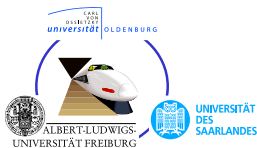


# KeYmaera: A Hybrid Theorem Prover for Hybrid Systems

André Platzer    **Jan-David Quesel**

University of Oldenburg, Department of Computing Science, Germany

International Joint Conference on Automated Reasoning, Sydney 2008

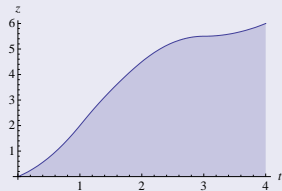
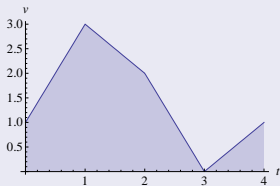
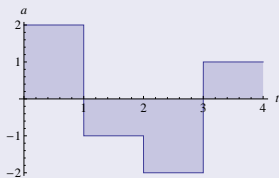


Deutsche  
Forschungsgemeinschaft  
**DFG**

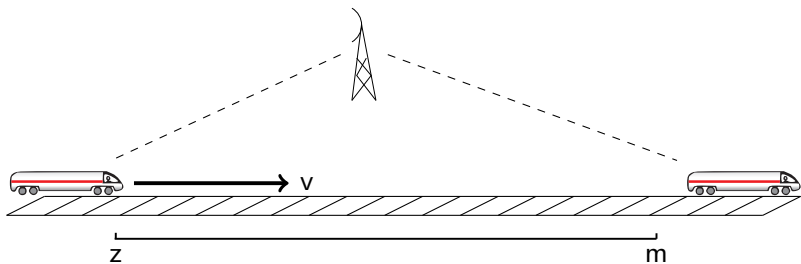
## KeYmaera: Verification tool for hybrid systems

### Hybrid System

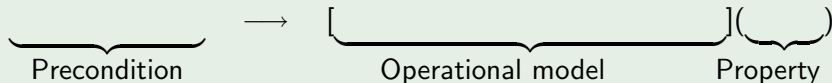
- Continuous evolutions (differential equations)
- Discrete jumps (control decisions)

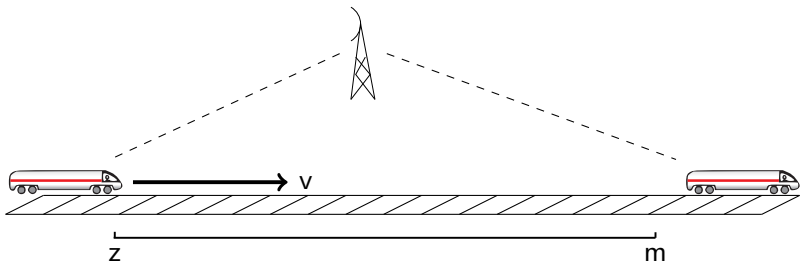


# Differential Dynamic Logic ( $d\mathcal{L}$ )



## Example

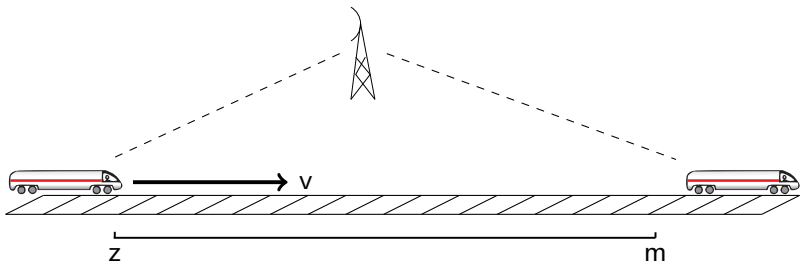




## Example

$$\underbrace{v^2 \leq 2b(m - z)}_{\text{Precondition}} \longrightarrow \underbrace{\quad}_{\text{Operational model}} \underbrace{\quad}_{\text{Property}} (z \leq m)$$

