Generalized Craig Interpolation for Stochastic Satisfiability Modulo Theory problems*

Ahmed Mahdi and Martin Fränzle

Carl von Ossietzky Universität,
Ammerländer Heerstraße 114-118, 26111 Oldenburg, Germany
{mahdi|fraenzle}@informatik.uni-oldenburg.de

Abstract. Craig interpolation is widely used in solving reachability and model-checking problems by SAT or SMT techniques, as it permits the computation of invariants as well as discovery of meaningful predicates in CEGAR loops based on predicate abstraction. Extending such algorithms from the qualitative to the quantitative setting of probabilistic models seems desirable. In 2012, Teige et al. [1] succeeded to define an adequate notion of generalized, stochastic interpolants and to expose an algorithm for efficiently computing them for stochastic Boolean satisfiability problems, i.e., SSAT. In this work we present a notion of *Generalized Craig Interpolant* for the stochastic SAT modulo theories framework, i.e., SSMT, and introduce a mechanism to compute such stochastic interpolants for non-polynomial SSMT problems based on a *sound* and, w.r.t. the arithmetic reasoner, *relatively complete* resolution calculus. The algorithm computes interpolants in SAT, SMT, SSAT, and SSMT problems. As this extends the scope of SSMT-based model-checking of probabilistic hybrid automata from the bounded to the unbounded case, we demonstrate our interpolation principle on an unbounded probabilistic reachability problem in a probabilistic hybrid automaton.

1 Introduction

Stochastic satisfiability modulo theories (SSMT) was proposed in 2008 [2] in order to extend SMT-based bounded model-checking to probabilistic hybrid systems. SSMT extends the satisfiability modulo theories (SMT) problem by randomized quantification or, equivalently, generalizes the stochastic boolean satisfiability problem (SSAT) [3] to background theories. An SSMT formula consists of a quantifier prefix and an SMT formula. The quantifier prefix is an alternating sequence of existentially quantified variables and variables bound by randomized quantifiers. All the quantified variables have discrete (finite) domains. Due to the presence of probabilistic assignments due to randomized quantification, the semantics of an SSMT formula Φ is no longer qualitative in the sense that Φ is satisfiable or unsatisfiable, as for propositional or predicate logic, but rather *quantitative* [2, 4]. For an SSMT formula Φ, we ask for the maximum probability

* Research supported by the German Research Council (DFG) as part of the Transregional Collaborative Research Center SFB/TR 14 AVACS (http://www.avacs.org).

of satisfaction or, if formulated as a decision problem, whether this probability of satisfaction exceeds a threshold. Intuitively, a solution of Φ is a strategy in form of a tree suggesting optimal assignments to the existential variables depending on the probabilistically determined values of preceding randomized variables, in order to maximize the probability of satisfying the SMT formula. SSMT as proposed by Fränzle et al. [2] can encode bounded probabilistic reachability problems of probabilistic hybrid automaton (PHA) over discrete time. That means many practical problems exhibiting uncertainty can be described as SSMT problems or sometimes even its propositional subset SSAT, in particular probabilistic planning problems [5, 6], belief networks [7], trust management [8], or depth-bounded PHA reachability [2, 9] and stability problems [4]. Probabilistic bounded model-checking (PBMC) problems, for example, ask whether the probability of *reaching bad states* from the PHA's initial states stays below a given threshold, irrespective of how non-determinism in the PHA is resolved. Solving a PBMC problem can be achieved by taking its equivalent SSMT encoding and solving it with an SSMT solver, like Teige's SiSAT tool [4].

Non-polynomial SSMT problems, i.e., SSMT formulae involving transcendental arithmetic, are generally undecidable due to the undecidable underlying arithmetic theory. There are some decidable classes of SSMT however; e.g., SSMT formulae without free variables due to the finite domains of bound variables, or SSMT formulae over decidable background theories, like linear order [10]. Undecidability implies that the Craig interpolation problem also cannot be solved exactly in general. In this paper, we propose a Craig interpolation procedure for SSMT that is sound and complete when the theory is linear order, and we extend it to non-polynomial SSMT by using interval constraint propagation (ICP) [11], then obviously sacrificing completeness, yet maintaining soundness.

Essentially, we first use ICP for reducing the general, non-polynomial SSMT problem to an SSMT problem of linear order over the reals. As an unsatisfied SSMT problem may well have satisfying assignments —just not sufficiently many to exceed the target probability threshold—, we then have to compute a *generalized interpolant*, which is a Craig interpolant for $A \wedge (B \wedge \neg S_{A,B})$, where $S_{A,B}$ represents the satisfying assignments of the formula $A \wedge B$. We do so by extending Púdlak's rules [12] to compute that generalized Craig interpolant. Instrumental to that adaptation of Púdlak's rules is the observation that the theory of linear order, with simple bounds as its atoms, admits a resolution rule akin to the propositional counterpart.

Related Work: Teige in [1] proposed generalized Craig interpolation for stochastic boolean satisfiability (SSAT) problems. Our work extends this to SSMT involving non-polynomial arithmetic constraints. Kupferschmid in [13] was the first to suggest Craig interpolation for non-polynomial and thus undecidable SMT problems by means of ICP and resolution in SMT of linear order. Our approach employs the same mechanism for dealing with arithmetic constraints, but extends the approach to SSMT problems, thus necessitating computation of generalized rather than traditional Craig interpolants. Numerous authors proposed different mechanisms to compute Craig interpolants for SAT

and decidable SMT problems, e.g., [14–20]. A recent approach for computing small CNF interpolants [21] could be integrated with our work, then replacing Púdlak's rules.

