EPiC Series in Computing

Volume 48, 2017, Pages 134–142

ARCH17. 4th International Workshop on Applied
Verification of Continuous and Hybrid Systems

ARCH-COMP17 Category Report: Bounded Model Checking of Hybrid Systems with Piecewise Constant Dynamics

Lei Bu[1], Rajarshi Ray[2], and Stefan Schupp[3]

[1] State Key Laboratory of Novel Software Techniques,
Nanjing University, Nanjing, Jiangsu, P.R. China
bulei@nju.edu.cn

[2] National Institute of Technology Meghalaya, Shillong, India
rajarshi.ray@nitm.ac.in

[3] RWTH Aachen University, Theory of hybrid systems, Aachen, Germany
stefan.schupp@cs.rwth-aachen.de

Abstract

This report presents results of a friendly competition for formal verification of continuous and hybrid systems with linear continuous dynamics. The friendly competition took place as part of the workshop Applied Verification for Continuous and Hybrid Systems (ARCH) in 2017. In its first edition, three tools have been applied to solve three different benchmark problems in the category of*bounded model checking of hybrid systems with piecewise constant dynamics* (in alphabetical order): BACH, HyDRA, and XSpeed. The result is a snapshot of the current landscape of tools and the types of benchmarks they are particularly suited for. Due to the diversity of problems, we are not ranking tools and we also welcome more tools to join in this friendly competition in the future event.

1 Introduction

Disclaimer The presented report of the ARCH friendly competition for *bounded model checking of hybrid systems with piecewise constant dynamics* aims at providing a landscape of the current capabilities of verification tools. We would like to stress that each tool has unique strengths—not all of the specificities can be highlighted within a single report. To reach a consensus in what benchmarks are used, some tools may benefit more from the presented choice than others. Meanwhile, in order to invite more tools to join the competition, the flow condition is restricted to the format of $dx/dt = a$, instead of more general form of $dx/dt \in [a, b]$. The obtained results have not been independently verified, but the authors trust each other's results. To establish trustworthiness of the results, the code with which the results have been obtained is publicly available.

This report summarizes results obtained in the 2017 friendly competition of the ARCH workshop[1] for bounded model checking of hybrid systems with piecewise constant dynamics. More specifically, the flow condition is restricted to the format of $dx/dt = a$, instead of more general form of $dx/dt \in [a, b]$. Tool developers run their tools summarized in Sec. 2 on different benchmark problems presented in Sec. 3 and report the results obtained from their own machines also in Sec. 3.

The results reported by each participant have not been checked by an independent authority and are obtained on the machines of the tool developers. Thus, one has to factor in the computational power of the used processors summarized in Sec. A as well as the efficiency of the programming language of the tools. It is not the goal of the friendly competition to rank the results, the goal is to present the landscape of existing solutions in a breadth that is not possible by scientific publications in classical venues. Those would require the presentation of novel techniques, while this report showcases the current state of the art.

The selection of the benchmarks has been conducted within the forum of the ARCH website (cps-vo.org/group/ARCH), which is visible for registered users and registration is open for anybody. All tools presented in this report use some form of reachability analysis. This, however, is not a constraint set by the organizers of the friendly competition. We hope to encourage further tool developers to showcase their results in future editions.

2 Participating Tools

The tools participating in the category *Bounded Model Checking of Hybrid Systems with Piecewise Constant Dynamics* are introduced below in alphabetical order.

BACH BACH [3, 2] is a bounded reachability checker for Linear Hybrid Automata (LHA) model, Hybrid Systems with Piecewise Constant Dynamics (HPWC) in the term of ARCH competition. The tool provides GUI for LHA modeling and also bounded reachability checkers for both single automaton and automata network. Be different from classical bounded checkers of LHA, which encodes the "complete" bounded state space of the system into a huge SMT problem, BACH conducts the bounded checking in a "path-oriented" layered style. It finds potential paths which can reach the target location on the graph structure first, then encodes the feasibility of such path into a linear programming problem and solve it afterwards. In this way, as the number of paths in the discrete graph structure of an LHA under a given bound is finite, all candidate paths can be enumerated and checked one by one to tackle the bounded reachability analysis of LHA. Furthermore, the memory usage is well controlled as it only encodes and solves one path at a time. Meanwhile, BACH provides an efficient way to locate the infeasible path segment core when a path is reported as infeasible to guide the backtracking in the graph structure traversing to achieve good performance [11]. Such infeasible path segments can also be used to derive complete state arguments under certain conditions [12].

