

Refactoring, Refinement, and Reasoning

A Logical Characterization for Hybrid Systems

Stefan Mitsch^{1,2} Jan-David Quesel¹ André Platzer¹

¹Computer Science Department, Carnegie Mellon University

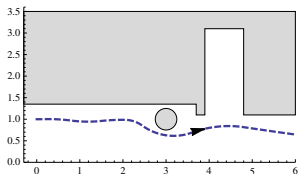
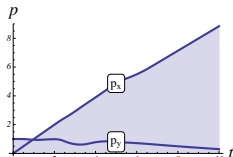
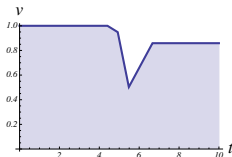
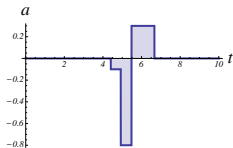
²Cooperative Information Systems, Johannes Kepler University

May 14, 2014

Hybrid Systems

Hybrid Systems are Challenging

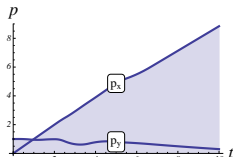
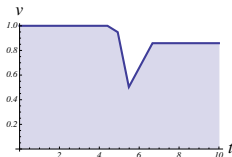
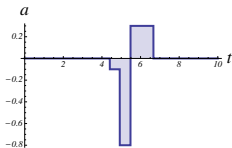
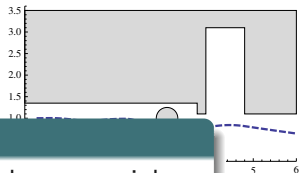
- ▶ Computation + Physical behavior
- ▶ Sensor uncertainty
- ▶ Disturbance
- ▶ Computation delay
- ▶ Many components



Hybrid Systems

Hybrid Systems are Challenging

- ▶ Computation + Physical behavior
 - ▶ Sensing **Challenge**
 - ▶ Disturbance Hybrid systems are almost impossible to get right without proper analysis
 - ▶ Control
 - ▶ Many
- **Formal verification**



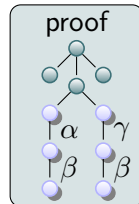
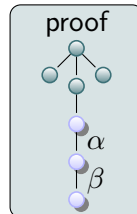
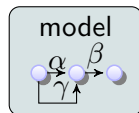
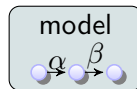
Formal Verification

Hybrid System Theorem Proving

- ▶ Symbolic execution of model
- ▶ Model structure reflected in proof
- ▶ Correctness properties

Safety Always stay safe

Liveness Ultimately complete
a task



Our Tools

KeYmaera Hybrid systems theorem prover

Synx Hybrid systems modeling

Iterative Development

Hybrid Systems Theorem Proving is Challenging

- ▶ Differential equations
- ▶ Complicated arithmetic

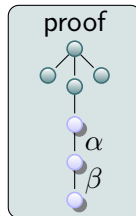
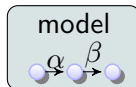
Iterative Development

Hybrid Systems Theorem Proving is Challenging

- ▶ Differential equations
- ▶ Complicated arithmetic

Manage complexity

- ▶ Start simple — verify



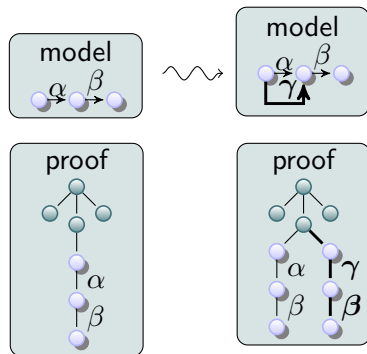
Iterative Development

Hybrid Systems Theorem Proving is Challenging

- ▶ Differential equations
- ▶ Complicated arithmetic

Manage complexity

- ▶ Start simple — verify
- ▶ Improve — verify — repeat



Iterative Development

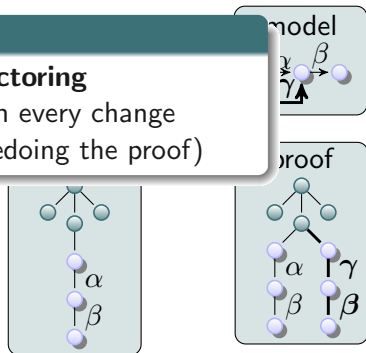
Hybrid Systems Theorem Proving is Challenging

