KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems

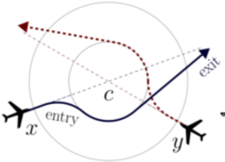
Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, André Platzer

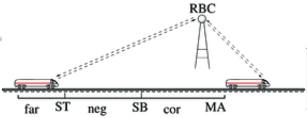
Presented at CADE-25

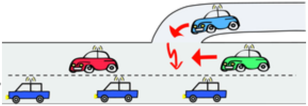
August 7, 2015

Milieu

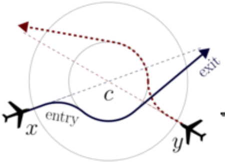
Safety-critical control software is now a fact of every-day life.

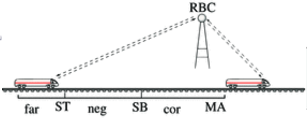


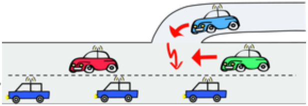




Safety-critical control software is now a fact of every-day life.







How can we design cyber-physical systems people can bet their lives on?
– Jeanette Wing

A Prototypical Hybrid System

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Hybrid System

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline for a $\varphi \rightarrow [\{\text{ctrl}; \text{plant}\}^*]\psi$ Model:

1. Propositional Reasoning

A Prototypical Hybrid System

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*]v \geq 0$$

A Prototypical Proof Outline for a $\varphi \rightarrow [\{\text{ctrl}; \text{plant}\}^*]\psi$ Model:

1. Propositional Reasoning
2. Identify System Loop Invariant

A Prototypical Hybrid System

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline for a $\varphi \rightarrow [\{\text{ctrl}; \text{plant}\}^*]\psi$ Model:

1. Propositional Reasoning
2. Identify System Loop Invariant
3. Symbolically Execute Control Program

A Prototypical Hybrid System

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline for a $\varphi \rightarrow [\{\text{ctrl}; \text{plant}\}^*]\psi$ Model:

1. Propositional Reasoning
2. Identify System Loop Invariant
3. Symbolically Execute Control Program
4. Solve ODE or identify Differential Invariant(s)

A Prototypical Hybrid System

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline for a $\varphi \rightarrow [\{\text{ctrl}; \text{plant}\}^*]\psi$ Model:

1. Propositional Reasoning
2. Identify System Loop Invariant
3. Symbolically Execute Control Program
4. Solve ODE or identify Differential Invariant(s)
5. Appeal to Decision Procedure for Real Arithmetic

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline:

1. Propositional Reasoning
2. Identify System Loop Invariant
3. Symbolically Execute Control Program
4. Solve ODE or identify Differential Invariant(s)
5. Appeal to Decision Procedure for Real Arithmetic

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline:

`ImplyR` &

2. Identify System Loop Invariant
3. Symbolically Execute Control Program
4. Solve ODE or identify Differential Invariant(s)
5. Appeal to Decision Procedure for Real Arithmetic

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline:

`ImplyR` &

`Loop("`$v \geq 0$`")` &

3. Symbolically Execute Control Program
4. Solve ODE or identify Differential Invariant(s)
5. Appeal to Decision Procedure for Real Arithmetic

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*]v \geq 0$$

A Prototypical Proof Outline:

`ImplyR &`

`Loop("`$v \geq 0$`") &`

`Seq & Choice & BoxAssign &`

4. Solve ODE or identify Differential Invariant(s)
5. Appeal to Decision Procedure for Real Arithmetic

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \wedge A > 0 \wedge B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*]v \geq 0$$

A Prototypical Proof Outline:

```
ImplyR &
Loop("v ≥ 0") &
Seq & Choice & BoxAssign &
DiffInv("v ≥ 0") &
```

5. Appeal to Decision Procedure for Real Arithmetic

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline:

```
ImplyR &
Loop("v ≥ 0") &
Seq & Choice & BoxAssign &
DiffInv("v ≥ 0") &
Arithmetic & Close
```

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \wedge A > 0 \wedge B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*]v \geq 0$$

A Prototypical Proof Outline:

```
Prop &                                   ⇐
Loop("v ≥ 0") &
SymbolicExecution &                      ⇐
DiffInv("v ≥ 0") &
Arithmetic & Close
```

