EPiC Series in Computer Science

Volume 37, 2015, Pages 2–10

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

Verification of Stochastic Systems by Stochastic Satisfiability Modulo Theories with Continuous Domain*

Yang Gao[1] and Martin Fränzle[2]

[1] FK II, Department Informatik, Universität Oldenburg, Germany
yang.gao@uni-oldenburg.de
[2] FK II, Department Informatik, Universität Oldenburg, Germany
fraenzle@informatik.uni-oldenburg.de

Abstract

Stochastic Satisfiability Modulo Theories (SSMT) [1] is a quantitative extension of classical Satisfiability Modulo Theories (SMT) inspired by stochastic logics. It extends SMT by the usual as well as randomized quantifiers, facilitating capture of stochastic game properties in the logic, like reachability analysis of hybrid-state Markov decision processes. Solving for SSMT formulae with quantification over finite and thus discrete domain has been addressed by Tino Teige et al. [2]. In our work, we extend their work to SSMT over continuous quantifier domains (CSSMT) in order to enable capture of, e.g., continuous disturbances and uncertainty in hybrid systems. We extend the semantics of SSMT and introduce a corresponding solving procedure. A discussion regarding to reachability analysis is given to demonstrate applicability of our framework to reachability problems in hybrid systems.

1 Motivation and Definitions

The idea of modelling uncertainty using randomized quantification was first proposed within the framework of propositional satisfiability (SAT) by Papadimitriou, yielding Stochastic SAT (SSAT) featuring both classical quantifiers and randomized quantifiers [3]. This work has been lifted to Satisfiability Modulo Theories (SMT) by Fränzle, Teige et al. [1, 2] in order to symbolically reason about reachability problems of probabilistic hybrid automata (PHA). Instead of reporting true or false, an SSAT/SSMT formula Φ has a probability as semantics. A serious limitation of the SSMT-solving approach pioneered by Teige [4] is that all randomized quantifiers are confined to range over finite domains. As this implies that the carriers of probability distributions have to be finite, a large number of phenomena cannot be expressed within the current SSMT framework, such as continuous noise or measurement error in hybrid systems. To overcome this limitation, we relax the constraints on the domains of randomized variables so that continuous probability distributions are admitted.

*This research is funded by the German Research Foundation through the Research Training Group DFG-GRK 1765: "System Correctness under Adverse Conditions" (SCARE, scare.uni-oldenburg.de) and the Transregional Collaborative Research Center SFB-TR 14 "Automatic Verification and Analysis of Complex Systems" (AVACS, www.avacs.org).

Our approach is based on a combination of the DPLL($\mathcal{T}$) [5] and ICP (Interval Constraint Propagation, [6, 7]) algorithms, as first implemented in the iSAT solver for rich arithmetic SMT problems over the $\mathbb{R}^n$ [8, 9], and on branch-and-prune rules for the quantifiers generalizing those suggested in [8, 4]. We extend these methods so that they can deal with SSMT formulae with continuous quantifier domains. Our solving procedure therefore is divided into three layers: an SMT layer manipulating the Boolean structure of the "matrix"[1] of the formula, an interval constraint solving layer reasoning over the conjunctive constraint systems in the theory part of the formula, and a stochastic SMT layer reasoning about the quantifier prefix. Each layer is defined by a set of rules to generate, split, and combine so-called *computation cells*, where a computation cell is a box-shaped part of the $\mathbb{R}^n$, i.e., the problem domain of the constraints. The solver thereby approximates the exact satisfaction probability of the formula under investigation and terminates with a conclusive result whenever the approximation gets tight enough to conclusively answer the question whether the satisfaction probability is above or below a certain target specified.

