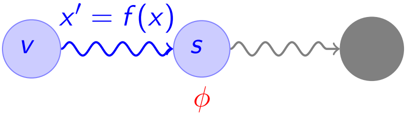
$$\frac{\phi \land [x := \theta]\phi}{[x := \theta]\Box\phi}$$



$$\frac{[\alpha]\Box\phi \land [\alpha][\beta]\Box\phi}{[\alpha; \beta]\Box\phi}$$

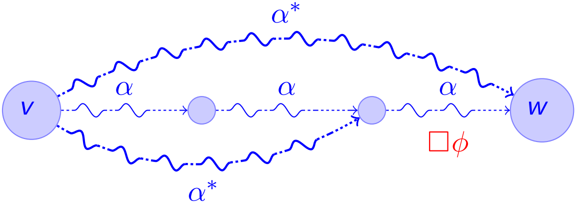


$$\frac{[x' = \theta]\phi}{[x' = \theta]\Box\phi}$$



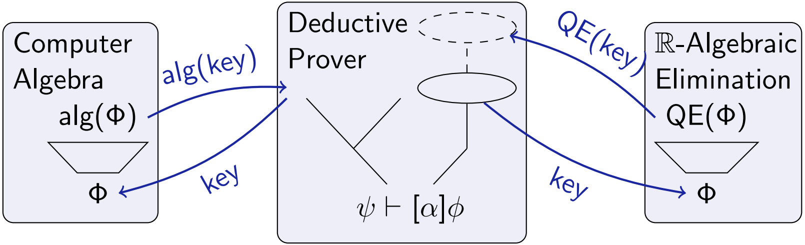
Modular Verification Calculus for Temporal dTL

$$\frac{[\alpha^*][\alpha]\Box\phi}{[\alpha^*]\Box\phi}$$

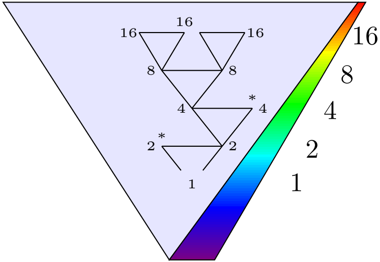


Outline (Algorithmic Refinement)

KeYmaera Verification Architecture

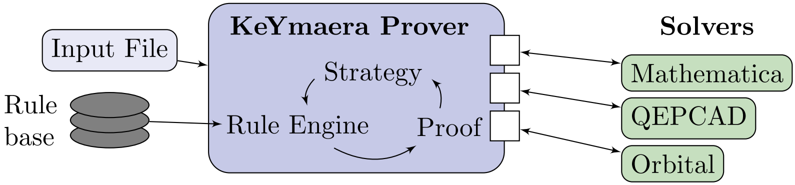


56 interactions?



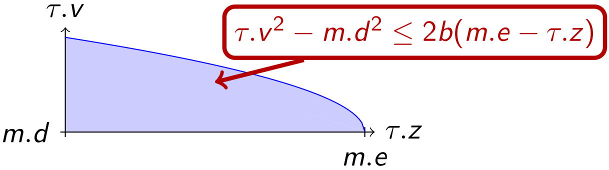
0–1 interactions!

KeYmaera Prover Architecture



Outline

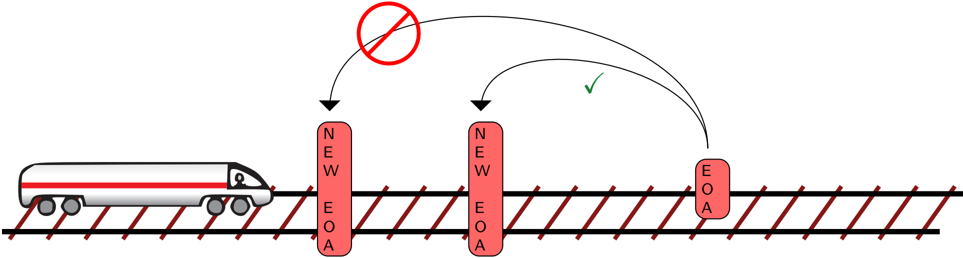
ETCS Controllability



Proposition (Controllability)

$$[\tau.z' = \tau.v, \tau.v' = -b \wedge \tau.v \geq 0](\tau.z \geq m.e \rightarrow \tau.v \leq m.d)$$
$$\equiv \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)$$

