Formal Composition Technology for Time-Triggered Systems

Pat Lincoln, John Rushby, Ashish Tiwari

{lincoln,rushby,tiwari}@csl.sri.com

http://www.csl.sri.com/.

Computer Science Laboratory

SRI International

333 Ravenswood

Menlo Park, CA 94025

Administrative

Title:	Formal Composition Technology for Time-Triggered Systems
PM:	John Bay
PI:	John Rushby
Address:	Computer Science Laboratory SRI International Menlo Park, CA 94025 (650) 859 5456, `rushby@csl.sri.com`
Contract No.:	F33615-00-C-1700
AO Number:	K232
End Date:	May 2003
Agent:	AFRL

Subcontractors and Collaborators

Subcontractors: No subcontractors

Collaborations:

- MoBIES:
  - Vanderbilt University: Design and development of interchange format and translators for analysis tools
  - U Penn, CMU: Model exchanges via HSIF
  - Kestrel: Parser for Stateflow
- SEC: Stanford
- BioSpice: SRI
- Outside: Honeywell, NASA

Problem Description

- Develop tools and techniques for automated formal analysis and assurance of models of embedded hybrid systems
- Develop invisible formal methods technology for integration with modeling tools based on lightweight theorem proving and symbolic reasoning

Program Objectives

- Provide analysis tools that integrate with the design process for development of embedded systems
- Success criteria: automated analysis on models from the OEP challenge problems

Tool Description

Name:	SAL tool suite
Description:	Analysis of safety properties of input models
Input:	Model in the SAL language/SAL XML
	Assumptions: Polynomial hybrid systems
Output:	Abstract system, Other verification results (typechecking), Counter-examples, etc
Interfaces:	Technology: HSIF (VU, UCB, Penn, CMU)
	Now: Can translate from HSIF into SAL
	Future: Translate from SAL to HSIF
	∴ connects with tools with HSIF interface
Non-MoBIES:	Future planned interface to SBML, etc

OEP Participation

OEP: Berkeley V2V

Technical POC: Mike Drew

Contributions:

- Design of the HSIF interchange format in collaboration with Vanderbilt, Berkeley, U Penn, CMU
- Translator from HSIF into SAL
- Verified simple V2V examples using a novel technique of doing reachability for linear systems

Project Status: Approach

Our present technical approach to verification of safety properties for hybrid system models is:

- Use invisible formal methods to check for certain kinds of inductive properties for the given model, e.g. type safety, completeness of specification, etc
- Use automated abstraction techniques to get a discrete transition system abstract model from a hybrid system model
- Output the abstract system (for other tools to use)
- Use the configurable explicit state model-checker to perform analysis on the abstract model
- Translate from and to the SAL modeling language to extend its interface

Project Status: Progress

- Developed a translator from HSIF to SAL
- Experimenting with the V2V challenge problems, we observed that effective use the abstraction tool for hybrid systems requires a good set of seed polynomials
- Developed new theoretical results to do non-trivial reachability computation on classes of linear systems that do not fall in the class of systems with a decidable reachability problem such as
  - nilpotent systems,
  - diagonalizable with rational eigenvalues,
  - diagonalizable with imaginary eigenvalues

