EPiC Series in Computer Science

Volume 37, 2015, Pages 40–48

Symbolic and Numerical Methods for Reachability Analysis, 1st International Workshop, SNR 2015

ProbReach: A Tool for Guaranteed Reachability Analysis of Stochastic Hybrid Systems

Fedor Shmarov[1] and Paolo Zuliani[1]

School of Computing Science, Newcastle University, Newcastle upon Tyne, U.K.
f.shmarov, paolo.zuliani@ncl.ac.uk

Abstract

In this paper we present a summary of our work on probability reachability in stochastic hybrid systems (SHS). In particular, we give an overview of `ProbReach`, a tool for computing probabilistic reachability in SHS which we introduced in [18]. We also present and an overview of our recent theoretical extensions and modification of the tool.

1 Introduction

- *What problem we address.* We study bounded reachability in stochastic hybrid systems (SHS). Our tool `ProbReach` assesses quantitative properties of SHS with random initial parameters, such as the probability of reaching a predefined unsafe region in a finite number of discrete transitions.
- *What kind of SHS we can analyse.* `ProbReach` performs reachability analysis in hybrid systems with random continuous initial parameters. These can be system parameters or initial conditions which are chosen in the initial state and remain unchanged throughout the system evolution. For continuous dynamics, we can analyse any Lipschitz-continuous ODEs.
- *What has been added.* Recently, we have extended the range of the systems that can be handled by `ProbReach`. The tool now supports hybrid systems with discrete initial parameters and continuous nondeterministic initial parameters.
- *What guarantees* `ProbReach` *provides.* Given a SHS with random continuous parameters and an arbitrarily small $\epsilon > 0$, `ProbReach` returns an interval of size not larger than ϵ containing the *exact* bounded reachability probability. This result is guaranteed to be numerically correct, *i.e.*, free from floating-point inaccuracies. Introducing discrete random parameters to the system will not affect the guarantees provided by `ProbReach`. However, if the system features only discrete random parameters then this guarantee does not hold. This happens because probability distributions over discrete random parameters are not continuous. Therefore, an arbitrary precision cannot be provided. Introducing nondeterministic continuous parameters affects the guarantees the tool provides, as well. This happens because nondeterministic parameters do not have any probability measure

at all. In this case, `ProbReach` computes an enclosure which is guaranteed to contain all the possible reachability probabilities. In general, such an enclosure may have size larger than ϵ.

- *The intuition behind our technique.* `ProbReach` employs a validated integration procedure to obtain a partition over the random parameters in such a way that the guarantees described above hold. This partition is then used to enclose the probability value by computing under- and over-approximations. For this we use the `dReach`[12, 8] tool to do (standard) qualitative bounded reachability analysis in hybrid systems.

Related Work. There are several tools for computing probabilistic reachability in stochastic hybrid systems. The `SiSAT` tool [5] solves probabilistic bounded reachability in a numerically guaranteed manner, but it does not currently support continuous random parameters. Its extension [2] handles continuous parameters through sampling from the continuous state space providing statistical guarantees, while `ProbReach` gives formal guarantees. The tool `FAUST`^{2} [19] uses abstraction to verify nondeterministic continuous-state Markov models, although currently for discrete-time models only. `ProHVer` computes an upper bound for the maximal reachability probability [21], and handles continuous random parameters via discrete overapproximation [4], while `ProbReach` calculates *both* bounds of the entire probability interval.

Tools such as `UPPAAL` [14] and `PRISM` [13] are powerful model checkers for probabilistic timed automata. `UPPAAL` employs a statistical model checking approach for computing probability values in nonlinear hybrid systems while `PRISM` utilises symbolic model checking for probabilistic timed automata.

Also, several approaches for solving probabilistic reachability have been recently presented. In [3] the authors present a technique using validated ODE solver for computing p-boxes in dynamic nonlinear systems (not hybrid) with finite-support random parameters. A wide class of hybrid systems with continuous nondeterministic parameters are considered in [1, 20, 16]. However, they handle continuous state through finite discretisation providing approximated numerical solutions. `ProbReach` instead works with continuous time and space giving full mathematical/numeric guarantees.

2 Methodology

In this section we give an overview of the technical details of the main algorithm and of the tool implementation.