Definition 1.1. An SSMT formula with continuous domain (CSSMT) is of the form: $\Phi = \mathcal{Q} : \varphi$, where:

- $\mathcal{Q} = Q_1 x_1 \in dom(x_1) \dots Q_n x_n \in dom(x_n)$ is a sequence of quantified variables, $dom(x_i)$ denotes the domain of variable x_i, which are intervals over the reals, Q_i is either an existential quantifier $\exists$ or a randomized quantifier Y_{π_i} with integrable probability density function over the reals π_i satisfying $\int_{dom(x_i)} \pi_i(x_i)dx_i = 1$.
- φ is an SMT formula over a quantifier-free non-linear arithmetic theory $\mathcal{T}$. Without loss of generality, we assume that φ is in conjunctive normal form (CNF), i.e., φ is a conjunction of clauses, and a clause is a disjuction of (atomic) arithmetic predicates. φ is also called the *matrix* of the formula.

Definition 1.2. The semantics of a CSSMT formula $\Phi = \mathcal{Q} : \varphi$ is defined by the maximum probability of satisfaction $Pr(\Phi)$ as follows, where ε denotes the empty quantifier prefix:

- $Pr(\varepsilon : \varphi) = 0$ if φ is unsatisfiable and $Pr(\varepsilon : \varphi) = 1$ if φ is satisfiable.
- $Pr(\exists x_i \in dom(x_i) \dots Q_n x_n \in dom(x_n) : \varphi)$
  $= \sup_{v \in dom(x_i)} Pr(Q_{i+1} x_{i+1} \in dom(x_{i+1}) \dots Q_n x_n \in dom(x_n) : \varphi[v/x_i])$.
- $Pr(\mathrm{Y}_{\pi_i} x_i \in dom(x_i) \dots Q_n x_n \in dom(x_n) : \varphi)$
  $= \int_{v \in dom(x_i)} Pr(Q_{i+1} x_{i+1} \in dom(x_{i+1}) \dots Q_n x_n \in dom(x_n) : \varphi[v/x_i])\pi_i(v)dv$.

Example 1.1. Fig. 1 constructs a tree according to the semantics of CSSMT formula, where $\mathcal{N}(0, 1)$ refers to the standard normal distribution. Semantically, Φ determines the maximum probability s.t. there are values for x which are between $[-1, 1]$ s.t. for normal distributed values of y the matrix is satisfiable. We branch the domain of x into three parts, and for each part, we branch the domain of y into two parts. Take the leftmost branch as an example, the matrix can not be satisfied and we mark the probability of satisfaction as 0. At last we propagate the probability according to the corresponding quantifiers, for example, y is normal distributed, so the probability when y takes value from $(-\infty, 0]$ is 0.5. If we combine the probability from bottom to top and choose maximum value among the three branches for x (since x is bounded by existential quantifier), then we get that the probability of satisfaction of Φ is 1.

2 Solving Procedure for CSSMT

The problem we consider is formalised as follow:

[1] In SSAT parlance, this is the body of the formula after rewriting it to prenex form and stripping all the quantifiers.

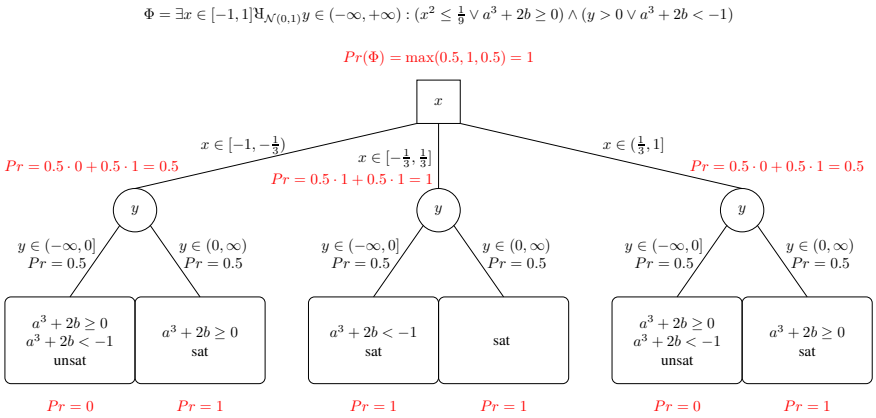


Figure 1: Semantics of a CSSMT formula depicted as a tree.