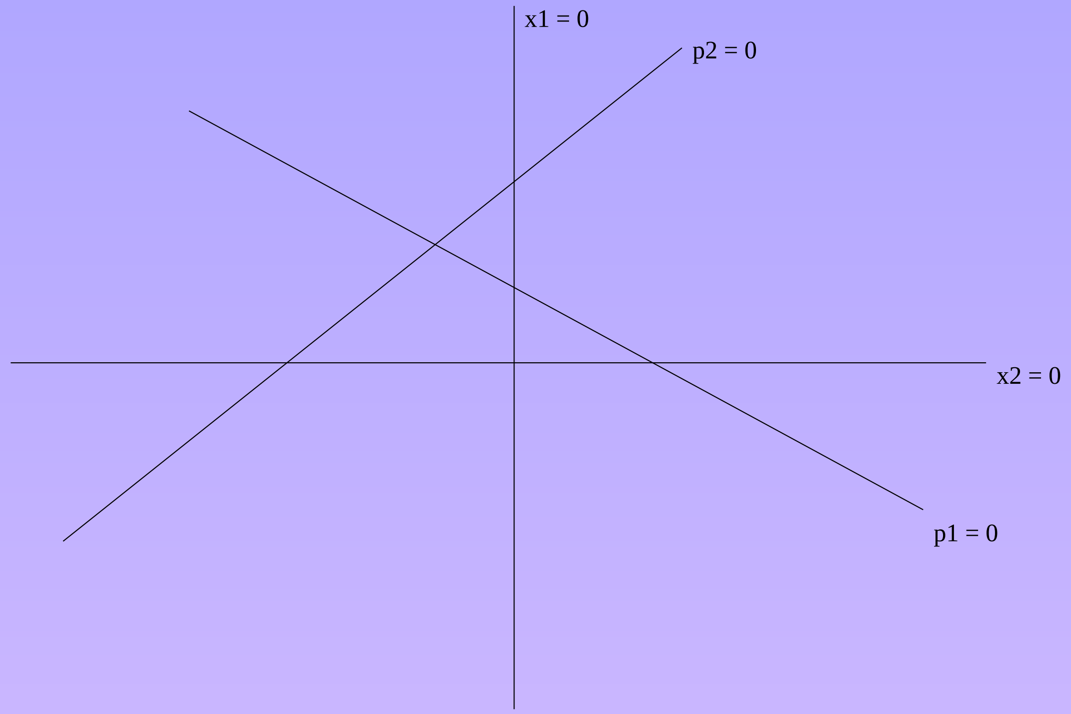
  The new result suggests seed polynomials for the abstraction tool

Project Status: Progress

- Developed a specialized model-checker for dealing with models created by abstraction
  - model-checker is aware that certain states might not be feasible
  - model-checker interfaces with an external routine to check for feasibility of a particular state
- Implemented the new technique of abstraction and model-checking using a new fast decision procedure for polynomial formulas
- Demonstrated the applicability of the new techniques on collision avoidance examples from the V2V auto OEP

Project Status: Progress

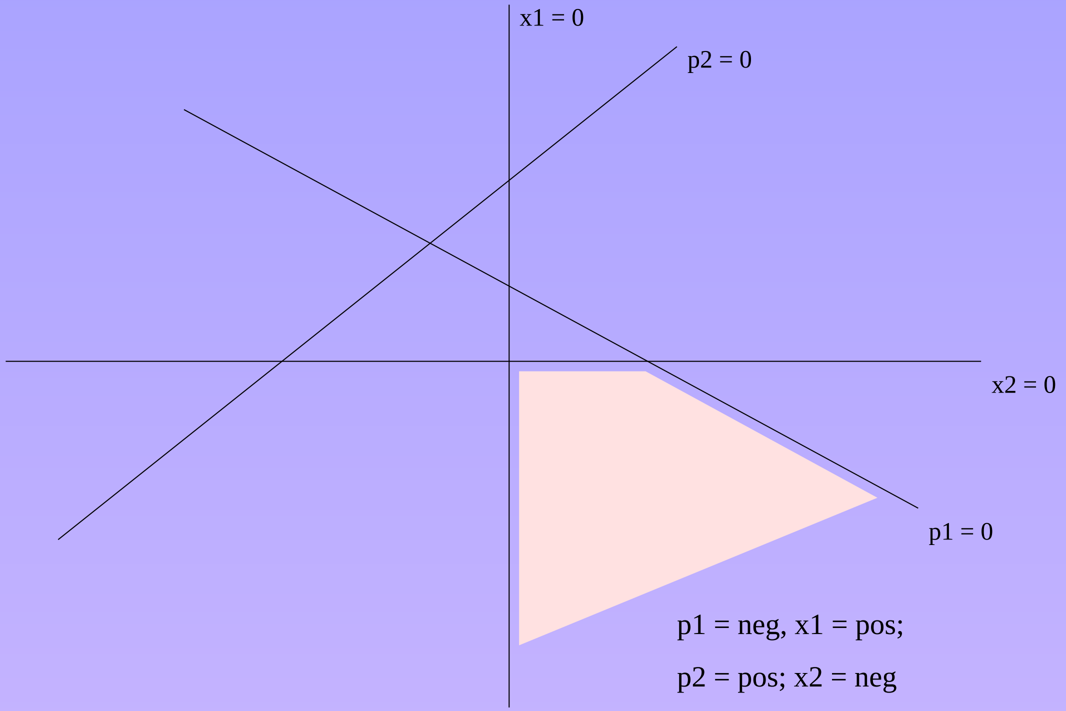
Recall the abstraction technique. A continuous dynamical system with two state variables, with a concrete state space $\Re^2$:



