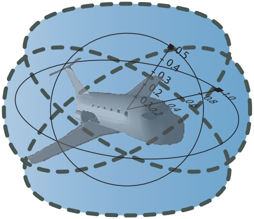
15-819/18-879: Hybrid Systems Analysis & Theorem Proving

09: Train Control Verification

André Platzer

aplatzer@cs.cmu.edu
Carnegie Mellon University, Pittsburgh, PA



Outline

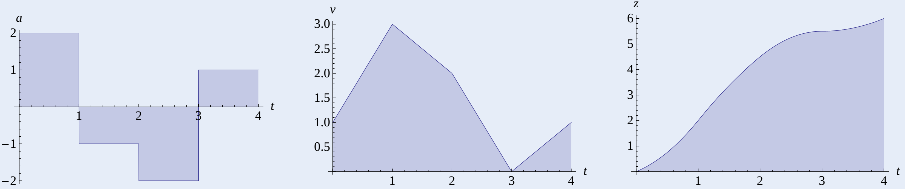
Outline

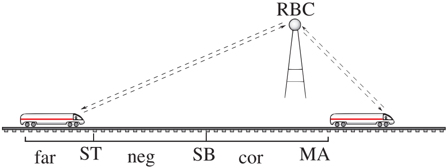
ETCS Control Verification

Problem

Hybrid System

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

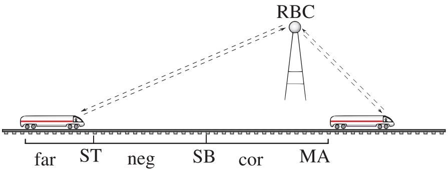




ETCS objectives:

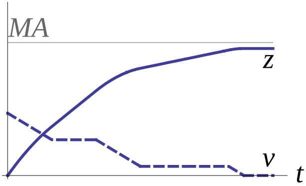
1. Collision free
2. Maximise throughput & velocity (300 km/h)
3. $2.1 * 10^6$ passengers/day

Verifying Parametric Hybrid Systems

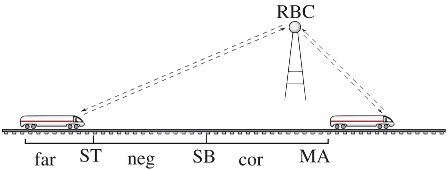


Parametric Hybrid Systems

continuous evolution along differential equations + discrete change

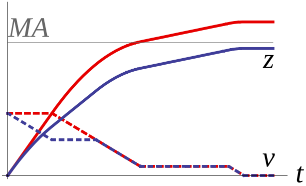


Verifying Parametric Hybrid Systems

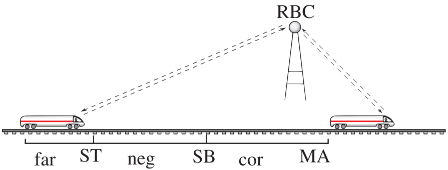


Parametric Hybrid Systems

continuous evolution along differential equations + discrete change

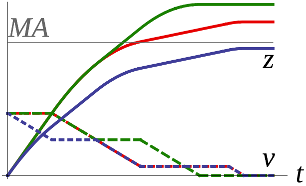


Verifying Parametric Hybrid Systems

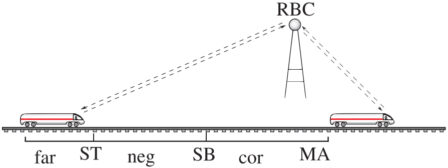


Parametric Hybrid Systems

continuous evolution along differential equations + discrete change



Verifying Parametric Hybrid Systems

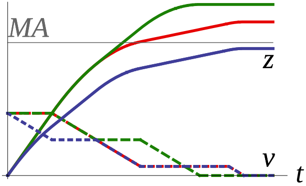


Parametric Hybrid Systems

continuous evolution along differential equations + discrete change

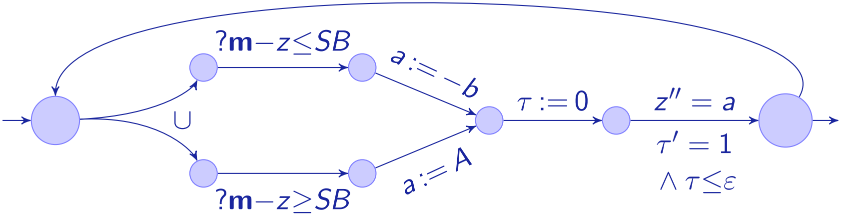
- Parameters have nonlinear influence
- Handle *SB* as free symbolic parameter?
- Challenge: verification (falsifying is "easy")
- Which constraints for *SB*?

$\forall \mathbf{m}\, \exists SB$ "train always safe"



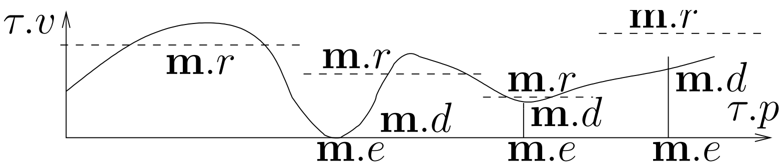
$$
\begin{aligned}
\text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\
\mathit{cor} &\equiv (?\mathbf{m} - z \le SB; a := -b) \cup (?\mathbf{m} - z \ge SB; a := A) \\
\mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \land v \ge 0 \land \tau \le \varepsilon)
\end{aligned}
$$

$$\begin{aligned} \text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\ \mathit{cor} &\equiv (?\mathbf{m} - z \le SB;\, a := -b) \cup (?\mathbf{m} - z \ge SB;\, a := A) \\ \mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \land v \ge 0 \land \tau \le \varepsilon) \end{aligned}$$



Outline

2D Movement Authorities

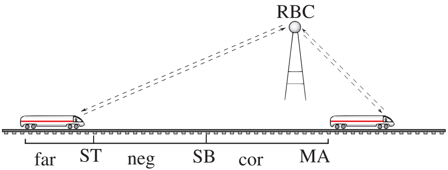


- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.



Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

- To simplify notation, assume trains are points.

Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.

Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be positions of all the trains 1 to n at ζ.

Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be positions of all the trains 1 to n at ζ.
- Let M_i be the MA-range, i.e., the set of positions on the track for which train i has currently been issued MA.

Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be positions of all the trains 1 to n at ζ.
- Let M_i be the MA-range, i.e., the set of positions on the track for which train i has currently been issued MA.
- Suppose there was a collision at time ζ.

Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be positions of all the trains 1 to n at ζ.
- Let M_i be the MA-range, i.e., the set of positions on the track for which train i has currently been issued MA.
- Suppose there was a collision at time ζ.
- Then $z_i = z_j$ at ζ for some $i, j \in \mathbb{N}$.

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be positions of all the trains 1 to n at ζ.
- Let M_i be the MA-range, i.e., the set of positions on the track for which train i has currently been issued MA.
- Suppose there was a collision at time ζ.
- Then $z_i = z_j$ at ζ for some $i, j \in \mathbb{N}$.
- However, by assumption, $z_i \in M_i$ and $z_j \in M_j$ at ζ, thus $M_i \cap M_j \neq \emptyset$,

Separation Principle

Lemma (Principle of separation by movement authorities)

If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide.

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \dots, z_n$ be positions of all the trains 1 to n at ζ.
- Let M_i be the MA-range, i.e., the set of positions on the track for which train i has currently been issued MA.
- Suppose there was a collision at time ζ.
- Then $z_i = z_j$ at ζ for some $i, j \in \mathbb{N}$.
- However, by assumption, $z_i \in M_i$ and $z_j \in M_j$ at ζ, thus $M_i \cap M_j \neq \emptyset$,
- This contradicts the assumption of disjoint MA.

Model/State Variables

Train τ:

- $\tau.v$ Position
- $\tau.v$ Speed
- $\tau.a$ Acceleration
- (t model time)

RBC + MA:

- $\mathbf{m}.e$ End of Authority
- $\mathbf{m}.d$ Speed limit
- $\mathbf{m}.r$ Recommended speed
- $rbc.message$ Channel

Parameters:

- SB Start Braking
- ST Start Talking
- b Braking power/deceleration
- A Maximum acceleration
- ε Maximum cycle time
- Δ Maximum expected communication delay

Parametric Skeleton of ETCS Cooperation Protocol

$$
\begin{array}{ll}
ETCS_{skel} & : (train \cup rbc)^* \\
train & : spd;\ atp;\ drive \\
spd & : \quad (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ? -b \le \tau.a \le A) \\
 & \quad \cup (?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ? -b \le \tau.a \le 0) \\
atp & : \mathtt{if}(\mathbf{m}.e - \tau.p \le SB \lor rbc.message = emergency)\, \tau.a := -b \\
drive & : t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \land \tau.v \ge 0 \land t \le \varepsilon) \\
rbc & : (rbc.message := emergency) \cup (\mathbf{m} := *;\ ?\mathbf{m}.r > 0)
\end{array}
$$

Parametric Skeleton of ETCS Cooperation Protocol

$$
\begin{array}{ll}
ETCS_{skel} & : (train \cup rbc)^* \\
train & : spd;\ atp;\ drive \\
spd & : \quad (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ? -b \le \tau.a \le A) \\
 & \quad \cup (?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ? -b \le \tau.a \le 0) \\
atp & : \mathtt{if}(\mathbf{m}.e - \tau.p \le SB \lor rbc.message = emergency)\, \tau.a := -b \\
drive & : t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \land \tau.v \ge 0 \land t \le \varepsilon) \\
rbc & : (rbc.message := emergency)\ \cup\ (\mathbf{m} := *;\ ?\mathbf{m}.r > 0)
\end{array}
$$

Verify safety?

Parametric Skeleton of ETCS Cooperation Protocol

$$
\begin{array}{ll}
ETCS_{skel} & : (train \cup rbc)^* \\
train & : spd;\ atp;\ drive \\
spd & : \quad (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le A) \\
 & \quad \cup(?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le 0) \\
atp & : \mathtt{if}(\mathbf{m}.e - \tau.p \le SB \lor rbc.message = emergency)\, \tau.a := -b \\
drive & : t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \land \tau.v \ge 0 \land t \le \varepsilon) \\
rbc & : (rbc.message := emergency)\ \cup\ (\mathbf{m} := *;\ ?\mathbf{m}.r > 0)
\end{array}
$$

Verify safety?