# Differential Dynamic Logic (dL)

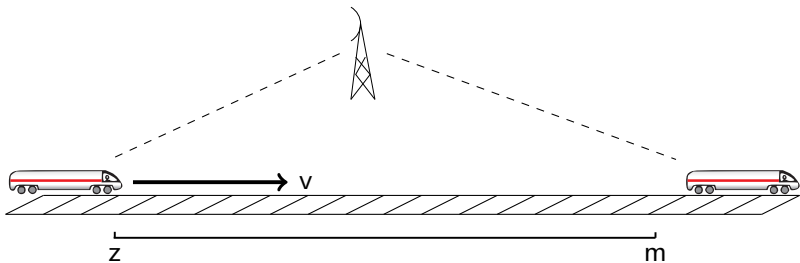


## Example

$$\underbrace{v^2 \leq 2b(m - z)}_{\text{Precondition}} \longrightarrow \left[ \underbrace{z' = v, v' = a}_{\text{Operational model}} \right] \underbrace{(z \leq m)}_{\text{Property}}$$

Continuous evolution:  
differential equation

# Differential Dynamic Logic (dL)

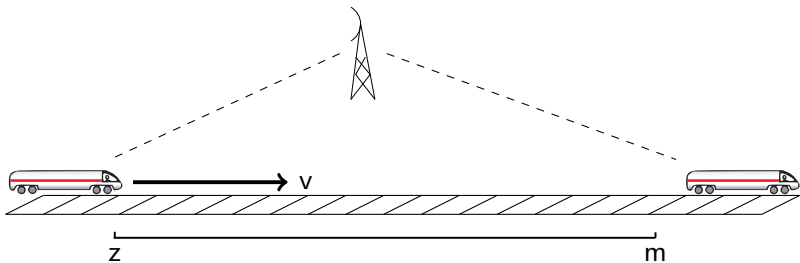


## Example

$$\underbrace{v^2 \leq 2b(m - z)}_{\text{Precondition}} \longrightarrow \underbrace{[a := *; z' = v, v' = a]}_{\text{Operational model}} \underbrace{(z \leq m)}_{\text{Property}}$$

Random assignment

# Differential Dynamic Logic (dL)



## Example

$$\underbrace{v^2 \leq 2b(m - z)}_{\text{Precondition}} \longrightarrow \underbrace{[a := *; ?a \leq -b; z' = v, v' = a]}_{\text{Operational model}} \underbrace{(z \leq m)}_{\text{Property}}$$

Test

## d $\mathcal{L}$ Formulas

$$\phi ::= \theta_1 \sim \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \forall x\phi \mid \exists x\phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

## Hybrid Program

## Effect

 $\alpha; \beta$ 

sequential composition

 $\alpha \cup \beta$ 

nondeterministic choice

 $\alpha^*$ 

nondeterministic repetition

 $x := \theta$ 

discrete assignment (jump)

 $x := *$ 

nondeterministic assignment

 $(x'_1 = \theta_1, \dots, x'_n = \theta_n, F)$ continuous evolution of  $x_i$  $?F$ check if formula  $F$  holds

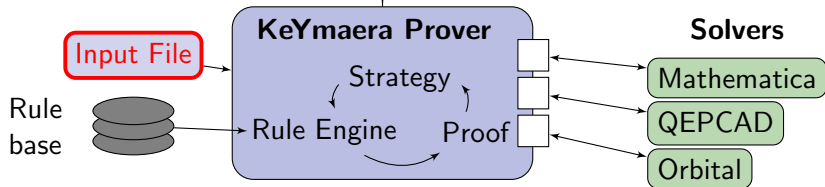
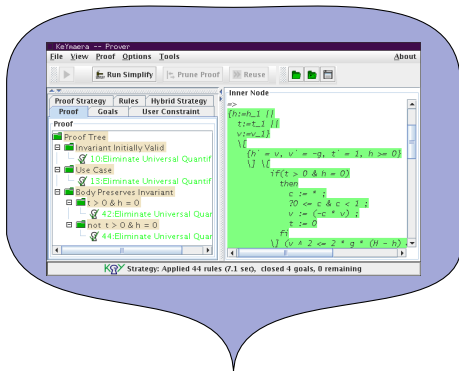
A. Platzer.

Differential Dynamic Logic for Hybrid Systems.