Given a CSSMT formula $\Phi = \mathcal{Q} : \phi$, a reference probability δ, and an accuracy ε, the solving procedure shall return

- *"GE", if $Pr(\Phi)$ is greater than or equal to $\delta + \varepsilon$;*
- *"LE", if $Pr(\Phi)$ is less than or equal to $\delta - \varepsilon$;*
- *"GE" or "Inconclusive", if $Pr(\Phi) \in [\delta, \delta + \varepsilon]$;*
- *"LE" or "Inconclusive", if $Pr(\Phi) \in [\delta - \varepsilon, \delta]$.*

In order to do so, the algorithm is equipped with the following structures:

- C: a set collecting the constraints which must be satisfied in the current phase.
- ρ: an ordered list (corresponding to the order of variables in $\mathcal{Q}$) which records the interval valuation for each variable.
- H: a set of computation cells. Intuitively, a computation cell is a convex "box" attached with a probability estimation.

The algorithm will start its deduction sequence, which is given by the DPLL rules at the outermost level, which in turn builds on the rules at the constraint solving and the SSMT layer. All the rules share the same structure: the manipulations are based on the set H which contains the computation cells. When the cells meet the premises, they will update, split or combine according to the conclusions.

2.1 SMT Level.

Rule (INI) adds the first computation cell to H, which contains: 1) the formula $\mathcal{Q} : \phi$ to be decided; 2) ρ is an initial evaluation for each variable; 3) the constraints C which must be satisfied, initially an empty set; 4) a superscript $(p, q)_i = (0, 1)_1$ over-approximating the satisfaction probability of the remaining formula when chopping off the quantifier prefix before variable x_i. For the Rule (INI), $(0, 1)_1$ means that no quantifiers have been resolved at the moment and the lower- and upper-estimation are 0 and 1 respectively.

$$\frac{}{H \rightarrow H \cup \{(Q : \phi, \rho, \emptyset)^{(0,1)_1}\}} \quad \text{(INI)}$$

If all the disjuncts except one (l') in some clause ($L \vee l'$) can not be satisfied w.r.t. the current evaluation ρ, then this remaining "unit" must hold (added to C). Rule (UP) corresponds to the unit

propagation in the DPLL framework.

$$\frac{(L \vee l') \in \phi, \rho \not\models L}{H' \cup \{(Q:\phi,\rho,C)^{(p,q)_i}\} \to H' \cup \{(Q:\phi,\rho,C\cdot\langle l'\rangle)^{(p,q)_i}\}} \quad \text{(UP)}$$

If the range of a variable x_j can be narrowed according to the constraints C and the current evaluation ρ by means of ICP (Interval constraint propagation, [6, 7]), and if ρ is not yet hull consistent w.r.t. to the new bound (using the notation $\not\models_{hc}$, intuitively, hull consistency means no interval narrowing can be further performed by using ICP), we update the evaluation set and the probability estimation according to the narrowing $\rho \overset{C}{\rightsquigarrow} (x_j \sim b)$ of x_j computed by ICP:

$$\frac{\rho \overset{C}{\rightsquigarrow} (x_j \sim b), \rho \not\models_{hc} (x_j \sim b)}{H' \cup \{(Q:\Phi,\rho,C)^{(p,q)_i}\} \to H' \cup \{(Q:\Phi, update_\rho(x_j \sim b), C)^{renewal_{\rho_j}(p,q)_i}\}} \quad \text{(ICP)}$$

where

$$update_\rho(x_j \sim b)(x_i) = \begin{cases} \rho(x_j) \cap \{z | z \sim b\}, & if\ x_i = x_j \\ \rho(x_j), & otherwise \end{cases}$$

Intuitively, the *update* operator narrows the bound of variable x_j and leaves other variables unchanged. The corresponding change in the probability estimate induced by narrowing a —potentially randomized— variable x_j is reflected by

$$renewal_{\rho_j}(p,q)_i = \begin{cases} (p,q)_i, & if\ x_j \prec x_i \\ \mathbb{P}(\rho(x_i) \times \cdots \times \rho(x_j) \cap \{z | z \sim b\} \times \cdots \rho(x_n))_i, & otherwise \end{cases}$$

where $\prec$ corresponds to the order of variables appearing in Q, $\mathbb{P}(I_i \times \cdots \times I_n)$ is a safe, interval-arithmetic based probability estimation which returns an interval over-approximating the measure of $I_i \times \cdots \times I_n$ under the distributions attached to the quantifiers.