- ▶ Differential equations
- ▶ Challenge

Proof-aware refactoring

Manually instead of reverification on every change
(retain soundness without redoing the proof)

- ▶ State simplification
- ▶ Improve — verify — repeat



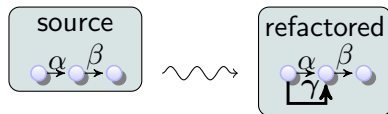
Proof-aware Refactoring

Refactoring Operation

- ▶ Transforms a source model into a refactored model
- ▶ Syntactic rewriting rule

For example

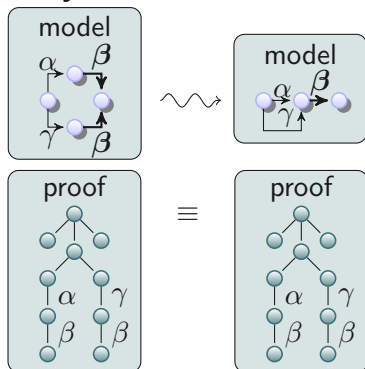
$$\frac{\text{conditions}}{\alpha; \beta \rightsquigarrow (\alpha \cup \gamma); \beta}$$



How to Retain Soundness

Structural Refactoring

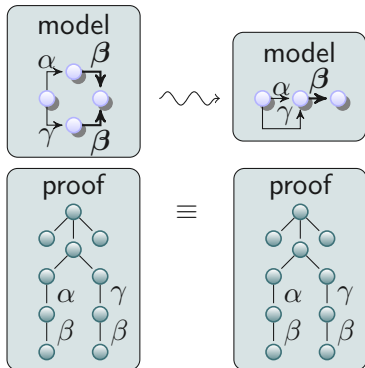
Always retains soundness



How to Retain Soundness

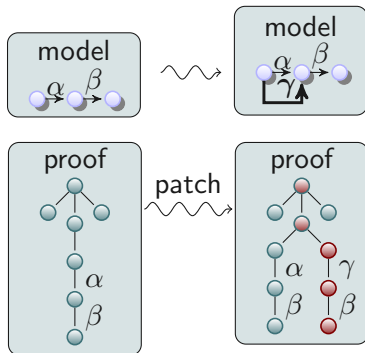
Structural Refactoring

Always retains soundness



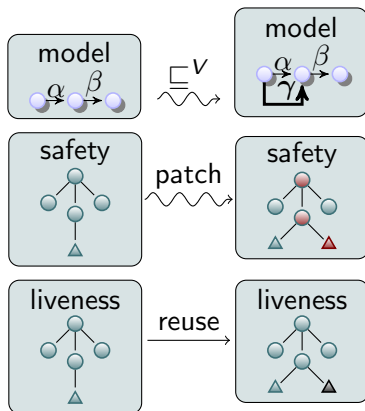
Behavioral Refactoring

Proof patch retains soundness



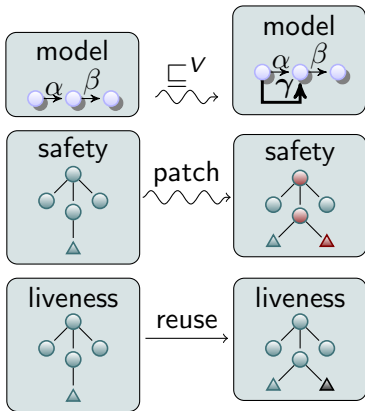
Patching Necessity by Correctness Property