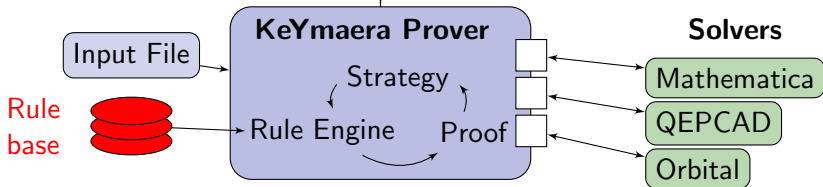
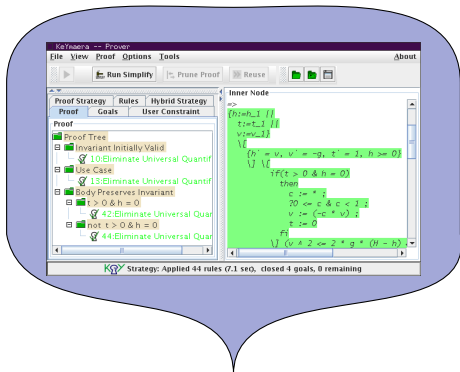
Journal of Automated Reasoning, 41(2), 2008, to appear.



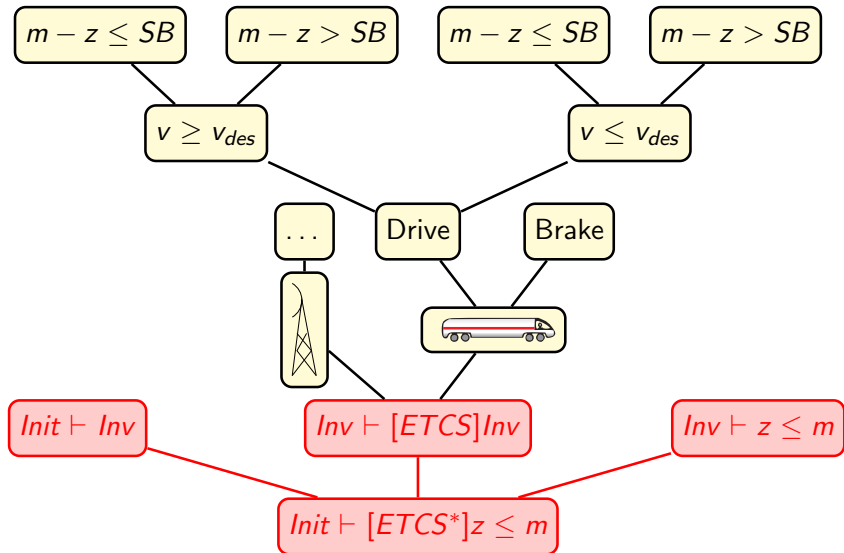
# KeYmaera Architecture



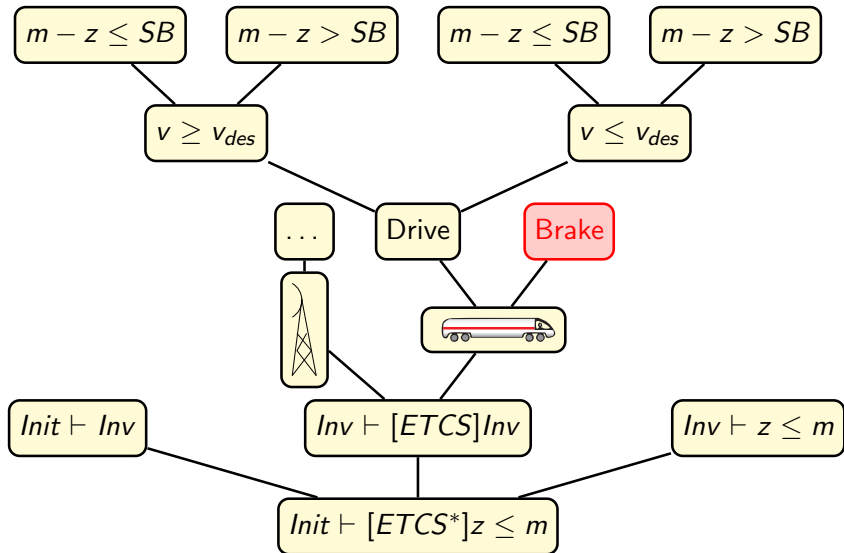
# KeYmaera Architecture



# Proof Sketch

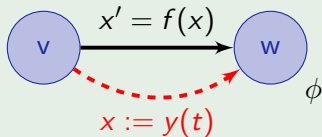


# Proof Sketch



## Example

$$\frac{\forall t \geq 0 [x := y(t)] \phi}{[x' = f(x)] \phi}$$

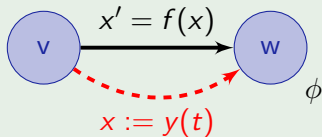


---

$$\dots \vdash [z' = v, v' = -b]z \leq m$$

## Example

$$\frac{\forall t \geq 0 [x := y(t)] \phi}{[x' = f(x)] \phi}$$