Partitioned w.r.t signs of polynomials x_1, x_2, p_1, and p_2.

Abstraction Algorithm

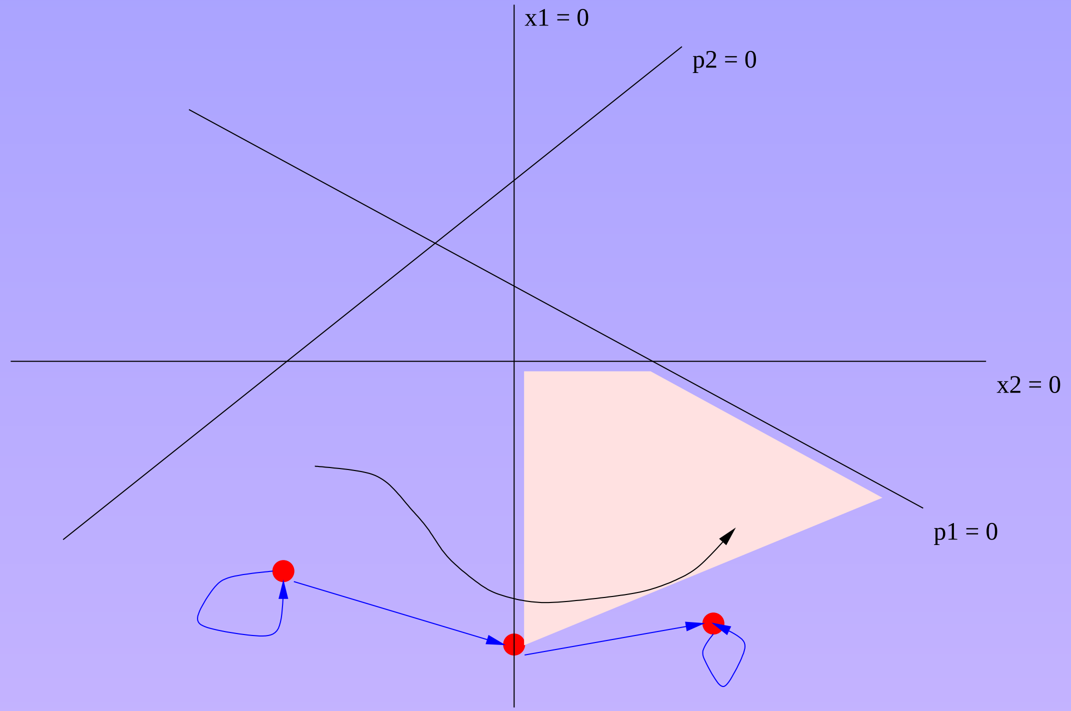
Abstract states correspond to subsets of concrete states.



More polynomials would mean more abstract states.

Abstraction Algorithm

Abstract transitions overapproximate concrete transitions.



Total number of abstract states = $3^4 = 81$, but feasible abstract states = $11 + 16 + 6 = 33$

Choosing Partition Polynomials

For a linear system, say specified by matrix A, use the eigenvector of the transpose A^T corresponding to a real eigenvalue.