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \wedge A > 0 \wedge B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline:

```
Prop &
Loop("v ≥ 0") &
SymbolicExecution &
DiffInv(DIGen) &                                    ⇐
Arithmetic & Close
```

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline:

```
Prop &
Loop(LoopInvGen) &                                   ⇐
SymbolicExecution &
DiffInv(DIGen) &
Arithmetic & Close
```

Motivation: Sketching and Searching

Theorem

$$v \geq 0 \land A > 0 \land B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline:

```
Prop &
Loop(LoopInvGen) &
SymbolicExecution &
DiffInv(DIGen) &
Arithmetic & Close
```
$\Leftarrow$

Motivation: Sketching and Searching

Theorem

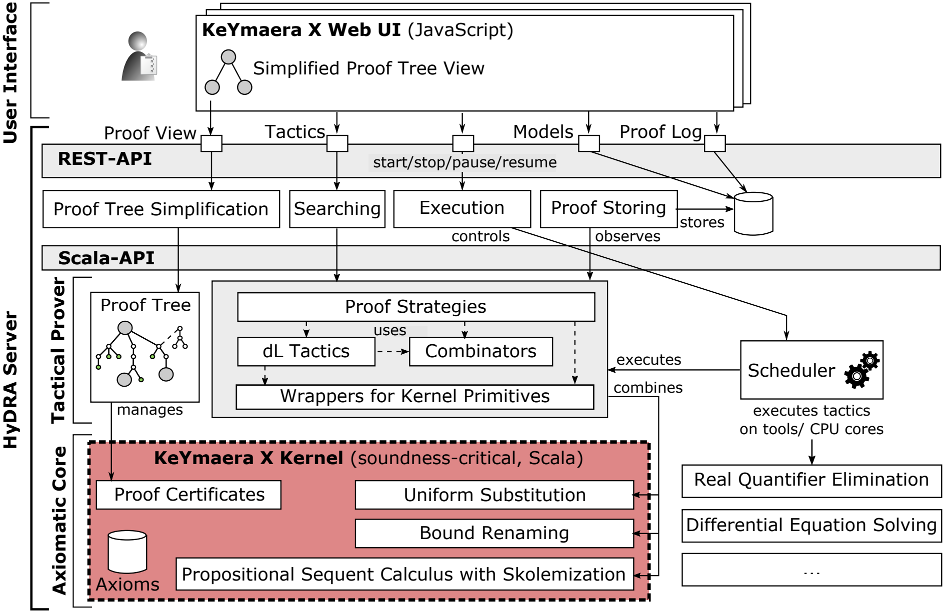
$$v \geq 0 \wedge A > 0 \wedge B > 0 \rightarrow$$
$$[\{\{a := A \cup a := -B\}; \{x' = v, v' = a \& v \geq 0\}\}^*] v \geq 0$$

A Prototypical Proof Outline:

```
Prop &
Loop(LoopInvGen) &
SymbolicExecution &
DiffInv(DIGen) &
Arithmetic & Close
```

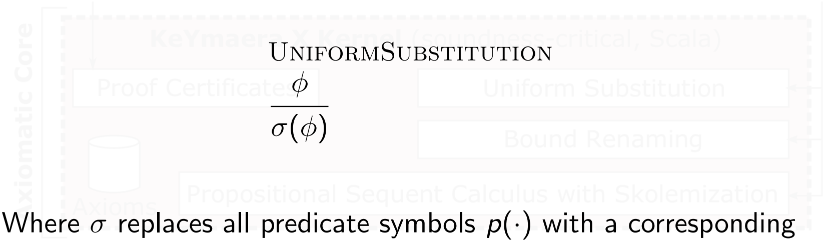
Contributions

Small Core	Increases trust, enables experimentation
Tactics	Bridging between a Hilbert-style Logic and a Gentzen-style deduction systems
Extensible	New logics, proof rules, axioms
Customizable	New interfaces (CPS Education, usability research, industry applications)



User Interface
KeYmaera X Web UI (JavaScript)
Simplified Proof Tree View
Proof View
Tactics
Models
Proof Log
REST-API
start/stop/pause/resume
Proof Tree Simplification
Searching
Execution
Proof Storing
stores
controls
observes
Scala-API
HyDRA Server
Tactical Prover
Proof Tree
Proof Strategies
uses
dL Tactics
Combinators
Wrappers for Kernel Primitives
executes
combines
Scheduler
executes tactics
on tools/ CPU cores
manages
Axiomatic Core
KeYmaera X Kernel (soundness-critical, Scala)
Proof Certificates
Uniform Substitution
Bound Renaming
Axioms
Propositional Sequent Calculus with Skolemization
Real Quantifier Elimination
Differential Equation Solving
...