ETCS RBC Controllability



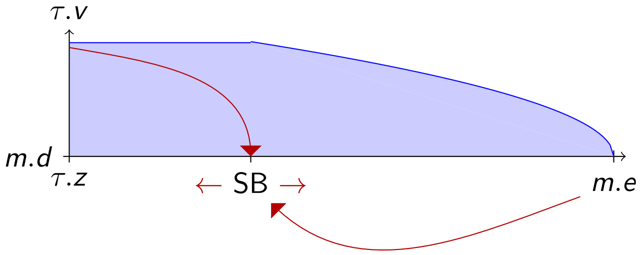
Proposition (RBC Controllability)

$m.d \geq 0 \land b > 0 \rightarrow [m_0 := m;\ RBC] \Big($

$m_0.d^2 - m.d^2 \leq 2b(m.e - m_0.e) \land m_0.d \geq 0 \land m.d \geq 0 \leftrightarrow$

$\forall \tau \left((\langle m := m_0 \rangle \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)) \rightarrow \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.\right.$

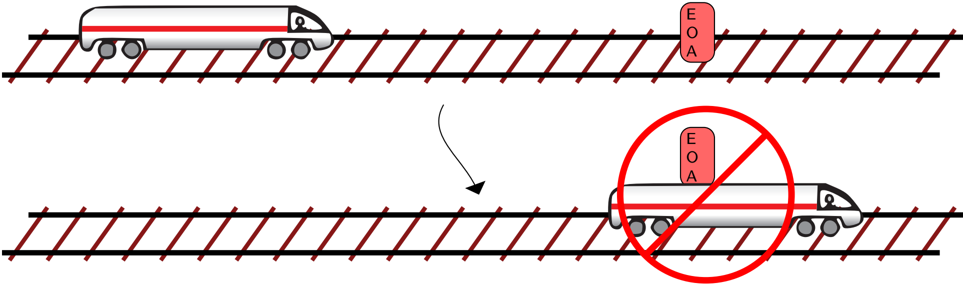
ETCS Reactivity



Proposition (Reactivity)

$$\Big(\forall m.e\,\forall \tau.z\,\big(m.e - \tau.z \geq SB \land \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z) \rightarrow$$
$$[\tau.a := A;\ drive]\,\tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z)\big)\Big)$$
$$\equiv SB \geq \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right)$$

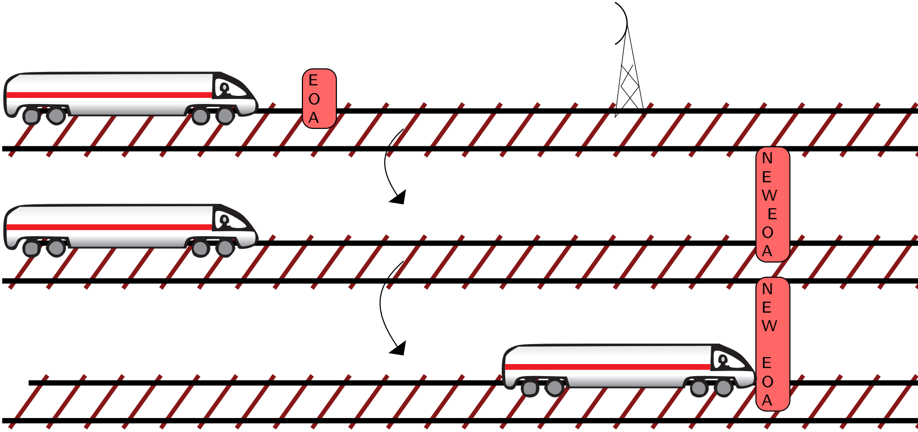
ETCS Safety



Proposition (Safety)

$$\tau.v^2 - m.d^2 \leq 2b(m.e - \tau.z) \rightarrow$$
$$[ETCS](\tau.z \geq m.e \rightarrow \tau.v \leq m.d)$$

ETCS Liveness



Proposition (Liveness)

$$\tau.v > 0 \land \varepsilon > 0 \rightarrow \forall P \, \langle ETCS \rangle \, \tau.z \geq P$$

Full European Train Control System (ETCS)

provable automatically!