2.1 Verified integration

We consider continuous random parameters defined by their probability density function (pdf). In order to calculate a bounded reachability probability value for systems involving random variables, we need to integrate their pdf over the Borel set B containing all the parameter values for which the system reaches the unsafe region. We will tackle the problem of computing set B later in this Section. Here, we describe the procedure used to compute the probability value. The integration problem we need to solve is calculating the value of a definite integral:

$$\mathcal{I}([a,b]) = \int_a^b f(x)dx$$

up to a desired precision ϵ. In other words, we need to obtain an interval $[\mathcal{I}]([a,b])$ around the integral value such that $|[\mathcal{I}]([a,b])| \leq \epsilon$. Any known technique for calculating exactly a

definite integral can be utilised, and in `ProbReach` we use the 1/3 Simpson rule. We use an interval extension of the integrand function to obtain an interval version of the Simpson rule [6]. Considering the integration error presented by the fourth derivative, we can obtain an enclosure containing the *exact* value of the integral. The interval formula is given below:

$$\mathcal{I}([a,b]) \in [\mathcal{I}]([a,b]) = \frac{b-a}{6}([f](a) + 4[f](\frac{a+b}{2}))+ \\ [f](b)) - \frac{(b-a)^5}{2880}[f]^{(4)}([a,b])$$

where $[\mathcal{I}]$ and $[f]$ are interval extensions of functions $\mathcal{I}$ and f which is assumed to satisfy the required integrability and differentiability conditions. This formula by itself does not imply that $|[\mathcal{I}]([a,b])| \leq \epsilon$. In order to deal with this problem we finitely partition the $[a,b]$ interval. Then by the definition of integral:

$$\mathcal{I}([a,b]) \in \Sigma_{i=1}^{n}[\mathcal{I}]([x]_i)$$

where n is a number of intervals $[x]_i$'s that partition $[a,b]$. The partition should be obtained in such a way that for each $[x]_i$ the following holds:

$$|[\mathcal{I}]([x]_i)| \leq \frac{|[x]_i|}{b-a}\epsilon.$$

This is sufficient to guarantee that the original integral $\mathcal{I}([a,b])$ is computed with precision ϵ.

2.2 Decision procedure

A decision procedure is used to determine the set B needed to compute validated integration. A key part is encoding bounded reachability in hybrid systems as a first-order logic formula, which can then solved by the δ-complete decision procedure `dReach` [7, 8] using the notion of δ-weakening of a logical formula.

The intuition behind our approach is to perform an evaluation of a weaker (decidable) formula and on that basis make a conclusion about the initial formula. Given an arbitrary first order bounded formula, a δ-complete procedure returns **unsat** if the formula is false and δ-**sat** if its weakening is true. Hence, unlike **unsat**, δ-**sat** is a *weak* answer as it does not imply the satisfiability of the original formula. We use this fact to define our decision procedure. Let us consider two formulas ϕ and ϕ^C, defined as follows:

- $\phi([x])$ is **true** iff $\exists x' \in [x] : x' \in B$
- $\phi^C([x])$ is **true** iff $\exists x' \in [x] : x' \notin B$.

Note that $\phi^C([x])$ is not a logical negation of the formula $\phi([x])$ as $\phi^C([x]) \not\Rightarrow \neg\phi([x])$ and $\phi([x]) \not\Rightarrow \neg\phi^C([x])$

Verifying now both formulas using `dReach` on each interval $[x]$ in the partition obtained by verified integration, we obtain four outcomes which can be interpreted as follows:

- $\phi([x])$ is **unsat** — all points of $[x]$ are outside the Borel set B *for sure*, and $[x]$ is used for calculating P_{upper} (probability overapproximation). The integral of the probability density over the interval $[x]$ is subtracted from P_{upper}, which is initially set to 1.
- $\phi([x])$ is δ-**sat** — there is a value in the interval $[x]$ such that the system reaches the unsafe region or its δ-weakening.

- $\phi^C([x])$ is **unsat** — the entire $[x]$ lies in set B *for sure*, and $[x]$ is used for calculating P_{lower} (probability underapproximation). The integral of the probability density over the interval $[x]$ is added to P_{under} which is initially set to 0.
- $\phi^C([x])$ is δ**-sat** – there is a value in the interval $[x]$ such that the system stays outside the unsafe region or its weakening within the k-th step.