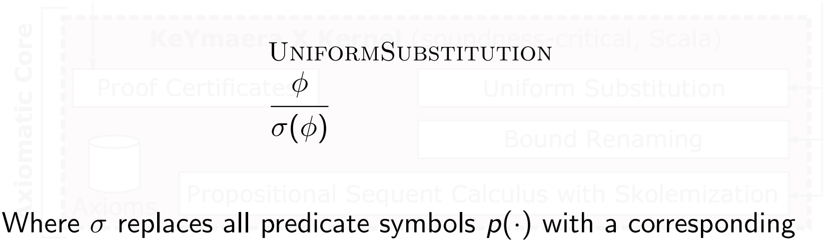
Core: Uniform Substitution



$$\textsc{UniformSubstitution}\quad \frac{\phi}{\sigma(\phi)}$$

Where σ replaces all predicate symbols $p(\cdot)$ with a corresponding formula.

Core: Uniform Substitution



$$\frac{\phi}{\sigma(\phi)}\ \textsc{UniformSubstitution}$$

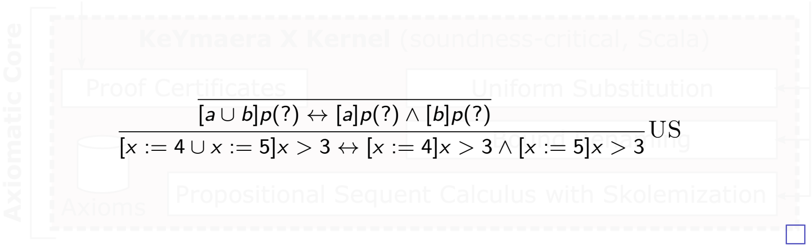
Where σ replaces all predicate symbols $p(\cdot)$ with a corresponding formula.
Similarly for other syntactic objects (e.g., program constants a).

Core: Uniform Substitution

Theorem

$[x := 4 \cup x := 5]x > 3 \leftrightarrow [x := 4]x > 3 \wedge [x := 5]x > 3$

Proof.



$$\dfrac{\overline{[a \cup b]p(?) \leftrightarrow [a]p(?) \wedge [b]p(?)}}{[x := 4 \cup x := 5]x > 3 \leftrightarrow [x := 4]x > 3 \wedge [x := 5]x > 3}\,\mathrm{US}$$

□

Definition of substitution σ:

$$a \leadsto [x := 4]$$
$$b \leadsto [x := 5]$$
$$p(?) \leadsto x > 3$$

Core: Uniform Substitution

Theorem

$[x := 4 \cup x := 5]x > 3 \leftrightarrow [x := 4]x > 3 \wedge [x := 5]x > 3$

Proof.

$$\dfrac{\dfrac{*}{[a \cup b]p(?) \leftrightarrow [a]p(?) \wedge [b]p(?)}\,\textsc{Axiom}}{[x := 4 \cup x := 5]x > 3 \leftrightarrow [x := 4]x > 3 \wedge [x := 5]x > 3}\,\mathrm{US}$$

□

Definition of substitution σ:

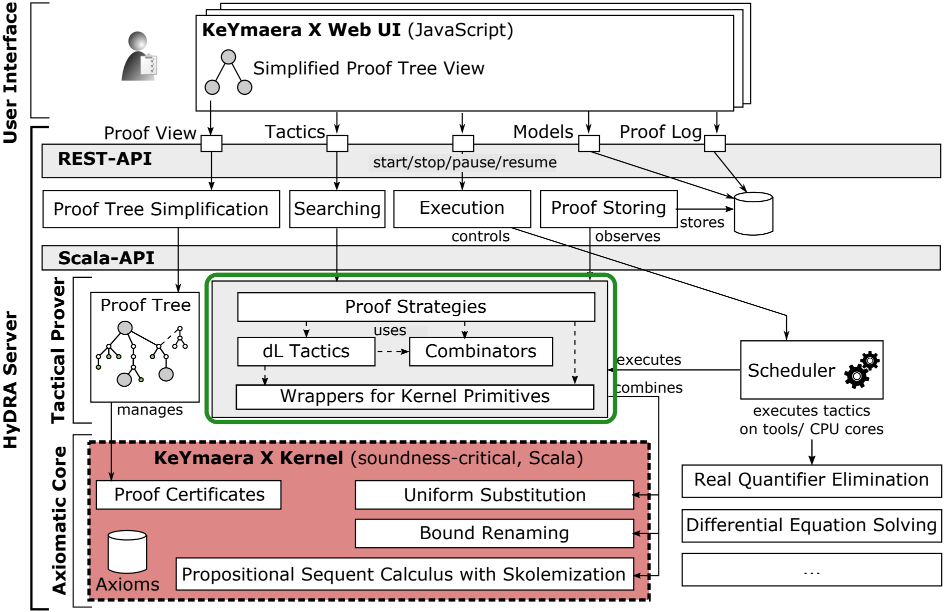
$$a \rightsquigarrow [x := 4]$$
$$b \rightsquigarrow [x := 5]$$
$$p(?) \rightsquigarrow x > 3$$

The Axiom File contains very nearly verbatim copies of axioms from papers:

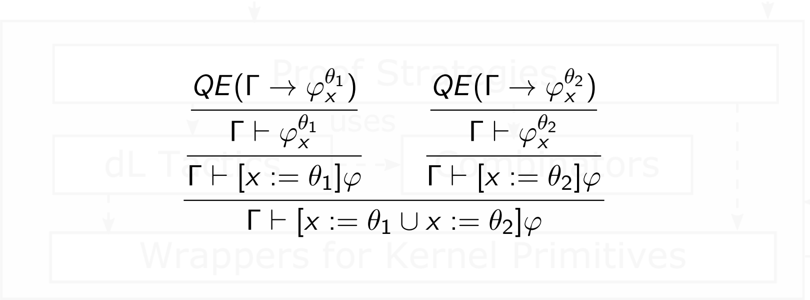
```
Axiom "K␣modal␣modus␣ponens".
  [a;](p(?)->q(?)) -> (([a;]p(?)) -> ([a;]q(?)))
End.

Axiom "DC␣differential␣cut".
  ([c&H(?);]p(?) <-> [c&(H(?)&r(?));]p(?)) <- [c&H(?);]r(?)
End.

Axiom "[++]␣choice".
  [a ++ b]p(?) <-> ([a;]p(?) & [b;]p(?)).
End.
```

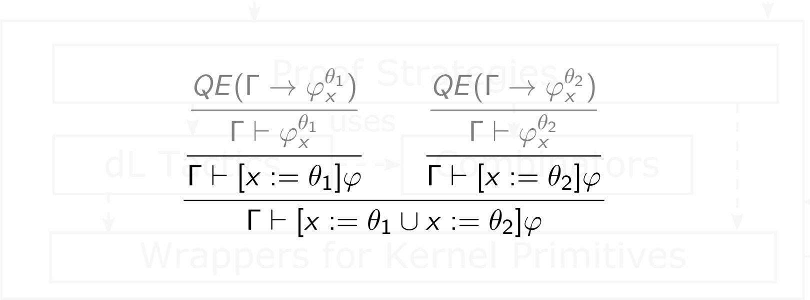

User Interface
KeYmaera X Web UI (JavaScript)
Simplified Proof Tree View
Proof View
Tactics
Models
Proof Log
REST-API
start/stop/pause/resume
Proof Tree Simplification
Searching
Execution
Proof Storing
stores
controls
observes
Scala-API
HyDRA Server
Tactical Prover
Proof Tree
manages
Proof Strategies
uses
dL Tactics
Combinators
Wrappers for Kernel Primitives
executes
combines
Scheduler
executes tactics
on tools/ CPU cores
Axiomatic Core
KeYmaera X Kernel (soundness-critical, Scala)
Proof Certificates
Uniform Substitution
Bound Renaming
Axioms
Propositional Sequent Calculus with Skolemization
Real Quantifier Elimination
Differential Equation Solving
...