This paper is structured as follows. In Section 2 we define the syntax and semantics of stochastic satisfiability modulo theories. Section 3 presents the SSMT-resolution calculus. In Section 4 we define generalized Craig interpolants for SSMT and expose a computation procedure. Section 5 demonstrates use of SSMT interpolation in probabilistic model-checking, with full details given in [22]. Finally, Section 6 presents the conclusion.

2 Stochastic Satisfiability Modulo Theory (SSMT)

In this section, we introduce the syntax and semantics of stochastic satisfiability modulo theories (SSMT) formulae, as originally proposed in [2].

Definition 1 (Syntax of SSMT). *A stochastic satisfiability modulo theories (SSMT) formula Φ is of the form $\mathcal{Q} : \varphi$ where*

1. *φ is an arbitrary SMT formula with respect to the theory of non-polynomial arithmetic over the reals and integers, called the* matrix *of the formula, and*
2. *$\mathcal{Q} = Q_1 x_1 \in \mathcal{D}_{x_1} \odot .... \odot Q_n x_n \in \mathcal{D}_{x_n}$ is a quantifier prefix binding some variables $x_i \in Var(\varphi)$ over finite domains $\mathcal{D}_{x_i}$ by a sequence of existential and randomized quantifiers Q_i; i.e., $\exists$ and $\mathcal{Y}$ respectively.*

Free, i.e., unbound by quantifiers, variables are permitted in SSMT formulae. For simplicity, we assume that the matrix φ of an SSMT formula $\mathcal{Q} : \varphi$ is in CNF form, as one can convert any formula to a CNF of linear size by introducing auxiliary variables [23].

Definition 2 (Semantics of SSMT). *The semantics of an SSMT formula Φ is given by its maximum probability of satisfaction $Pr(\Phi)$ defined as follows:*

$$Pr(\varepsilon : \varphi) = \begin{cases} 0 \text{ if } \varphi \text{ is unsatisfiable,} \\ 1 \text{ if } \varphi \text{ is satisfiable,} \end{cases}$$

$$Pr(\exists x \in \mathcal{D}_x \odot \mathcal{Q} : \varphi) = max_{v \in D_x} Pr(\mathcal{Q} : \varphi[v/x]),$$

$$Pr(\mathcal{Y}^{d_x} x \in \mathcal{D}_x \odot \mathcal{Q} : \varphi) = \sum_{v \in \mathcal{D}_x} d_x(v) \cdot Pr(\mathcal{Q} : \varphi[v/x]).$$

The semantics of an SSMT formula is a $1\frac{1}{2}$ player game shown in Fig. 1. In naïve SSMT solving, the quantifier tree would be fully unravelled and all resulting instances of the matrix passed to an SMT solver. Pruning rules also shown in Fig. 1 yet permit to skip investigating a major portion of the instances in general.

3 Resolution for SSMT

The existing SSMT solving algorithms of Teige [4] are tightly integrated with the CDCL(ICP)[1] proof search of the iSAT tool [24] and do, in principle, traverse the

[1] CDCL = conflict-driven clause learning, ICP = interval constraint propagation.

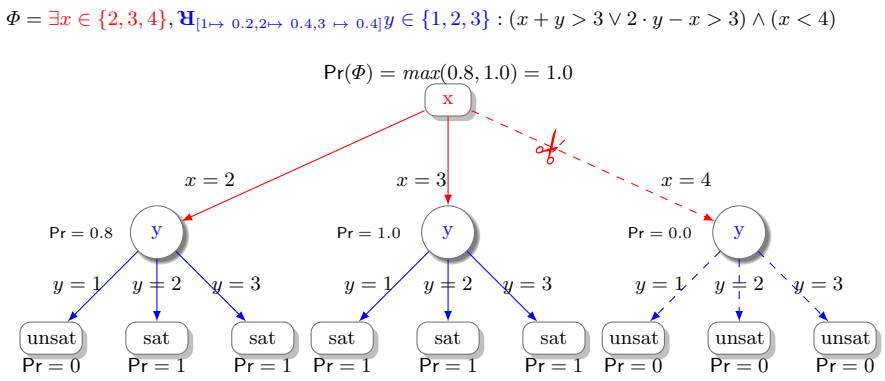


Fig. 1: $1\frac{1}{2}$ player game semantics of an SSMT formula. In recursive solvers, traversal of the dashed part of the quantifier tree will be skipped due to pruning [4].

quantifier tree of the formula as in Fig. 1 to recursively compute the maximum satisfaction probability bottom-up. Note that this does by no means imply that they are bound to traverse the whole, exponentially sized quantifier tree, as Teige proposed various mechanisms to drastically prune that tree and thus accelerate the actual computation. In the contrast to the CDCL(ICP) approach, the SSMT resolution calculus, as proposed by the authors of this paper in [10] based on Teige's SSAT resolution [25], solves SSMT problems by a resolution mechanism. SSMT-resolution works by deriving attributed clauses c^p, where c is a clause and p a probability. When such a clause c^p is derived during resolution, it expresses that the maximum probability of violation of c is p. If the probabilistic variant $\emptyset^p$ of a conflict clause happens to be derived at the end of resolution, then the maximum probability that the formula holds is p. The related SSAT-resolution calculus proposed by Teige [25, 1] is *sound* and *complete*. The same applies for SSMT resolution if the theory is confined to linear order over the reals, yet if (e.g., non-polynomial) arithmetic is involved, the resolution calculus of SSMT is *sound* but only *relatively complete* with interval constraint propagation (ICP) [26] being its "oracle" for resolving arithmetic [10].