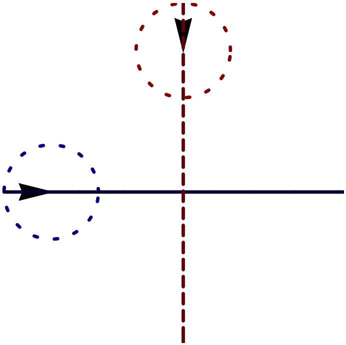
$$
\begin{array}{lll}
\text{spec} & : & \tau.v^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \tau.p) \land \tau.v \ge 0 \land \mathbf{m}.d \ge 0 \land b > 0 \\
& & \to \ [\text{ETCS}](\tau.p \ge \mathbf{m}.e \to \tau.v \le \mathbf{m}.d) \\
\text{ETCS} & : & (\text{train} \cup \text{rbc})^* \\
\text{train} & : & \text{spd};\ \text{atp};\ \text{move} \\
\text{spd} & : & \ (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le A) \\
& & \cup(?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ?0 > \tau.a \ge -b) \\
\text{atp} & : & SB := \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right); \\
& & \ (?(\mathbf{m}.e - \tau.p \le SB \lor rbc.message = emergency);\ \tau.a := -b) \\
& & \cup(?\mathbf{m}.e - \tau.p \ge SB \land rbc.message \ne emergency) \\
\text{move} & : & t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \land \tau.v \ge 0 \land t \le \varepsilon) \\
\text{rbc} & : & \ (rbc.message := emergency) \\
& & \cup\ \big(\mathbf{m}_0 := \mathbf{m}; \mathbf{m} := *; \\
& & \quad ?\mathbf{m}.r \ge 0 \land \mathbf{m}.d \ge 0 \land \mathbf{m}_0.d^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \mathbf{m}_0.e)\big)
\end{array}
$$

Full European Train Control System (ETCS)

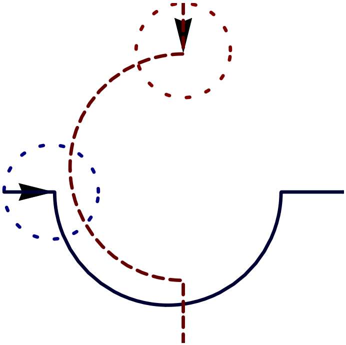
```
 state = 0,
 2 * b * (m - z) >= v ^ 2 - d ^ 2,
 v >= 0, d >= 0, v >= 0, ep >  0, b >  0, amax >  0, d >= 0
==>
    v <= vdes
 -> \forall R a_3;
      (   a_3 >= 0 & a_3 <= amax
       ->   (       m - z
                <= (amax / b + 1) * ep * v
                 + (v ^ 2 - d ^ 2) / (2 * b)
                 + (amax / b + 1) * amax * ep ^ 2 / 2
             -> \forall R t0;
                  (   t0 >= 0
                   -> \forall R ts0;  (0 <= ts0 & ts0 <= t0 -> -b * ts0 + v >= 0 & ts0 + 0 <= ep)
                   ->      2 * b * (m - 1 / 2 * (-b * t0 ^ 2 + 2 * t0 * v + 2 * z))
                        >= (-b * t0 + v) ^ 2
                         - d ^ 2
                      & -b * t0 + v >= 0
                      & d >= 0))
          & (       m - z
                >  (amax / b + 1) * ep * v
                 + (v ^ 2 - d ^ 2) / (2 * b)
                 + (amax / b + 1) * amax * ep ^ 2 / 2
             -> \forall R t2;
                  (   t2 >= 0
                   -> \forall R ts2;  (0 <= ts2 & ts2 <= t2 -> a_3 * ts2 + v >= 0 & ts2 + 0 <= ep)
                   ->      2 * b * (m - 1 / 2 * (a_3 * t2 ^ 2 + 2 * t2 * v + 2 * z))
                        >= (a_3 * t2 + v) ^ 2
                         - d ^ 2
                      & a_3 * t2 + v >= 0
                      & d >= 0)))
```

Outline

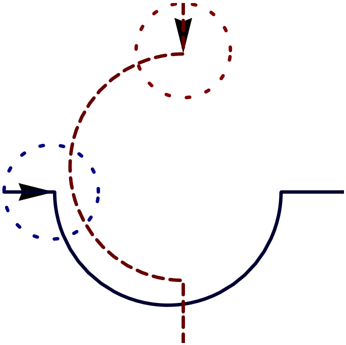
Air Traffic Control



Air Traffic Control



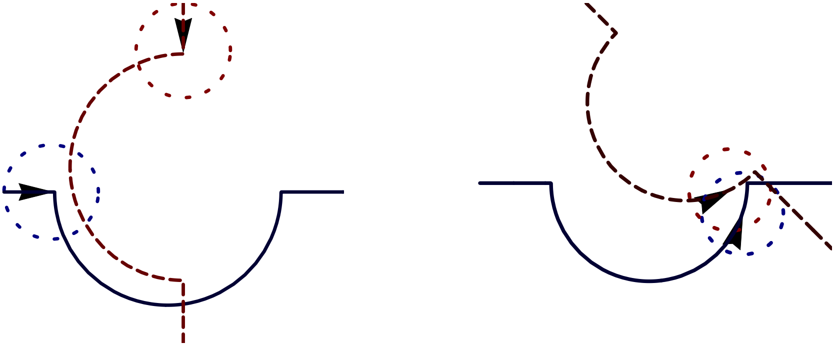
Air Traffic Control



Verification?