Example. Consider a cruise control:

$$\begin{bmatrix} \dot{v} \\ \dot{vf} \\ \dot{a} \\ \dot{gap} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -4 & 3 & -3 & 1 \\ -1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ vf \\ a \\ gap \end{bmatrix}$$

where v, a is the velocity and acceleration of this car, vf is the velocity of car in front, and gap is the distance between the two cars.

Example Contd

The transpose matrix A^T has one negative real eigenvalue λ.
If $\vec{r} = [r1, r2, r3, r4]^T$ is the eigenvector corresponding to λ, then consider the polynomial

$$p = r1 * v + r2 * vf + r3 * a + r4 * gap$$

Why is this special?

$$\begin{aligned} \frac{dp}{dt} &= \frac{d}{dt}([v, vf, a, gap]\vec{r}) \\ &= (A[v, vf, a, gap])^T \vec{r} \\ &= [v, vf, a, gap] A^T \vec{r} \\ &= [v, vf, a, gap] \lambda \vec{r} = \lambda p \end{aligned}$$

Progress: Results

Interesting consequences of this observation:

- Can do non-trivial reachability computation for linear systems with mixed eigenvalues
  - existing decidability results can not handle this class of systems
  - several systems in the V2V challenge problems can be handled using this new technique
  - we extract as much information from the system as available, and bridge the gap between the decidable and undecidable problems

Progress: Results

Additional interesting consequence:

- Do not need to explicitly compute the real eigenvalues or the eigenvector $\vec{r}$
  - the eigenvalue and eigenvector are easily seen to be algebraic
  - symbolic decision procedures for real closed fields can handle the algebraic expressions representing these eigenvalues
- Although the new idea applies specifically only to linear systems, it suggests ways to handle non-linear systems as well

Whats in the Demo

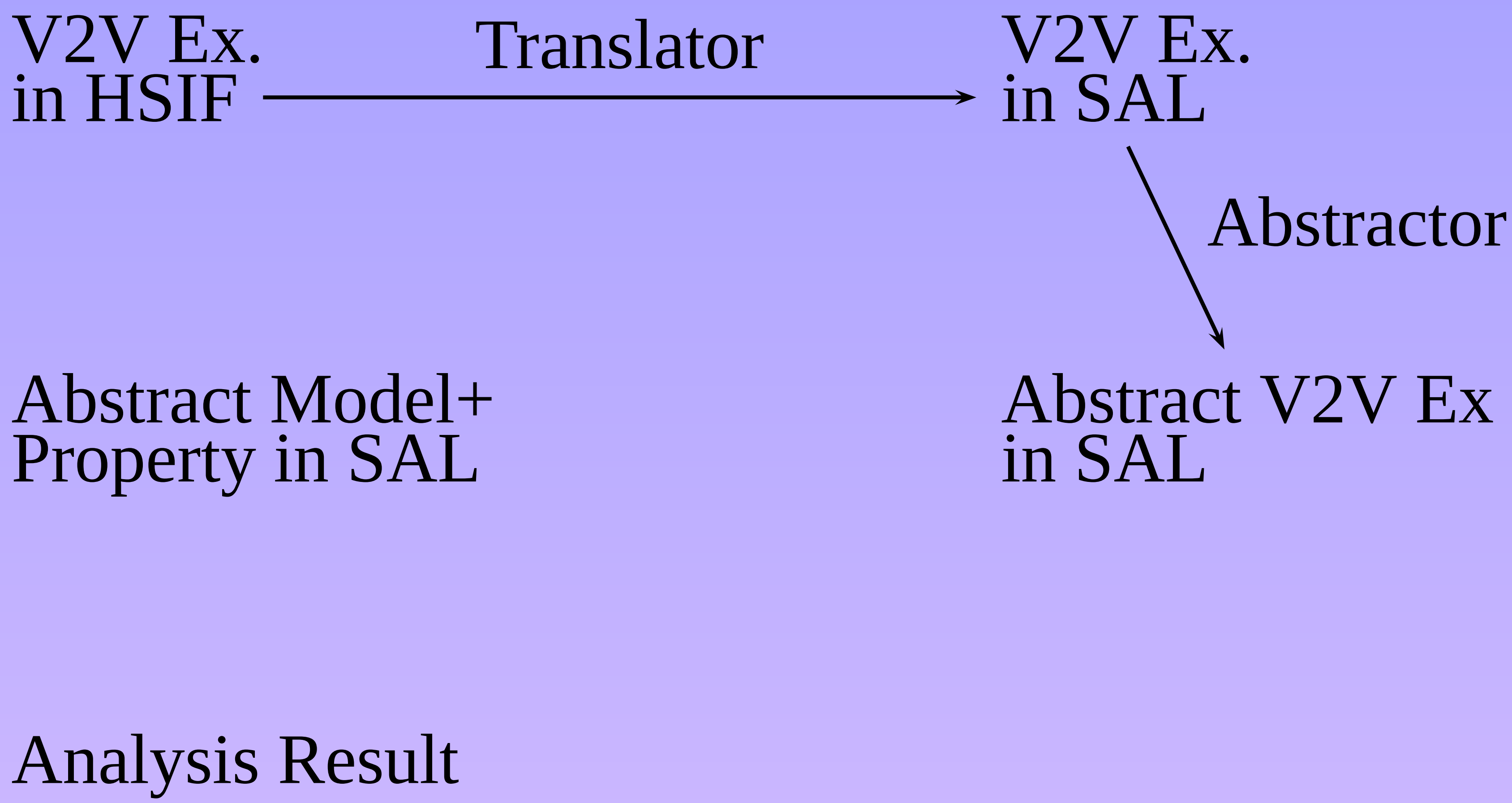
V2V Ex.
in HSIF → Translator → V2V Ex.
in SAL