$$[ETCS_{skel}](\tau.p \ge \mathbf{m}.e \to \tau.v \le \mathbf{m}.d)$$

$$
\begin{array}{lll}
ETCS_{skel} & : & (train \cup rbc)^* \\
train & : & spd;\ atp;\ drive \\
spd & : & \ \ (?\tau.v \leq \mathbf{m}.r;\ \tau.a := *;\ ?-b \leq \tau.a \leq A) \\
 & & \cup(?\tau.v \geq \mathbf{m}.r;\ \tau.a := *;\ ?-b \leq \tau.a \leq 0) \\
atp & : & \mathtt{if}(\mathbf{m}.e - \tau.p \leq SB \vee rbc.message = emergency)\, \tau.a := -b \\
drive & : & t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon) \\
rbc & : & (rbc.message := emergency)\ \cup\ (\mathbf{m} := *;\ ?\mathbf{m}.r > 0)
\end{array}
$$

Verify safety?

$$[ETCS_{skel}](\tau.p \geq \mathbf{m}.e \rightarrow \tau.v \leq \mathbf{m}.d)$$

Lots of counterexamples!

1. *Controllability discovery*: Start with uncontrolled system dynamics. Apply structural d$\mathcal{L}$ decomposition until FOL-formula is obtained revealing controllable state region, which specifies for which parameter combinations the system dynamics can be controlled safely by any control law.

Iterative Control Refinement Process

1. *Controllability discovery*: Start with uncontrolled system dynamics. Apply structural d$\mathcal{L}$ decomposition until FOL-formula is obtained revealing controllable state region, which specifies for which parameter combinations the system dynamics can be controlled safely by any control law.
2. *Control refinement*: Successively add partial control laws to the system while leaving its decision parameters (like SB or $\mathbf{m}$) free. Apply d$\mathcal{L}$ decomposition to discover parametric constraints which maintain controllability under these control laws.

Iterative Control Refinement Process

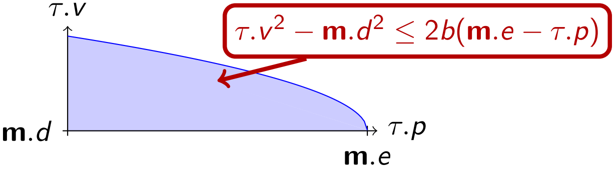
1. *Controllability discovery*: Start with uncontrolled system dynamics. Apply structural d$\mathcal{L}$ decomposition until FOL-formula is obtained revealing controllable state region, which specifies for which parameter combinations the system dynamics can be controlled safely by any control law.
2. *Control refinement*: Successively add partial control laws to the system while leaving its decision parameters (like SB or $\mathbf{m}$) free. Apply d$\mathcal{L}$ decomposition to discover parametric constraints which maintain controllability under these control laws.
3. *Safety convergence*: Repeat step 2 until resulting system proven safe.

Iterative Control Refinement Process

1. *Controllability discovery*: Start with uncontrolled system dynamics. Apply structural d$\mathcal{L}$ decomposition until FOL-formula is obtained revealing controllable state region, which specifies for which parameter combinations the system dynamics can be controlled safely by any control law.
2. *Control refinement*: Successively add partial control laws to the system while leaving its decision parameters (like SB or $\mathbf{m}$) free. Apply d$\mathcal{L}$ decomposition to discover parametric constraints which maintain controllability under these control laws.
3. *Safety convergence*: Repeat step 2 until resulting system proven safe.
4. *Liveness check*: Prove that discovered parametric constraints do not over-constrain system inconsistently by showing that it remains live.

Outline

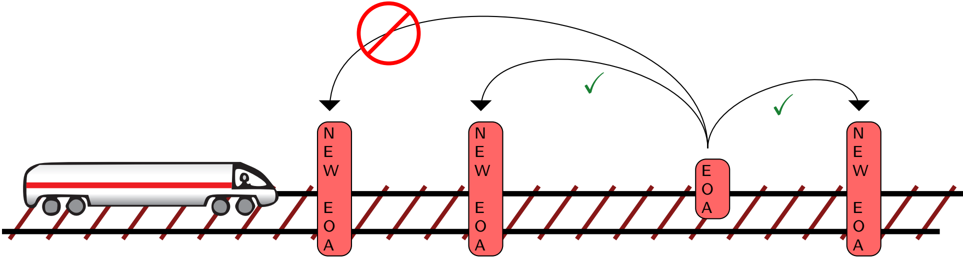
ETCS Controllability



Proposition (Controllability)

$$[\tau.p' = \tau.v, \tau.v' = -b \wedge \tau.v \geq 0](\tau.p \geq \mathbf{m}.e \rightarrow \tau.v \leq \mathbf{m}.d)$$

$$\equiv \mathcal{C} \;\equiv\; \tau.v^2 = \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \tau.p)$$

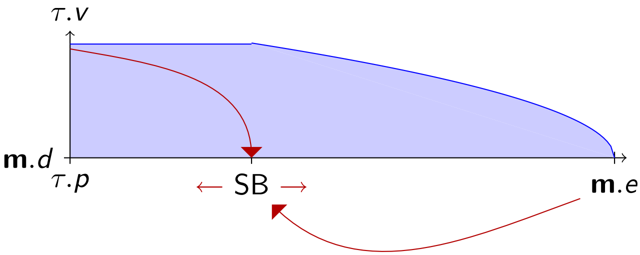


Proposition (RBC Controllability)

$$\mathbf{m}.d \geq 0 \land b > 0 \rightarrow [\mathbf{m}_0 := \mathbf{m};\ rbc] \Big($$

$$\mathcal{M} \equiv \mathbf{m}_0.d^2 - \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \mathbf{m}_0.e) \land \mathbf{m}_0.d \geq 0 \land \mathbf{m}.d \geq 0 \leftrightarrow$$

$$\forall \tau \left((\langle \mathbf{m} := \mathbf{m}_0 \rangle \mathcal{C}) \rightarrow \mathcal{C} \right) \Big)$$