looks correct

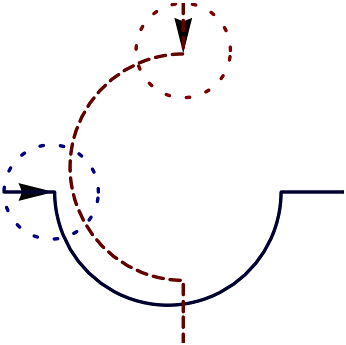
Air Traffic Control

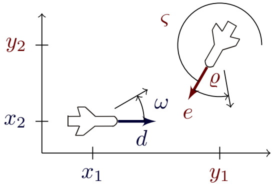


Verification?

looks correct NO!

Air Traffic Control



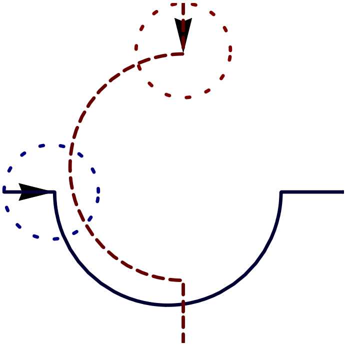


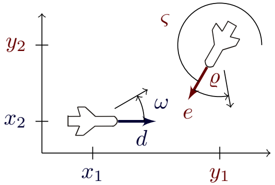
$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!

Air Traffic Control





$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

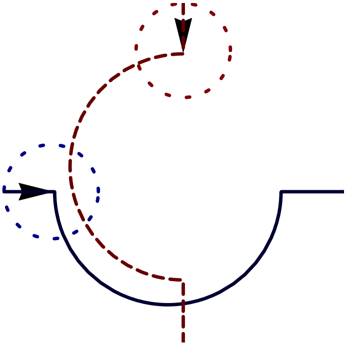
Example ("Solving" differential equations)

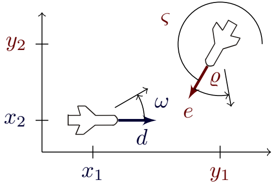
$$x_1(t) = \frac{1}{\omega\varpi}\big(x_1\omega\varpi\cos t\omega - v_2\omega\cos t\omega\sin\vartheta + v_2\omega\cos t\omega\cos t\varpi\sin\vartheta - v_1\varpi\sin t\omega$$

$$+ x_2\omega\varpi\sin t\omega - v_2\omega\cos\vartheta\cos t\varpi\sin t\omega - v_2\omega\sqrt{1-\sin\vartheta^2}\sin t\omega$$

$$+ v_2\omega\cos\vartheta\cos t\omega\sin t\varpi + v_2\omega\sin\vartheta\sin t\omega\sin t\varpi\big)\ldots$$

Air Traffic Control





$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

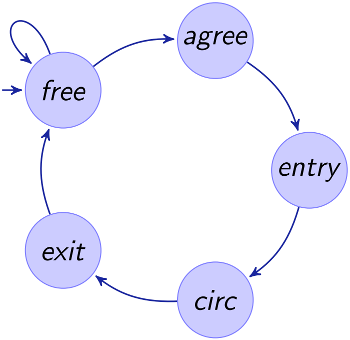
Example ("Solving" differential equations)

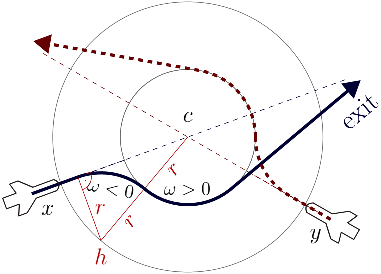
$$\forall t \geq 0 \quad \frac{1}{\omega\varpi}\big(x_1\omega\varpi \cos t\omega - v_2\omega \cos t\omega \sin\vartheta + v_2\omega \cos t\omega \cos t\varpi \sin\vartheta - v_1\varpi \sin t\omega$$

$$+ x_2\omega\varpi \sin t\omega - v_2\omega \cos\vartheta \cos t\varpi \sin t\omega - v_2\omega\sqrt{1-\sin\vartheta^2} \sin t\omega$$