All derived clauses c^p are forced to have a tight bound p in the sense that under each assignment which falsifies c, the satisfaction probability of the remaining subproblem is exactly p.[2] Before illustrating the resolution rules, we define the symbolic falsifying assignment $falsify_c$ that captures variable assignments falsifying a clause c. A simple bound $x \sim a \in \mathbb{SB}$ means that a variable x is restricted by comparison operator, i.e., $\sim \in \{>, \geq, <, \leq\}$, relative to value a, where the latter value is a real number. Also, we assign to each variable a domain which is a bounded interval. Let c be a non-tautological disjunction of simple bounds. We define the falsification function $falsify_c$ that falsifies c as following:

[2] In [10] we relaxed the condition to a probability of less than or equal to p. The stronger form used here makes interpolation simpler.

Definition 3 (Falsification function). *Let $\mathbb{C}$ be a set of all non-tautological clauses with a typical element c such that c consists of a disjunction of simple bounds, i.e., $sb_1 \vee ... \vee sb_n$. The falsification function $falsify_c : \mathbb{C} \to \mathbb{C}$ is defined as follows:*

- $falisfy_c(c) := \bigvee_{i=1}^{n} \mathit{ff}_s(sb_i)$,
- $\mathit{ff}_s : \mathbb{SB} \to \mathbb{SB}$ *s.t.* $\mathit{ff}_s(x \sim a) := x \sim' a$ *where* $\sim'$ *is the converse relation to* $\sim$*, e.g.,* $\leq'$ *is* $>$.

where $x \in X$, $a \in \mathbb{R}$, $\sim, \sim' \in \{\leq, <, \geq, >\}$ and x has a well-defined domain.

In order to extend the SSAT resolution rules to SSMT formulae, we assume w.l.o.g. that any clause c where resolution is applied consists of disjunctions of simple bounds only, as ICP yields a reduction to simple bounds by propagating arithemtic constraints into simple bounds [4, 10]. We will introduce four resolution rules that define the resolution calculus for SSMT problems. Rule RR.1 derives a clause c^0 from an original clause $c \in \varphi$ such that c is not a tautological clause. One can consider RR.1 correspond to the quantifier-free base case where φ is *false* under any assignment that falsifies c (cf. [10] for details).

$$\frac{\big(c \in \varphi\big)}{c^0} \tag{RR.1}$$

Rule RR.2 reflects the quantifier-free base case in which φ is true under any assignment that is conform to the partial assignment τ, since $\models \varphi[\tau(x_1)/x_1]...[\tau(x_i)/x_i]$. The constructed c^1 represents the negation of the satisfiable partial assignment τ of φ.

$$\frac{\left(\begin{array}{c} c \subseteq \{x \sim a \mid x \in \mathit{Var}(c)\}, \not\models c, \mathcal{Q}(c) = \mathcal{Q}_1 x_1 ... \mathcal{Q}_i x_i, \\ \text{for each } \tau : \mathit{Var}(\varphi) \downarrow_i \to \mathbb{SB} \text{ with } \forall x \in \mathit{Var}(\varphi) : \tau(x) \text{ in } \mathit{ff}_s(x \sim a) : \\ \models \varphi[\tau(x_1)/x_1]...[\tau(x_i)/x_i] \end{array}\right)}{c^1} \tag{RR.2}$$

Rule RR.3 computes the actual probability of a resolvent depending on the type of the quantifier governing the pivot variable, where a bound on the pivot variable is used as the resolution literal. Definition 2 enforces that the domain of any quantified variable is discrete, which implies that we can evaluate the probability by simply summing up or selecting the maximum of the probabilities of satisfying assignments for $\mathfrak{R}$- or $\exists$-quantified variable x, resp.

$$\frac{\left(\begin{array}{c} (x \sim a \vee c_1)^{p_1}, (x \sim' b \vee c_2)^{p_2}, (x \in \mathcal{D}_x \wedge x \sim a \wedge x \sim' b \vdash \texttt{False}) \\ \mathcal{Q}_x \in \mathcal{Q}, \not\models (c_1 \vee c_2) \\ p = \begin{cases} max(p_1, p_2) & \text{if } \mathcal{Q}_x = \exists x \in \mathcal{D}_x \\ p_1 \cdot Pr(x \sim' b) + p_2 \cdot Pr(x \sim a) & \text{if } \mathcal{Q}_x = \mathfrak{R}^{Pr} x \in \mathcal{D}_x \end{cases} \end{array}\right)}{(c_1 \vee c_2)^p} \tag{RR.3}$$

Rule RR.3e is a counterpart of RR.3 for free variables in SSMT formulae. All free variables are implicitly existentially quantified at innermost level, yet —in

contrast to explicit quantification— to continuous domains in general.

$$\frac{\begin{pmatrix}(x \sim a \vee c_1)^{p_1}, (x \sim' b \vee c_2)^{p_2}, \mathcal{Q}_x \notin \mathcal{Q}, x \text{ has domain } \mathcal{D}_x \\ (x \in \mathcal{D}_x \wedge x \sim a \wedge x \sim' b) \vdash \texttt{False}, \not\models (c_1 \vee c_2) \\ p = max(p_1, p_2)\end{pmatrix}}{(c_1 \vee c_2)^p} \tag{RR.3e}$$

Note that the SSMT-resolution calculus is *sound* and *relatively complete* w.r.t. to its underlying arithmetic reasoner ICP. On SSMT problems over the theory of linear order, SSMT resolution is *complete* (cf. [10, 22] for more details). An example of SSMT resolution is shown together with interpolation in Sect. 4.