As it was mentioned above, only **unsat** returned for either of the formulas guarantees the correctness of the interval validation, and therefore can be used to refine the probability interval. Hence, if both formulas are δ**-sat** then either a *false alarm* is obtained (when a formula which should be unsatisfiable is verified as δ-sat because of a relatively large value of δ used) or the analysed interval is *mixed* (*i.e.*, some points in $[r]$ belong to set B and some others do not) which means that the interval should be partitioned and verified again. Extra partitioning can be performed arbitrarily many times as it does not alter the correctness of the result and, in fact, it is necessary to provide the described guarantees. Moreover, this is unavoidable in general due to undecidability of bounded reachability in hybrid systems. The refinement of the probability interval continues until its length is smaller than or equal to ϵ.

2.3 Random variables with unbounded support

Several useful random variables (RVs) are defined over unbounded intervals (*e.g.*, normal distribution, exponential distribution). It was shown above how to deal with bounded RVs. In case of unbounded RVs we are making a trade-off. The main idea behind this is that given a desired length ϵ of the enclosure we choose a value $k \in (0, 1)$ to obtain an interval $[a, b]$ such that:

$$\int_a^b f(x)\,dx > 1 - k\epsilon.$$

The following is immediately true by the fact that the integral of a probability density function on interval $(-\infty, \infty)$ is equal to 1.

$$\int_{-\infty}^{a} f(x)\,dx + \int_b^{\infty} f(x)\,dx \leq k\epsilon.$$

Hence, even if all the values from $(-\infty, a] \cup [b, \infty)$ are in the set B, the size of the probability interval will at most increase by $k\epsilon$ without affecting the desired precision. Then, if the evaluation routine described above terminates when the length of the interval $[P_{lower}, P_{upper}]$ is shorter than $(1-k)\epsilon$, the probability is guaranteed to be contained inside an interval of length $k\epsilon + (1-k)\epsilon = \epsilon$. The pseudo-code of the algorithm implemented in *ProbReach* is presented in Algorithm 1.

It is also worth mentioning that at *any* point in time during the computation, the *exact* value of the reachability probability belongs to the interval $[P_{lower}, P_{upper}]$, which is output by `ProbReach` when the interval bounds change. This might be advantageous for time-critical verification scenarios, as the user can specify a computation timeout. Thus, despite the fact that the desired precision might not be achievable within the specified timeframe, the obtained result is still complete in the sense that the *exact* probability value is *guaranteed* to be inside the computed interval.

2.4 Multiple continuous random variables

It is clear how to handle SHS with one initial random parameter. Introducing multiple random continuous parameters does not affect the decision procedure. The only changes are needed for

Algorithm 1: *ProbReach* (one continuous random parameter)

Input : probability density f, $k \in (0,1) \cap \mathbb{Q}$, $\epsilon \in (0,1] \cap \mathbb{Q}$, formula ϕ, ϕ^C
Output: interval $[I]$: $\int_B f \in [I]$ and $width([I]) \leq \epsilon$

$\epsilon_{inf} = k\epsilon$
$\epsilon_{prob} = (1-k)\epsilon$
$[a,b] = bounds(f, \epsilon_{inf})$ {obtain truncation for unbounded RVs }
$B.push(integral(f, [a,b], \epsilon_{prob}))$ {get domain partition by validated integration}
$[P_{lower}] = [0.0, 0.0]$ {interval for under-approximation}
$[P_{upper}] = [1.0, 1.0]$ {interval for over-approximation}
while $\overline{[P_{upper}]} - \underline{[P_{lower}]} > \epsilon_{prob}$ **do**
  $D = \emptyset$ {a stack to store further interval partitions}
  while $size(B) > 0$ **do**
    $\{[x], [S]([x])\} = B.\text{pop}()$ {get an interval}
    if $\phi([x])$ == δ-*sat* **then** {call dReach to evaluate ϕ}
      if $\phi^C([x])$ == δ-*sat* **then** {call dReach to evaluate ϕ^C}
        // it might be a mixed interval - we need to split it
        D.push($\{[\underline{x}, mid([x])], [S([\underline{x}, mid([x])])]\}$)
        D.push($\{[mid([x]), \overline{x}], [S([mid([x]), \overline{x})]\}$)
      else $[P_{lower}] = [P_{lower}] + [S]([x])$ {increase under-approximation}
    else $[P_{upper}] = [P_{upper}] - [S]([x])$ {descrease over-approximation}
  $B = D$
$[P_{upper}] = [P_{upper}] + 1 - \int_a^b f(x)\,dx$ {add leftovers from the unbounded domain}
return $[\underline{[P_{lower}]}, \overline{[P_{upper}]}]$

obtaining a finer partition of each RV's domain. In particular, it is sufficient to compute the Cartesian product of the partitions of each random variable. Those are obtained by applying the validated integration procedure to each random parameter. The formula below shows which precision value ϵ should be used for integrating each random variable in order to guarantee an overall precision ϵ_{prod}.