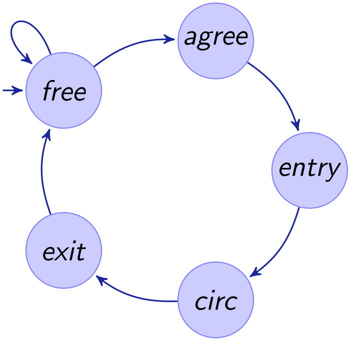
$$+ v_2\omega \cos\vartheta \cos t\omega \sin t\varpi + v_2\omega \sin\vartheta \sin t\omega \sin t\varpi\big) \ldots$$

Flyable Roundabout Maneuver: Overview

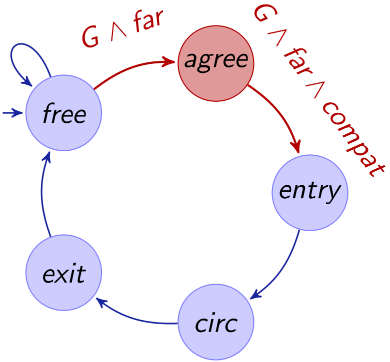




Fixedpoint Iterations for Air Traffic Control



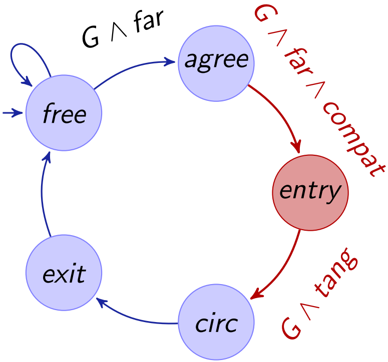
Fixedpoint Iterations for Air Traffic Control



Example (dℒ formula of verification subgoal)

$$safe \land far \rightarrow [agree](safe \land far \land compatible)$$

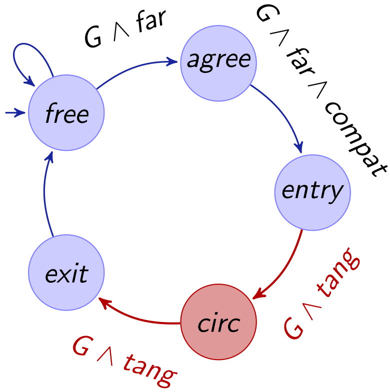
Fixedpoint Iterations for Air Traffic Control



Example (d$\mathcal{L}$ formula of verification subgoal)

$$\mathit{safe} \land \mathit{far} \land \mathit{compatible} \;\rightarrow\; [\mathit{entry}](\mathit{safe} \land \mathit{tangential})$$

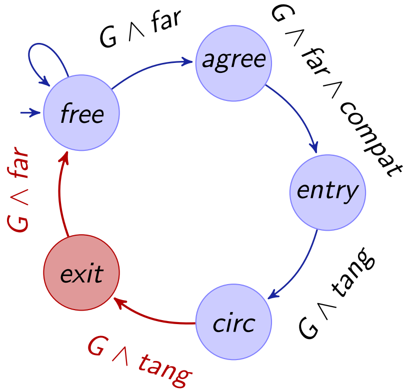
Fixedpoint Iterations for Air Traffic Control



Example (d$\mathcal{L}$ formula of verification subgoal)

$$safe \land tangential \;\rightarrow\; [circ](safe \land tangential)$$

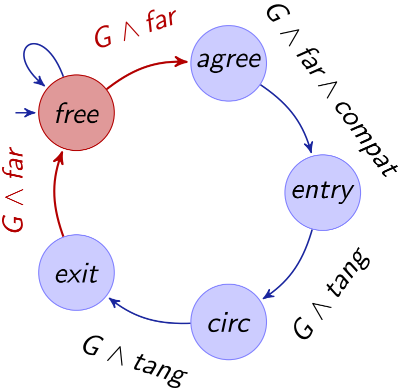
Fixedpoint Iterations for Air Traffic Control



Example (d$\mathcal{L}$ formula of verification subgoal)

$$safe \wedge tangential \rightarrow [exit](safe \wedge far)$$

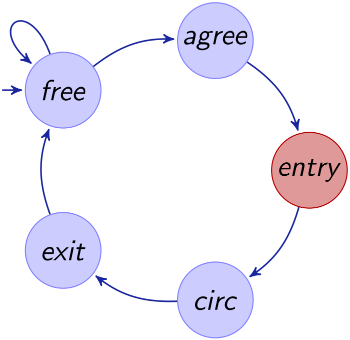
Fixedpoint Iterations for Air Traffic Control