When both rule (ICP) and rule (UP) do not yield further deductions, we say ϕ is inconclusive on ρ. We may then perform the splitting rule (SPL) to split the current computation cell into two cells (in practice, this can be achieved by splitting from middle point) and update ρ as well as the probability estimation accordingly.

$$\frac{\rho_j \neq \emptyset, \rho_j^1 \cup \rho_j^2 = \rho_j}{H' \cup \{(Q:\phi,\rho,C)^{(p,q)_i}\} \to H' \cup \{(Q:\phi,\rho'\cdot\langle\rho_j^1\rangle\cdot\rho'',C)^{renewal_{\rho_j^1}(p,q)_j}, (Q:\phi,\rho'\cdot\langle\rho_j^2\rangle\cdot\rho'',C)^{renewal_{\rho_j^2}(p,q)_j}\}} \quad \text{(SPL)}$$

2.2 Constraint Solving Level.

When a conflict is obtained, i.e. if ICP under the current evaluation ρ and constraints C narrows some variables to empty sets, or if ρ violates every part in one clause, the current computation cell can be safely marked with probability 0. This is reflected by rule (CFL):

$$\frac{\rho \overset{C}{\rightsquigarrow} (x_i = \emptyset) \text{ or } L \in \phi \wedge \rho \not\models L}{H' \cup \{(Q:\phi,\rho,C)^{(p,q)_i}\} \to H' \cup \{(Q:\phi,\rho,C)^{(0,0)_n}\}} \quad \text{(CFL)}$$

If the current evaluation ρ is hull consistent w.r.t. the actual constraint set C, a paving procedure [10] can be invoked to generate an inner approximation and an outer approximation of the actual solution set by sets of boxes (i.e. $\{(\cdot)\}^*$ means number of cells). By computing safe upper (lower, resp.) approximations on the probability measures of the outer (inner, resp.) approximations of the solution

sets, we obtain a safe interval estimate on the satisfaction probability. Rule (CNSIS) assigns these.

$$\frac{\rho \vDash_{hc} C}{H' \cup \{(Q:\phi,\rho,C)^{(p,q)_i}\} \to H' \cup \{(Q:\phi,\rho',C)^{(p',q')_n}\}^*} \quad \text{(CNSIS)}$$

2.3 Stochastic SMT Level.

Two computation cells can be *combinative* in that they estimate satisfaction probability w.r.t. adjacent intervals for the same variable x_i. In case that x_i is bound by $\exists$, combining the two cells yields the maximum probability (Rule ($\exists$-COM)); otherwise if bound by $\mathfrak{U}$, the two cells can be combined by adding their probabilities (Rule $\mathfrak{U}$-COM).

$$\frac{\rho_i^1 \uplus \rho_i^2 \text{ is the interval hull of } \rho_i^1 \text{ and } \rho_i^2}{\begin{array}{r} H' \cup \{(Q' \exists x_i Q'' : \phi, \rho' \cdot \langle \rho_i^1 \rangle \cdot \rho'', C)^{(p_1,q_1)_i}, (Q' \exists x_i Q'' : \phi, \rho' \cdot \langle \rho_i^2 \rangle \cdot \rho'', C)^{(p_2,q_2)_i}\} \to \\ H' \cup \{(Q : \phi, \rho' \cdot \langle \rho_i^1 \uplus \rho_i^2 \rangle \cdot \rho'', C)^{max((p_1,q_1)_i,(p_2,q_2)_i)}\} \end{array}} \quad (\exists\text{-COM})$$

