Computational Modeling and Analysis For Complex Systems

NSF Expedition in Computing

CMACS

Embedded Systems Challenge Problem

Bruce H. Krogh

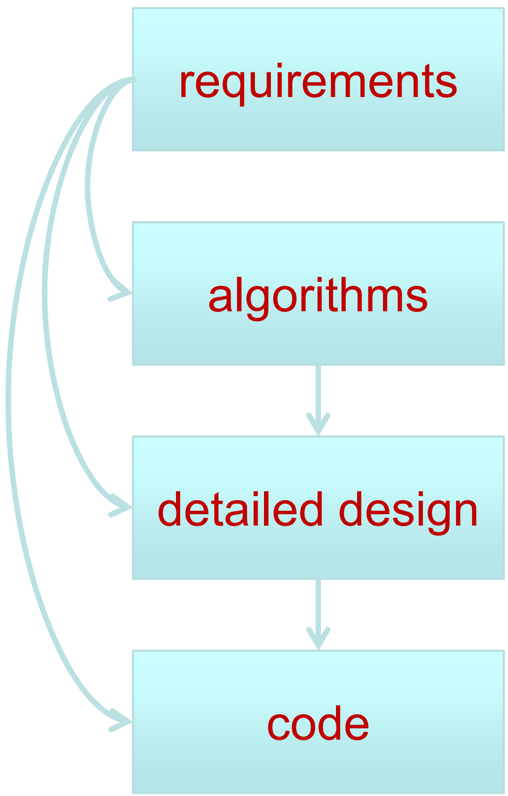
Carnegie Mellon University

2nd Year Review Meeting, Carnegie Mellon University

November 3, 2011

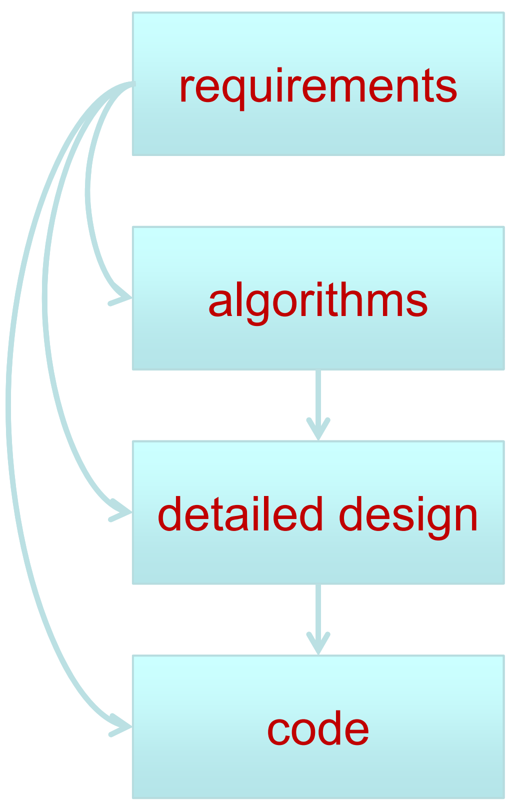
Embedded Control Systems

Design Flow



Embedded Control Systems

Design Flow



Challenges

- consistency
- correctness
- complexity
- coverage
- run-time correctness

Embedded Control Systems

Design Flow

- requirements
- algorithms
- detailed design
- code

Challenges

- consistency
- correctness
- complexity
- coverage
- run-time correctness

CMACS Research

- requirements reconstruction
- analysis of hybrid systems
  - theorem proving
  - compositionality
  - reachability
  - statistical model checking
- code verification
  - abstract interpretation
  - analysis-aware design
  - run-time verification

CMACS Embedded Systems Team

Scott Smolka
Stony Brook

Radu Grosu
Stony Brook

Rance Cleaveland
U Maryland

Andre Platzer
CMU

Klaus Havelund
NASA

Gerard Holzmann
NASA

Patrick Cousot
NYU

Impossible Without An Expeditions Project

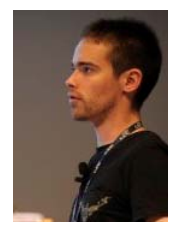
Steve Marcus
U Maryland

Bruce Krogh
CMU

Matthias Althoff
CMU

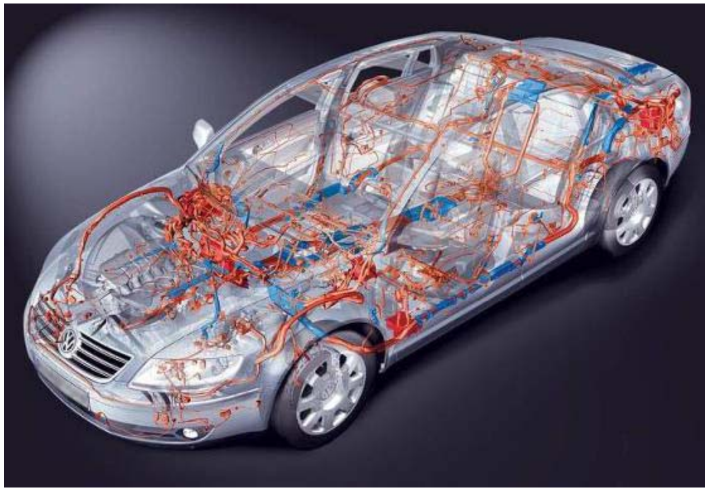
Ed Clarke
CMU

Paolo Zuliani
CMU



Colas Le Guernic
NYU

Requirements Reconstruction



Challenge Problem

Outdated requirements documents for automotive embedded systems

- due to system evolution
- limits ability to apply formal verification in future development

Approach

Use test data to re-create high-level descriptions of system behavior.

- apply machine learning: association-rule mining
- identify possible invariants satisfied by the system.

Technical Challenges

- quickly detecting and eliminating false invariants
- ensuring that correct invariants are indeed detected

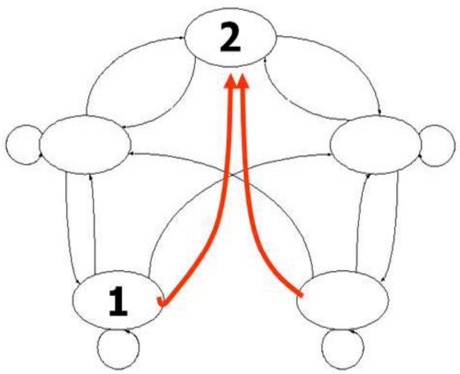
Research Team: *UMD*: Chris Ackermann, Rance Cleaveland, Sam Huang; *Fraunhofer*: Arnab Ray; *Robert Bosch*: Beth Latronico, Charles Shelton

Requirements Reconstruction – cont'd.