Sequent Calculus as Tactics



$$\dfrac{\dfrac{\dfrac{QE(\Gamma \rightarrow \varphi_x^{\theta_1})}{\Gamma \vdash \varphi_x^{\theta_1}}}{\Gamma \vdash [x := \theta_1]\varphi} \qquad \dfrac{\dfrac{QE(\Gamma \rightarrow \varphi_x^{\theta_2})}{\Gamma \vdash \varphi_x^{\theta_2}}}{\Gamma \vdash [x := \theta_2]\varphi}}{\Gamma \vdash [x := \theta_1 \cup x := \theta_2]\varphi}$$

Sequent Calculus as Tactics



$$\dfrac{\dfrac{\dfrac{QE(\Gamma \to \varphi_x^{\theta_1})}{\Gamma \vdash \varphi_x^{\theta_1}}}{\Gamma \vdash [x := \theta_1]\varphi} \qquad \dfrac{\dfrac{QE(\Gamma \to \varphi_x^{\theta_2})}{\Gamma \vdash \varphi_x^{\theta_2}}}{\Gamma \vdash [x := \theta_2]\varphi}}{\Gamma \vdash [x := \theta_1 \cup x := \theta_2]\varphi}$$

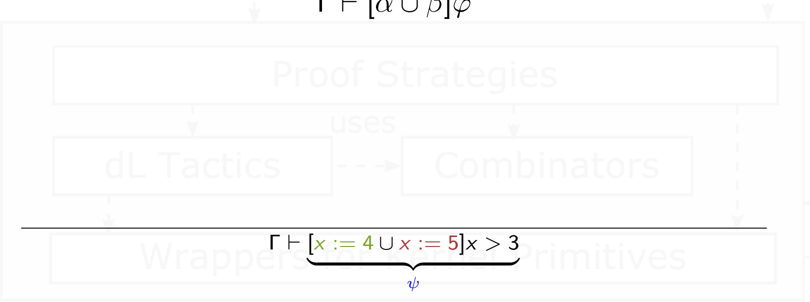
Tactical Proving

$$\textsc{BoxChoice}\ \frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$

Tactical Proving

BoxChoice

$$\frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$



$$\frac{}{\Gamma \vdash \underbrace{[x := 4 \cup x := 5]x > 3}_{\psi}}$$

Tactical Proving

$$\textsc{BoxChoice}\quad \frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$

$$\frac{\underline{[a \cup b]p(?) \leftrightarrow [a]p(?) \cup [b]p(?)}}{\Gamma \vdash \underbrace{[x := 4 \cup x := 5]x > 3}_{\psi}}$$

$\sigma =$

$$a \rightsquigarrow x := 4$$
$$b \rightsquigarrow x := 5$$
$$p(?) \rightsquigarrow x > 3$$

Tactical Proving

BoxChoice

$$\frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$

$$\frac{\dfrac{[a \cup b]p(?) \leftrightarrow [a]p(?) \cup [b]p(?)}{\psi \leftrightarrow [x := 4]x > 3 \wedge [x := 5]x > 3}}{\Gamma \vdash \underbrace{[x := 4 \cup x := 5]x > 3}_{\psi}}$$

$\sigma =$

$$a \rightsquigarrow x := 4$$

$$b \rightsquigarrow x := 5$$

$$p(?) \rightsquigarrow x > 3$$

Tactical Proving

BoxChoice

$$\frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$

$$\dfrac{\dfrac{[a \cup b]p(?) \leftrightarrow [a]p(?) \cup [b]p(?)}{\psi \leftrightarrow [x := 4]x > 3 \wedge [x := 5]x > 3} \qquad \dfrac{}{\Gamma, \psi \leftrightarrow [x := 4]x > 3 \wedge [x := 5]x > 3 \vdash \psi}}{\Gamma \vdash \underbrace{[x := 4 \cup x := 5]x > 3}_{\psi}}$$

