AVACS – Automatic Verification and Analysis of Complex Systems

REPORTS

of SFB/TR 14 AVACS

Editors: Board of SFB/TR 14 AVACS

Generalized Craig Interpolation for Stochastic Boolean Satisfiability Problems

by

Tino Teige Martin Fränzle

AVACS Technical Report No. 67
March 2011
ISSN: 1860-9821

Publisher: Sonderforschungsbereich/Transregio 14 AVACS
(Automatic Verification and Analysis of Complex Systems)
Editors: Bernd Becker, Werner Damm, Bernd Finkbeiner, Martin Fränzle,
Ernst-Rüdiger Olderog, Andreas Podelski
ATRs (AVACS Technical Reports) are freely downloadable from `www.avacs.org`

Author(s) contact: Martin Fränzle (`mf@avacs.org`).

Generalized Craig Interpolation for Stochastic Boolean Satisfiability Problems*

Tino Teige and Martin Fränzle

Carl von Ossietzky Universität, Oldenburg, Germany
{teige|fraenzle}@informatik.uni-oldenburg.de

Abstract. The stochastic Boolean satisfiability (SSAT) problem has been introduced by Papadimitriou in 1985 when adding a probabilistic model of uncertainty to propositional satisfiability through randomized quantification. SSAT has many applications, among them bounded model checking (BMC) of symbolically represented Markov decision processes. This paper identifies a notion of *Craig interpolant* for the SSAT framework and develops an algorithm for computing such interpolants based on SSAT resolution. As a potential application, we address the use of interpolation in SSAT-based BMC, turning the falsification procedure into a verification approach for probabilistic safety properties.

1 Introduction

Papadimitriou [1] has proposed the idea of modeling uncertainty within propositional satisfiability (SAT) by adding *randomized* quantification to the problem description. The resultant *stochastic Boolean satisfiability* (SSAT) problems consist of a quantifier prefix followed by a propositional formula. The quantifier prefix is an alternating sequence of existentially quantified variables and variables bound by randomized quantifiers. The meaning of a randomized variable x is that x takes value `true` with a certain probability p and value `false` with the complementary probability $1-p$. Due to the presence of such probabilistic assignments, the semantics of an SSAT formula Φ no longer is qualitative in the sense that Φ is satisfiable or unsatisfiable, but rather *quantitative* in the sense that we are interested in the *maximum probability of satisfaction* of Φ. Intuitively, a solution of Φ is a strategy for assigning the existential variables, i.e. a tree of assignments to the existential variables depending on the probabilistically determined values of preceding randomized variables, such that the assignments maximize the probability of satisfying the propositional formula.

In recent years, the SSAT framework has attracted interest within the Artificial Intelligence community, as many problems from that area involving uncertainty have concise descriptions as SSAT problems, in particular probabilistic planning problems [2–4]. Inspired by that work, other communities have started to exploit SSAT and closely related formalisms within their domains. The Constraint Programming community is working on *stochastic constraint satisfaction* problems [5, 6] to address, a.o., multi-objective decision making under uncertainty [7]. Recently, a technique for the symbolic analysis of probabilistic hybrid systems based on stochastic satisfiability has been suggested by the authors [8, 9]. To this end, SSAT has been extended by embedded theory reasoning over arithmetic theories, as known from *satisfiability modulo theories* (SMT) [10], which yields the notion of *stochastic*

* An abridged version of the material contained herein, omitting most of the proofs and examples, has been accepted for publication in the proceedings of the "Seventeenth International Conference on Tools and Algorithms for the Construction and Analysis of Systems" (TACAS), to be published by Springer Verlag in spring 2011.

satisfiability modulo theories (SSMT). By the expressive power of SSMT, bounded probabilistic reachability problems of uncertain hybrid systems can be phrased symbolically as SSMT formulae yielding the same probability of satisfaction. As this bounded model checking approach yields valid lower bounds lb for the probability of reaching undesirable system states along unbounded runs, it is able to *falsify* probabilistic safety requirements of shape "a system error occurs with probability at most 0.1‰".

Though the general SSAT problem is PSPACE-complete, the plethora of real-world applications calls for practically efficient algorithms. The first SSAT algorithm, suggested by Littman [11], extends the Davis-Putnam-Logemann-Loveland (DPLL) procedure [12, 13] for SAT with appropriate quantifier handling and algorithmic optimizations like *thresholding*. Majercik improved the DPLL-style SSAT algorithm by *non-chronological backtracking* [14]. Unlike these explicit tree-traversal approaches and motivated by work on *resolution* for propositional and first-order formulae [15] and for QBF formulae [16], the authors have recently developed an alternative SSAT procedure based on resolution [17].