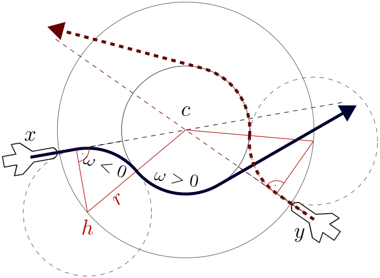


Example (d$\mathcal{L}$ formula of verification subgoal)

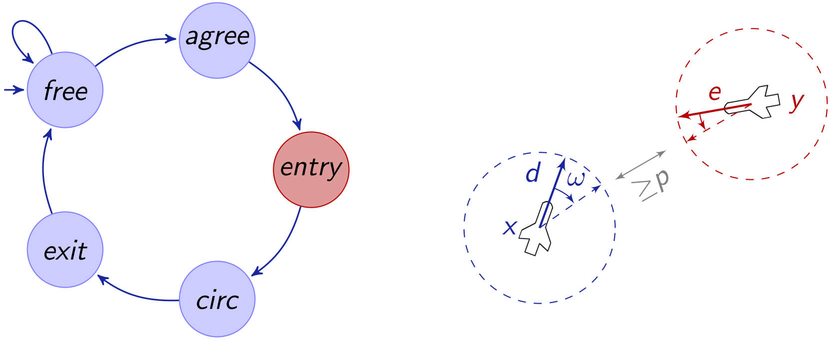
$$safe \land far \;\rightarrow\; [free](safe \land far)$$

Flyable Roundabout Maneuver: Entry

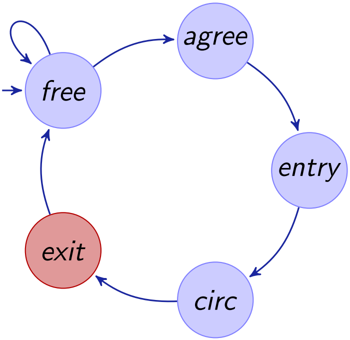


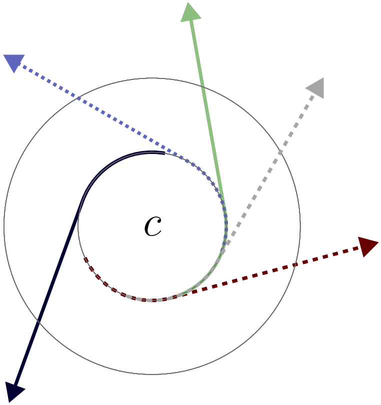


Flyable Roundabout Maneuver: Entry

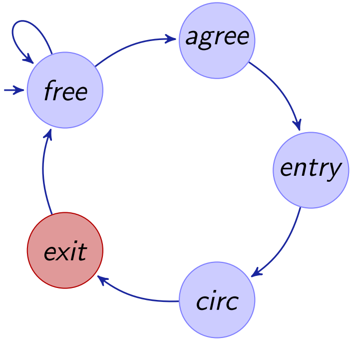


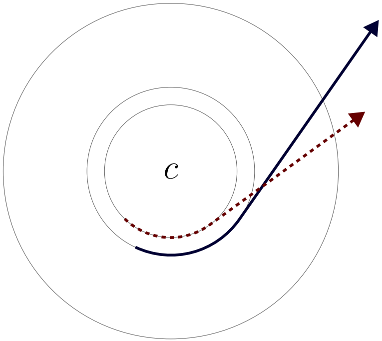
Flyable Roundabout Maneuver: Exit





Flyable Roundabout Maneuver: Exit





Tangential Roundabout Collision Avoidance Maneuver

provable automatically!

$$
\begin{aligned}
\psi &\equiv \phi \rightarrow [trm^*]\phi \\
\phi &\equiv \|x-y\|^2 \geq p^2 \equiv (x_1-y_1)^2 + (x_2-y_2)^2 \geq p^2 \\
trm &\equiv free;\quad entry;\quad \mathcal{F}(\omega) \wedge \mathcal{G}(\omega) \\
free &\equiv \exists\omega\,\mathcal{F}(\omega) \wedge \exists\varpi\,\mathcal{G}(\varpi) \wedge \phi \\
entry &\equiv \exists u\,\omega := u;\ \exists c\,(d := \omega(x-c)^\perp \wedge e := \omega(y-c)^\perp)
\end{aligned}
$$