Proposition (Reactivity)

$$\Big(\forall \mathbf{m}.e\, \forall \tau.p\, \big(\mathbf{m}.e - \tau.p \geq SB \land \mathcal{C} \rightarrow [\tau.a := A;\ drive]\, \mathcal{C}\big)\Big)$$

$$\equiv SB \geq \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\, \tau.v\right)$$

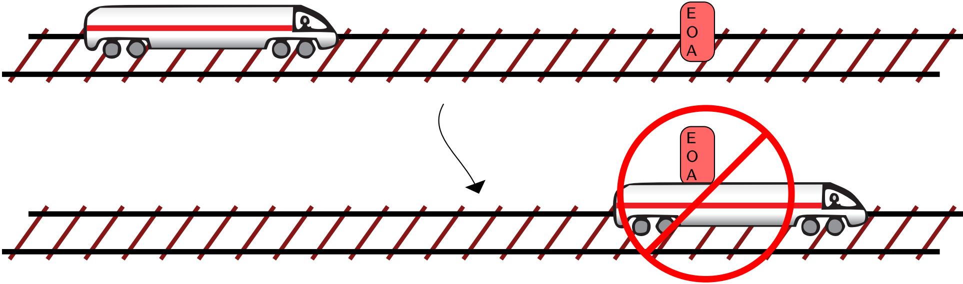
Refined ETCS Control

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

Refined ETCS Control

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

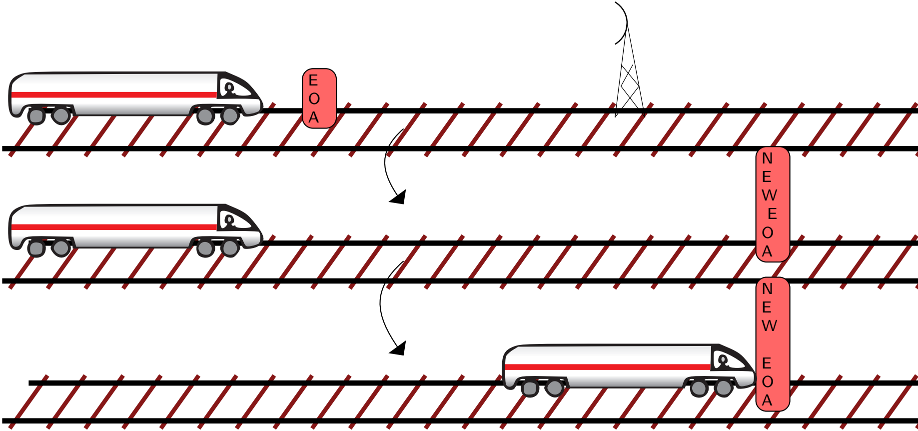
$$\tau.v^2 = \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \tau.p) \to [ETCS_{aug}](\tau.p \ge \mathbf{m}.e \to \tau.v \le \mathbf{m}.d)$$



Proposition (Safety)

$$\mathcal{C} \to$$
$$[ETCS](\tau.p \geq \mathbf{m}.e \to \tau.v \leq \mathbf{m}.d)$$

ETCS Liveness



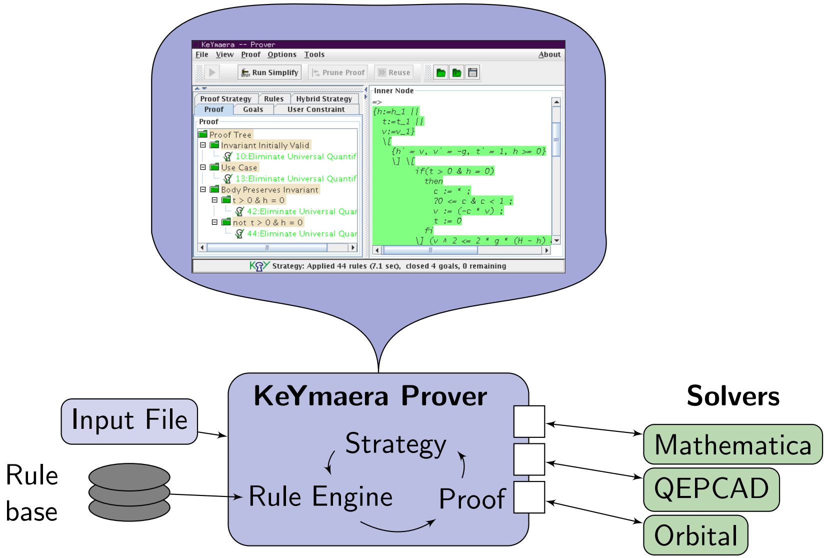
Proposition (Liveness)