4 Interpolation for SSMT

Craig interpolation is a logical concept suggested by Craig in 1957 [27] that has been widely used in model theory and automatic verification. In its classical, non-probabilistic form, a Craig interpolant provides a reason for mutual inconsistency between two formulae. Formally, it is defined as follows:

Definition 4 (Craig Interpolation). *Given two propositional logic formulae A and B in a logics $\mathcal{L}$ such that $\models_{\mathcal{L}} A \rightarrow \neg B$, a* Craig interpolant for (A, B) *is a quantifier-free $\mathcal{L}$-formula $\mathcal{I}$ such that $\models_{\mathcal{L}} A \rightarrow \mathcal{I}$, $\models_{\mathcal{L}} \mathcal{I} \rightarrow \neg B$, and the (necessarily free) variables of $\mathcal{I}$ form a subset of the shared (and thus free) variables between A and B, i.e., $Var(\mathcal{I}) \subseteq Var(A) \cap Var(B)$.*

Depending on the logics $\mathcal{L}$, such Craig interpolants, which provide a reason why A is not satisfiable together with B, can be computed by various mechanisms. If $\mathcal{L}$ admits quantifier-elimination, then this can in principle be used; various more efficient schemes have been devised for propositional logic and for SAT-modulo-theory by exploiting the connection between resolution and variable elimination [12, 28]. Following the latter line, Teige et al. [1] succeeded to generalize the Púdlak rules [12] from the propositional SAT case to stochastic SAT, where a more general definition of interpolant is needed, based on S-resolution [25] for SSAT. In the sequel of this paper, we will do the same for SSMT, thereby exploiting SSMT resolution [10].

4.1 Generalized Craig Interpolants

Traditional interpolation requires that $A \wedge B$ is unsatisfiable for the formulae A and B to interpolate. The precondition $A \wedge B \models \texttt{False}$, which would translated to $Pr(A \wedge B) = 0$ in a stochastic setting, however is too restrictive for use in probabilistic model-checking, as a residual chance of failure — which amounts to satisfying a path condition $A \wedge B$ in that context — is well acceptable in many engineering problems [4, 1]. As an example consider the quantitative safety target "The probability that a plane will crash is at most 10^{-9} per year". For a violation of this quantitative safety goal, we cannot find a classical interpolant in general.

Teige proposed a general concept which can be used to form an adequate lattice of interpolants for stochastic problems.

Definition 5 (Generalized Craig Interpolant [1]). *Let A and B be some SMT formulae where $V_A := Var(A) \setminus Var(B) = \{a_1, ..., a_\alpha\}$, $V_B := Var(B) \setminus Var(A) = \{b_1, ..., b_\beta\}$, $V_{A,B} := Var(A) \cap Var(B)$, $A^\exists = \exists a_1, ..., a_\alpha : A$, and $\overline{B}^\forall = \neg\exists b_1, ..., b_\beta : B$. An SMT formula $\mathcal{I}$ is called a generalized Craig interpolant for (A, B) if and only if the following properties are satisfied: $Var(\mathcal{I}) \subseteq V_{A,B}$, $\models_\mathcal{L} (A^\exists \wedge \overline{B}^\forall) \rightarrow \mathcal{I}$, and $\models_\mathcal{L} \mathcal{I} \rightarrow (A^\exists \vee \overline{B}^\forall)$*

For SMT calculi admitting quantifier elimination, like the linear fragments of integer [29] and rational [30] as well as the polynomial fragment of real arithmetic [31, 32], the four quantifier-free SMT formulae equivalent to $A^\exists \wedge \overline{B}^\forall$, to $A^\exists$, to $\overline{B}^\forall$, and to $A^\exists \vee \overline{B}^\forall$ can serve as generalized Craig interpolants for (A, B). These fragments of arithmetic are, however, very confined. A — necessarily incomplete— interpolation procedure can, however, be obtained for the non-polynomial case based on ICP, which reduces arithmetic reasoning to bound reasoning, i.e., to the decidable case of the theory of linear order over the reals and integers.

An interpolation procedure for SMT involving transcendental functions based on the latter principle has been pioneered by Kupferschmid et al. [13] without, however, addressing the stochastic case of generalized Craig interpolants (GCI). GCI for the propositional case of SSAT, on the other hand, have been explored by Teige et al. [1]. We will here reconcile these lines in order to compute GCI for SSMT.

4.2 Computation of Generalized Craig Interpolants

In this subsection, we present a formal way of computing the Craig interpolants for SSMT formulae by defining certain rules based on the SSMT resolution calculus. In order to compute systemically the Craig interpolants, one can use Púdlak's technique [12] (symmetric) or McMillan's technique [14] (asymmetric) which are both built on top of the resolution calculus for propositional logic.