Add Behavior

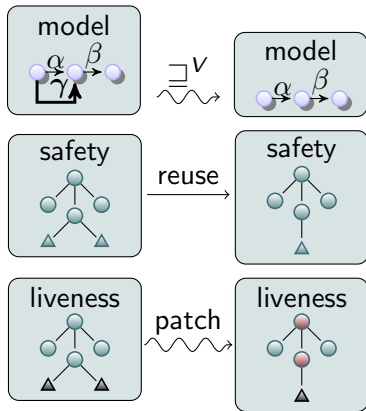


Patching Necessity by Correctness Property

Add Behavior



Remove Behavior



Patching Necessity by Correctness Property

Add Behavior

Remove Behavior

Projective Relational Refinement

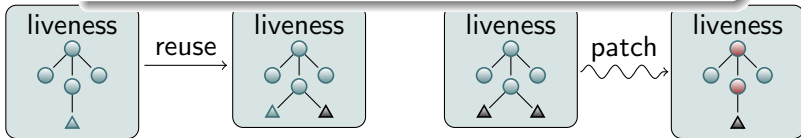
$$\alpha \sqsubseteq^V \gamma \text{ iff } \rho(\alpha)|_V \subseteq \rho(\gamma)|_V$$

$\rho(\alpha)$ reachability relation of α

α, γ hybrid systems models

$V \subseteq \Sigma$ relevant set of variables

$|_V$ projection of relations or states to the variables in V



Sound Refactoring Catalog

Structural Refactorings

- ▶ Extract Common Program
- ▶ Extract Continuous Dynamics
- ▶ Drop Implied Evolution Domain Constraint

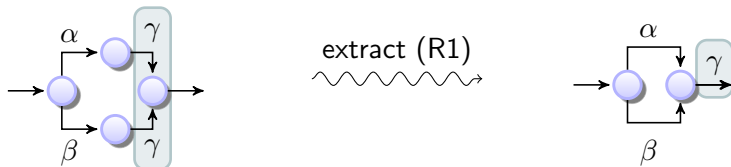
Behavioral Refactorings

- ▶ Introduce Control Path
- ▶ Introduce Complementary Continuous Dynamics
- ▶ Event- to Time-Triggered Architecture

Extract Common Program

Motivation Reduce model duplication

Mechanics (R1) $(\alpha; \gamma) \cup (\beta; \gamma) \rightsquigarrow (\alpha \cup \beta); \gamma$



Variation Inline program (R2) $(\alpha \cup \beta); \gamma \rightsquigarrow (\alpha; \gamma) \cup (\beta; \gamma)$

Proof patch

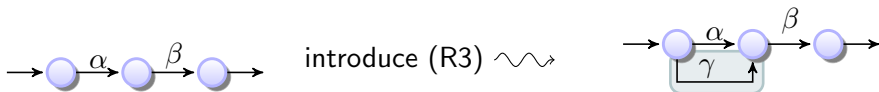
Safety None

Liveness None

Introduce Control Path

Motivation Add control decisions

Mechanics (R3) $\alpha; \beta \rightsquigarrow (\alpha \cup \gamma); \beta$



Variation Remove Control Path (R4) $(\alpha \cup \gamma); \beta \rightsquigarrow \alpha; \beta$

Proof patch

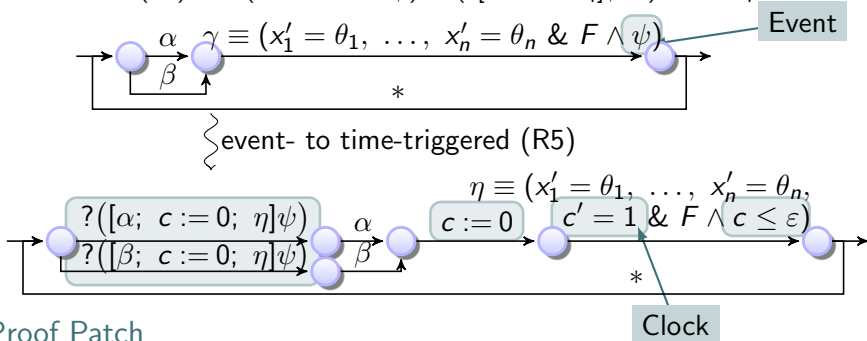
Safety Prove safety of the added branch

Liveness None

Event- to Time-Triggered Architecture

Motivation Derive a time-triggered controller

Mechanics (R5) $\alpha; (x' = \theta \ \& \ F \wedge \psi) \rightsquigarrow (?[\alpha; c := 0; \eta]\psi; \alpha); c := 0; \eta$