Major Advances

Applied *instrumentation-based verification* (model checking technique)

- identifies false invariants
- ensures test data satisfies coverage constraints
- ensures coverage of proposed invariants



Results to date

For a large production automotive control subsystem

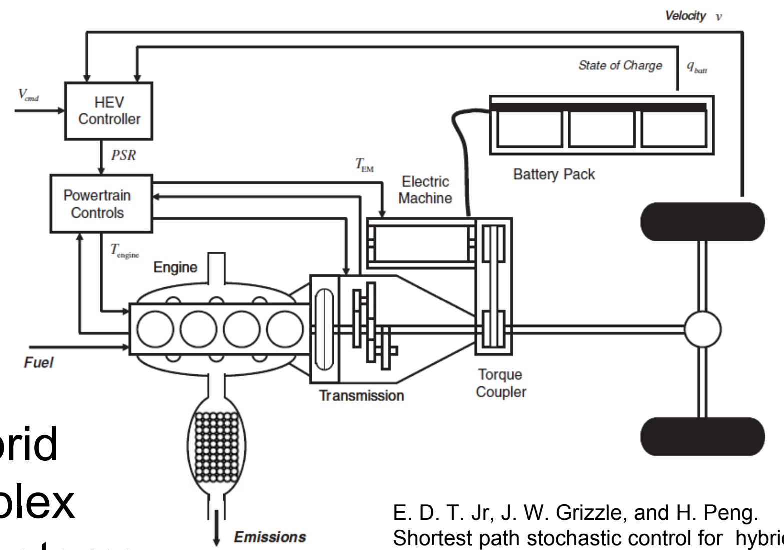
- 41 of 42 invariants recovered for one module
- found 2 invariants not stated in the requirements
- only 1 incorrectly declared invariant not detected.

Current work

- genetic algorithms for inferring temporal properties
- larger pilot study involving 10 automotive control subsystems

C. Ackermann, R. Cleaveland, S. Huang, A. Ray, C. Shelton, and E. Latronico. Automatic requirement extraction from test cases. *First International Conference on Runtime Verification*, LNCS vol 6418, pp. 1-15, Malta, Nov 2010.

Composition of Hybrid Systems



E. D. T. Jr, J. W. Grizzle, and H. Peng. Shortest path stochastic control for hybrid electric vehicles. Int. J. Robust Nonlinear Control, 18(14):1409–1429, December 2007.

Challenge Problem

How can component-based models of automotive embedded control systems be composed and analyzed in a rigorous way based on formal methods?

Approach

Generalize ideas of Process Algebra to hybrid (dynamical) systems to analyze/verify complex systems in terms of simpler, reusable subsystems

Technical Challenges

- theories of composition has received relatively little attention for hybrid systems
- need new mathematical frameworks supporting the rich array of mechanisms used to build composite embedded systems in practice

Research Team: *UMD*: Rance Cleaveland (CS), Steve Marcus (ECE), Peter Fontana (CS), James Ferlez (ECE)

Composition of Hybrid Systems – cont'd.

Major Advances

New mathematical model of system behavior that generalizes methods of Process Algebra to hybrid systems

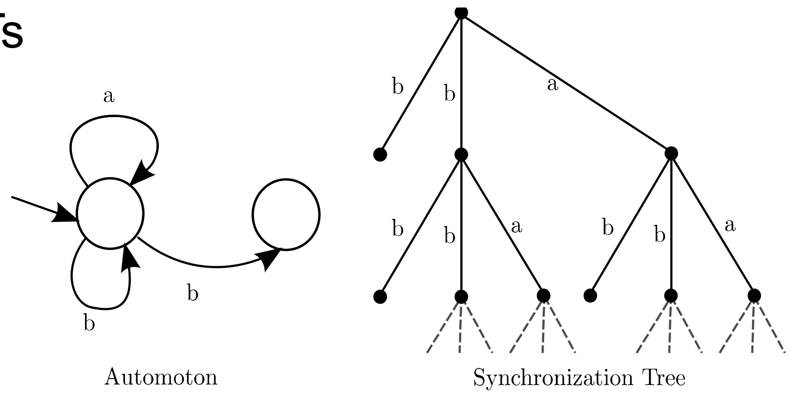
- asynchronous parallel composition
- synthesis of ideas from computer science (process algebra) and control (the behavioral methodology of Willems, van der Schaft, etc.)

Results to date

- generalized synchronization trees (GSTs) for hybrid systems
- preliminary algebraic properties of GSTs
- paper in progress

Current work

- further algebraic properties of GSTs
- types of generalized composition
- control law synthesis



Design Verification

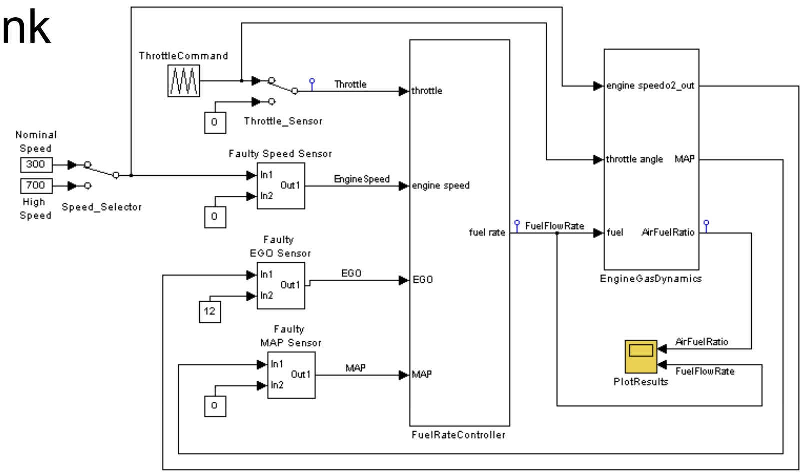
Challenge Problem

Verification of stochastic Stateflow/Simulink models

E.g.

$\Phi = \neg F100\ G1(\text{FuelFlowRate} = 0)$

Prob (Sys $\models \Phi$) = .9779 $\pm$ .01



Approach

Prob (Φ)?

simulation + model checking + statistical estimation

Prob (Φ) > θ?

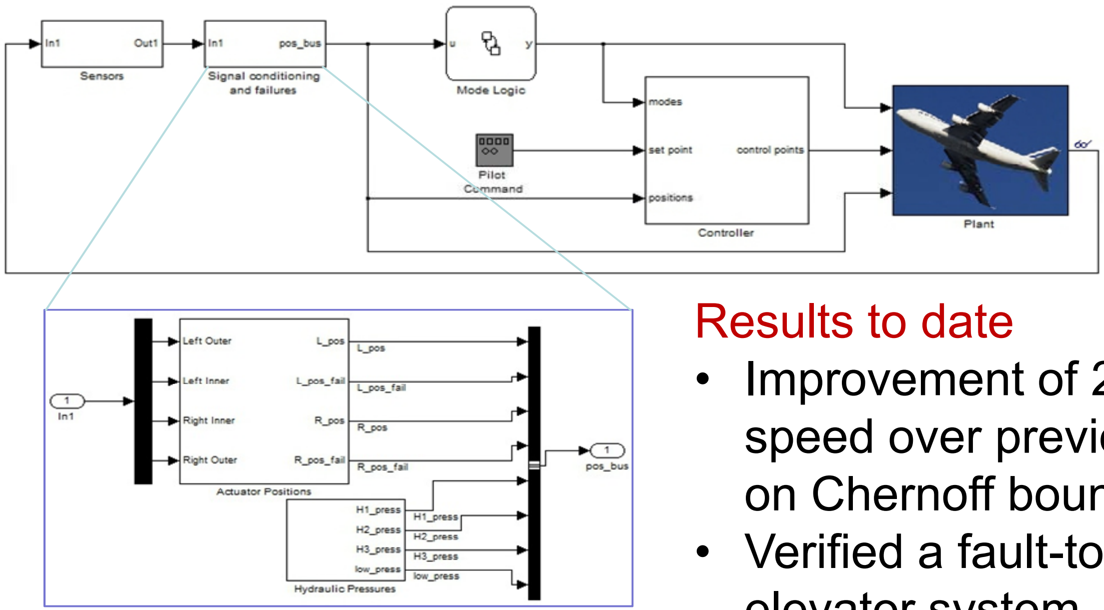
simulation + model checking + statistical hypothesis testing