We use SSMT resolution for computing generalized Craig interpolants. For this purpose, the rules of SSMT resolution are extended to deal with pairs $(c^p, \mathcal{I})$ of annotated clauses c^p and an SMT formulae $\mathcal{I}$, where $\mathcal{I}$ represents a partial generalized interpolant [1, 13]. Whenever a pair $(\emptyset^p, \mathcal{I})$ denoting the empty clause is derived, a generalized Craig interpolant for the given SSMT formula has been computed. We compute the interpolant according to the three rules GR.1, GR.2, and GR.3 given below. The first Rule GR.1 represents a base case assigning initial interpolants to each clause of A and B.

$$\frac{\begin{array}{c} c \vdash_{\text{RR.1}} c^0, \\ \mathcal{I} = \begin{cases} \text{False}, & c \in A \\ \text{True}, & c \in B \end{cases} \end{array}}{(c^0, \mathcal{I})} \quad \text{(GR.1)}$$

Rule GR.2 does not exist in non-stochastic interpolation, as it refers to rule RR.2 of SSMT resolution, where the partial assignment satisfies $A \wedge B$, which

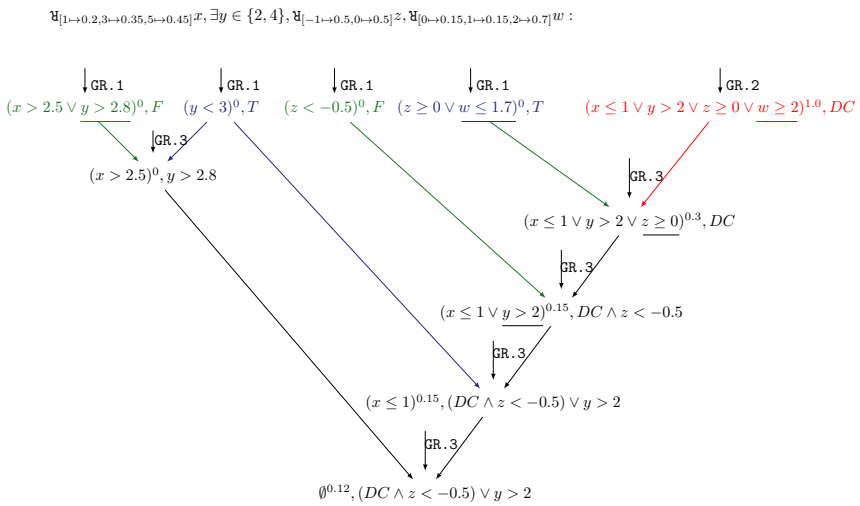


Fig. 2: Generalized Craig interpolant for Example 1. The green part is A and the blue one is B. The red part represents $\neg S_{A,B}$ with a don't-care interpolant.

is impossible in the traditional setting. If we take the negation of the satisfying assignments of $A \wedge B$; i.e., $\neg S_{A,B}$, then $A \wedge \neg S_{A,B}$, and $\neg S_{A,B} \wedge B$ are unsatisfiable. Therefore, we can choose the interpolant freely over the shared variable between A and B, i.e., $V_{A,B}$.

$$\frac{\vdash_{\text{RR.2}} c^1 \qquad \mathcal{I} \text{ is any formula over } V_{A,B}}{(c^1, \mathcal{I})} \tag{GR.2}$$

The third rule extends Púdlak's rule for resolution in the direction of SMT simple bounds. Whenever we have two disjoint simple bounds in different clauses, we can apply SSMT resolution, i.e., one of rules RR.3 or RR.3e.

$$\frac{\begin{array}{r}((x \sim a \vee c_1)^{p_1}, \mathcal{I}_1), ((x \sim' b \vee c_2)^{p_2}, \mathcal{I}_2), \\ (x \sim a \vee c_1)^{p_1}, (x \sim' b \vee c_2)^{p_2} \vdash_{\text{RR.3(e)}} (c_1 \vee c_2)^p, \\ \mathcal{I} = \begin{cases} \mathcal{I}_1 \vee \mathcal{I}_2 & \text{if } x \in V_A \\ \mathcal{I}_1 \wedge \mathcal{I}_2 & \text{if } x \in V_B \\ (x \sim a \vee \mathcal{I}_1) \wedge (x \sim' b \vee \mathcal{I}_2) & \text{if } x \in V_{A,B} \end{cases}\end{array}}{((c_1 \vee c_2)^p, \mathcal{I})} \tag{GR.3}$$

Lemma 1. *Let $\Phi = \mathcal{Q} : (A \wedge B)$ with $\mathcal{Q} = \mathcal{Q}_1 x_1 ... \mathcal{Q}_n x_n$ be some SSMT formula, and the pair $(c^p, \mathcal{I})$ be derivable from Φ by interpolating SSMT-resolution, where $\mathcal{Q}(c) = \mathcal{Q}_1 x_1 ... \mathcal{Q}_i x_i$. Then, for each $\tau : Var(\varphi) \downarrow_i := \{x_1, ..., x_i\}$ for $i \leq n$ with $\forall x \in Var(c) : \tau(x) = ff_s(x \sim a)$, where $x \sim a \in c$, it holds that:*

1. *$Var(\mathcal{I}) \subseteq V_{A,B}$,*

2. $Pr(\mathcal{Q}_{i+1}x_{i+1}...\mathcal{Q}_n x_n : (A \wedge \neg S_{A,B} \wedge \neg \mathcal{I})[\tau(x_1)/x_1]...[\tau(x_i)/x_i]) = 0$, *and*
3. $Pr(\mathcal{Q}_{i+1}x_{i+1}...\mathcal{Q}_n x_n : (\mathcal{I} \wedge B \wedge \neg S_{A,B})[\tau(x_1)/x_1]...[\tau(x_i)/x_i]) = 0$.

The proof of this Lemma is stated in [22]. By using the previous lemma with the relatively complete SSMT resolution calculus, we get the following corollary:

Corollary 1 (Generating generalized SSMT interpolants). *If interpolating SSMT resolution derives* $(\emptyset^p, \mathcal{I})$ *from an SSMT formula* $\Phi = \mathcal{Q} : (A \wedge B)$, *then* $\mathcal{I}$ *is a generalized Craig interpolant for* (A, B) *witnessing* $Pr(\Phi) = p$.