Proof Patch

Safety Composes several refactorings + prove safety of derived tests

Summary

Benefits of Proof-aware Refactorings

- ▶ easier to evolve correct systems
- ▶ easier to get simple systems correct
- ▶ still want to handle complex systems, but not pay the price of reverification
- ▶ **co-evolve model and proof**

Future Work

Refactoring Catalog

- ▶ Pull and merge tests
- ▶ Weaken/strengthen test
- ▶ Switch sequence
- ▶ Introduce computation delay
- ▶ Introduce uncertainty
- ▶ Introduce disturbance
- ▶ Change norm ($2/\infty$ norm)
- ▶ ...

Theory

- ▶ Liveness proof patches
- ▶ Distance measurement
- ▶ Refinement based on games

Implementation

- ▶ S φ nX and KeYmaera
- ▶ Background proving

Thank you!

Stefan Mitsch

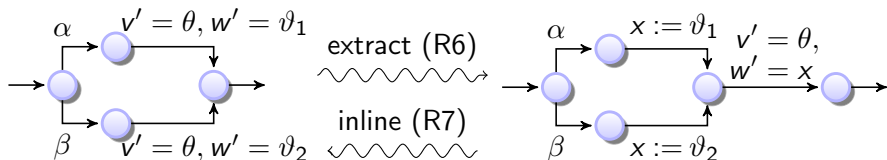
smitsch@cs.cmu.edu

<http://www.cs.cmu.edu/~smitsch>

Extract Continuous Dynamics

$$(R6) \quad \frac{\forall v \in V(\theta) \cup \bigcup_{i \in I} V(\vartheta_i). v \notin BV(\mathcal{D}(\theta)) \cup \bigcup_{i \in I} BV(\mathcal{D}(\vartheta_i))}{\bigcup_{i \in I} (\alpha_i; (v' = \theta, w' = \vartheta_i)) \rightsquigarrow (\bigcup_{i \in I} (\alpha_i; x := \vartheta_i)); (v' = \theta, w' = x)}$$

$$(R7) \quad (\bigcup_{i \in I} (\alpha_i; x := \vartheta_i)); (v' = \theta, w' = x) \rightsquigarrow \bigcup_{i \in I} (\alpha_i; (v' = \theta, w' = \vartheta_i))$$

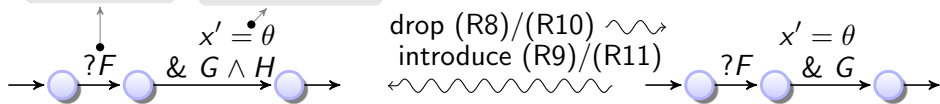


Proof obligations

None, because the original program and the refactored program are **observationally equivalent**.

Drop Implied Evolution Domain Constraint

$$\begin{array}{l}
 \text{(R8)} \quad \frac{F \rightarrow H \quad F \rightarrow [x' = \theta \& G]H}{?F; x' = \theta \& G \wedge H \rightsquigarrow ?F; x' = \theta \& G} \quad \text{(R9)} \quad \frac{?F; x' = \theta \& G}{?F; x' = \theta \& G \wedge H} \\
 \text{(R10)} \quad \frac{?F; x' = \theta \& G \wedge H}{?F; x' = \theta \& G} \quad \text{(R11)} \quad \frac{F \rightarrow H \quad F \rightarrow [x' = \theta \& G]H}{?F; x' = \theta \& G \rightsquigarrow ?F; x' = \theta \& G \wedge H}
 \end{array}$$

 $\models F \rightarrow H$ θ preserves H 

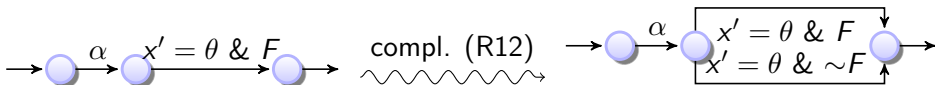
Proof obligations

Liveness None, because projective partial refinement, i. e.,
 $(?F; x' = \theta \& G \wedge H) \sqsubseteq_F^V (?F; x' = \theta \& G)$ holds.