HyDRA The Hybrid systems Dynamic Reachability Analysis (HyDRA) tool implements flow-pipe construction based reachability analysis for linear hybrid automata. The tool is built on top of HyPro [6, 10], a C++ library for reachability analysis. HyPro provides different implementations of state set representations tailored for reachability analysis such as boxes, convex polyhedra, support functions, or zonotopes, all sharing a common interface. This interface allows to easily exchange the utilized state set representation in HyDRA. We use this to

[1] Workshop on Applied Verification for Continuous and Hybrid Systems (ARCH), cps-vo.org/group/ARCH

extend state-of-the art reachability analysis by CEGAR-like parameter refinement loops, which (among other parameters) allow to vary the used set representation. Furthermore, HyDRA incorporates the capability to explore different branches of the search tree in parallel. Being in an early state of development, HyDRA already shows promising results on some benchmarks, although there is still room for improvements. An official first release is planned.

XSpeed The tool *XSpeed* implements algorithms for reachability analysis for continuous and hybrid systems with linear dynamics. The focus of the tool is to exploit the modern multicore architectures to enhance the performance of reachability analysis through parallel computations. XSpeed realizes two algorithms to enhance the performance of reachability analysis of purely continuous systems. The first is the parallel support function sampling algorithm and the second is the time-slicing algorithm [8, 9]. The performance of hybrid systems reachability analysis is enhanced using the adapted G.J. Holzmann's algorithm and the task parallel algorithm, both of which proposes variants of parallel breadth first exploration of the hybrid automaton [5].

3 Verification of Benchmarks

We have agreed on three benchmarks, each one of them having unique features. The *Motorcade* [7] instance used in the competition is a single model with small number of variables and locations. *Navigation* [4] instance is much larger than *Motorcade* in the aspect of locations. We also present a composed system, which is the famous Fischer's protocol [1] to see the performance of different tools of handling composed system. Next, let us briefly discuss the specificities and results of each benchmark problem.

Types of Inputs As the HBMC category focuses on HPWC model this year, all the three benchmarks are HPWC models with dynamic laws in the form of $dx/dt = a$, instead of more general form of $dx/dt \in [a, b]$. Therefore, the models used in the competition are a little different from their original versions. This restriction may causes different behavior in some benchmarks as well. For example, in NAV model, the point is supposed to move in different directions in the original model with $dx/dt \in [-a, a]$. However, in our version, as it is fixed to $dx/dt = a$, it can only goes in one direction.

3.1 Motorcade, a.k.a. Adaptive Cruise Controller

3.1.1 Model

The first automaton is a central arbiter for a automated motorcade in a highway introduced in [7]. The model is shown below in Fig.1. The system works as follows: when two vehicles come within a distance 4 of each other, a collision may happen. Then the arbiter asks the approaching car to slow down and the leading car to speed up. When the distance between the two vehicles involved in the possible collision exceeds 4, the arbiter model goes back to the dynamics of the cruise mode.

The size of this model can be easily expanded by introducing more cars into the system, which will increase new locations and variables in the model. However, in order to make the benchmark suitable for most of the competitors, we select a system with only 5 vehicles for competition, in another word, the benchmark model has 6 locations and 5 variables.