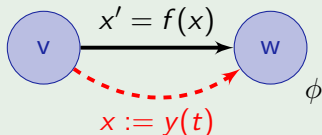


$$\dots \vdash \forall t \geq 0 [z := -\frac{1}{2}bt^2 + tv + z] z \leq m$$

$$\dots \vdash [z' = v, v' = -b] z \leq m$$

## Example

$$\frac{\forall t \geq 0 [x := y(t)] \phi}{[x' = f(x)] \phi}$$

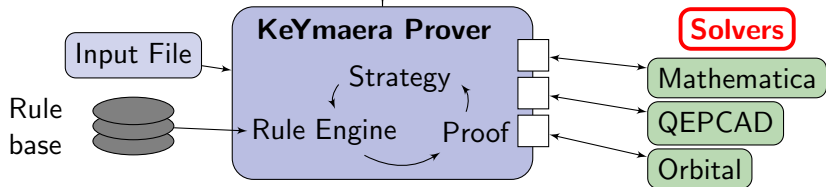
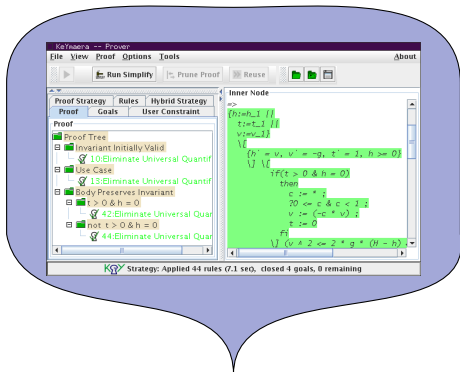


$$\dots \vdash \forall t \geq 0 (-\frac{1}{2}bt^2 + tv + z \leq m)$$

$$\dots \vdash \forall t \geq 0 [z := -\frac{1}{2}bt^2 + tv + z]z \leq m$$

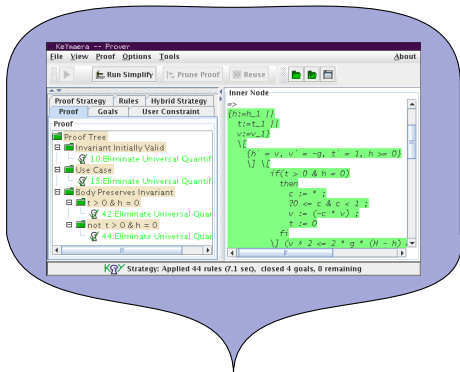
$$\dots \vdash [z' = v, v' = -b]z \leq m$$