Safety Show that H is a differential invariant

Introduce Complementary Continuous Dynamics

$$(R12) \quad \alpha; x' = \theta \& F \rightsquigarrow \alpha; (x' = \theta \& F \cup x' = \theta \& \sim F)$$

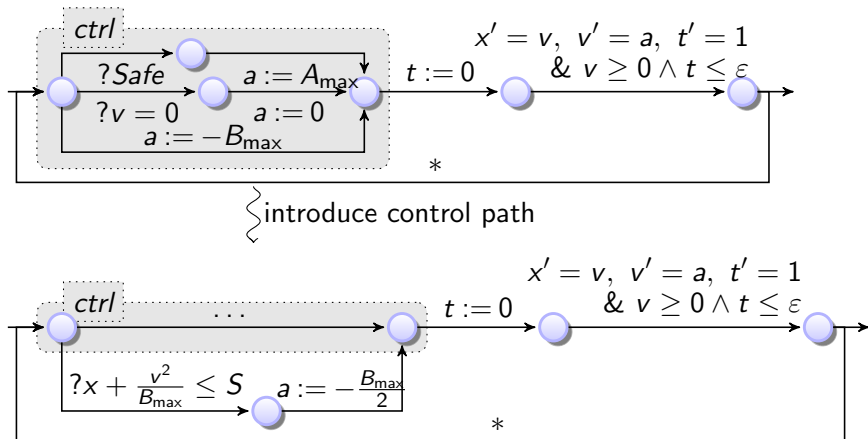


Proof obligations

Liveness None

Safety Show that the controller with subsequent complementary dynamics only reaches states that are already reachable with the original dynamics, i. e., show $(\langle \alpha; x' = \theta \& \sim F \rangle \Upsilon_V) \rightarrow (\langle \alpha; x' = \theta \& F \rangle \Upsilon_V)$