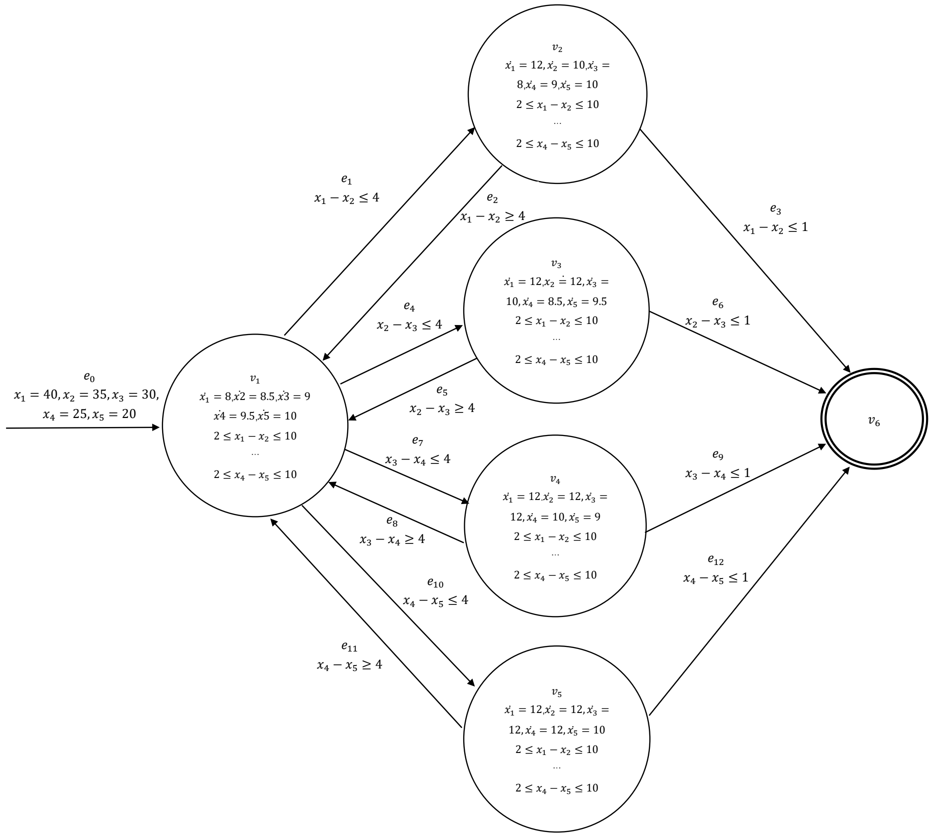

v_2
$\dot{x_1} = 12, \dot{x_2} = 10, \dot{x_3} =$
$8, \dot{x_4} = 9, \dot{x_5} = 10$
$2 \leq x_1 - x_2 \leq 10$
…
$2 \leq x_4 - x_5 \leq 10$
e_1
$x_1 - x_2 \leq 4$
e_2
$x_1 - x_2 \geq 4$
e_3
$x_1 - x_2 \leq 1$
v_3
$\dot{x_1} = 12, \dot{x_2} = 12, \dot{x_3} =$
$10, \dot{x_4} = 8.5, \dot{x_5} = 9.5$
$2 \leq x_1 - x_2 \leq 10$
…
$2 \leq x_4 - x_5 \leq 10$
e_4
$x_2 - x_3 \leq 4$
e_6
$x_2 - x_3 \leq 1$
e_0
$x_1 = 40, x_2 = 35, x_3 = 30,$
$x_4 = 25, x_5 = 20$
v_1
$\dot{x_1} = 8, \dot{x}2 = 8.5, \dot{x}3 = 9$
$\dot{x}4 = 9.5, \dot{x}5 = 10$
$2 \leq x_1 - x_2 \leq 10$
…
$2 \leq x_4 - x_5 \leq 10$
e_5
$x_2 - x_3 \geq 4$
v_6
e_7
$x_3 - x_4 \leq 4$
e_9
$x_3 - x_4 \leq 1$
v_4
$\dot{x_1} = 12. \dot{x_2} = 12, \dot{x_3} =$
$12, \dot{x_4} = 10, \dot{x_5} = 9$
$2 \leq x_1 - x_2 \leq 10$
…
$2 \leq x_4 - x_5 \leq 10$
e_8
$x_3 - x_4 \geq 4$
e_{12}
$x_4 - x_5 \leq 1$
e_{10}
$x_4 - x_5 \leq 4$
e_{11}
$x_4 - x_5 \geq 4$
v_5
$\dot{x_1} = 12, \dot{x_2} = 12, \dot{x_3} =$
$12, \dot{x_4} = 12, \dot{x_5} = 10$
$2 \leq x_1 - x_2 \leq 10$
…
$2 \leq x_4 - x_5 \leq 10$

Figure 1: Motorcade-5 vehicles

3.1.2 Specification

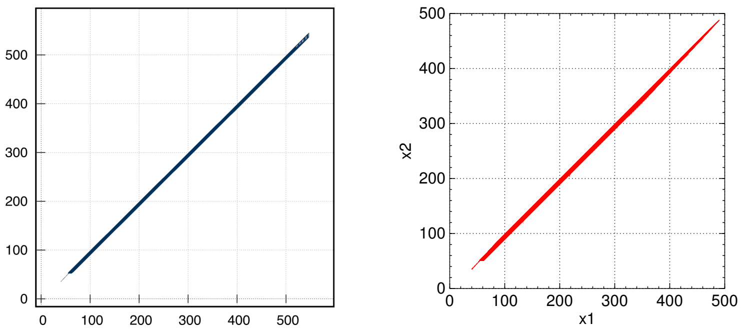
The safety property of this system is that no two vehicles collide with each other, in another word, whether location $Error\, v_6$ is reachable in bound 20.

