Efficient Dynamic Error Reduction for Hybrid Systems Reachability Analysis

Stefan Schupp Erika Ábrahám
RWTH Aachen University, Germany

Motivation

Safety verification for hybrid systems has come a long way, but:

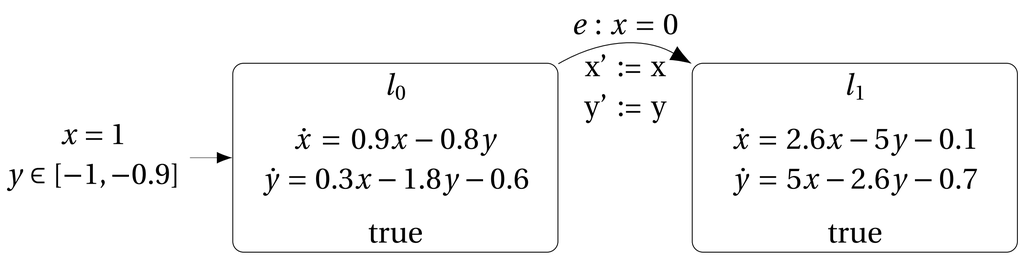
- **Analysis parameters:** Tuning of analysis parameters requires expert knowledge.
- **Refinement:** Verification fails → restart whole analysis with adjusted parameters.

Partial path refinement in a nutshell

- **Analysis parameters:** Ordered sequence of analysis parameter configurations.
- **Refinement:** Only partial refinement instead of full refinement.
- **Incremental:** Reuse information from previous refinement runs.

Hybrid Automata [1]

Example: Hybrid automaton



- *Time evolution:* Variables change according to ODE in current mode.
- *Discrete jumps:* Guarded switching between modes.

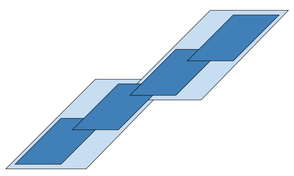
Flowpipe Construction

Reachability analysis: Over-approximate the set of reachable states R of system S for time horizon T by a set R'. Check R' against bad states P_{bad}.
Verification: $R' \cap P_{bad} = \emptyset \Rightarrow S$ safe, otherwise: *unknown* (failure).

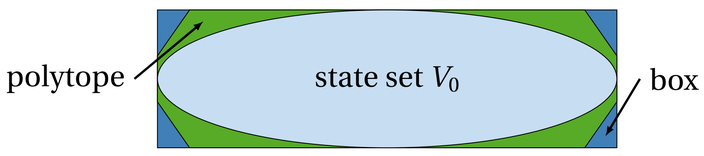
Flowpipe construction: Use geometric or symbolic state set representations to over-approximate R.

Some analysis parameters affecting over-approximation R':

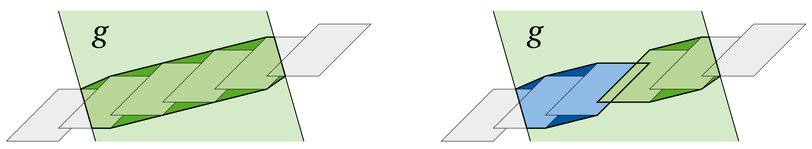
- Discretize T : $\delta = \frac{T}{N}$, large $\delta \rightarrow$ less precision, faster.



- State set representations (e.g. boxes, convex polytopes, support functions) [2].



- Discrete jumps: aggregate (left) or cluster state sets (right).

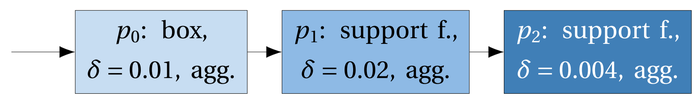


In general: Precision vs. computational effort.

Error Reduction: Partial Path Refinement

- **Parameter configuration p_i:** One set of relevant analysis parameters.
- **Strategy:** Finite, ordered sequence of parameter configurations p_i.