$\sigma =$

$$a \rightsquigarrow x := 4$$

$$b \rightsquigarrow x := 5$$

$$p(?) \rightsquigarrow x > 3$$

Tactical Proving

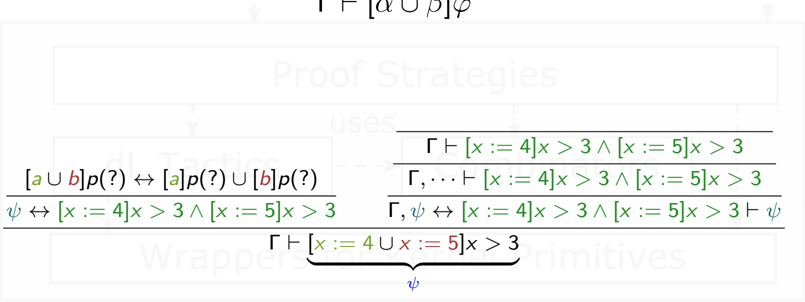
$$\text{BoxChoice}\quad \frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$

$$\dfrac{\dfrac{[a \cup b]p(?) \leftrightarrow [a]p(?) \cup [b]p(?)}{\psi \leftrightarrow [x := 4]x > 3 \land [x := 5]x > 3} \qquad \dfrac{\Gamma, \cdots \vdash [x := 4]x > 3 \land [x := 5]x > 3}{\Gamma, \psi \leftrightarrow [x := 4]x > 3 \land [x := 5]x > 3 \vdash \psi}}{\Gamma \vdash \underbrace{[x := 4 \cup x := 5]x > 3}_{\psi}}$$

$$\sigma = \begin{aligned} a &\rightsquigarrow x := 4 \\ b &\rightsquigarrow x := 5 \\ p(?) &\rightsquigarrow x > 3 \end{aligned}$$

Tactical Proving

$$\textsc{BoxChoice}\quad \frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$



$$\frac{\dfrac{[a \cup b]p(?) \leftrightarrow [a]p(?) \cup [b]p(?)}{\psi \leftrightarrow [x := 4]x > 3 \land [x := 5]x > 3} \qquad \dfrac{\dfrac{\dfrac{}{\Gamma \vdash [x := 4]x > 3 \land [x := 5]x > 3}}{\Gamma, \cdots \vdash [x := 4]x > 3 \land [x := 5]x > 3}}{\Gamma, \psi \leftrightarrow [x := 4]x > 3 \land [x := 5]x > 3 \vdash \psi}}{\Gamma \vdash \underbrace{[x := 4 \cup x := 5]x > 3}_{\psi}}$$

$\sigma =$

$$a \rightsquigarrow x := 4$$

$$b \rightsquigarrow x := 5$$

$$p(?) \rightsquigarrow x > 3$$

Tactical Proving

BoxChoice

$$\frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$

$$\cfrac{\cfrac{[a \cup b]p(?) \leftrightarrow [a]p(?) \cup [b]p(?)}{\psi \leftrightarrow [x := 4]x > 3 \land [x := 5]x > 3} \qquad \cfrac{\cfrac{\cfrac{\Gamma \vdash [x := 4]x > 3 \qquad \Gamma \vdash [x := 5]x > 3}{\Gamma \vdash [x := 4]x > 3 \land [x := 5]x > 3}}{\Gamma, \cdots \vdash [x := 4]x > 3 \land [x := 5]x > 3}}{\Gamma, \psi \leftrightarrow [x := 4]x > 3 \land [x := 5]x > 3 \vdash \psi}}{\Gamma \vdash \underbrace{[x := 4 \cup x := 5]x > 3}_{\psi}}$$

$\sigma =$

$$a \rightsquigarrow x := 4$$

$$b \rightsquigarrow x := 5$$

$$p(?) \rightsquigarrow x > 3$$

Hilbert and Gentzen Meet at Church

$$\textsc{BoxChoice}\quad \frac{\Gamma \vdash [\alpha]\varphi \qquad \Gamma \vdash [\beta]\varphi}{\Gamma \vdash [\alpha \cup \beta]\varphi}$$

$$\dfrac{\dfrac{[a \cup b]p(?) \leftrightarrow [a]p(?) \cup [b]p(?)}{\psi \leftrightarrow [x := 4]x > 3 \land [x := 5]x > 3} \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash [x := 4]x > 3 \qquad \Gamma \vdash [x := 5]x > 3}{\Gamma \vdash [x := 4]x > 3 \land [x := 5]x > 3}}{\Gamma, \cdots \vdash [x := 4]x > 3 \land [x := 5]x > 3}}{\Gamma, \psi \leftrightarrow [x := 4]x > 3 \land [x := 5]x > 3 \vdash \psi}}{\Gamma \vdash \underbrace{[x := 4 \cup x := 5]x > 3}_{\psi}}$$