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

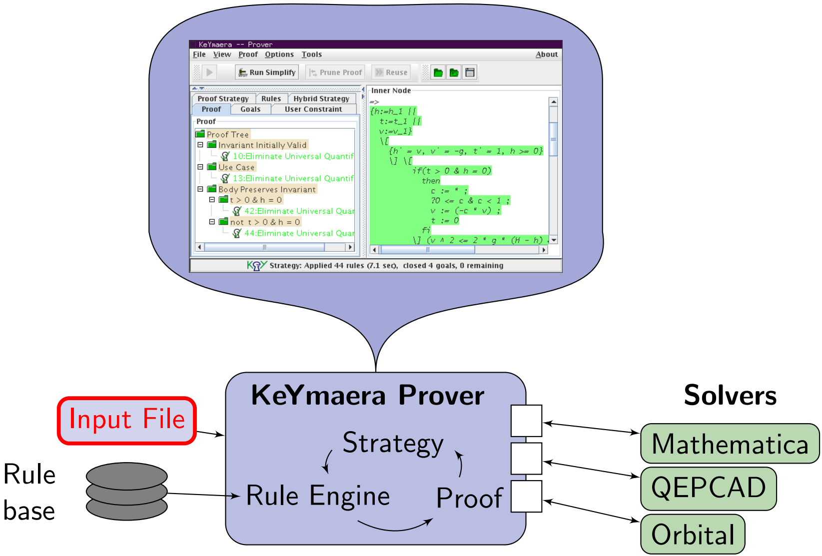
Outline

1. Motivation
2. Train Control
   - Separation Principle
   - Parametric ETCS
3. Parametric European Train Control System
   - Controllability
   - Reactivity
   - Refined Control
   - Safety
   - Liveness
4. Proving ETCS in KeYmaera
   - Architecture
   - KeYmaera Problem Input
   - KeYmaera Rule Base
   - Real Arithmetic, Computer Algebra and Automation
   - Experiments

KeYmaera Architecture



KeYmaera Architecture



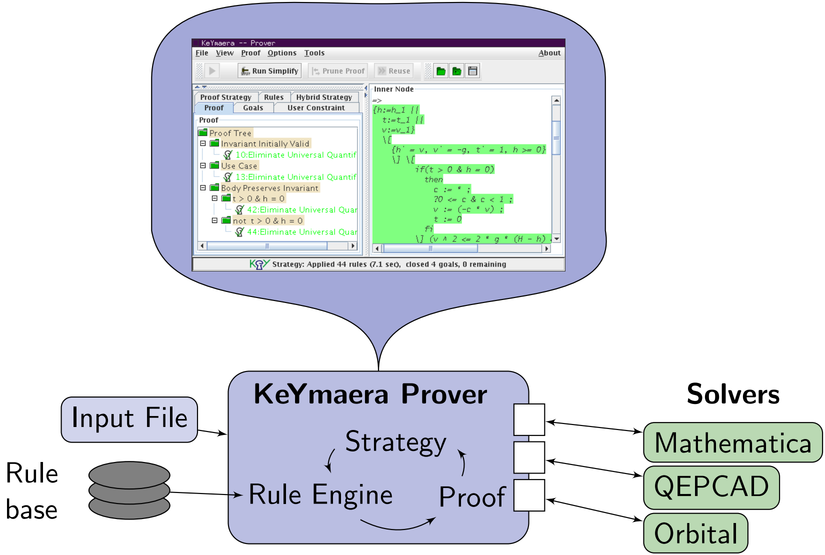
KeYmaera Problem Specification Input File .key

```
\functions {
  R ep;     R b;    R A;
}
\problem {
\[ R SB, a, v, z, t, m; \] (
  ( v^2 <= 2*b*(m-z) & b > 0 & A>=0)
->
 \[(
  SB := (v^2)/(2*b) + ((A/b)+1)*((A/2)*ep^2 + ep*v);
  ((?m - z <= SB; a:= -b)
   ++ (?m - z >= SB; a:=A));
  t:=0;
  {z`=v, v` = a, t`=1, (v >= 0 & t <= ep)}
  )*
  \] (z <= m)
)
}
```

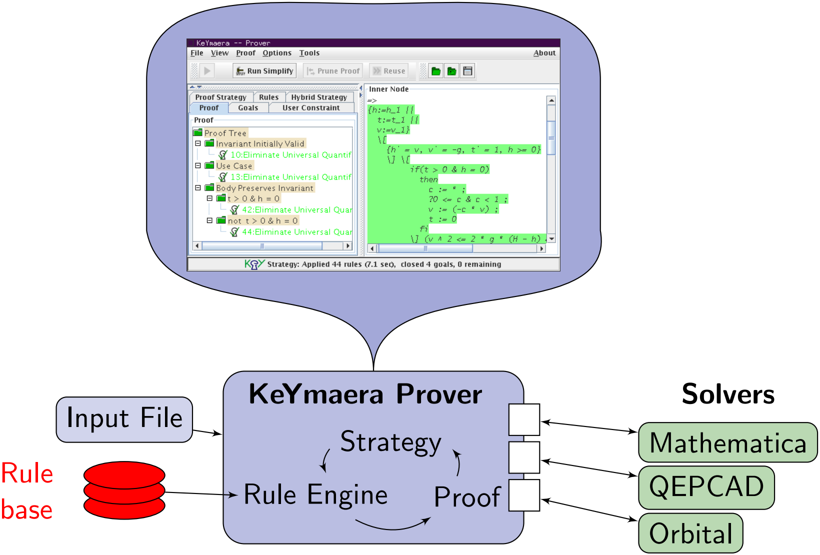
KeYmaera Problem Specification Input File .key

```
\functions { R b;     R A;    R ep; }
\problem {
\[ R r,SB,mo,t,a,v,z,m,d,do,drive,brake,state;drive:=0
   (v^2-d^2 <= 2*b*(m-z) & d>=0 & b>0 & A>=0 & ep>=0)
-> \[((
      (do:=d; mo:=m; m:=*; d:=*; r:=*;
       ?d>=0 & do^2-d^2<=2*b*(m-mo) & r>=0)
      ++ (state := brake)
     ) ++ (
       ((?v <= r; a:=*; ?a >=-b & a <= A)
       ++ (?v >= r; a:=*; ?a <0 & a >= -b));
       SB := (v^2-d^2)/(2*b) + (A/b+1)*(A/2*ep^2+ep*v);
       if (m-z <= SB | state=brake) then a:= b fi;
       (t:=0; {z'=v,v'=a,t'=1, (v>=0 & t<=ep)})
     )
     )*\] (z>=m -> v<=d)    )
}
```