The previous corollary follows directly due to Def. 5.

Corollary 2 (Controlling strength of SSMT interpolants). *If* $\mathcal{I} =$ `true` *is used within each application of Rule GR.2, then* $Pr(\mathcal{Q} : (A \wedge \neg\mathcal{I})) = 0$. *If* $\mathcal{I} =$ `false` *is used within each application of Rule GR.2, then* $Pr(\mathcal{Q} : (B \wedge \mathcal{I})) = 0$.

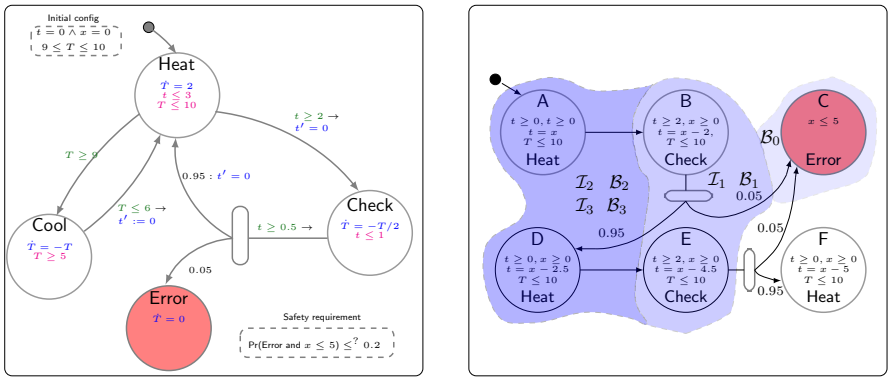
Proof. The proof of this corollary follows the previous lemma. The complete proof is stated for the SSAT case in [1] and adapts easily to SSMT.

Example 1. In order to get the idea of computing the Craig interpolants for SSMT problems, let us consider the following formula: $\aleph_{[1\mapsto 0.2,3\mapsto 0.35,5\mapsto 0.45]}$ $x,\ \exists y \in \{2,4\}\ \aleph_{[-1\mapsto 0.5,\ 0\mapsto 0.5]},\ z\ \aleph_{[0\mapsto 0.15,1\mapsto 0.15,2\mapsto 0.7]} w : A \wedge B$ where $A = (z < -0.5) \wedge (x > 2.5 \vee y > 2.8)$ and $B = (y < 3) \wedge (z \geq 0 \vee w \leq 1.7)$. Fig. 2 shows formally how the generalized Craig interpolant is computed. *DC* stands for a *don't care* formula which can replaced by true or false, a.o. If we replace *DC* with *true*, then the interpolant becomes $z < -0.5 \vee y > 2$ which is implied by A. Likewise, if it is replaced by *false*, then the resulting interpolant $y > 2$ implies the negation of B as in Corollary 2.

5 Interpolation-based probabilistic model checking

In this section we demonstrate an application of generalized Craig interpolation to quantitative model-checking of probabilistic hybrid automata. Probabilistic hybrid automata (PHA) are Markov decision processes (MDPs) over infinite state space, with arithmetic-logical transition guards and actions. These permit a straightforward encoding by SSMT formulae as proposed in [2, 1]. Let us consider that we are given some set T of target states in the PHA model, and we try to maximize the probability of reaching these states over all policies resolving the non-determinism in the PHA model. Applications would be that T represents bad (or good) states and that we are asked to assure that the maximum probability of reaching bad (good, resp.) states in the model does not violate a certain safety target (exceeds a desired service level, resp.).

The encoding of PHA into SSMT formulae pioneered in [2] directly applies to PHA capturing continuous dynamics by pre-post relations. For PHA containing ordinary differential equations, one has to add ICP for ODE, as suggested in [33] and integrated into SSMT solving in [9], or one has to resort to abstraction of ODE into pre-post relations by tools like PHAVer, as pursued in ProHVer [34, 35]. For the thermostat case study presented in Fig. 3a, we use the latter approach, obtaining the abstraction depicted in Fig. 3b and taken from [34].



(a) PHA model of a thermostat involving ordinary differential equations

(b) Abstraction of the thermostat model using pre-post relations instead [34]

Fig. 3: Thermostat case-study discussed in [34, 35]

5.1 Probabilistic bounded model-checking (PBMC)

The idea of interpolation-based bounded model checking is to encode *the step-bounded reachability problem* as an SSMT formula. In each step, the transition relation, the non-deterministic choices, and the probabilistic choices are encoded, where the first one is achieved by an SMT formula, while the latter two require existential and randomized quantification respectively. Furthermore, the initial states and target states are encoded by predicates.

5.2 Interpolation-based unbounded model-checking

In order to use generalized interpolation in unbounded probabilistic model-checking, first one needs to *encode* the model's transiiton relation by a SMT representation. Then one generates a probabilistic bounded model-checking problem (PBMC) in SSMT [2] and determines whether the targets are reachable with probability exceeding the safety target within some step bound k. Should this not be the case, one can use generalized Craig interpolation to compute an *over-approximation of the states backward reachable* from the targets within that step bound. Technically, we interpolate between the initial state predicate and the k-fold iteration of the transition relation plus the target predicate, albeit under quantification. PBMC is iterated for increasingly larger k until either the safety property is falsified or the generalized Craig interpolant (CGI) stabilizes, i.e., a superset of all states backward reachable from the target has been computed.