# KeYmaera Architecture





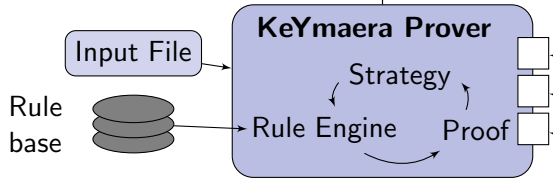
# KeYmaera Architecture



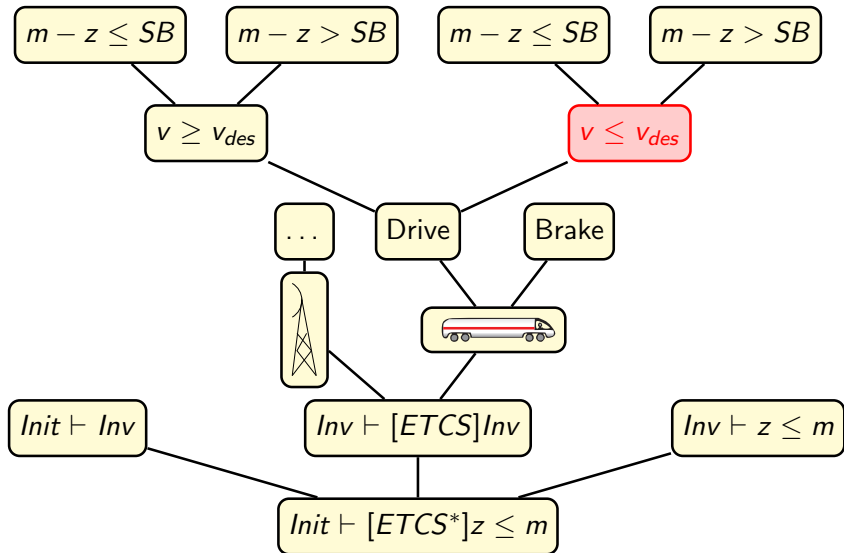
Quantifier elimination

Solvers

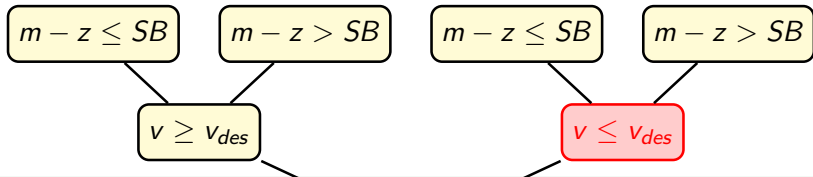
- Mathematica
- QEPCAD
- Orbital



# Proof Sketch



# Proof Sketch



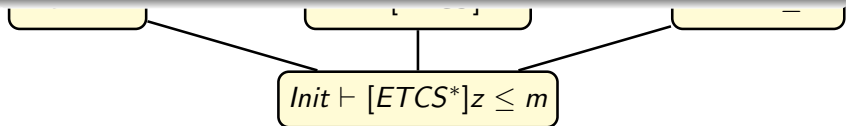
## Example

$$m - z \geq \left(\frac{A}{b} + 1\right) (\varepsilon v + \frac{A}{2}\varepsilon^2) + \frac{v^2 - d^2}{2b} \wedge 0 \leq a \leq A \wedge 0 \leq v \leq v_{des} \\ \wedge v^2 - d^2 \leq 2b(m - z) \wedge d \geq 0 \wedge \varepsilon > 0 \wedge b > 0 \wedge A > 0$$

⊢

$$\forall t \geq 0 ((\forall 0 \leq \tilde{t} \leq t (a\tilde{t} + v \geq 0 \wedge \tilde{t} \leq \varepsilon))$$

$$\rightarrow (at + v)^2 - d^2 \leq 2b(m - (\frac{1}{2}at + tv + z)) \wedge at + v \geq 0 \wedge d \geq 0)$$

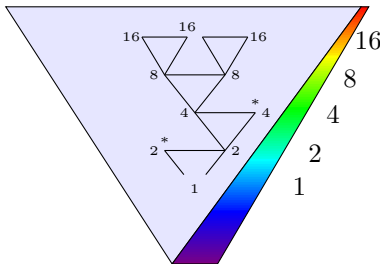


# Iterative Background Closure



- Quantifier elimination is doubly exponential
- Choice conflict:
  - 1 Apply quantifier elimination
  - 2 Split using

$$\frac{\vdash F \quad \vdash G}{\vdash F \wedge G}$$



# Experimental Results



Case Study	Interact	Steps	IBC(s)	Eager QE(s)
ETCS essentials	0	46	47.8	$\infty$
	1	46	6.6	8.8
ETCS complete	0	163	2045.2	$\infty$
	1	168	23.3	$\infty$
ETCS reactivity	0	49	76.2	$\infty$
ETCS liveness	3	112	17.6	16.0
Aircraft TRM	0	94	10.9	$\infty$
	1	94	1.2	1.2
TRM 3 Planes	0	187	171.8	$\infty$
	1	187	21.2	$\infty$
TRM 4 Planes	0	255	704.3	$\infty$
	1	255	170	$\infty$
Water tank	0	-	$\infty$	$\infty$
	1	375	2.0	2.0