Given a stochastic hybrid system with n independent continuous random initial parameters and a desired size of the probability interval $\epsilon_{prod} \in (0,1] \cap \mathbb{Q}$ it is sufficient to integrate each random variable with precision ϵ satisfying the formula below:

$$\epsilon_{prod} \geq \binom{n}{1}\epsilon + \binom{n}{2}\epsilon^2 + ... + \binom{n}{i}\epsilon^i + ... + \binom{n}{n-1}\epsilon^{n-1} + \binom{n}{n}\epsilon^n \tag{1}$$

where $\binom{n}{i}$ is the binomial coefficient. The decision procedure remains unchanged and it is applied to each *box* from the Cartesian product. In case a box is evaluated as *mixed*, then all its dimensions are bisected forming 2^n boxes, which are verified in the same manner. The algorithm stops upon reaching length ϵ_{prod} for the probability interval.

2.5 Implementation

`ProbReach` is implemented in C++. It employs the `CAPD`[1] library to compute interval extensions of the probability density function and its derivative in the verified integration procedure,

[1] http://capd.ii.uj.edu.pl/

and to perform computation of the probability interval. Also, `ProbReach` uses the `IBEX`[2] library to solve formula (1). `ProbReach` is parallelised using `OpenMP` to increase its performance. `ProbReach` utilises `dReal` [8] and `dReach` [12] as standalone applications in *plug-and-play* manner to solve standard (*i.e.*, non-probabilistic) bounded reachability. `ProbReach` source code and static binaries are publicly available at https://github.com/dreal/probreach/. More details on the `ProbReach` implementation are available in [18].

3 Experiments

We have applied `ProbReach` to a number of nonlinear hybrid system models. Below we introduce the two most complex of them: an insulin-glucose regulatory system model and personalized prostate cancer therapy model. The results of all experiments were validated by Monte Carlo simulation and confidence intervals with the Chernoff-Hoeffding bound [9].

Insulin-glucose regulatory model. This model represents an insulin-glucose regulatory system for patients with type-1 diabetes. `ProbReach` was applied to the model introduced in [17] based on Hovorka's glucoregulatory model [10]. The described system is rather sophisticated. It considers parameters such as the amount of carbohydrates consumed with food and their glycemic index. The glucose level is constantly monitored and upon reaching a defined upper threshold the insulin pump starts working. The pump stops when the glucose level goes below the lower threshold. In order to give an idea about the model complexity, we present the system of ODEs governing the system dynamics:

$$
\begin{aligned}
\frac{dQ_1}{dt} &= -F_{01}^c - x_1 Q_1 + k_{12} Q_2 - F_R + EGP_0(1 - x_3) + 0.18UG \\
\frac{dQ_2}{dt} &= x_1 Q_1 - (k_{12} + x_2) Q_2 \\
\frac{dS_1}{dt} &= u - \frac{S_1}{t_{maxI}} \\
\frac{dS_2}{dt} &= \frac{S_1 - S_2}{t_{maxI}} \\
\frac{dI}{dt} &= \frac{S_2}{t_{maxI} V_I} - k_e I \\
\frac{dx_1}{dt} &= -k_{a1} x_1 + k_{b1} I \\
\frac{dx_2}{dt} &= -k_{a2} x_2 + k_{b2} I \\
\frac{dx_3}{dt} &= -k_{a3} x_3 + k_{b3} I \\
F_{01}^c &= \frac{F_{01} G}{0.85(G + 1)} \\
G &= \frac{Q_1}{V_G}
\end{aligned}
$$