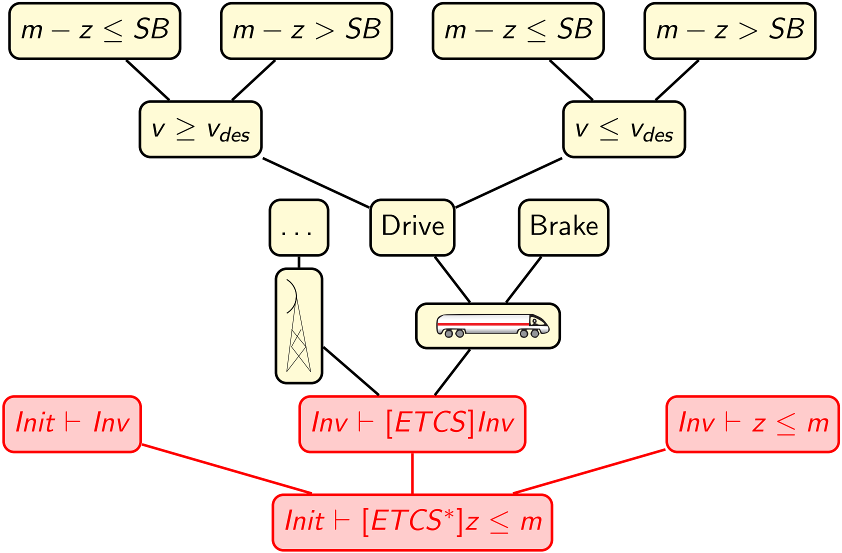
KeYmaera Architecture



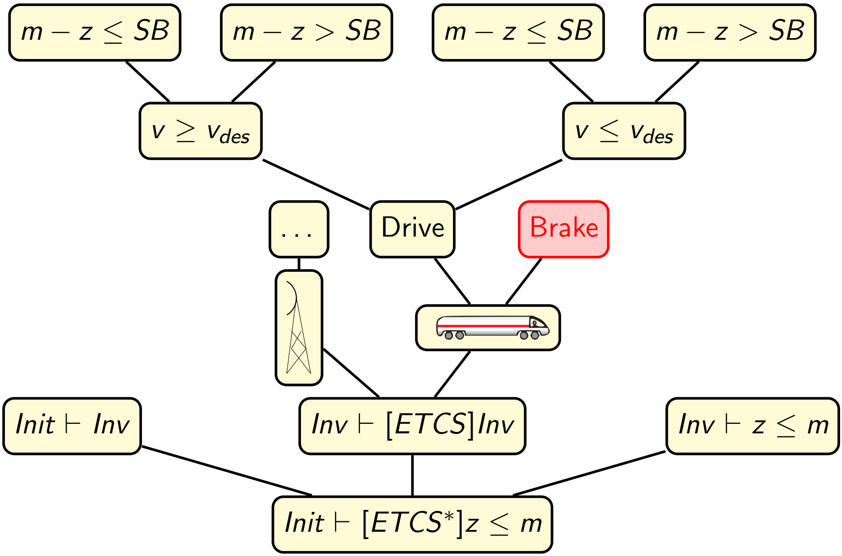
KeYmaera Architecture



Proof Sketch



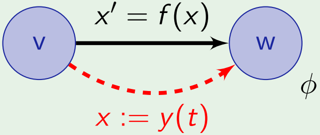
Proof Sketch



Handling Differential Equations

Example

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

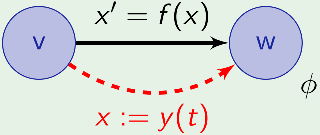


$$\frac{}{\ldots \vdash [z' = v, v' = -b] z \leq m}$$

Handling Differential Equations

Example

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

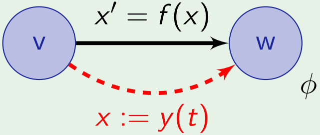


$$\frac{\overline{\ldots \vdash \forall t \geq 0\,[z := -\frac{1}{2}bt^2 + tv + z]z \leq m}}{\ldots \vdash [z' = v, v' = -b]z \leq m}$$

Handling Differential Equations

Example

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



$$\cfrac{\cfrac{\ldots \vdash \forall t \geq 0\ \left(-\frac{1}{2}bt^2 + tv + z \leq m\right)}{\ldots \vdash \forall t \geq 0\,[z := -\frac{1}{2}bt^2 + tv + z]z \leq m}}{\ldots \vdash [z' = v, v' = -b]z \leq m}$$