Example: Introduce Moderate Braking

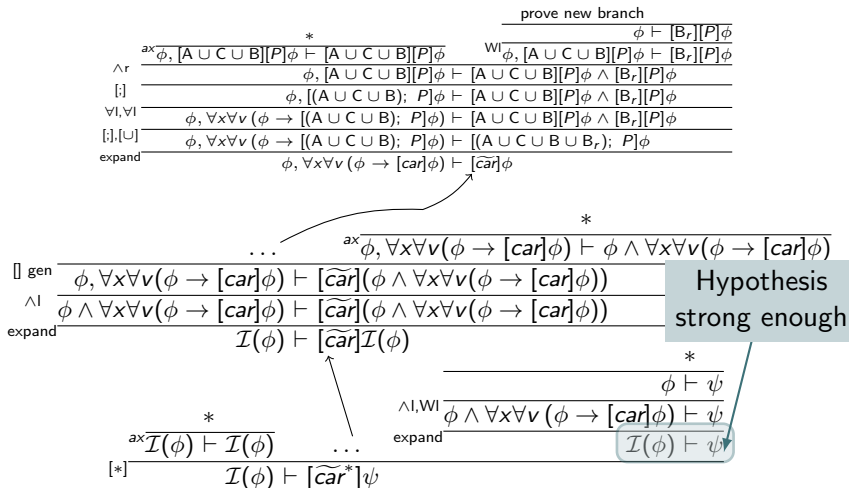


Introduce Moderate Braking: Auxiliary Safety Proof

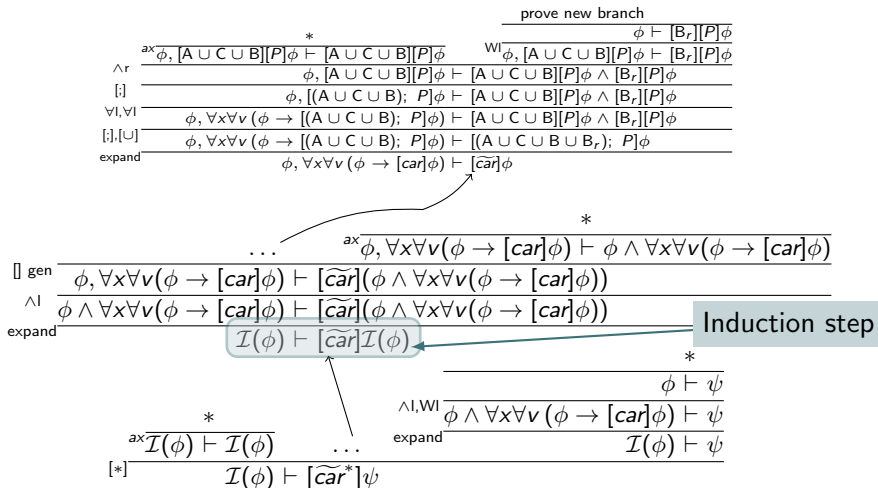
$$\begin{array}{c}
 \text{prove new branch} \\
 \frac{}{\phi \vdash [B_r][P]\phi} \\
 \hline
 \text{WI} \frac{\phi, [A \cup C \cup B][P]\phi \vdash [B_r][P]\phi}{\phi, [A \cup C \cup B][P]\phi \vdash [A \cup C \cup B][P]\phi \wedge [B_r][P]\phi} \\
 \hline
 \text{ax} \frac{\phi, [A \cup C \cup B][P]\phi \vdash [A \cup C \cup B][P]\phi}{\phi, [A \cup C \cup B][P]\phi \vdash [A \cup C \cup B][P]\phi \wedge [B_r][P]\phi} \\
 \hline
 \text{[i]} \frac{}{\phi, [(A \cup C \cup B); P]\phi \vdash [A \cup C \cup B][P]\phi \wedge [B_r][P]\phi} \\
 \hline
 \text{VI, VI} \frac{}{\phi, \forall x \forall v (\phi \rightarrow [(A \cup C \cup B); P]\phi) \vdash [A \cup C \cup B][P]\phi \wedge [B_r][P]\phi} \\
 \hline
 \text{[i], [U]} \frac{}{\phi, \forall x \forall v (\phi \rightarrow [(A \cup C \cup B); P]\phi) \vdash [(A \cup C \cup B)_r]; P]\phi} \\
 \hline
 \text{expand} \frac{}{\phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}]\phi} \\
 \hline
 \dots \frac{\text{ax} \phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash \phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi)}{\phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}](\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi))} \\
 \hline
 \text{[gen]} \frac{}{\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}](\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi))} \\
 \hline
 \text{expand} \frac{}{I(\phi) \vdash [\widetilde{car}]I(\phi)} \\
 \hline
 \dots \\
 \hline
 \text{ax} \frac{}{I(\phi) \vdash I(\phi)} \\
 \hline
 \text{[*]} \frac{}{I(\phi) \vdash [\widetilde{car}^*]\psi} \\
 \hline
 \text{expand} \frac{\text{[i], WI} \frac{}{\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi) \vdash \psi}}{I(\phi) \vdash \psi} \\
 \hline
 \text{[*]} \frac{}{\phi \vdash \psi}
 \end{array}$$