$$\frac{\rho_i^1 \uplus \rho_i^2 \text{ is the interval hull of } \rho_i^1 \text{ and } \rho_i^2}{\begin{array}{r} H' \cup \{(Q' \mathfrak{U} x_i Q'' : \phi, \rho' \cdot \langle \rho_i^1 \rangle \cdot \rho'', C)^{(p_1,q_1)_i}, (Q' \mathfrak{U} x_i Q'' : \phi, \rho' \cdot \langle \rho_i^2 \rangle \cdot \rho'', C)^{(p_2,q_2)_i}\} \to \\ H' \cup \{(Q : \phi, \rho' \cdot \langle \rho_i^1 \uplus \rho_i^2 \rangle \cdot \rho'', C)^{(p_1,q_1)_i + (p_2,q_2)_i}\} \end{array}} \quad (\mathfrak{U}\text{-COM})$$

where the *interval hull* of two sets I_1 and I_2 here is the smallest interval which contains I_1 and I_2.

If all the computation cells w.r.t. the same variable have been tackled, the probability should be propagated to the preceding variable in the variable order. Rule (LFT) checks all the computation cells in H, and will propagate if all its siblings have been combined.

$$\frac{\forall (Q:\phi,\rho',C')^{(\cdot,\cdot)_j} \in H' : j \neq i}{H' \cup \{(Q:\phi,\rho,C)^{(p,q)_i}\} \to H' \cup \{(Q:\phi,\rho,C)^{(p,q)_{i-1}}\}} \quad \text{(LFT)}$$

2.4 Termination.

Whenever the estimated probability interval at the level of the first variable x_1 becomes less and equal than the reference probability δ, the original formula is concluded to satisfy $P(\Phi) \leq \delta$. Rule (LE) then reports "LE"; rule (GE) does the equivalent for the converse case.

$$\frac{q \leq \delta}{H' \cup \{(Q:\phi,\rho,C)^{(p,q)_1}\} \to \text{LE}} \quad \text{(LE)}$$

$$\frac{p \geq \delta}{H' \cup \{(Q:\phi,\rho,C)^{(p,q)_1}\} \to \text{GE}} \quad \text{(GE)}$$

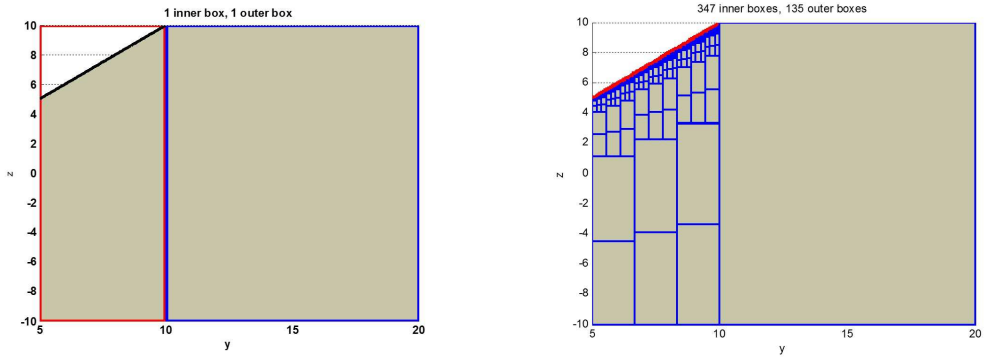
If the above two cases cannot be be judged under the accuracy ε, the evaluation of the formula remains inconclusive w.r.t. δ :

$$\frac{q > \delta \wedge p < \delta \wedge |p-q| < \varepsilon}{H' \cup \{(Q:\phi,\rho,C)^{(p,q)_1}\} \to \text{INCON}} \quad \text{(INCON)}$$

Whenever none of the above three termination rules applies, we have to go back to the SMT level and generate more cells by (SPL).