KeYmaera Rule Base

$$\frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \land \psi, \Delta}$$

```
and_right {
  \find (==> b & c)
  \replacewith(==> b);
  \replacewith(==> c)
  \heuristics(split,beta)
};
```

KeYmaera Rule Base

$$\frac{\Gamma \vdash [\alpha]\phi, \Delta \quad \Gamma \vdash [\beta]\phi, \Delta}{\Gamma \vdash [\alpha \cup \beta]\phi, \Delta}$$

```
box_choice_right {
  \find (==> \[ #dl ++ #dl2 \](post))
  \replacewith(==> \[#dl\](post));
  \replacewith(==> \[#dl2\](post))
  \heuristics(simplify_prog)
};
```

KeYmaera Rule Base

$$\frac{\Gamma \vdash \langle \mathcal{S}(t) \rangle \phi, \Delta}{\Gamma \vdash [x_1' = \theta_1, .., x_n' = \theta_n]\phi, \Delta}$$

```
ODESolve_right {
  \find (==> \[ #simpleode \](post))
  \replacewith(==> #ODESolve(\[#simpleode\](post)))
  \heuristics(diff_solve, diff_rule)
  \displayname "ODESolve"
};
```

$$\frac{\Gamma \vdash \langle \mathcal{S}(t) \rangle \phi, \Delta}{\Gamma \vdash [x_1' = \theta_1, \ldots, x_n' = \theta_n]\phi, \Delta}$$

```
ODESolve_right {
  \find (==> \[ #simpleode \](post))
  \replacewith(==> #ODESolve(\[#simpleode\](post)))
  \heuristics(diff_solve, diff_rule)
  \displayname "ODESolve"
};
```

Using meta-operator #ODESolve implemented in Java

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

```
all_left {
  \find (\forall u; b ==>)
  \replacewith ({\subst u; q}(b) ==>)
  \heuristics(gamma)
};
ex_left {
  \find (\exists u; b ==>)
  \varcond (\new(sk, \dependingOn(b)))
  \replacewith ({\subst u; sk}b ==>)
  \heuristics(delta)
};
```

KeYmaera Rule Base

$$\frac{\vdash \mathrm{QE}(\forall X\,(\Phi(X) \vdash \Psi(X)))}{\Phi(s(X_1,\ldots,X_n)) \vdash \Psi(s(X_1,\ldots,X_n))}$$

$$\frac{\vdash \mathrm{QE}(\exists X\ \bigwedge_i(\Phi_i \vdash \Psi_i))}{\Phi_1 \vdash \Psi_1 \quad \ldots \quad \Phi_n \vdash \Psi_n}$$

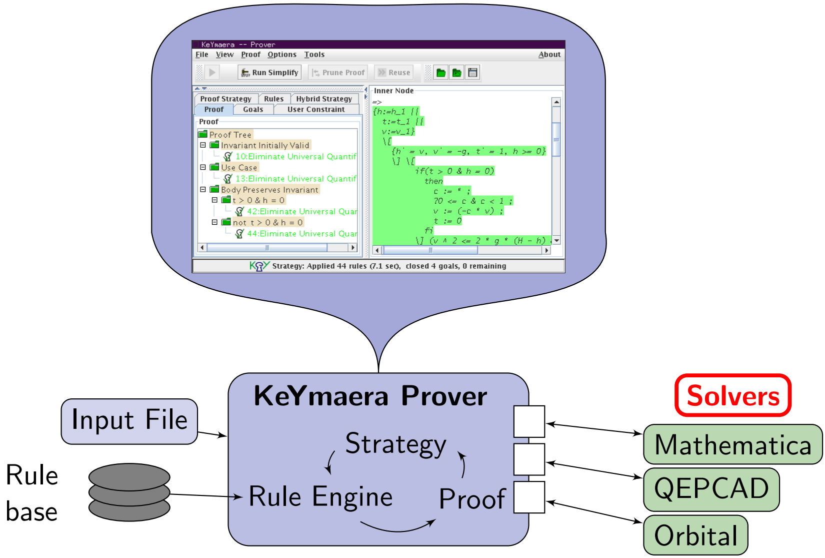
KeYmaera Rule Base

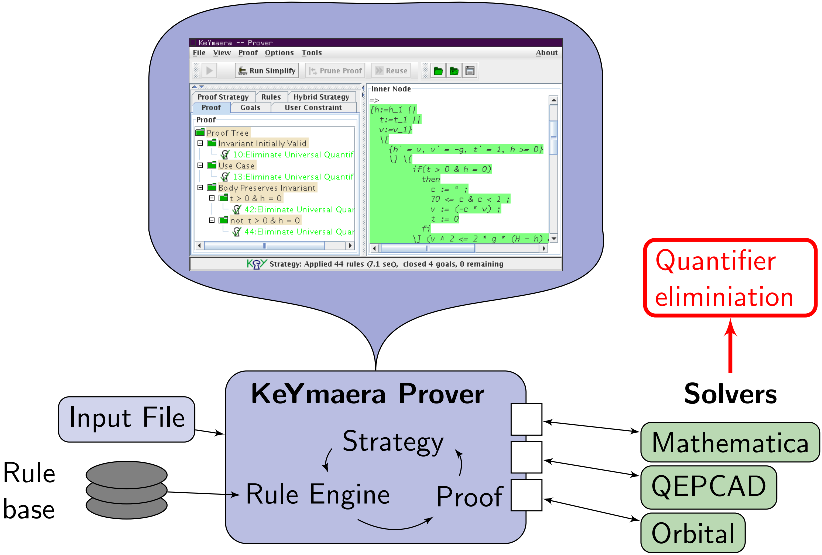
$$\frac{\vdash \mathrm{QE}(\forall X\,(\Phi(X) \vdash \Psi(X)))}{\Phi(s(X_1,\ldots,X_n)) \vdash \Psi(s(X_1,\ldots,X_n))}$$