Base case

Introduce Moderate Braking: Auxiliary Safety Proof

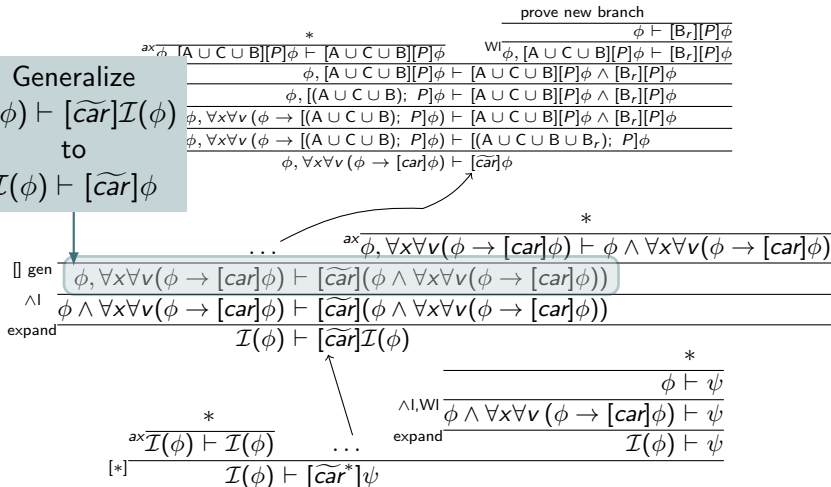


Introduce Moderate Braking: Auxiliary Safety Proof



Introduce Moderate Braking: Auxiliary Safety Proof

Generalize
 $I(\phi) \vdash [\widetilde{car}]I(\phi)$
 to
 $I(\phi) \vdash [\widetilde{car}]\phi$



Introduce Moderate Braking: Auxiliary Safety Proof

$$\begin{array}{c}
 \text{ax} \\
 \wedge r \\
 [i] \\
 \forall I, \forall I \\
 [i].[U] \\
 \text{expand}
 \end{array}
 \frac{
 \begin{array}{c}
 * \\
 \phi, [AUCUB][P]\phi \vdash [AUCUB][P]\phi \\
 \phi, [AUCUB][P]\phi \vdash [AUCUB][P]\phi \wedge [B_r][P]\phi \\
 \phi, [(AUCUB); P]\phi \vdash [AUCUB][P]\phi \wedge [B_r][P]\phi \\
 \phi, \forall x \forall v (\phi \rightarrow [(AUCUB); P]\phi) \vdash [AUCUB][P]\phi \wedge [B_r][P]\phi \\
 \phi, \forall x \forall v (\phi \rightarrow [(AUCUB); P]\phi) \vdash [(AUCUB \cup B_r); P]\phi
 \end{array}
 }{
 \phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}]\phi
 }$$

Standard rules
for programs \cup and $;$

$$\begin{array}{c}
 \text{expand} \\
 \text{expand} \\
 \text{expand} \\
 [*]
 \end{array}
 \frac{
 \begin{array}{c}
 \dots \\
 \text{ax} \\
 \phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash \phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi) \\
 \vdash [car]\phi \vdash [\widetilde{car}](\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi)) \\
 \vdash [car]\phi \vdash [\widetilde{car}](\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi))
 \end{array}
 }{
 \mathcal{I}(\phi) \vdash [\widetilde{car}]\mathcal{I}(\phi)
 }$$

$$\frac{
 \begin{array}{c}
 * \\
 \phi \vdash \psi \\
 \wedge I, \text{WI} \\
 \phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi) \vdash \psi \\
 \text{expand} \\
 \mathcal{I}(\phi) \vdash \psi
 \end{array}
 }{
 \mathcal{I}(\phi) \vdash [\widetilde{car}^*]\psi
 }$$

Introduce Moderate Braking: Auxiliary Safety Proof

prove new branch

$$\begin{array}{c}
 \text{ax} \frac{\phi, [AUCUB][P]\phi \vdash [AUCUB][P]\phi}{\phi, [AUCUB][P]\phi \vdash [B_r][P]\phi} \\
 \wedge r \frac{\phi, [AUCUB][P]\phi \vdash [AUCUB][P]\phi \wedge [B_r][P]\phi}{\phi, [AUCUB][P]\phi \vdash [AUCUB][P]\phi} \\
 [i] \frac{\phi, [(AUCUB); P]\phi \vdash [AUCUB][P]\phi \wedge [B_r][P]\phi}{\phi, [(AUCUB); P]\phi \vdash [(AUCUB) \cup B_r]; P]\phi} \\
 \text{Exploit } \phi \rightarrow [car]\phi \text{ already proven} \quad \frac{(\phi \rightarrow [(AUCUB); P]\phi) \vdash [AUCUB][P]\phi \wedge [B_r][P]\phi}{(\phi \rightarrow [(AUCUB); P]\phi) \vdash [(AUCUB) \cup B_r]; P]\phi} \\
 \phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}]\phi
 \end{array}$$