Example 2.1. Consider the CSSMT formula $\Phi = \exists x \in [-10,10]\, \mathfrak{U} y \in \mathcal{U}[5,25]\, \mathfrak{U} z \in \mathcal{U}[-10,10] : (x > 3 \vee y < 1) \wedge (z > x^2 + 2 \vee y \leq 20) \wedge (x^2 > 49 \vee y > 7x) \wedge (x < 6 \vee y \geq z)$, where y and z are uniformly distributed with range $[5,25]$ and $[-10,10]$ correspondingly. The initial configurations are $C = \emptyset$, $H = \emptyset$ and $\rho = ([-10,10],[5,25],[-10,10])$, we set $\delta = 0.45$ to be the reference probability.

By applying Rule (INI), we add the first computation cell $(\Phi, ([-10, 10], [5, 25], [-10, 10]), \emptyset)^{(0,1)_1}$ to the set H. According to the Rule (UP), the formula $x > 3$ is added to C as a constraint which must be satisfied. Interval constraint propagation is then performed so that the domain of x is narrowed, which yields proof state $(\Phi, ((3, 10], [5, 25], [-10, 10]), \{x > 3\})^{(0,1)_1}$. The current evaluation makes $z > x^2 + 1$ unsatisfiable, so $y \leq 20$ will be added to C (Rule (UP)), the domain of y is then narrowed to $[5, 20]$ (Rule (ICP)), since y is bounded by $\mathfrak{U}$, we need update the probability estimation, this yields $(\Phi, ((3, 10], [5, 20], [-10, 10]), \{x > 3, y \leq 20\})^{(0,0.75)_1}$, we cannot guarantee that there are solutions in $[5, 20]$, so the lower bound is 0, for the upper bound we can conclude that it will not exceed 0.75 since y is uniformly distributed and only the values in $[5, 20]$ will be considered. The next step is to apply the rule (SPL). We choose x and split its interval into two parts, giving $H = \{(\Phi, ((3, 7), [5, 20], [-10, 10]), \{x > 3, y \leq 20\})^{(0,0.75)_1}, (\Phi, ([7, 10], [5, 20], [-10, 10]), \{x > 3, y \leq 20\})^{(0,0.75)_1}\}$. Since the evaluation violates the clause $x^2 > 49 \vee y > 7x$, the first computation cell is marked with probability 0 according to (CFL). For the second cell, by performing rule (UP) we get $(\Phi, ([7, 10], [5, 20], [-10, 10]), \{x > 3, y \leq 20, x^2 > 49, y \geq z\})^{(0,0.75)_1}$. Now the constraints C is hull consistent w.r.t. the current evaluation ρ. For the sake of demonstration, we generate one inner box and one outer box manually in this illustrating example (in practice, we employ RealPaver for this task.), i.e., $(\Phi, ([7, 10], [5, 10], [-10, 10]), \{x > 3, y \leq 20, x^2 > 49, y \geq z\})^{(0,0.33*0.75)_3}$ (all the points in the inner box are satisfied w.r.t. $y \geq z$, so we can estimate the real probability by lower- and upper-bounds which are close to the real one) and $(\Phi, ([7, 10], (10, 20], [-10, 10]), \{x > 3, y \leq 20, x^2 > 49, y \geq z\})^{(0.66*0.75,0.67*0.75)_3}$ (the outer box contains both solutions and non-solutions, so the lower bound has to be assigned to 0 and upper bound be the maximum), which over and under approximate the solutions for C w.r.t. ρ respectively. As has been depicted in Fig. 2-(a), a light gray area is shown, where the formula Φ is satisfiable. The red box is the corresponding outer box and blue is an inner.



(a) 1 inner box and 1 outer box.

(b) 347 inner boxes and 135 outer boxes.

Figure 2: Inner and outer approximations for constraint solving problem: $\{x > 3, y \leq 20, x^2 > 49, y \geq z\}$ where $x \in [7, 10]$, $y \in [5, 20]$ and $z \in [-10, 10]$.

Now we have three cells and try to propagate the probability, as depicted in Fig. 3.

The given δ for this running example is 0.45, according to the Rule (GE), we know that $Pr(\Phi) > 0.45$. The decision procedure terminates here. Now let us consider a higher reference probability, i.e., $\delta = 0.70$, a tighter approximation can be achieved by generating more boxes. As shown in Fig. 2-(b), we use RealPaver [10], which is a modeling language implementing interval-based algorithms to process systems of nonlinear constraints over the real numbers, to generate the inner boxes and outer boxes so that a better result can be obtained. By doing so, we get a tighter approximation, which is $[0.7181, 0.7191]$.