Research Team: *CMU*: Ed Clarke, Paolo Zuliani, Andre Plazer; *TU Dresden: Christel Baier*

Design Verification – cont'd.

Major Advances

- Efficient Bayesian estimation and hypothesis testing techniques
- Importance Sampling (IS) and Cross-Entropy (CE) with statistical MC



Results to date

- Improvement of 2-3 orders of magnitude in speed over previous methods (techniques based on Chernoff bound)
- Verified a fault-tolerant controller for an aircraft elevator system

P. Zuliani, A. Platzer, E. M. Clarke. Bayesian Statistical Model Checking with Application to Stateflow/Simulink Verification. In HSCC 2010, pages 243-252.

E. M. Clarke and P. Zuliani. Statistical Model Checking for Cyber-Physical Systems. In ATVA 2011, LNCS 6996, pages 1-12.

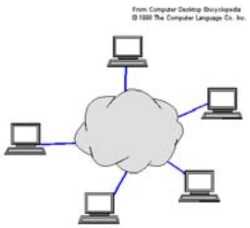
P. Zuliani, C. Baier, E.M. Clarke. Rare-Event Verification for Stochastic Hybrid Systems. Submitted

Embedded Software Verification

Challenge Problems

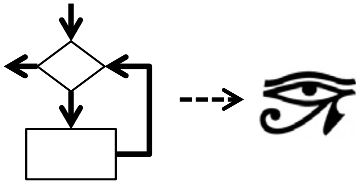
Scale model checking algorithms to handle unmodified industrial size software as used for safety critical embedded systems (aerospace/automotive/medical)

Improve runtime verification techniques by creating more expressive specification languages with efficient monitoring algorithms, and designing specification learning and trace visualization techniques.



Approach

- develop new analysis-aware software design methods
- develop new context aware verification methods
- target massive use of parallelism



Research Team: *JPL/CalTech*: Klaus Havelund,Gerard Holzmann, Mihai Florian (Caltech CS, grad student), Ed Gamble

NASA/JPL Laboratory for Reliable Software

Embedded Software Verification– cont'd.

Current work

- direct verification of real-time priority-based scheduling algorithms
- new multi-core and cloud-based model checking algorithms
  - performance is expected to scale linearly with the number of available processing elements (cores, CPUs, and/or GPU engines),
  - potential for orders of magnitude improvements on large compute farms
- new efficient rule-based methods for runtime verification based on pattern matching

M. Florian. A Framework for Systematic Testing of Multi-threaded Applications, Proc. 17th IEEE Pacific Rim Int. Symposium on Dependable Computing (PRDC 2011).

M. McKelvin, and G.J. Holzmann, Model checking multitask applications for OSEK compliant real-time operating systems, Proc. 17th IEEE Pacific Rim Int. Symposium on Dependable Computing (PRDC 2011), Pasadena, CA, Dec. 12-14, 2011.

G.J. Holzmann, R. Joshi, and A. Groce. Swarm verification techniques. IEEE Trans. on Software Engineering, accepted for publication, 2011.

S. D. Stoller, E. Bartocci, J. Seyster, R. Grosu, K. Havelund, S. A. Smolka, and E. Zadok. Runtime Verification with State Estimation. The 2nd International Conference on Runtime Verification (RV 2011). San Francisco, California, USA, September 27-30, 2011. LNCS (won best paper award).

.

Advances in aerospace applications

- The paper

  Julien Bertrane, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, & Xavier Rival.
  Static Analysis and Verification of Aerospace Software by Abstract Interpretation. In *AIAA Infotech@Aerospace 2010*, Atlanta, Georgia. American Institute of Aeronautics and Astronautics, 20—22 April 2010. © AIAA.

  received the AIAA intelligent systems best paper award 2010

- All control/command software of a European aircraft manufacturer now mandatorily verified by abstract-interpretation based static analysis (in conformance with DO-178-C)

- Progress on the static verification of parallel processes

Advances in abstract interpretation

Significant advances on

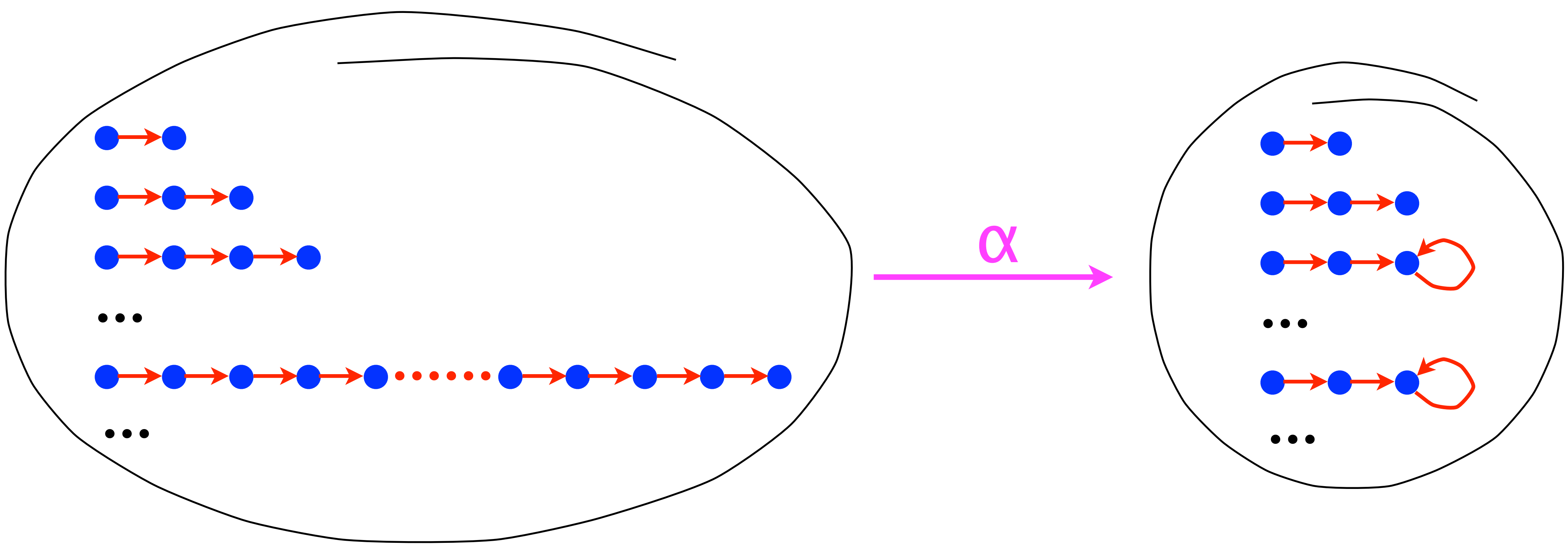
- Under-approximation
- Combination of algebraic and logical abstractions
- Probabilistic abstraction
- Termination/liveness

have been done for infinite state systems.