$\sigma =$

$$a \rightsquigarrow x := 4$$

$$b \rightsquigarrow x := 5$$

$$p(?) \rightsquigarrow x > 3$$

Hilbert and Gentzen Meet at Church

Contextual Box Assignment

```
CtxCut("φ ↔ ··· ∧ ··· ")
& onBranch(
  ("Show",
    USubst(a ⇝ x := 4, b ⇝ x := 5, p ⇝ x > 3)
    & AxiomCtx("[++] choice")
  ),
  ("Use",
    CtxEquiv(ante.length, 0) & AndR
  )
)
```

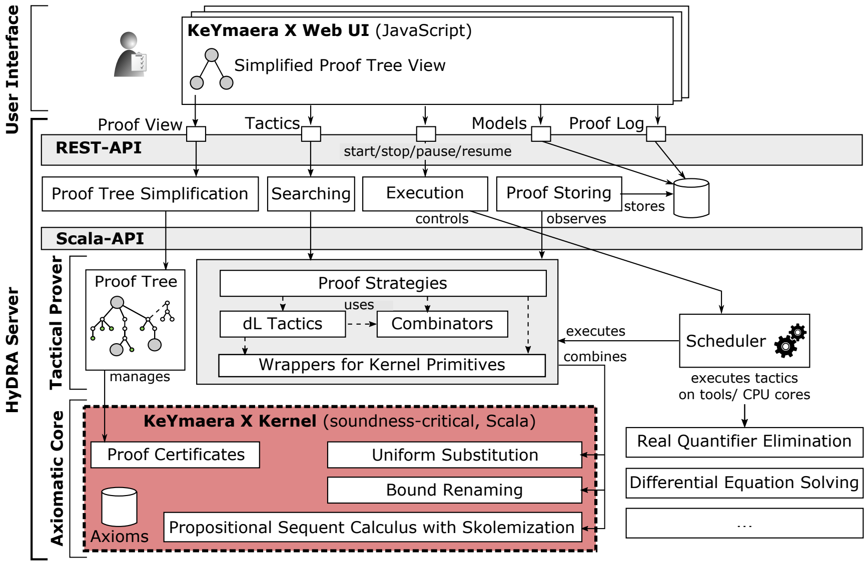
$[a \cup$ | > 3

$\psi \leftrightarrow$ | 3 | $\vdash \psi$

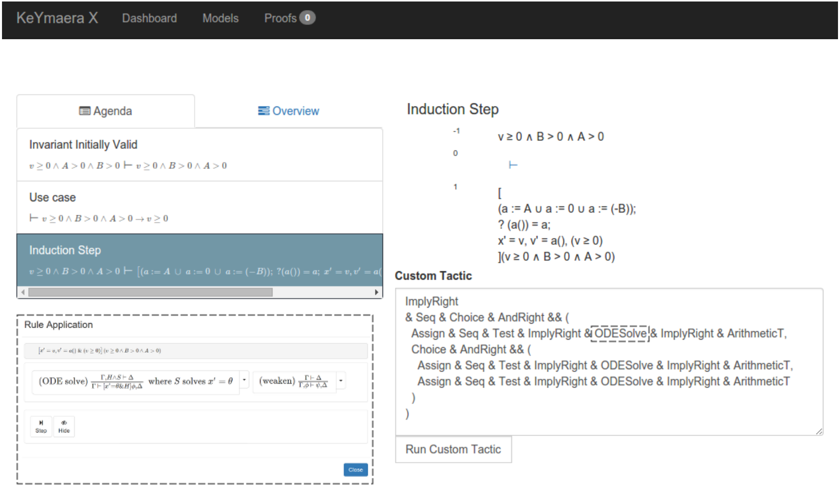
$\sigma =$

$b \rightsquigarrow x := 5$

$p(?) \rightsquigarrow x > 3$


User Interface
KeYmaera X Web UI (JavaScript)
Simplified Proof Tree View
Proof View
Tactics
Models
Proof Log
REST-API
start/stop/pause/resume
Proof Tree Simplification
Searching
Execution
Proof Storing
stores
controls
observes
Scala-API
HyDRA Server
Tactical Prover
Proof Tree
manages
Proof Strategies
uses
dL Tactics
Combinators
Wrappers for Kernel Primitives
executes
combines
Scheduler
executes tactics
on tools/ CPU cores
Real Quantifier Elimination
Differential Equation Solving
...
Axiomatic Core
KeYmaera X Kernel (soundness-critical, Scala)
Proof Certificates
Uniform Substitution
Bound Renaming
Axioms
Propositional Sequent Calculus with Skolemization