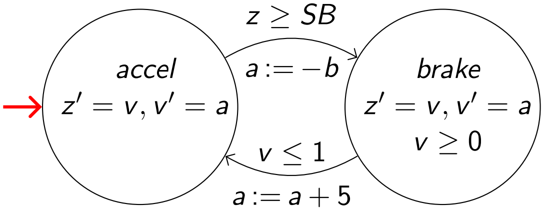
$$
\mathcal{F}(\omega) \equiv \begin{pmatrix} x_1' = v\cos\vartheta & = d_1 \\ \wedge\, x_2' = v\sin\vartheta & = d_2 \\ \wedge\, d_1' = v(-\sin\vartheta)\vartheta' & = -\omega d_2 \\ \wedge\, d_2' = v(\cos\vartheta)\vartheta' & = \quad \omega d_1 \end{pmatrix} \qquad \mathcal{G}(\varpi) \equiv \begin{pmatrix} y_1' = e_1 \\ \wedge\, y_2' = e_2 \\ \wedge\, e_1' = -\varpi e_2 \\ \wedge\, e_2' = \quad \varpi e_1 \end{pmatrix}
$$

provable automatically!

$$
\begin{aligned}
\psi \quad & \equiv \phi \to [trm^*]\phi \\
\phi \quad & \equiv (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \land (y_1 - z_1)^2 + (y_2 - z_2)^2 \geq p^2 \\
& \quad \land (x_1 - z_1)^2 + (x_2 - z_2)^2 \geq p^2 \land (x_1 - u_1)^2 + (x_2 - u_2)^2 \geq p^2 \\
& \quad \land (y_1 - u_1)^2 + (y_2 - u_2)^2 \geq p^2 \land (z_1 - u_1)^2 + (z_2 - u_2)^2 \geq p^2 \\
trm \quad & \equiv free; \quad entry; \\
& \quad x_1' = d_1 \land x_2' = d_2 \land d_1' = -\omega_x d_2 \land d_2' = \omega_x d_1 \\
& \quad \land y_1' = e_1 \land y_2' = e_2 \land e_1' = -\omega_y e_2 \land e_2' = \omega_y e_1 \\
& \quad \land z_1' = f_1 \land z_2' = f_2 \land f_1' = -\omega_z f_2 \land f_2' = \omega_z f_1 \\
& \quad \land u_1' = g_1 \land u_2' = g_2 \land g_1' = -\omega_u g_2 \land g_2' = \omega_u g_1 \\
free \quad & \equiv (\omega_x := *; \quad \omega_y := *; \quad \omega_z := *; \quad \omega_u := *; \\
& \quad x_1' = d_1 \land x_2' = d_2 \land d_1' = -\omega_x d_2 \land d_2' = \omega_x d_1 \\
& \quad \land y_1' = e_1 \land y_2' = e_2 \land e_1' = -\omega_y e_2 \land e_2' = \omega_y e_1 \\
& \quad \land z_1' = f_1 \land z_2' = f_2 \land f_1' = -\omega_z f_2 \land f_2' = \omega_z f_1 \\
& \quad \land u_1' = g_1 \land u_2' = g_2 \land g_1' = -\omega_u g_2 \land g_2' = \omega_u g_1 \land \phi)^* \\
entry \quad & \equiv \omega := *; \; c := *; \\
& \quad d_1 := -\omega(x_2 - c_2); \; d_2 := \omega(x_1 - c_1); \\
& \quad e_1 := -\omega(y_1 - c_1); \; e_2 := \omega(y_2 - c_2); \\
& \quad f_1 := -\omega(z_1 - c_1); \; f_2 := \omega(z_2 - c_2); \\
& \quad g_1 := -\omega(u_1 - c_1); \; g_2 := \omega(u_2 - c_2)
\end{aligned}
$$

Outline

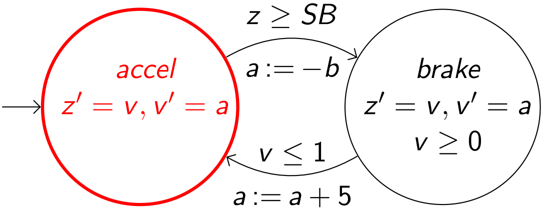
Embedding Hybrid Automata as Hybrid Programs