Difficulty of the problems

- Abstraction to finite / bounded executions is impossible (unsound, ineffective, ...)

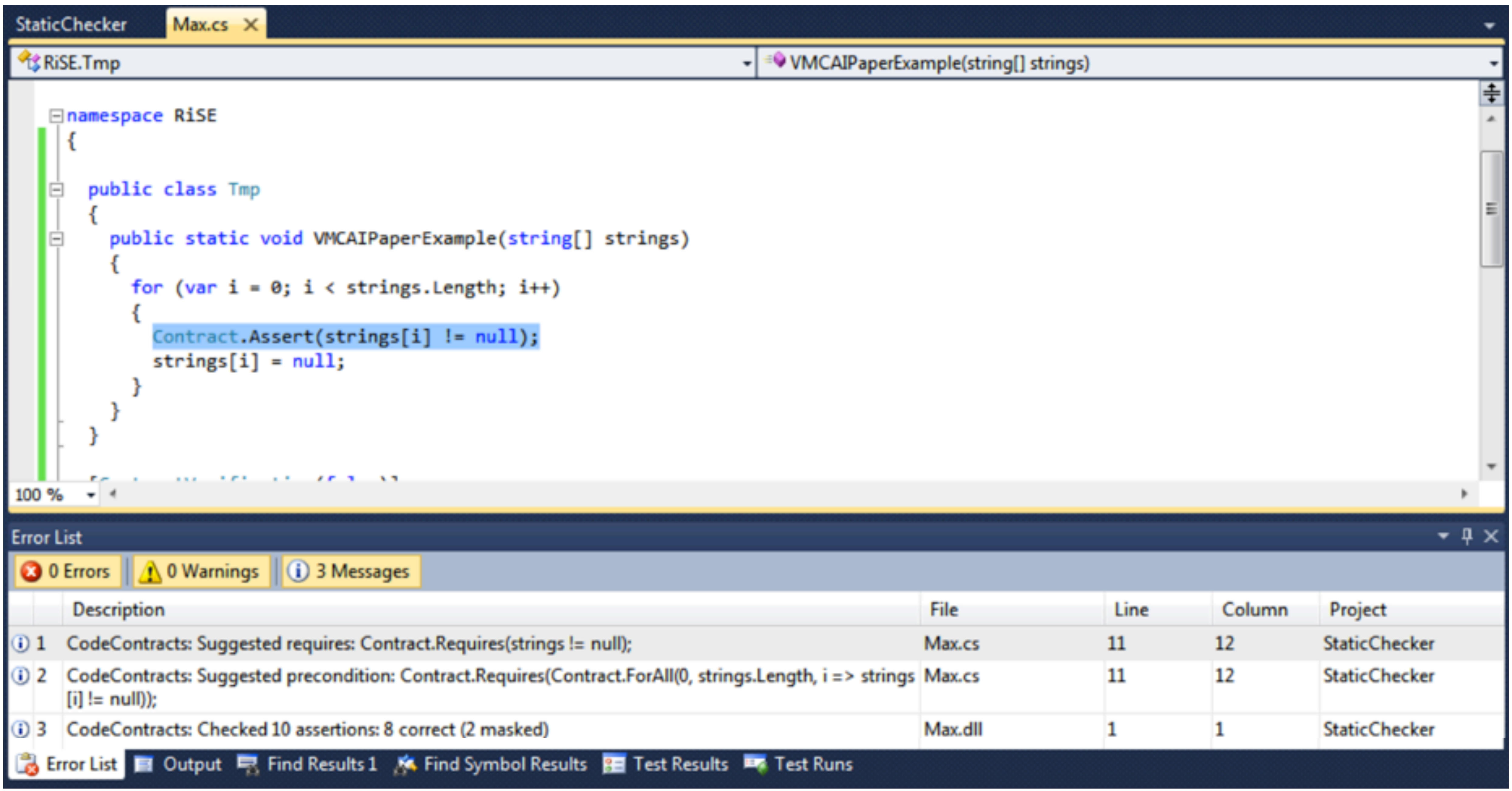
  Example: [non]-termination of *unbounded* programs



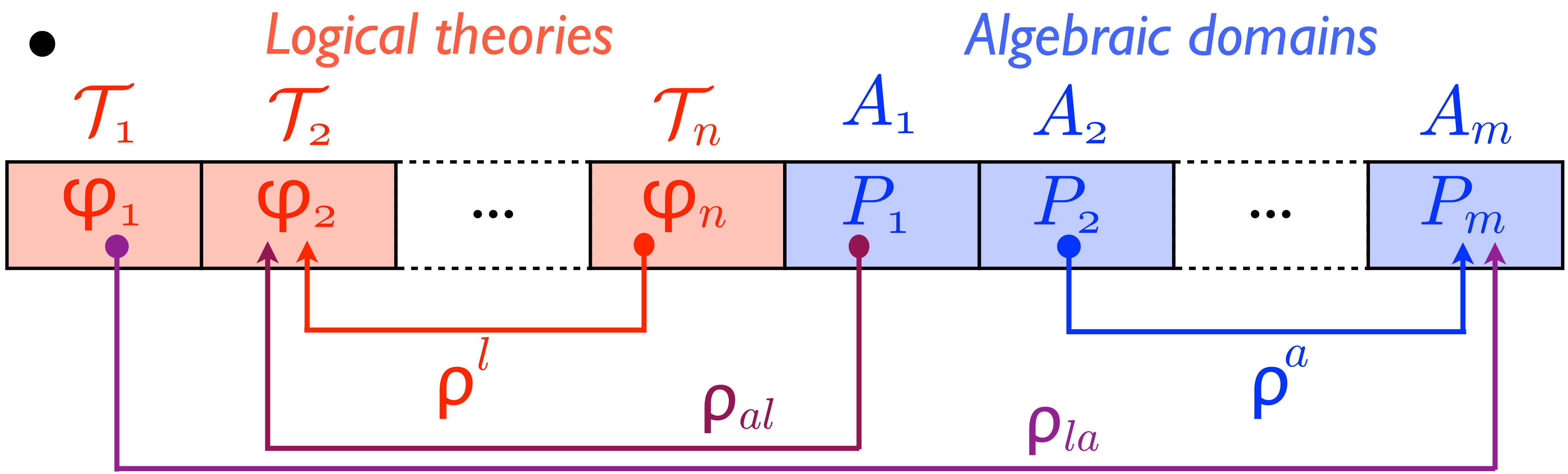
- Abstraction must be infinite, which is extremely difficult