Randomising some parameters we can calculate the probability that glucose level (G) will get back to normal ($G \leq 10$) within some time from the start of insulin infusion. We conducted the experiment with one continuous random initial parameter ($x_3(0)$, normally distributed) which took about 100 hours to compute ($\epsilon = 0.0001$) and returned the *numerically guaranteed* interval

[2] http://www.ibex-lib.org/

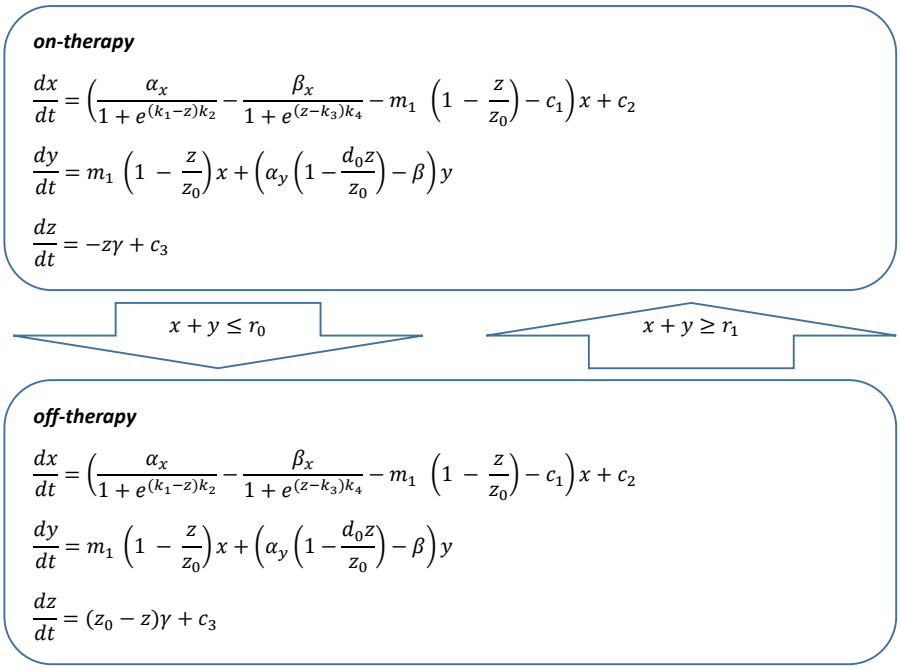

Figure 1: Personalized prostate cancer therapy model

[0.999657, 0.999712]. The confidence interval obtained by Monte Carlo simulation was [0.99706, 1], with coverage probability 0.99 and width 0.005.

Personalized prostate cancer therapy. We consider a model of personalised prostate cancer therapy introduced by Ideta *et al.* [11] and improved by Liu *et al.* [15]. The patient's prostate-specific antigen (PSA) level is monitored throughout the therapy. When the PSA level reaches an upper threshold, the patient starts receiving treatment (*on-therapy* stage) until the PSA level decreases to a lower threshold (*off-therapy*). The main aim of the therapy is to delay cancer relapse for as long as possible. The model of the therapy is given in Figure 1 (a full explanation of the model and its parameters can be found in [15]). Mode 1 is the *on-therapy* stage, and it continues until the PSA level (measured by $x + y$ in Fig.(1)) is above threshold r_0. Then the system makes a transition to the *off-therapy* mode which continues until the PSA level is below r_1. We explored the scenario where one of the model continuous parameters was normally distributed. We computed the probability of cancer relapse (*i.e.*, $y \geq 1$) within 100 days from the start of the therapy. The computation took about 15 minutes ($\epsilon = 0.001$) and returned the guaranteed interval [0.47380981, 0.47441201]. The confidence interval obtained by Monte Carlo simulation was [0.4648111, 0.4848111], with coverage probability 0.99 and interval width 0.02.

4 Recent Work

We have recently added to `ProbReach` support for discrete random parameters and nondeterministic continuous parameters. The user can specify any combinations of random/nondeterministic parameters. When nondeterministic parameters are present in the model, the precision ϵ controls the size of the minimum parameter box examined by `ProbReach`. Also, for models

with discrete random parameters only, ϵ is not considered at all.

We are currently re-running all our experiments with the latest `dReal3`. From the first results we are observing a speed-up of at least 50% with respect to the CPU times reported in [18]. This is due to the fact that validated ODE solving is a major source of computational complexity in our approach. Also, the complexity depends on the number of parameters in the model, since in the worst case the number of boxes to examine increases exponentially with the number of parameters. We remark that both sources of complexity are in general unavoidable, as it happens similarly with the state explosion problem experienced by model checking.