$\infty \hat{=}$  more than five hours

# Experimental Results



Case Study	Interact	Steps	IBC(s)	Eager QE(s)
ETCS essentials	0	46	47.8	$\infty$
	1	46	6.6	8.8
ETCS complete	0	163	2045.2	$\infty$
	1	168	23.3	$\infty$
ETCS reactivity	0	49	76.2	$\infty$
ETCS liveness	3	112	17.6	16.0
Aircraft TRM	0	94	10.9	$\infty$
	1	94	1.2	1.2
TRM 3 Planes	0	187	171.8	$\infty$
	1	187	21.2	$\infty$
TRM 4 Planes	0	255	704.3	$\infty$
	1	255	170	$\infty$
Water tank	0	-	$\infty$	$\infty$
	1	375	2.0	2.0

$\infty \hat{=}$  more than five hours

## KeYmaera Summary

- Hybrid theorem prover for verifying hybrid systems
  - Differential Dynamic Logic (dL) and Hybrid Programs
  - Sequent calculus
  - Quantifier elimination, computer algebra
- Automatic proof strategies (95 – 100%)
- Plugin-architecture for backends

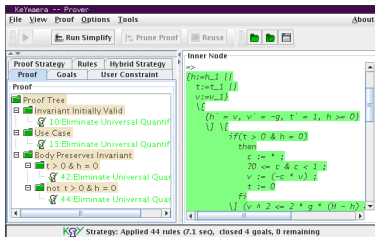
The screenshot shows the KeYmaera Prover interface. The main window is titled "KeYmaera -- Prover" and has a menu bar with "File", "View", "Proof", "Options", and "Tools". Below the menu bar are buttons for "Run Simplify", "Prune Proof", and "Reuse". The interface is divided into several panes:

- Proof Strategy:** Shows "Proof", "Goals", and "User Constraint".
- Proof Tree:** A tree view showing the current state of the proof. It includes nodes like "Invariant Initially Valid", "10:Eliminate Universal Quantif", "Use Case", "13:Eliminate Universal Quantif", "Body Preserves Invariant", "t > 0 & h = 0", "42:Eliminate Universal Quas", "not t > 0 & h = 0", and "44:Eliminate Universal Quar".
- Inner Node:** A text editor showing the current goal or state. The code is:

```
=>
{?i=h-1 | h
 t:=t-1 | f
 v:=v-1 | f
 \d
 {h = v, v' = -g, t' = 1, h >= 0}
 \d \{
 ??(t > 0 & h = 0)
 then
 c := +
 ?0' <= c & c < 1 ;
 v := (-c * v) ;
 t := 0
 \d
 \} (v * 2 <= 2 * g * (h - h))
```

At the bottom, a status bar indicates: "Strategy: Applied 44 rules (7.1 sec, closed 4 goals, 0 remaining)".

# Conclusions



## Features

- Verify controllability, reactivity, safety and liveness properties
- Counterexample generation
- Automatic invariant discovery
- Handling differential inequalities ( $z'' \leq a$ )
- Support for proof annotations
- Equational Gröbner basis verification support







A. Platzer.

Differential Dynamic Logic for Verifying Parametric Hybrid Systems.  
In N. Olivetti, editor, *TABLEAUX*, volume 4548 of *LNCS*, pages 216–232. Springer, 2007.



A. Platzer.

Differential Dynamic Logic for Hybrid Systems.  
*Journal of Automated Reasoning*, 41(2), 2008, to appear.



A. Platzer and E. M. Clarke.

Computing Differential Invariants of Hybrid Systems as Fixedpoints.  
In A. Gupta and S. Malik, editors, *CAV*, volume 5123 of *LNCS*, pages 176–189, 2008.



A. Platzer and J.-D. Quesel.

Logical Verification and Systematic Parametric Analysis in Train Control.

In M. Egerstedt and B. Mishra, editors, *HSCC*, volume 4981 of *LNCS*, pages 646–649. Springer, 2008.