Under-approximation

- Previously: explore finite parts of a finite subset of executions
- New: algebraic approach to handle infinitely many infinite executions
- Example: pre-conditions ensuring the *presence* of errors



Combining algebraic & logical abstractions

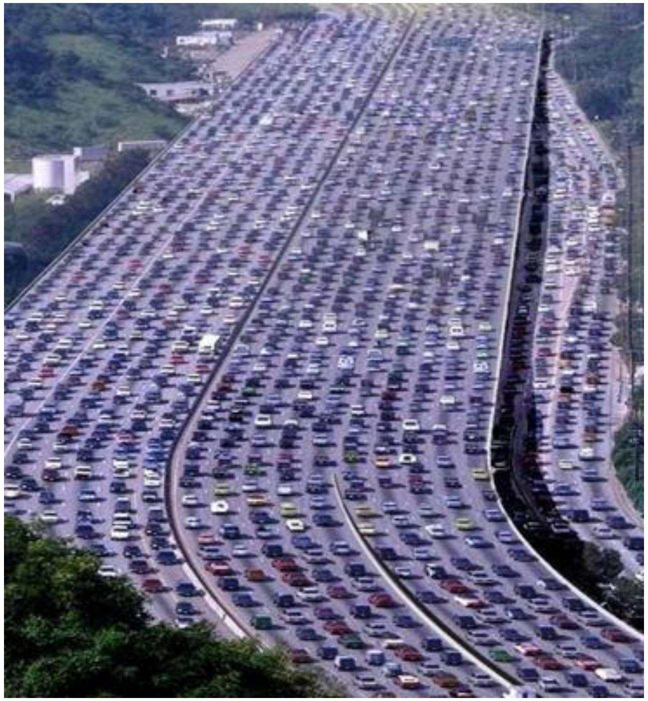
- A *new understanding* of the Nelson-Oppen procedure to combine logical theories in SMT solvers/provers as an algebraic reduced product
- 

- When checking satisfiability of $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$, the Nelson-Oppen procedure generates (dis)-equalities that can be propagated by ρ_{la} to reduce the P_i, $i=1,\ldots,m$
- $\alpha_i(\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n)$ can be propagated by ρ_{la} to reduce the P_i, $i=1,\ldots,m$
- The purification to theory $\mathcal{T}_i$ of $\gamma_i(P_i)$ can be propagated to φ_i by ρ_{al} in order to reduce it to $\varphi_i \wedge \gamma_i(P_i)$ (in $\mathcal{T}_i$)

Termination

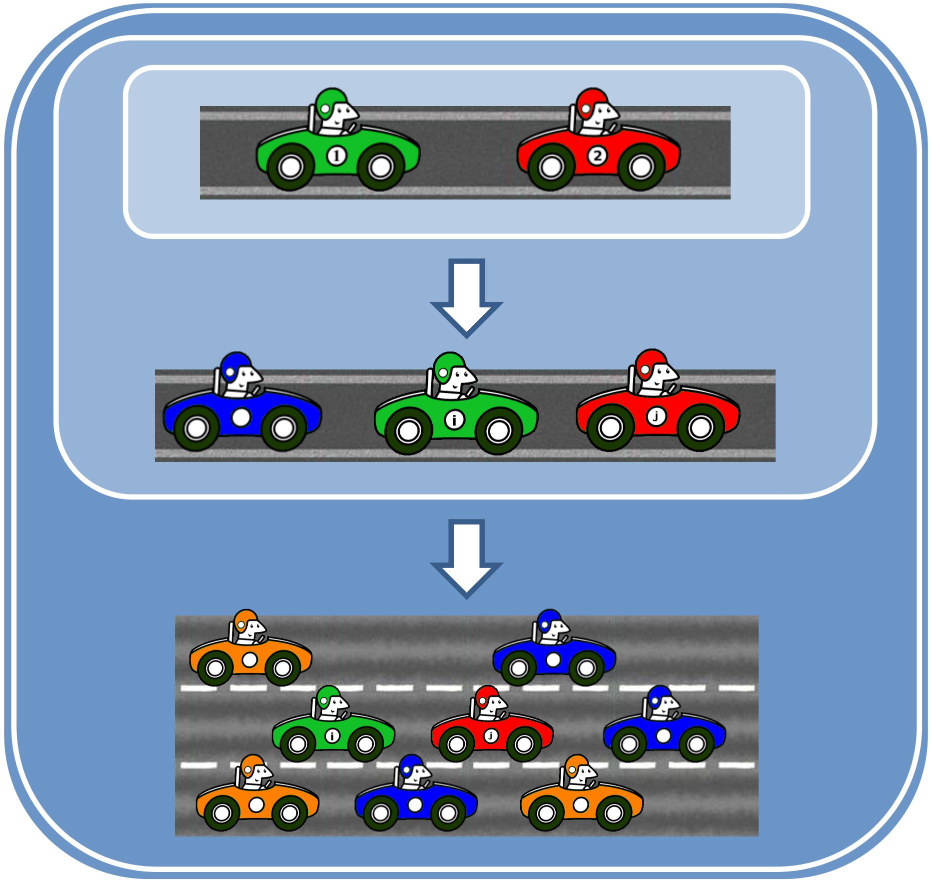
- Previously: recent progress on automatic proof of termination for *small, simple and pure programs* (no abstraction needed)
- Challenge: scale automatic program termination methods to large, *complex, and realistic programs* by integrating *abstraction*
- New advances:
  - Trace segments as a new basis for inductively formulating program properties
  - Fixpoint definition of a collecting semantics for termination/liveness
  - Systematic ways for constructing termination proofs, by construction of abstract fixpoints (e.g. variant functions)
  - Includes weak fairness