5 Conclusions

We have presented a summary of our recent work on probabilistic bounded reachability for stochastic hybrid systems with random and nondeterministic parameters. We have implemented our technique in the open source tool `ProbReach`. Two key features of our approach are: we compute numerically guaranteed enclosures, and we can handle hybrid systems with continuous dynamics described by any Lipschitz-continuous ODEs.

6 Acknowledgements

This work has been supported by award N00014-13-1-0090 of the US Office of Naval Research.

References

[1] Alessandro Abate, Joost-Pieter Katoen, John Lygeros, and Maria Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6):624 – 641, 2010.

[2] Christian Ellen, Sebastian Gerwinn, and Martin Fränzle. Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains. *International Journal on Software Tools for Technology Transfer (STTT)*, 2014. *to appear.*

[3] Joshua A. Enszer and Mark A. Stadtherr. Verified solution and propagation of uncertainty in physiological models. *Reliable Computing*, 15:168–178, 2010.

[4] Martin Fränzle, Ernst Moritz Hahn, Holger Hermanns, Nicolás Wolovick, and Lijun Zhang. Measurability and safety verification for stochastic hybrid systems. In *HSCC*, pages 43–52, 2011.

[5] Martin Fränzle, Tino Teige, and Andreas Eggers. Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. *J. Log. Algebr. Program.*, 79(7):436–466, 2010.

[6] Sérgio Galdino. Interval integration revisited. *Open Journal of Applied Sciences*, 2(4B):108–111, 2012.

[7] Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. Delta-complete decision procedures for satisfiability over the reals. In *IJCAR*, pages 286–300, 2012.

[8] Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *CADE-24*, volume 7898 of *LNCS*, pages 208–214, 2013.

[9] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58(301):13–30, 1963.

[10] Roman Hovorka et al. Partitioning glucose distribution/transport, disposal, and endogenous production during IVGTT. *American Journal of Physiology: Endocrinology and Metabolism*, 282(5):E992 – E1007, 2002.

[11] Aiko Miyamura Ideta, Gouhei Tanaka, Takumi Takeuchi, and Kazuyuki Aihara. A mathematical model of intermittent androgen suppression for prostate cancer. *Journal of Nonlinear Science*, 18(6):593–614, 2008.

[12] Soonho Kong, Sicun Gao, Wei Chen, and Edmund M. Clarke. dReach: Delta-reachability analysis for hybrid systems. In *TACAS*, 2015. *to appear.*

[13] Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV*, volume 6806 of *LNCS*, pages 585–591, 2011.

[14] Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, 1:134–152, 1997.

[15] Bing Liu, Soonho Kong, Sicun Gao, Paolo Zuliani, and Edmund M. Clarke. Towards personalized cancer therapy using delta-reachability analysis. In *HSCC*, pages 227–232. ACM, 2015.

[16] Federico Ramponi, Debasish Chatterjee, Sean Summers, and John Lygeros. On the connections between PCTL and dynamic programming. In *HSCC*, pages 253–262. ACM, 2010.

[17] Sriram Sankaranarayanan and Georgios Fainekos. Simulating insulin infusion pump risks by in-silico modeling of the insulin-glucose regulatory system. In *CMSB*, volume 7605 of *LNCS*, pages 322–341, 2012.

[18] Fedor Shmarov and Paolo Zuliani. ProbReach: Verified probabilistic δ-reachability for stochastic hybrid systems. In *HSCC*, pages 134–139. ACM, 2015.

[19] Sadegh Esmaeil Zadeh Soudjani, C. Gevaerts, and Alessandro Abate. FAUST^2: Formal abstractions of uncountable-state stochastic processes. In *TACAS*, 2015. *to appear.*

[20] Ilya Tkachev and Alessandro Abate. Formula-free finite abstractions for linear temporal verification of stochastic hybrid systems. In *HSCC*, pages 283–292. ACM, 2013.

[21] Lijun Zhang, Zhikun She, Stefan Ratschan, Holger Hermanns, and Ernst Moritz Hahn. Safety verification for probabilistic hybrid systems. In *CAV*, volume 6174 of *LNCS*, pages 196–211, 2010.