Let us consider the PHA of Fig. 3(a) modelling a thermostat system. Using its safe abstraction Fig. 3(b), we want to verify whether *the maximum probability to reach the location* `Error` *within 5 time units is at most* $\frac{1}{5}$. Note that the property is expressed in terms of time units rather than computation steps. As there is no immediate correspondence between time units and computation steps, this

verification problem cannot be solved by PBMC, but rather requires unbounded reachability computation by GCI.

In the abstract model, the probability to reach the error states within 5 time units is 0.0975, which is less than $\frac{1}{5}$ and thus acceptable. To determine this probability, we encode the abstraction of the thermostat as an SSMT formula and then compute overapproximations of the backward reachable states incrementally by GCI until it stabilizes. The target is `C-Error` which cannot be reached from the initial `A-Heat` via a single transition. In the first interpolation, the target `C-Error` together with a single transition relation represents the A part, while the initial state predicate `A-Heat` constitutes B. The first computed interpolant will thus equal all states except the initial one, providing a useless upper bound of 1 on the probability of eventually hitting the target. Successive interpolations for larger step numbers yield tighter approximations. In this model, the interpolant stabilizes after three iterations and yields a tight enough overapproximation of the backward reachable state set (cf. [22] for details).

Fig. 4 represents three results: the upper (red) curve represents the upper bound on the step-unbounded probability to reach location `Error` within 5 time units, as computed by GCI. The numbers on the horizontal axis here refer to the iteration (the number of steps), while the vertical axis refers to the computed probabilities. The middle (green) line represents the exact probability to reach location `Error` within 5 time units. The lower (blue) curve represents the lower bound on the probability to reach an `Error` state within 5 time units, as computed by PBMC. One may observe that upper and lower bounds almost coincide after step $k = 4$. In fact, interpolation then tells us that the reachability probability is below 0.1, i.e., well below the safety target. All details of this example are shown in [22].

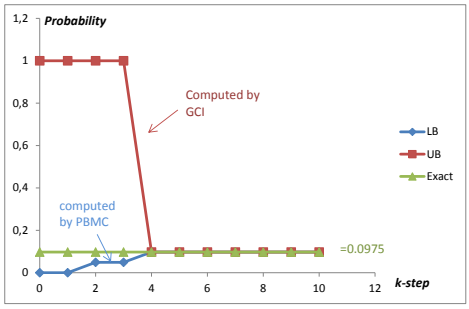


Fig. 4: Probability of reaching Error state within 5 time units with(out) interpolation

6 Conclusion and Future work

We have successfully extended the concept of generalized Craig interpolation (CGI) from stochastic SAT to stochastic SAT modulo theory. We exposed a rule set suitable for automatically computing CGIs in non-polynomial arithmetic SSMT problems. An application of CGI on unbounded probabilistic model-checking problems was demonstrated, where the step-bounded probabilistic reachability of PHAs is encoded symbolically as an SSMT problem and interpolation serves as a means for generalizing the findings to the unbounded case. This approach can straightforwardly be extended to *probabilistic stability* problems [1]. In *future work* we will integrate the interpolation procedure into the SiSAT tool [4] for automatic quantitative analysis of PHA.

References

1. Teige, T., Fränzle, M.: Generalized Craig interpolation. Logical Methods in Computer Science **8**(2) (2012)
2. Fränzle, M., Hermanns, H., Teige, T.: Stochastic satisfiability modulo theory: A novel technique for the analysis of probabilistic hybrid systems. In Egerstedt, M., Mishra, B., eds.: HSCC. Volume 4981 of Lecture Notes in Computer Science., Springer (2008) 172–186
3. Papadimitriou, C.H.: Games against nature. J. Comput. Syst. Sci. **31**(2) (1985) 288–301
4. Teige, T.: Stochastic Satisfiability Modulo Theories: A Symbolic Technique for the Analysis of Probabilistic Hybrid Systems. PhD thesis, Dpt. of Computing Science, Carl von Ossietzky Universität, Oldenburg, Germany (August 2012)
5. Majercik, S.M., Littman, M.L.: Maxplan: A new approach to probabilistic planning. In Simmons, R.G., Veloso, M.M., Smith, S.F., eds.: AIPS, AAAI (1998) 86–93
6. Majercik, S.M., Littman, M.L.: Contingent planning under uncertainty via stochastic satisfiability. Artif. Intell. **147**(1-2) (2003) 119–162
7. Bacchus, F., Dalmao, S., Pitassi, T.: DPLL with caching: A new algorithm for #sat and Bayesian inference. Electronic Colloquium on Computational Complexity (ECCC) **10**(003) (2003)
8. Freudenthal, E., Karamcheti, V.: QTM: Trust management with quantified stochastic attributes. Technical Report NYU Computer Science Technical Report TR2003-848, Courant Institute of Mathematical Sciences, New York University (2003)
9. Teige, T., Eggers, A., Fränzle, M.: Constraint-based analysis of concurrent probabilistic hybrid systems: An application to networked automation systems. Nonlinear Analysis: Hybrid Systems **5**(2) (2011) 343–366
10. Mahdi, A., Fränzle, M.: Resolution for stochastic SAT modulo theories. Technical report, Dpt. of Computing Science, Carl von Ossietzky Universität Oldenburg, Germany (December 2013)
11. Benhamou, F., Granvilliers, L.: Combining local consistency, symbolic rewriting and interval methods. In Calmet, J., Campbell, J.A., Pfalzgraf, J., eds.: AISMC. Volume 1138 of Lecture Notes in Computer Science., Springer (1996) 144–159
12. Pudlák, P.: Lower bounds for resolution and cutting plane proofs and monotone computations. J. Symb. Log. **62**(3) (1997) 981–998
13. Kupferschmid, S., Becker, B.: Craig interpolation in the presence of non-linear constraints. In Fahrenberg, U., Tripakis, S., eds.: FORMATS. Volume 6919 of Lecture Notes in Computer Science., Springer (2011) 240–255
14. McMillan, K.L.: Interpolation and SAT-based model checking. In Hunt Jr., W.A., Somenzi, F., eds.: CAV. Volume 2725 of Lecture Notes in Computer Science., Springer (2003) 1–13
15. Christ, J., Hoenicke, J., Nutz, A.: Proof tree preserving interpolation. In Piterman, N., Smolka, S.A., eds.: TACAS. Volume 7795 of Lecture Notes in Computer Science., Springer (2013) 124–138
16. Brillout, A., Kroening, D., Wahl, T.: Craig interpolation for quantifier-free Presburger arithmetic. CoRR **abs/0811.3521** (2008)
17. Griggio, A., Thi Thieu Hoa Le, Sebastiani, R.: Efficient interpolant generation in satisfiability modulo linear integer arithmetic. In Abdulla, P.A., Leino, K.R.M., eds.: TACAS. Volume 6605 of Lecture Notes in Computer Science., Springer (2011) 143–157