3.1.3 Results

First of all, as different tools may have different settings, we have a specific column "remark" listed in all the forms. If the tool is executed directly according to the specification, it will be marked as "-". Otherwise, the setting will be marked there.

Plots Following are the plot file of the reachable set generated by HyDRA and XSpeed correspondingly. As BACH is not a fixed-point computation based checker, it does not provide such functionality.

Computation Times The computation times of various tools for the Motorcade benchmark are listed in Tab. 1. The performance for *XSpeed* is given for an exploration till bound 12, using



(a) HyDRA (BND 6)

(b) XSpeed (BND 6)

Figure 2: Plots of the reachable set for the Motorcade benchmark.

a time step of $\delta = 4e4$ and Octagonal template directions. *XSpeed* does not reach the target error state in bound 12.

Table 1: Computation Times on the Motorcade Benchmarks with Default Bound 20.

	computation time in [s]		**platform**	
tool	**time**	**remark**	**language**	**machine (see Sec. A)**
BACH	0.08	-	C++	M_{BACH}
HyDRA	17.6	Bound=6, Time=40000, $\delta = 0.1$	C++	M_{HyDRA}
XSpeed	6.76	Bound=12, DIR=OCT, $\delta = 4e4$	C++	M_{XSpeed}

3.2 Navigation (NAV)

3.2.1 Model

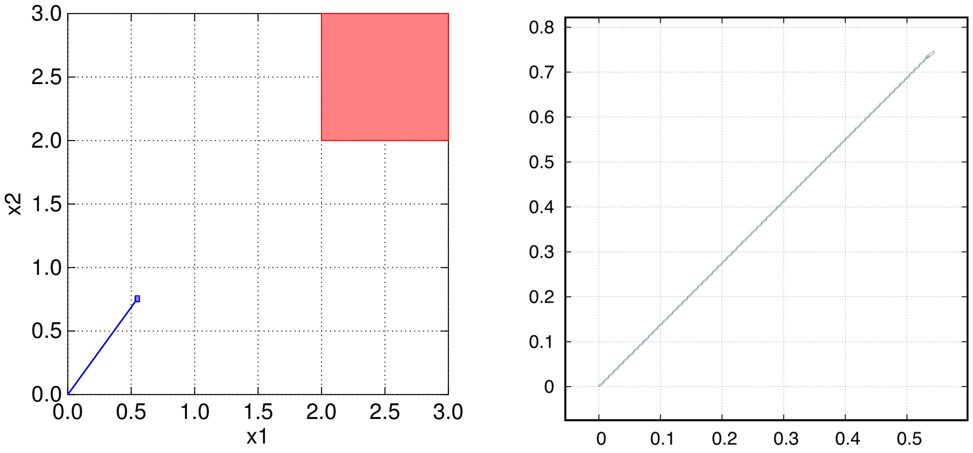
The navigation example is also a single automaton. It models the motion of a point robot in a n-dimensional cube. The cube is partitioned into m^n rectangular regions and each such region is associated with a vector field described by the flow equations. We use a generalization method introduced in [4] to generate a n-dimensional navigation model. Similar with the motorcade model, in order to generate a not too complex model, we set both m and n as 3, in another word 27 regions/locations. As the model is too large to put in the paper, we will omit the graphical presentation here.

3.2.2 Specification

The specification is to check whether there is a behavior of the system which can reach the specific state in the farthest corner. In the benchmark model, Whether l_{222} is reachable in bound 20.

3.2.3 Result

Plots The plots of the Navigation benchmark generated by XSpeed and HyDRA are shown in Fig. 3(a) and 3(b).



(a) XSpeed (BND 20)

(b) HyDRA (BND 20)

Figure 3: For XSpeed, reachable set of x_1, x_2 is shown in blue. The target region is shown in red. XSpeed shows that the target is not reachable in bound 20.

Computation Times The computation times of various tools for the NAV benchmark are listed in Tab. 2. The parameters δ and DIR in XSpeed denotes the time step and direction template used for the result.