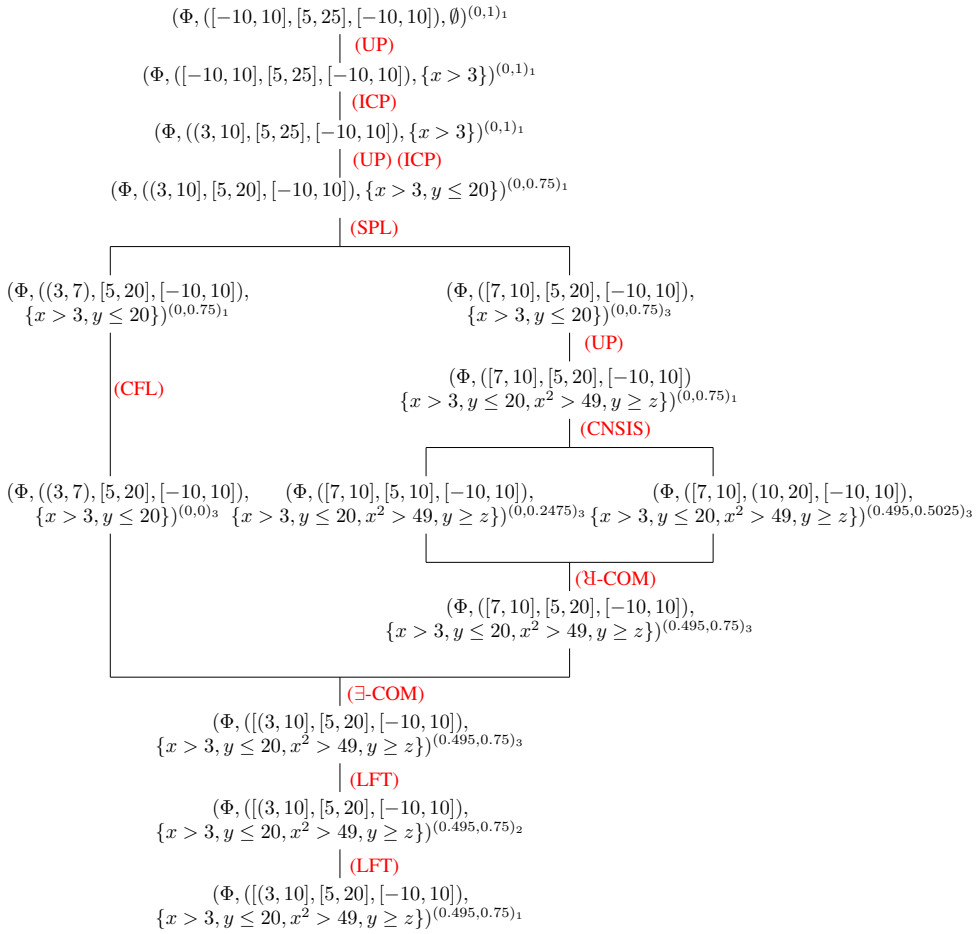


Figure 3: Solving procedure for Example 2.1

3 Reachability Analysis by Using CSSMT

CSSMT is capable to analyze the reachability properties for systems with stochastic behavior, e.g., temperature regulation problem which has been considered in [11] where it was modeled by discrete time stochastic hybrid systems (DTSHS) and investigated by using dynamic programming (DP). Instead, in our recent work [12] the CSSMT framework was adopted. The framework can be concluded as follows:

- Formalize the initial conditions ($\mathcal{I}$), transition relations ($\mathcal{T}$) and the goal ($\mathcal{G}$) the system should achieve, as the conjunction of constraints, i.e, $Q : \mathcal{I} \wedge \mathcal{T} \wedge \mathcal{G}$, where Q is a sequence of quantifiers. Due to the continuity and randomness of some variables, i.e., random delays regarding to the switching among different states, the disturbance and noise introduced by measurement and environment etc, the formula $Q : \mathcal{I} \wedge \mathcal{T} \wedge \mathcal{G}$ belongs to CSSMT;
- Perform the solving procedure so that we can obtain a probability estimation for $Pr(Q : \mathcal{I} \wedge \mathcal{T} \wedge \mathcal{G})$, which tells us how probable the system can reach the goal $\mathcal{G}$ starting from the states satisfying $\mathcal{I}$. The candidate intervals which lead to the maximum probability of satisfaction can be interpreted

as "optimal" decisions or configurations.

In [12], we considered the problem of regulating the temperature of a room during some time horizon $[0, N]$ by a thermostat that can switch a heater on or off. The goal of the regulation problem is to determine a control strategy that maximizes the probability that the average room temperature is driven close to a given temperature with an admissible tolerance. The idea is same as what we mentioned above, we translate the initial condition, transition relations and desired sets into a CSSMT formula $\mathcal{Q} : \Phi$, then perform the CSSMT solving steps to obtain the maximum probability of satisfaction w.r.t. the formula. At last we extract the values of control action from the branches which lead to the maximum probability. The implementation is done in MATLAB, however it is just a prototype implementation which can not be generalized to other case studies, a full CSSMT solver is expected to be implemented in order to support stochastic modeling so that a larger class of properties can be handled. The implementation is currently an on-going work, which may partially be based on iSAT/SiSAT [9, 13] and Realpaver [10] in order to obtain safe bounds. In this paper, we only discussed the solving procedure by using DPLL and ICP, yet conflict driven clause learning (CDCL) performs better due to to non-chronological backjumping and memorization of reasons for inconsistencies. CDCL will be considered in the tool implementation for efficiency. As has been mentioned, the solving procedure for CSSMT is handled by computation cells equipped with probability estimations. The cells can be combined at any time when combinative, this structure makes parallel computation possible, i.e., the SMT level and Stochastic level can be separated and handled in parallel.

References

[1] Martin Fränzle, Holger Hermanns, and Tino Teige. Stochastic satisfiability modulo theory: A novel technique for the analysis of probabilistic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 172–186. Springer, 2008.

[2] Tino Teige and Martin Fränzle. Stochastic satisfiability modulo theories for non-linear arithmetic. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 248–262. Springer, 2008.

[3] Christos H Papadimitriou. Games against nature. *Journal of Computer and System Sciences*, 31(2):288–301, 1985.

[4] Tino Teige. *Stochastic satisfiability modulo theories: a symbolic technique for the analysis of probabilistic hybrid systems*. PhD thesis, Universität Oldenburg, 2012.

[5] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving sat and sat modulo theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL($\mathcal{T}$). *Journal of the ACM (JACM)*, 53(6):937–977, 2006.

[6] Francesca Rossi, Peter Van Beek, and Toby Walsh. *Handbook of constraint programming*. Elsevier, 2006.

[7] Pascal Van Hentenryck, David McAllester, and Deepak Kapur. Solving polynomial systems using a branch and prune approach. *SIAM Journal on Numerical Analysis*, 34(2):797–827, 1997.

[8] Martin Fränzle, Christian Herde, Tino Teige, Stefan Ratschan, and Tobias Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *JSAT*, 1(3-4):209–236, 2007.

[9] iSAT Homepage. https://projects.avacs.org/projects/isat/. [Online; accessed April 2015].

[10] Laurent Granvilliers and Frédéric Benhamou. Realpaver: an interval solver using constraint satisfaction techniques. *ACM Transactions on Mathematical Software (TOMS)*, 32(1):138–156, 2006.

[11] Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.

[12] Yang Gao and Martin Fränzle. A solving procedure for stochastic satisfiability modulo theories with continuous domain. In *Quantitative Evaluation of Systems (QEST)*. Springer, to appear, 2015.

[13] SiSAT Homepage. https://projects.avacs.org/projects/sisat/. [Online; accessed April 2015].