18. Goel, A., Krstic, S., Tinelli, C.: Ground interpolation for combined theories. In Schmidt, R.A., ed.: CADE. Volume 5663 of Lecture Notes in Computer Science., Springer (2009) 183–198
19. Cimatti, A., Griggio, A., Sebastiani, R.: Efficient generation of Craig interpolants in satisfiability modulo theories. ACM Trans. Comput. Log. **12**(1) (2010) 7
20. Lynch, C., Tang, Y.: Interpolants for linear arithmetic in SMT. In Cha, S.D., Choi, J.Y., Kim, M., Lee, I., Viswanathan, M., eds.: ATVA. Volume 5311 of Lecture Notes in Computer Science., Springer (2008) 156–170
21. Vizel, Y., Ryvchin, V., Nadel, A.: Efficient generation of small interpolants in CNF. In Sharygina, N., Veith, H., eds.: CAV. Volume 8044 of Lecture Notes in Computer Science., Springer (2013) 330–346
22. Mahdi, A., Fränzle, M.: Generalized Craig interpolation for SSMT. Technical report, Dpt. of Comuting Sceince, Carl von Ossietzky Universität, Oldenburg, Germany (2014)
23. Tseitin, G.S.: On the complexity of derivation in propositional calculus. In Siekmann, J., Wrightson, G., eds.: Automation of Reasoning 2: Classical Papers on Computational Logic 1967-1970. Springer, Berlin, Heidelberg (1983) 466–483
24. Fränzle, M., Herde, C., Ratschan, S., Schubert, T., Teige, T.: Efficient solving of large non-linear arithmetic constraint systems with complex Boolean structure. Journal on Satisfiability, Boolean Modeling and Computation – Special Issue on SAT/CP Integration **1** (2007) 209–236
25. Teige, T., Fränzle, M.: Resolution for stochastic Boolean satisfiability. In Fermüller, C.G., Voronkov, A., eds.: LPAR (Yogyakarta). Volume 6397 of Lecture Notes in Computer Science., Springer (2010) 625–639
26. Benhamou, F., McAllester, D.A., Hentenryck, P.V.: CLP(Intervals) revisited. In Bruynooghe, M., ed.: ILPS, MIT Press (1994) 124–138
27. Craig, W.: Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. J. Symb. Log. **22**(3) (1957) 269–285
28. Esparza, J., Kiefer, S., Schwoon, S.: Abstraction refinement with Craig interpolation and symbolic pushdown systems. In Hermanns, H., Palsberg, J., eds.: TACAS. Volume 3920 of Lecture Notes in Computer Science., Springer (2006) 489–503
29. Cooper, D.: Theorem proving in arithmetic without multiplication. Machine Intelligence **7** (1972) 91–99
30. Ferrante, J., Rackoff, C.: A decision procedure for the first order theory of real addition with order. SIAM J. Comput. **4**(1) (1975) 69–76
31. Tarski, A.: A decision method for elementary algebra and geometry. RAND Corporation, Santa Monica, Calif. (1948)
32. Davenport, J.H., Heintz, J.: Real quantifier elimination is doubly exponential. J. Symb. Comput. **5**(1/2) (1988) 29–35
33. Eggers, A., Fränzle, M., Herde, C.: SAT modulo ODE: A direct SAT approach to hybrid systems. In Cha, S.S., Choi, J.Y., Kim, M., Lee, I., Viswanathan, M., eds.: Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis (ATVA'08). Volume 5311 of Lecture Notes in Computer Science (LNCS)., Springer-Verlag (2008) 171–185
34. Zhang, L., She, Z., Ratschan, S., Hermanns, H., Hahn, E.M.: Safety verification for probabilistic hybrid systems. In Touili, T., Cook, B., Jackson, P., eds.: CAV. Volume 6174 of Lecture Notes in Computer Science., Springer (2010) 196–211
35. Fränzle, M., Hahn, E.M., Hermanns, H., Wolovick, N., Zhang, L.: Measurability and safety verification for stochastic hybrid systems. In Caccamo, M., Frazzoli, E., Grosu, R., eds.: HSCC, ACM (2011) 43–52