Distributed and Compositional Hybrid Systems



Hierarchical and Compositional Verification

Hierarchical Modularity



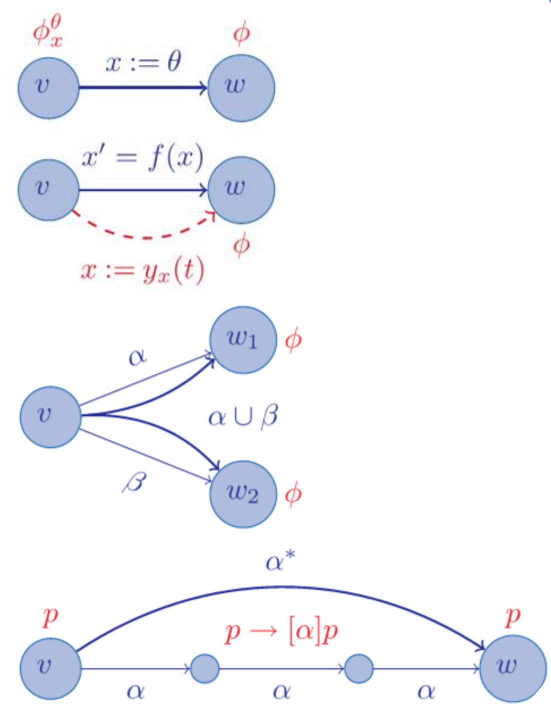
Decompositions

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

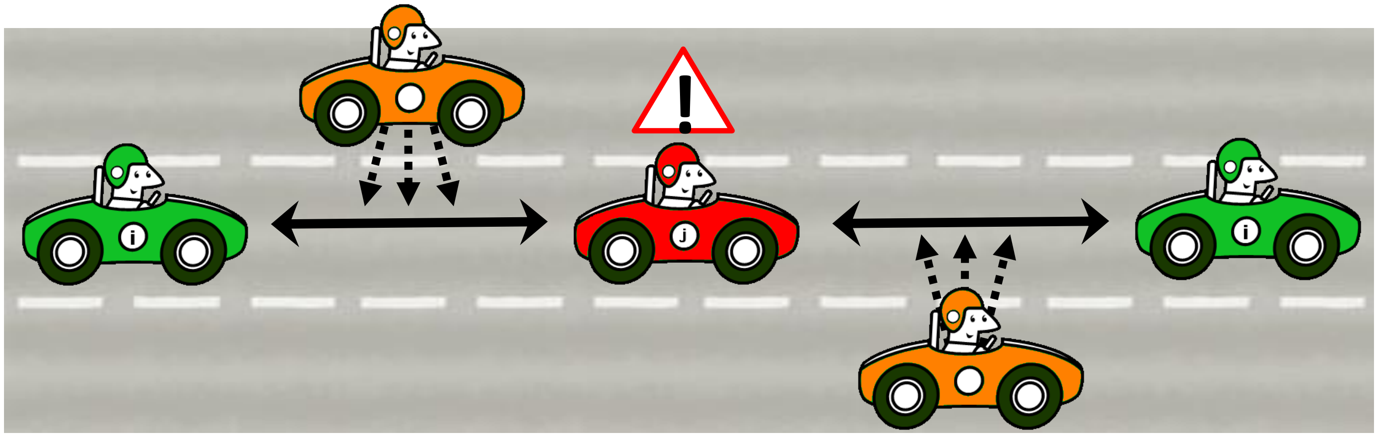
$$\frac{\exists t \geq 0\, \langle x := y_x(t) \rangle \phi}{\langle x' = f(x) \rangle \phi}$$

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

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



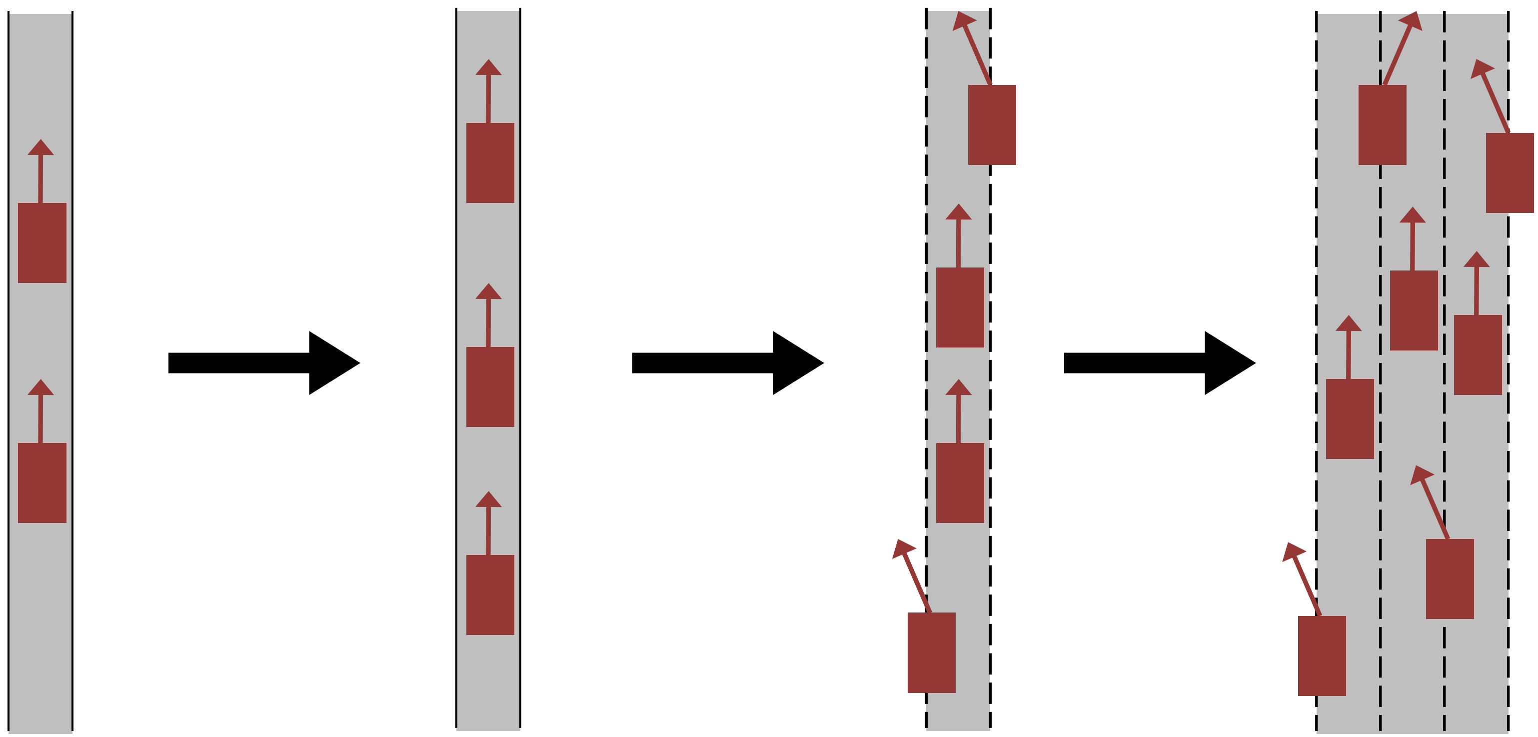
How Can We Prove Complex Highways?



Sensor limits on actual cars are always local.
Sometimes a maneuver may look safe locally…
But is a terrible idea when implemented globally because of unsafe emergent behavior.