Table 2: Computation Times on the NAV Benchmark with Default Bound 20.

	computation time in [s]		**platform**	
tool	**time**	**remark**	**language**	**machine (see Sec. A)**
BACH	44.76	uc=20	C++	M_{BACH}
HyDRA	0.02	$\delta = 0.01$	C++	M_{HyDRA}
XSpeed	0.14	$\delta = 0.001$, DIR=BOX	C++	M_{XSpeed}

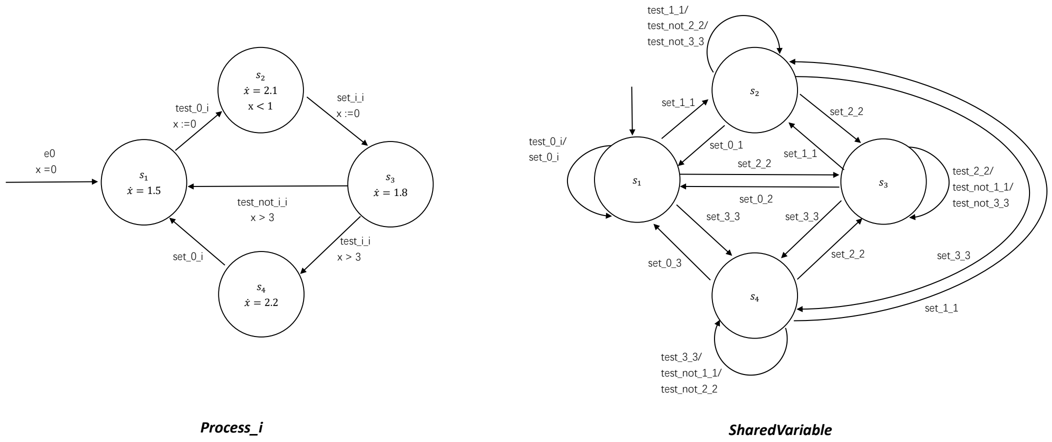
3.3 Fisher's Protocol

3.3.1 Model

The Fischer Protocol [1] system consists of several competing processes which all attempt to enter the critical section. The automaton we use to model a single process is shown in Fig. 4. As this is a classical shared variable problem, in order to handle it in the context of our tool (synchronize with shared labels), we build a LHA: Shared Variable (SV) to represent all the evaluation and reset actions on the shared variable. In the competition, the benchmark has 3 processes, in another word 4 automaton.

3.3.2 Specification

The specification is to check whether it is possible for all the processes enter the critical section together with bound 12.



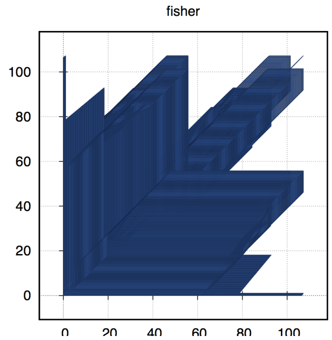
Process_i

SharedVariable

Figure 4: Fischer Protocol

3.3.3 Results

Plots The plots of the Fisher benchmark generated by HyDRA for $process_2, process_1$ with bound 10 are shown in Fig. 5.



(a) HyDRA (BND 10)

Figure 5: Reachable set of $process_2$ and $process_1$ is shown in blue.

Computation Times The computation times of various tools for the Fischer benchmark are listed in Tab. 3.

4 Conclusion and Outlook

This report presents the results on a first friendly competition for the *bounded model checking of hybrid systems with piecewise constant dynamics* as part of the ARCH'17 workshop. The

Table 3: Computation Times on the Fischer Benchmark with Default Bound 12.

	computation time in [s]		**platform**	
tool	**time**	**remark**	**language**	**machine (see Sec. A)**
BACH	2.25	-	C++	M_{BACH}
HyDRA	423	Bound=10, $\delta = 0.1$	C++	M_{HyDRA}
XSpeed	-	-	C++	M_{XSpeed}

reports of other categories can be found in the proceedings and on the ARCH website: cps-vo.org/group/ARCH.

A most interesting observation of the results is that different techniques/tools have different specialities. For exmaple, BACH is a path-oriented BMC checker, while HyDRA and XSpeed are fixed-point computation based. In the experiments, on case like Motorcade, which structure is not very complex but may have many possible execution trajectories. BACH finishes the checking quickly, while fixed-point computation based method may have difficulty in the computation.