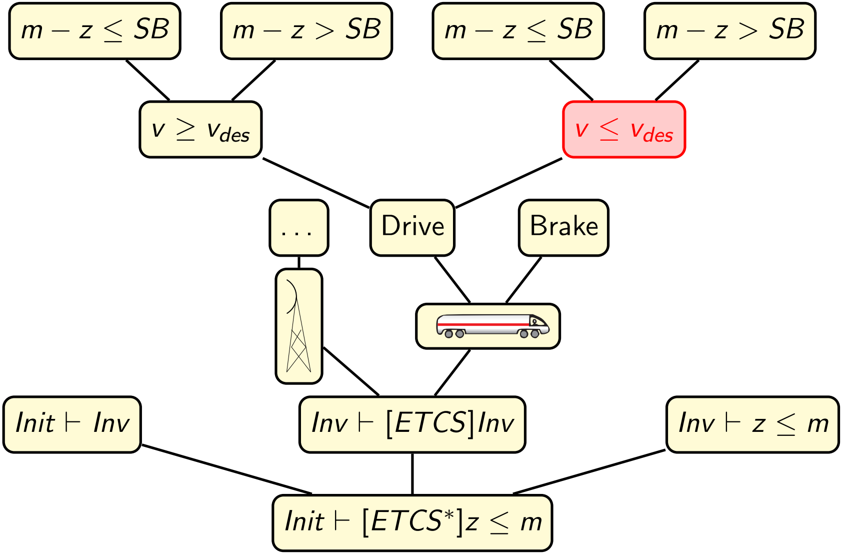
$$\frac{\vdash \mathrm{QE}(\exists X\ \bigwedge_i(\Phi_i \vdash \Psi_i))}{\Phi_1 \vdash \Psi_1 \quad \ldots \quad \Phi_n \vdash \Psi_n}$$

Using built-in rule implemented in Java

KeYmaera Architecture




KeYmaera -- Prover
File View Proof Options Tools About
Run Simplify
Prune Proof
Reuse
Proof Strategy Rules Hybrid Strategy
Proof Goals User Constraint
Proof Tree
Invariant Initially Valid
10:Eliminate Universal Quantif
Use Case
13:Eliminate Universal Quantif
Body Preserves Invariant
t > 0 & h = 0
42:Eliminate Universal Quar
not t > 0 & h = 0
44:Eliminate Universal Quar
Inner Node
Strategy: Applied 44 rules (7.1 sec), closed 4 goals, 0 remaining
Quantifier eliminiation
KeYmaera Prover
Input File
Strategy
Rule base
Rule Engine
Proof
Solvers
Mathematica
QEPCAD
Orbital


$m - z \le SB$
$m - z > SB$
$m - z \le SB$
$m - z > SB$
$v \ge v_{des}$
$v \le v_{des}$
...
Drive
Brake
$Init \vdash Inv$
$Inv \vdash [ETCS]Inv$
$Inv \vdash z \le m$
$Init \vdash [ETCS^*]z \le m$

Proof Sketch

$m - z \leq SB$ $m - z > SB$ $m - z \leq SB$ $m - z > SB$

$v \geq v_{des}$ $v \leq v_{des}$

Example

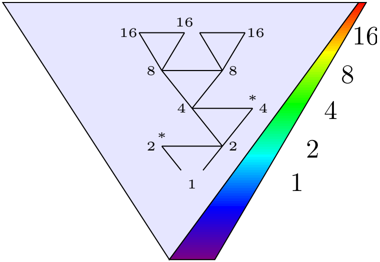
$$m - z \geq \left(\frac{A}{b} + 1\right)\left(\varepsilon v + \frac{A}{2}\varepsilon^2\right) + \frac{v^2 - d^2}{2b} \land 0 \leq a \leq A \land 0 \leq v \leq vdes$$
$$\land v^2 - d^2 \leq 2b(m - z) \land d \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0$$
$$\vdash$$
$$\forall t \geq 0\, ((\forall 0 \leq \tilde{t} \leq t\ (a\tilde{t} + v \geq 0 \land \tilde{t} \leq \varepsilon))$$
$$\rightarrow (at + v)^2 - d^2 \leq 2b(m - (\tfrac{1}{2}at + tv + z)) \land at + v \geq 0 \land d \geq 0)$$

$Init \vdash [ETCS^*]z \leq m$

Quantifier Elimination and Proof Strategies

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

$$\frac{\vdash A \quad \vdash B}{\vdash A \land B}$$



Experimental Results

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

$\infty \mathrel{\widehat{=}}$ more than five hours

Experimental Results

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

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

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

A. Platzer and J.-D. Quesel.
KeYmaera: A hybrid theorem prover for hybrid systems.
In A. Armando, P. Baumgartner, and G. Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 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.