Car Control Proof Sketch

Local Lane Control → Global Lane Control → Local Highway Control → Global Highway Control



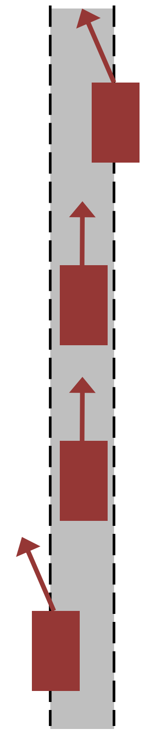
Local Lane Control	Global Lane Control	Local Highway Control	Global Highway Control
2 vehicles 1 lane no lane change	n vehicles 1 lane no lane change	n vehicles 1 lane lane changes	n vehicles m lanes lane changes

Car Control: Local Highway Control

Verified:

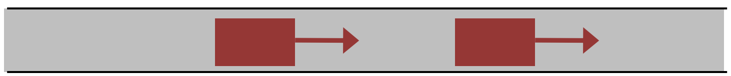
$$\forall i : C(i \ll L(i)) \;\rightarrow\; [\mathtt{lhc}]\forall i : C(i \ll L^*(i))$$

$$
\begin{aligned}
\mathtt{lhc} &\equiv (delete^*; create^*; ctrl^n; dyn^n)^* \\
create &\equiv n := new;\; ?((F(n) \ll n) \land (n \ll L(n))) \\
(n := new) &\equiv n := *;\; ?(\mathrm{E}(n) = 0);\; \mathrm{E}(n) := 1 \\
(F(n) \ll n) &\equiv \forall j : C\,(L(j) = n \rightarrow (j \ll n)) \\
delete &\equiv n := *;\; ?(\mathrm{E}(n) = 1);\; \mathrm{E}(n) := 0 \\
ctrl^n &\equiv \forall i : C\,(ctrl(i)) \\
ctrl(i) &\equiv (a(i) := *; ?(-B \le a(i) \le -b)) \\
&\quad\;\cup\; (?\mathbf{Safe}_\varepsilon(i);\; a(i) := *;\; ?(-B \le a(i) \le A)) \\
\mathbf{Safe}_\varepsilon(i) &\equiv x(i) + \frac{v(i)^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v(i)\right) < x(L(i)) + \frac{v(L(i))^2}{2B} \\
dyn^n &\equiv (t := 0;\; \forall i : C\,(dyn(i)), t' = 1, t \le \varepsilon) \\
dyn(i) &\equiv x'(i) = v(i), v'(i) = a(i), v(i) \ge 0
\end{aligned}
$$

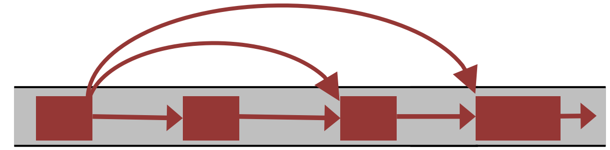


Initial Conditions → [Model] Requirements

Proof: Local Highway Control

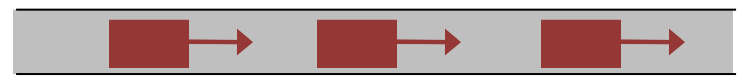


$$\forall i\; x(i) \ll x\big(L(i)\big) \rightarrow [glc]\; \forall i\; x(i) \ll x(L(i))$$



$$\forall i\; x(i) \ll x\big(L(i)\big) \rightarrow\; \forall i\; x(i) \ll x(L^*(i))$$

(cut)



$$\forall i\; x(i) \ll L\big(x(i)\big) \rightarrow [glc]\forall i\; x(i) \ll L^*(x(i))$$

Transitivity

$$\forall i\; x(i) \ll L\big(x(i)\big) \rightarrow [create^*]\forall i\; x(i) \ll L^*(x(i))$$

Transitivity

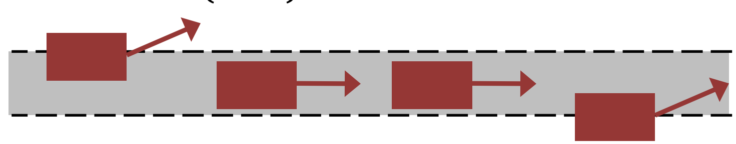
$$\forall i\; x(i) \ll L\big(x(i)\big) \rightarrow [delete^*]\forall i\; x(i) \ll L^*(x(i))$$

([] split)

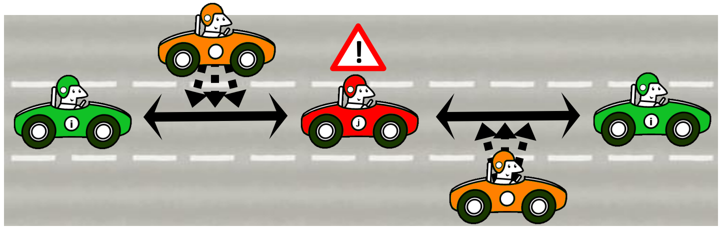
$$\forall i\; x(i) \ll L\big(x(i)\big) \rightarrow [delete^*][create^*][glc]\forall i\; x(i) \ll L^*(x(i))$$

([;])

$$\forall i\; x(i) \ll L\big(x(i)\big) \rightarrow [lhc]\forall i\; x(i) \ll L^*(x(i))$$



Conclusions



Challenges

- Infinite, continuous, and evolving state space, $\mathbb{R}^\infty$
- Continuous dynamics
- Discrete control decisions
- Distributed dynamics
- Arbitrary number of cars, changing over time
- Emergent behaviors

Solutions

- Quantifiers for distributed dynamics of cars
- Compositionality – using small problems to solve the big ones
- Hierarchical and modular proofs
- Variations in system design
- Future work: curved road dynamics and using differential invariants

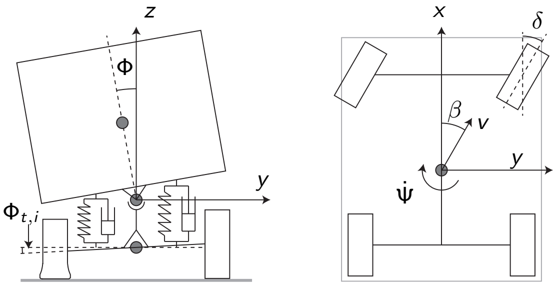
Rollover Verification of a Truck

Problem: Prove that truck cannot roll over under all possible maneuvers when the truck is braking ($a_x = -7$ m/s^2) and the lateral acceleration is bounded by $a_y \in [-4, 4]$ m/s^2

- Infinitely many maneuvers including all steering frequencies.
- Cannot be exhaustively tested by real experiments and simulations.

Challenges:

- Nonlinear cont. dynamics (8 cont. state variables)
- Uncertainty: Steering input
- Hybrid dynamics (gain scheduled controller)



Capturing Nonlinear Dynamics and Uncertain Inputs

Inherit problem: Only linear maps are structure-preserving for common set representations (ellipsoids, polyhedra, zonotopes, etc.)