Abstract Model+
Property in SAL

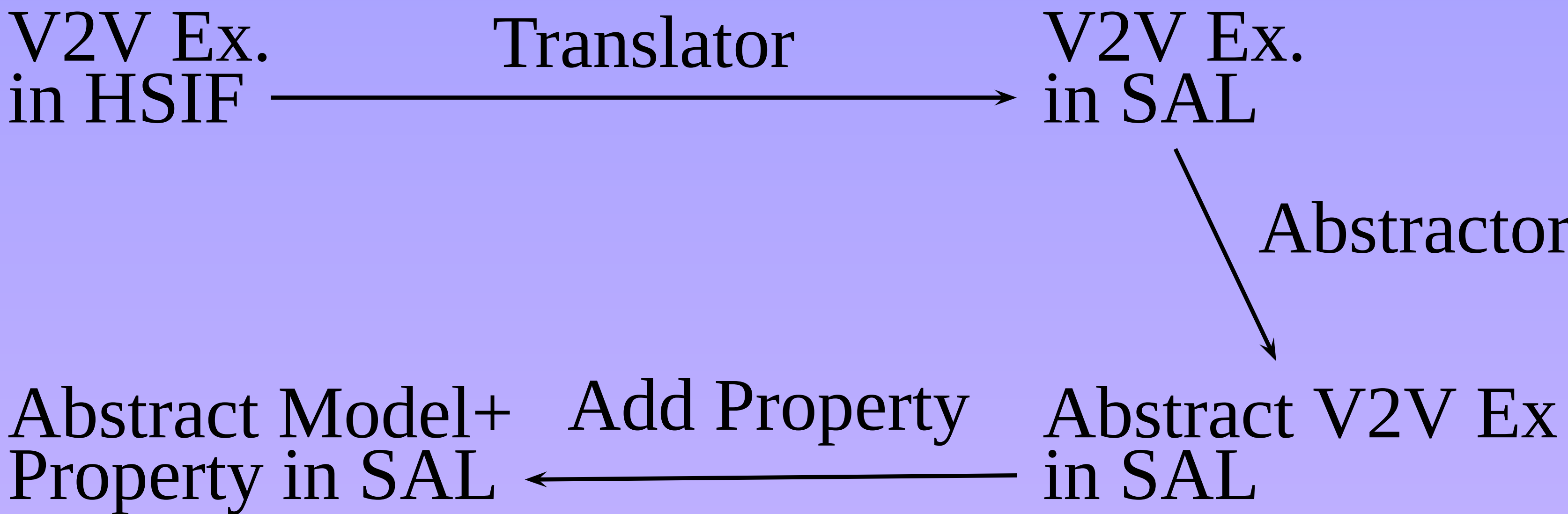
Abstract V2V Ex
in SAL

Analysis Result

Whats in the Demo

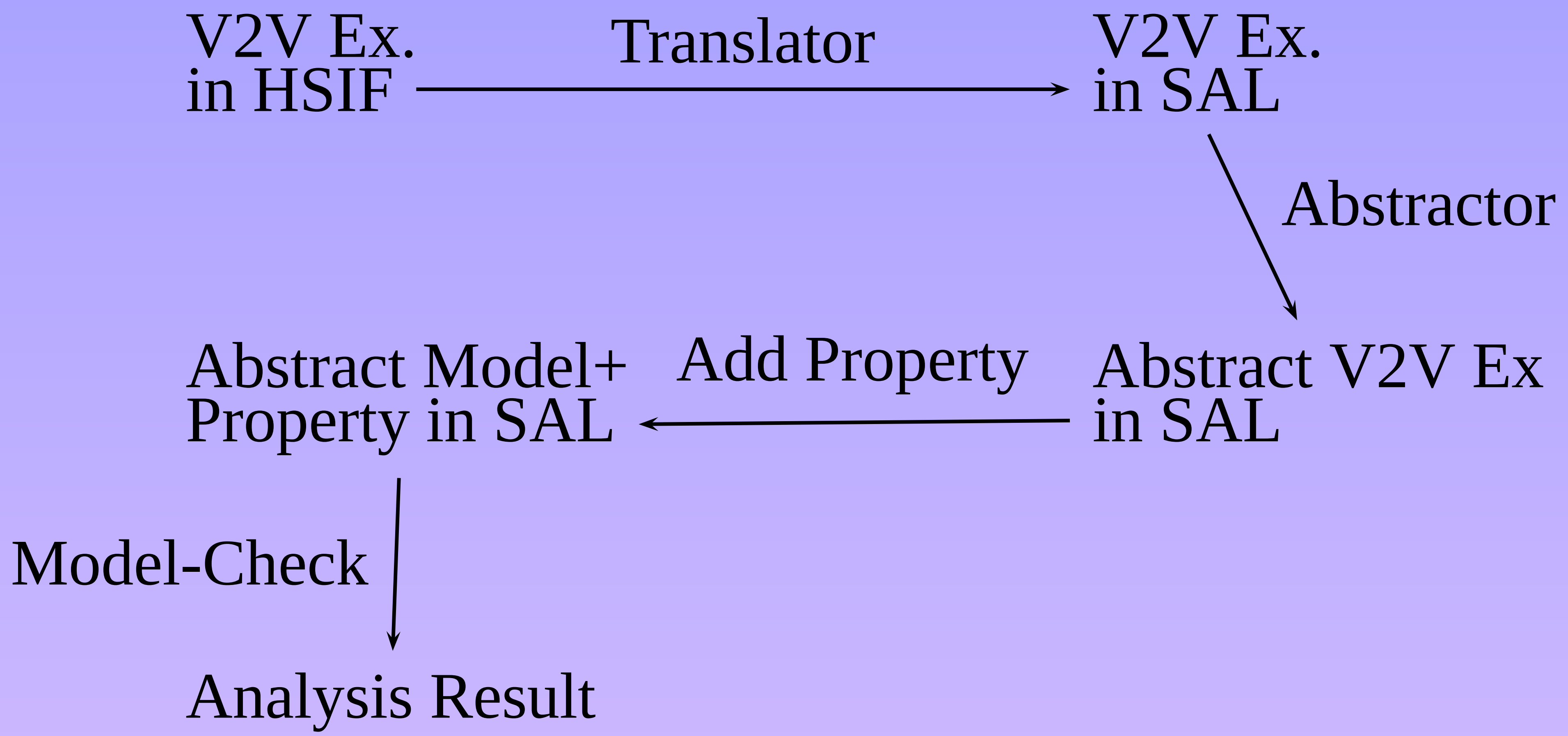


Whats in the Demo



Analysis Result

Whats in the Demo



Interpreting the Demo Results

- We do reachability analysis to show safety, and not just stability analysis of the given system
- We prove that the rear car would not collide with the car in front only assuming
  - a bound on the cars acceleration and deceleration
  - initial state of the two cars falls inside the assumed algebraic set
  - the leading car is moving at a constant, but unspecified, velocity
- Further analysis can be carried out using the same tools with different initial conditions and different properties

Project Status: Accomplishments

- Developed a translator from HSIF into SAL
- Used the translator to convert a HSIF specification of a V2V cruise control example into SAL
- Automatically abstracted the SAL specification to a simpler SAL specification