On the other hand, the NAV model shows a different story. The structure of the model is large. There are many potential paths on the structure of the model. However, due to the restriction of the flow condition, only one way is possible according to the state computation. Therefore, fixed-point computation based method runs very quickly on this case, while BACH has to check all the path and spend a long time.

We would like to introduce more complex models with high dimension and/or large initial set in the future event to see what kind of system can be analyzed by existing tools. We would also like to encourage other tool developers to consider participation in the next year.

5 Acknowledgments

The authors gratefully acknowledge financial support of the Joint NSFC-ISF Research Program, jointly funded by the National Natural Science Foundation of China and the Israel Science Foundation (No.61561146394), and the National Natural Science Foundation of China (No.61572249), and the National Institute of Technology Meghalaya, India and DST-SERB, Government of India under project grant No. YSS/2014/000623.

A Specification of Used Machines

A.1 M_{BACH}

- Processor: Intel(R) Core(TM)2 Quad CPU Q9500 @ 2.83GHz x 4
- Memory: 4 GB
- Average CPU Mark on www.cpubenchmark.net: 3636 (full), 1203 (single thread)

A.2 M_{HyDRA}

- Processor: Intel Core i7-4790K CPU @ 4.00GHz x 8

- Memory: 15.9 GB
- Average CPU Mark on www.cpubenchmark.net: 11185

A.3 $\mathbf{M}_{XSpeed}$

- Processor: Intel Core i7-4770 CPU @ 3.4GHz x 4
- Memory: 8 GB
- Average CPU Mark on www.cpubenchmark.net: 9806

References

[1] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.*, 138(1):3–34, 1995.

[2] Lei Bu, You Li, Linzhang Wang, Xin Chen, and Xuandong Li. BACH 2 : Bounded reachability checker for compositional linear hybrid systems. In *Design, Automation and Test in Europe, DATE 2010, Dresden, Germany, March 8-12, 2010*, pages 1512–1517, 2010.

[3] Lei Bu, You Li, Linzhang Wang, and Xuandong Li. BACH : Bounded reachability checker for linear hybrid automata. In *Formal Methods in Computer-Aided Design, FMCAD 2008, Portland, Oregon, USA, 17-20 November 2008*, pages 1–4, 2008.

[4] Roberto Giacobazzi, Josh Berdine, and Isabella Mastroeni, editors. *Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI 2013, Rome, Italy, January 20-22, 2013. Proceedings*, volume 7737 of *Lecture Notes in Computer Science*. Springer, 2013.

[5] Amit Gurung, Arup Deka, Ezio Bartocci, Sergiy Bogomolov, Radu Grosu, and Rajarshi Ray. Parallel reachability analysis for hybrid systems. In *2016 ACM/IEEE International Conference on Formal Methods and Models for System Design, MEMOCODE 2016, Kanpur, India, November 18-20, 2016*, pages 12–22. IEEE, 2016.

[6] Hypro project website. Available at http://ths.rwth-aachen.de/research/projects/hypro/.

[7] Sumit Kumar Jha, Bruce H. Krogh, James E. Weimer, and Edmund M. Clarke. Reachability for linear hybrid automata using iterative relaxation abstraction. In *Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings*, pages 287–300, 2007.

[8] Rajarshi Ray and Amit Gurung. Poster: Parallel state space exploration of linear systems with inputs using xspeed. In *Proc. of HSCC'15*, pages 285–286. ACM, 2015.

[9] Rajarshi Ray, Amit Gurung, Binayak Das, Ezio Bartocci, Sergiy Bogomolov, and Radu Grosu. XSpeed: Accelerating reachability analysis on multi-core processors. In *Proc. of HVC 2015*, volume 9434 of *LNCS*, pages 3–18, 2015.

[10] Stefan Schupp, Erika Abraham, Ibtissem Ben Makhlouf, and Stefan Kowalewski. HyPro: A C++ library for state set representations for hybrid systems reachability analysis. In *Proc. NFM'17*, volume 10227 of *LNCS*, pages 288–294. Springer, 2017.

[11] Dingbao Xie, Lei Bu, Jianhua Zhao, and Xuandong Li. SAT-LP-IIS joint-directed path-oriented bounded reachability analysis of linear hybrid automata. *Formal Methods in System Design*, 45(1):42–62, 2014.

[12] Dingbao Xie, Wen Xiong, Lei Bu, and Xuandong Li. Deriving unbounded reachability proof of linear hybrid automata during bounded checking procedure. *IEEE Trans. Computers*, 66(3):416–430, 2017.