Example continued: Strategy

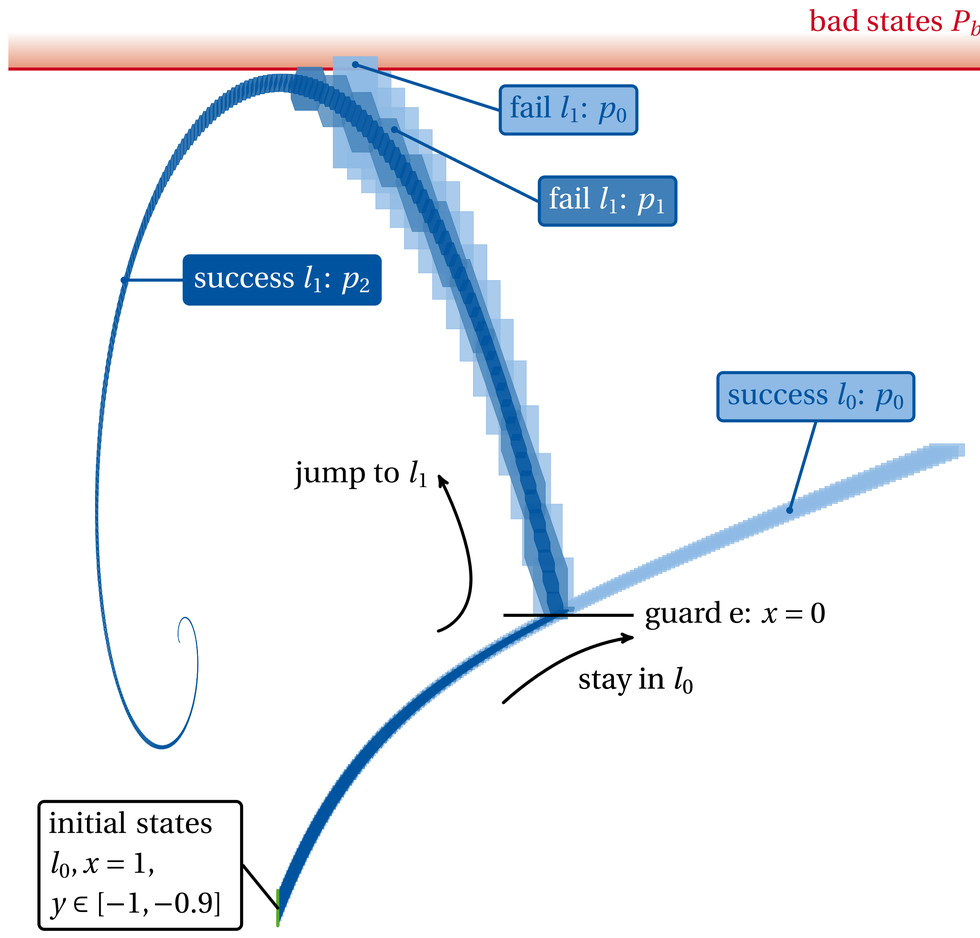


Partial path refinement

Idea:

- Start analysis with configuration p_0.
- Bad states reachable using configuration p_i: Refine counterexample path with configuration p_{i+1}.
- Store refinement information.
  → Counterexamples with shared path prefix: reuse refinements.
- Reuse information during refinement (e.g. time intervals for enabled transitions).

Example continued: Analysis



Observations:

- Coarse analysis (p_0) sufficient for mode l_0.
- Require configuration p_2 for mode l_1.
- Not visible: Reduced number of guard checks for configurations p_1, p_2 in mode l_0.

Future Work

- Increase usage of information obtained during refinements.
- Synthesize parameter configurations (at runtime).
- Introduce conditional strategies (strategy tree).
- Parallelization.

References

[1] Thomas A. Henzinger. "The theory of hybrid automata". In: *Proc. LICS'96*. IEEE Computer Society Press, 1996, pp. 278–292.

[2] Stefan Schupp et al. "HyPro: A C++ library for state set representations for hybrid systems reachability analysis". In: *Proc. of NFM'17*. Vol. 10227. LNCS. Springer, 2017, pp. 288–294.

Acknowledgments

This work was partially funded by the German research council (DFG) in the context of the project HyPro and the DFG Research Training Group 2236 UnRAVeL.

Download this poster