⇝

$$
\begin{aligned}
&q := accel;\\
&(\quad (?q = accel;\quad z' = v, v' = a)\\
&\cup\ (?q = accel \land z \geq SB;\quad a := -b;\quad q := brake;\quad ?v \geq 0)\\
&\cup\ (?q = brake;\quad z' = v, v' = a \land v \geq 0)\\
&\cup\ (?q = brake \land v \leq 1;\quad a := a + 5;\quad q := accel))^*
\end{aligned}
$$

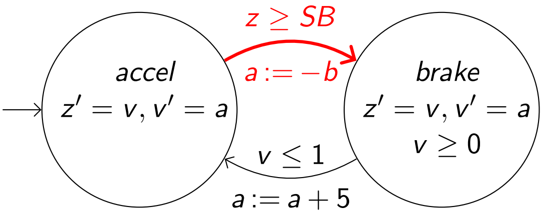
Embedding Hybrid Automata as Hybrid Programs



$\rightsquigarrow$

$$
\begin{aligned}
&q := accel;\\
&(\quad (?q = accel;\quad z' = v, v' = a)\\
&\cup\ (?q = accel \land z \geq SB;\quad a := -b;\quad q := brake;\quad ?v \geq 0)\\
&\cup\ (?q = brake;\quad z' = v, v' = a \land v \geq 0)\\
&\cup\ (?q = brake \land v \leq 1;\quad a := a + 5;\quad q := accel))^*
\end{aligned}
$$

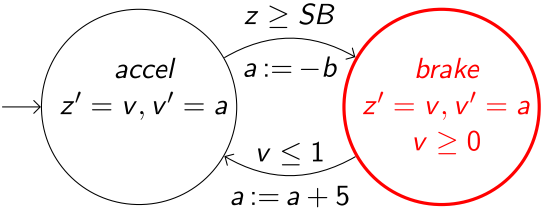
Embedding Hybrid Automata as Hybrid Programs



⇝

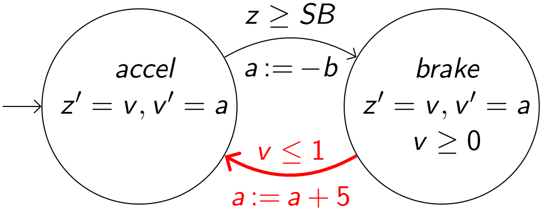
$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel;\ \ z' = v, v' = a) \\
& \cup\ (?q = accel \land z \geq SB;\ \ a := -b;\ \ q := brake;\ \ ?v \geq 0) \\
& \cup\ (?q = brake;\ \ z' = v, v' = a \land v \geq 0) \\
& \cup\ (?q = brake \land v \leq 1;\ \ a := a + 5;\ \ q := accel)\big)^*
\end{aligned}
$$

Embedding Hybrid Automata as Hybrid Programs



$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel;\quad z' = v, v' = a) \\
& \cup\ (?q = accel \land z \geq SB;\quad a := -b;\quad q := brake;\quad ?v \geq 0) \\
& \cup\ (?q = brake;\quad z' = v, v' = a \land v \geq 0) \\
& \cup\ (?q = brake \land v \leq 1;\quad a := a + 5;\quad q := accel)\big)^*
\end{aligned}
$$

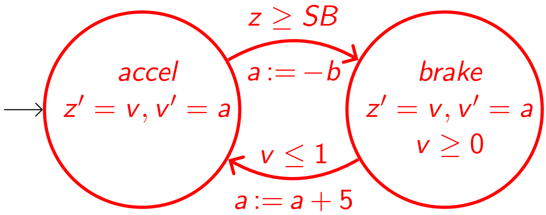
Embedding Hybrid Automata as Hybrid Programs



$\rightsquigarrow$

$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel; \quad z' = v, v' = a) \\
& \cup \ (?q = accel \land z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
& \cup \ (?q = brake; \quad z' = v, v' = a \land v \geq 0) \\
& \cup \ (?q = brake \land v \leq 1; \quad a := a + 5; \quad q := accel)\big)^*
\end{aligned}
$$

Embedding Hybrid Automata as Hybrid Programs



$\rightsquigarrow$

$$
\begin{aligned}
& q := accel; \\
& \big(\quad (?q = accel; \quad z' = v, v' = a) \\
& \cup \ (?q = accel \wedge z \geq SB; \quad a := -b; \quad q := brake; \quad ?v \geq 0) \\
& \cup \ (?q = brake; \quad z' = v, v' = a \wedge v \geq 0) \\
& \cup \ (?q = brake \wedge v \leq 1; \quad a := a + 5; \quad q := accel) \big)^*
\end{aligned}
$$