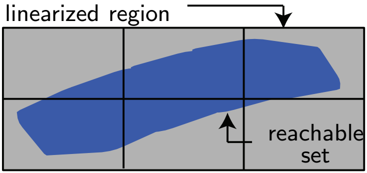
Solution: Abstract nonlinear dynamics to linear dynamics (x: state, u: input):

$$\dot{x} = f(x(t), u(t)) \in \left\{ A(t)x(t) + u(t) + v(t) \middle| A(t) \in \mathcal{A}, v(t) \in \mathcal{V} \right\}$$

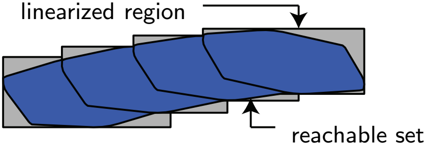
Dynamic abstraction using

- uncertain system matrix $\mathcal{A}$: [Althoff, Le Guernic, Krogh 2011]
- uncertain additional input $\mathcal{V}$: [Dang, Le Guernic, Maler 2011; Althoff et al. 2008]

Old technique: Static abstraction (coarser abstraction, guard intersection required):



New technique: Dynamic abstraction (tighter abstraction, no guard intersection required):

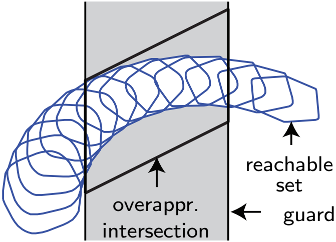


Capturing Switching Dynamics

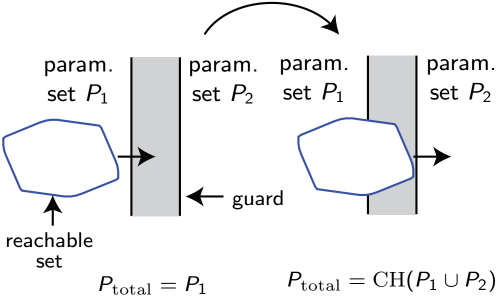
Hybrid reachability is limited by geometric intersections with guard sets, which is

- exact for polyhedra, but does not scale and is numerically unstable,
- efficient for other representations (template polyhedra, etc.), but conservative.

Old technique: Classical intersection computation possibly resulting in large overapproximation.



New technique: Compute with union of parameters when only the parameter set changes [Althoff, Le Guernic, Krogh 2011].



Dynamics of the Closed Loop System

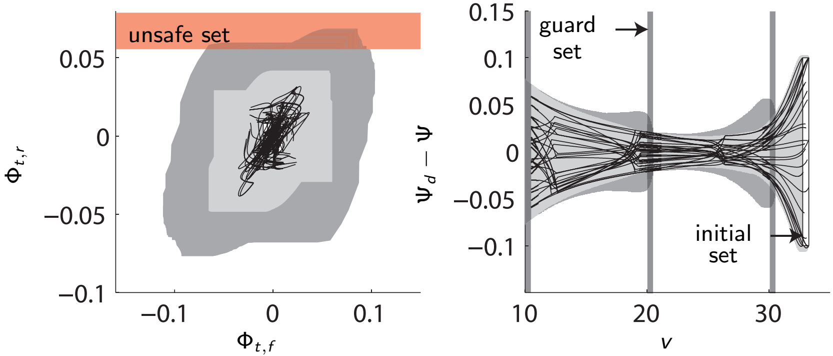
truck dynamics (blue variables are states, red ones are inputs) taken from [Gaspar et al. 2004]:

$$
\begin{aligned}
m x_7(\dot{x}_1 + x_2) - m_S h \dot{x}_4 &= Y_\beta x_1 + Y_{\dot{\Psi}}(x_7) x_2 + Y_\delta \delta \\
-I_{xz}\dot{x}_4 + I_{zz}\dot{x}_2 &= N_\beta x_1 + N_{\dot{\Psi}}(x_7) x_2 + N_\delta \delta \\
(I_{xx} + m_S h^2)\dot{x}_4 - I_{xz}\dot{x}_2 &= m_S g h x_3 + m_S h x_7(\dot{x}_1 + x_2) - k_f(x_3 - x_5) \\
&\quad -b_f(x_4 - \dot{x}_5) - k_r(x_3 - x_6) - b_r(x_4 - \dot{x}_6) \\
-r(Y_{\beta,f} x_1 + Y_{\dot{\Psi},f} x_2 + Y_\delta \delta) &= m_{u,f}(r - h_{u,f}) x_7(\dot{x}_1 + x_2) + m_{u,f} g h_{u,f} x_5 \\
&\quad - k_{t,f} x_5 + k_f(x_3 - x_5) + b_f(x_4 - \dot{x}_5) \\
-r(Y_{\beta,r} x_1 + Y_{\dot{\Psi},r} x_2) &= m_{u,r}(r - h_{u,r}) x_7(\dot{x}_1 + x_2) - m_{u,r} g h_{u,r} x_6 \\
&\quad - k_{t,r} x_6 + k_r(x_3 - x_6) + b_r(x_4 - \dot{x}_6) \\
\dot{x}_7 &= a_x.
\end{aligned}
$$

yaw controller: $\delta = k_1 e + k_2 \int e(t)\,dt, \quad e = \dot{\Psi}_d - \dot{\Psi} = \dot{\Psi}_d - x_2.$

velocity $x_7 \in$	$[10, 20]$ m/s	$[20, 30]$ m/s	$[30, \infty[$ m/s
controller	$k_1 = 0.4$	$k_1 = 0.5$	$k_1 = 0.6$
gains	$k_2 = 1.5$	$k_2 = 2$	$k_2 = 2.5$

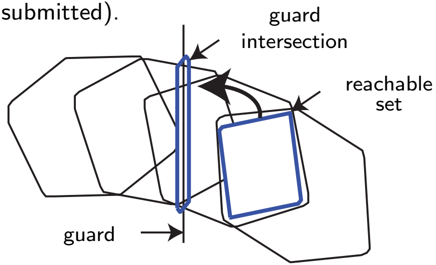
Reachable Set of the Truck



- Black lines: possible trajectories.
- Dark gray area: old technique; light gray area: new technique.
- Verification of safety only achieved by new technique.
- Computation time 38 s on an Intel i7 Processor with 6GB memory in MATLAB.

Other Advances for Hybrid Reachability Analysis

- Abstracting hybrid dynamics to uncertain linear dynamics. Allows verification of a phase-locked loop in the time of a few simulations [Althoff et al. 2011].
- Tightening the reachability results of linear system with uncertain parameters [Althoff, Krogh 2010].
- Introduction of zonotope bundles to mitigate shortcomings of zonotopes [Althoff, Krogh 2011].
- Development of a mapping enclosing the guard intersection of hyperplanes [Althoff, Krogh 2012] (submitted).



- Applications: phase-locked loop, RLC-circuits, autonomous cars, automotive powertrain, collision avoidance at intersections.

Embedded Control Systems

Design Flow

- requirements
- algorithms
- detailed design
- code

Challenges

- consistency
- correctness
- complexity
- coverage
- run-time correctness

CMACS Research

- requirements reconstruction
- analysis of hybrid systems
  - theorem proving
  - compositionality
  - reachability
  - statistical model checking
- code verification
  - abstract interpretation
  - analysis-aware design
  - run-time verification

Embedded Systems: Future Research Directions

- scalability for more complex systems
- compositional methods for hybrid systems
- advancing probabilistic/statistical methods
- integrated methods (theorem proving, model checking, abstract interpretation, probabilistic approaches)
- abstractions for real systems
- industry-scale case studies