Web-Based User Interface



Kernel Comparison

System	LOC	
KeYmaera X	**1 682**	
KeYmaera	65 989	
KeY	51 328	
HOL Light	396	
Isabelle/Pure	8 113	
Nuprl	15 000	+ 50 000
Coq	20 000	
HSolver	20 000	
Flow*	25 000	
PHAVer	30 000	
dReal	50 000	+ millions
SpaceEx	100 000	
HyCreate2	6 081	+ user model analysis

Disclaimer: These self-reported estimates of the soundness-critical lines of code are to be taken with a grain of salt. Different languages, capabilities, styles ...

Conclusion

Small Core Increases trust, enables experimentation

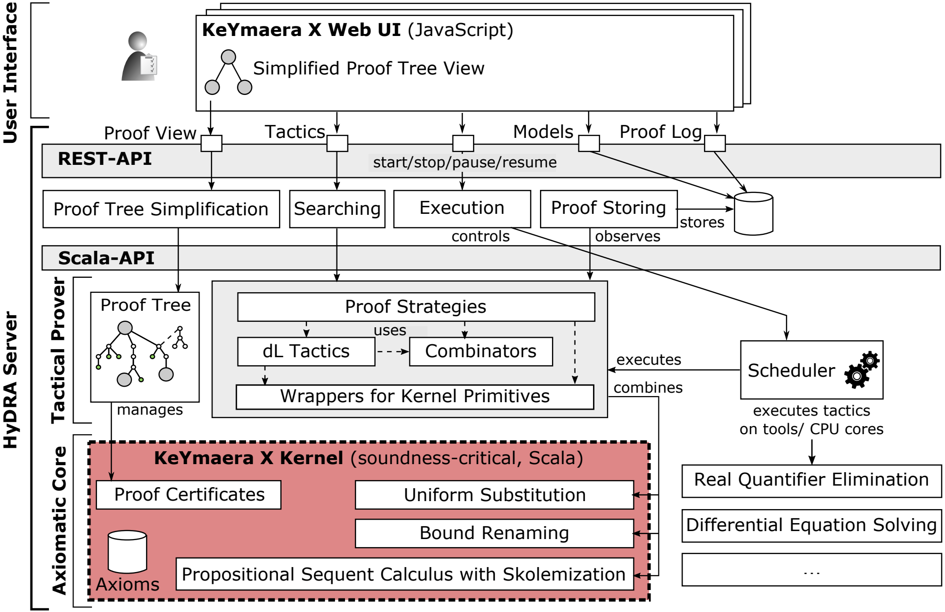
Tactics Bridging between a Hilbert-style Logic and a Gentzen-style deduction systems

Extensible New logics, proof rules, axioms

Customizable New interfaces (CPS Education, usability research, industry applications)

Thanks: Ran Ji, Jean-Baptiste Jeannin, Sarah Loos, João Martins, Khalil Ghorbal

Download: http://keymaeraX.org
Developer contact email: keymaerax@keymaerax.org


User Interface
KeYmaera X Web UI (JavaScript)
Simplified Proof Tree View
Proof View
Tactics
Models
Proof Log
REST-API
start/stop/pause/resume
Proof Tree Simplification
Searching
Execution
Proof Storing
stores
controls
observes
Scala-API
HyDRA Server
Tactical Prover
Proof Tree
Proof Strategies
uses
dL Tactics
Combinators
Wrappers for Kernel Primitives
manages
executes
combines
Scheduler
executes tactics
on tools/ CPU cores
Axiomatic Core
KeYmaera X Kernel (soundness-critical, Scala)
Proof Certificates
Uniform Substitution
Bound Renaming
Axioms
Propositional Sequent Calculus with Skolemization
Real Quantifier Elimination
Differential Equation Solving
...