$$\begin{array}{c}
 \dots \quad \text{ax} \frac{\phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash \phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi)}{\phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}](\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi))} \\
 [i] \text{ gen} \frac{\phi, \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}](\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi))}{\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}](\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi))} \\
 \wedge l \\
 \text{expand} \frac{\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi) \vdash [\widetilde{car}](\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi))}{I(\phi) \vdash [\widetilde{car}]I(\phi)} \\
 \text{ax} \frac{I(\phi) \vdash I(\phi)}{I(\phi) \vdash [\widetilde{car}]^* I(\phi)} \\
 \dots \quad \text{ax} \frac{\phi \vdash \psi}{\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi) \vdash \psi} \\
 \wedge l, WI \\
 \text{expand} \frac{\phi \wedge \forall x \forall v (\phi \rightarrow [car]\phi) \vdash \psi}{I(\phi) \vdash \psi} \\
 [*] \frac{I(\phi) \vdash I(\phi) \quad \dots \quad I(\phi) \vdash \psi}{I(\phi) \vdash [\widetilde{car}]^* \psi}
 \end{array}$$

Introduce Moderate Braking: Auxiliary Safety Proof

$$\begin{array}{c}
 \text{ax} \\
 \hline
 \phi, [\text{AUCUB}][P]\phi \vdash [\text{AUCUB}][P]\phi \\
 \wedge r \\
 \hline
 \phi, [\text{AUCUB}][P]\phi \vdash [\text{AUCUB}][P]\phi \wedge [B_r][P]\phi \\
 [i] \\
 \hline
 \phi, [(AUCUB); P]\phi \vdash [\text{AUCUB}][P]\phi \wedge [B_r][P]\phi \\
 \forall i, \forall i \\
 \hline
 \phi, \forall x \forall v (\phi \rightarrow [(AUCUB); P]\phi) \vdash [\text{AUCUB}][P]\phi \wedge [B_r][P]\phi \\
 [i].[\cup] \\
 \hline
 \phi, \forall x \forall v (\phi \rightarrow [(AUCUB); P]\phi) \vdash [(AUCUB \cup B_r); P]\phi \\
 \text{expand} \\
 \hline
 \phi, \forall x \forall v (\phi \rightarrow [\text{car}]\phi) \vdash [\widetilde{\text{car}}]\phi
 \end{array}$$

Prove new branch
prove new branch

$$\begin{array}{c}
 \dots \\
 \text{ax} \\
 \hline
 \phi, \forall x \forall v (\phi \rightarrow [\text{car}]\phi) \vdash \phi \wedge \forall x \forall v (\phi \rightarrow [\text{car}]\phi) \\
 [i] \text{ gen} \\
 \hline
 \phi, \forall x \forall v (\phi \rightarrow [\text{car}]\phi) \vdash [\widetilde{\text{car}}](\phi \wedge \forall x \forall v (\phi \rightarrow [\text{car}]\phi)) \\
 \wedge i \\
 \hline
 \phi \wedge \forall x \forall v (\phi \rightarrow [\text{car}]\phi) \vdash [\widetilde{\text{car}}](\phi \wedge \forall x \forall v (\phi \rightarrow [\text{car}]\phi)) \\
 \text{expand} \\
 \hline
 \mathcal{I}(\phi) \vdash [\widetilde{\text{car}}]\mathcal{I}(\phi)
 \end{array}$$

$$\begin{array}{c}
 \dots \\
 \text{ax} \\
 \hline
 \mathcal{I}(\phi) \vdash \mathcal{I}(\phi) \\
 [*] \\
 \hline
 \mathcal{I}(\phi) \vdash [\widetilde{\text{car}}^*]\psi
 \end{array}$$

$$\begin{array}{c}
 \dots \\
 \wedge i, \text{WI} \\
 \hline
 \phi \wedge \forall x \forall v (\phi \rightarrow [\text{car}]\phi) \vdash \psi \\
 \text{expand} \\
 \hline
 \mathcal{I}(\phi) \vdash \psi
 \end{array}$$