- Model-checked the abstraction for safety properties
- Proved collision avoidance for many different controllers developed by the OEP

Publication:

- "Invisible formal methods for embedded control systems", To appear in Proceedings of the IEEE
- Most of the new work is unpublished as yet

Project Plans

For the next 6 months

- Develop tools to perform lightweight formal analysis of large models specified in, say Simulink/Stateflow
- Develop new insights to perform analysis of non-linear systems by generalizing some of the approaches we developed for linear systems
- Build more features into the abstraction tool to handle compositions of hybrid automata automatically
- Continue experimental work by taking a bigger example from the V2V OEP with many different modes and non-trivial mode transitions

Project Schedule and Milestones

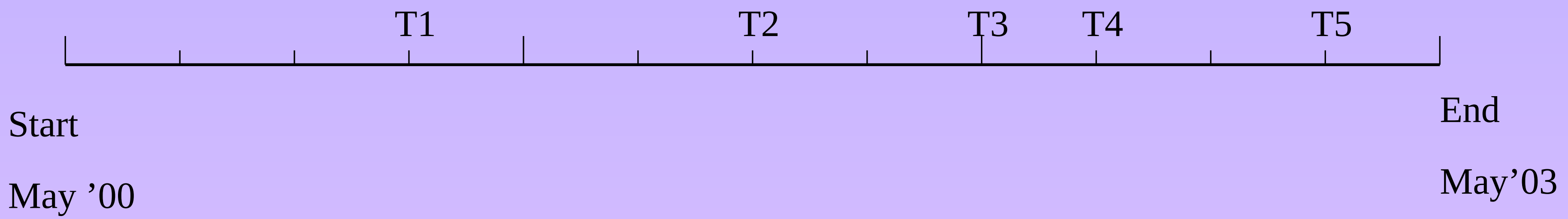
T1 . Semantics of Stateflow and checking Stateflow diagrams for simple properties

T2 . Invariant checking and typechecking for SAL specifications of hybrid system models

T3 . Model-checking tools to explore state-space of abstracted systems

T4 . Abstraction technique enhancements and composition

T5 . Interface of tools with other tools and Simulink/Stateflow and further experimentation



Project Schedule

- Development of analysis tools has been an incremental process—new techniques were (and are being) implemented and tested on challenge problems
- Focus on analysis has continued longer than initially expected
- Revisiting the lightweight methods to connect earlier work and some new work with the Matlab tools: given the availability of translators developed by other Mobies participants

Technology Transition/Transfer

- SAL/PVS integration with decision procedure for real closed fields to ICASE for verification of aircraft collision avoidance algorithms
- Rockwell-Collins considering SAL to integrate their various development and analysis tools
- BioSpice program