In this paper, we investigate the concept of Craig interpolation for SSAT. Given two formulae A and B for which $A \Rightarrow B$ is true, a *Craig interpolant* [18] $\mathcal{I}$ is a formula over variables common to A and B that "lies in between" A and B in the sense that $A \Rightarrow \mathcal{I}$ and $\mathcal{I} \Rightarrow B$. In the automatic hardware and software verification communities, Craig interpolation has found widespread use in model checking algorithms, both as a means of extracting reasons for non-concretizability of a counterexample obtained on an abstraction as well as for obtaining a symbolic description of reachable state sets. In McMillan's approach [19, 20], interpolants are used to symbolically describe an overapproximation of the step-bounded reachable state set. If the sequence of interpolants thus obtained stabilizes eventually, i.e. no additional state is found to be reachable, then the corresponding state-set predicate R has all reachable system states as its models. The safety property that states satisfying B (where B is a predicate) are never reachable, is then verified by checking $R \wedge B$ for unsatisfiability.

Given McMillan's verification procedure for non-probabilistic systems, it is natural to ask whether a corresponding probabilistic counterpart can be developed, i.e. a *verification procedure for probabilistic systems based on Craig interpolation for stochastic SAT*. Such an approach would complement the aforementioned falsification procedure for probabilistic systems based on SSAT/SSMT. In this paper, we suggest a solution to the issue above. After a formal introduction of SSAT in Section 2, we recall (and adapt slightly) the resolution calculus for SSAT from [17] in Section 3. We suggest a generalization of the notion of Craig interpolants suitable for SSAT as well as an algorithm based on SSAT resolution to compute such generalized interpolants (Section 4). Finally, we propose an interpolation-based approach to probabilistic model checking that is able to *verify* probabilistic safety requirements (Section 5) and illustrate the applicability of this verification procedure on a small example.

2 Stochastic Boolean satisfiability

A *stochastic Boolean satisfiability* (SSAT) formula is of the form $\Phi = \mathcal{Q} : \varphi$ with a prefix $\mathcal{Q} = Q_1 x_1 \dots Q_n x_n$ of quantified propositional variables x_i, where Q_i is either an existential quantifier $\exists$ or a randomized quantifier R^{p_i} with a rational constant $0 < p_i < 1$, and a propositional formula φ s.t. $Var(\varphi) \subseteq \{x_1, \dots, x_n\}$, where $Var(\varphi)$ denotes the set of all (necessarily free) variables occurring in φ. W.l.o.g., we can assume that φ is in *conjunctive normal form* (CNF), i.e. a conjunction of disjunctions of propositional literals. A *literal* ℓ is a propositional variable, i.e.

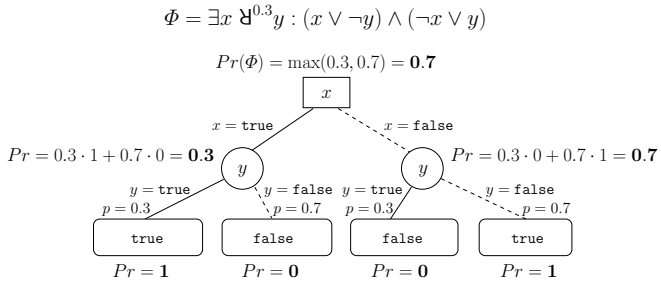

Fig. 1. Semantics of an SSAT formula depicted as a tree.

$\ell = x_i$, or its negation, i.e. $\ell = \neg x_i$. A *clause* is a disjunction of literals. Throughout the paper and w.l.o.g., we require that a clause does not contain the same literal more than once as $\ell \vee \ell \equiv \ell$. Consequently, we may also identify a clause with its set of literals. The semantics of Φ, as illustrated in Fig. 1, is defined by the *maximum probability of satisfaction* $Pr(\Phi)$ as follows.

$$
\begin{array}{lcl}
Pr(\varepsilon : \varphi) & = & \begin{cases} 0 & \text{if } \varphi \text{ is logically equivalent to } \mathtt{false} \\ 1 & \text{if } \varphi \text{ is logically equivalent to } \mathtt{true} \end{cases} \\
Pr(\exists x\ \mathcal{Q} : \varphi) & = & \max(Pr(\mathcal{Q} : \varphi[\mathtt{true}/x]), Pr(\mathcal{Q} : \varphi[\mathtt{false}/x])) \\
Pr(\mathrm{R}^p x\ \mathcal{Q} : \varphi) & = & p \cdot Pr(\mathcal{Q} : \varphi[\mathtt{true}/x]) \ + \ (1-p) \cdot Pr(\mathcal{Q} : \varphi[\mathtt{false}/x])
\end{array}
$$

Note that the semantics is well-defined as Φ has no free variables s.t. all variables have been substituted by the constants `true` and `false` when reaching the quantifier-free base case.

3 Resolution for SSAT

As basis of the SSAT interpolation procedure introduced in Section 4, we recall the sound and complete resolution calculus for SSAT from [17], subsequently called *S-resolution*. In contrast to SSAT algorithms implementing a DPLL-based backtracking procedure, thereby explicitly traversing the tree given by the quantifier prefix and recursively computing the individual satisfaction probabilities for each subtree by the scheme illustrated in Fig. 1, S-resolution follows the idea of *resolution* for propositional and first-order formulae [15] and for QBF formulae [16] by deriving new clauses c^p annotated with probabilities $0 \leq p \leq 1$. S-resolution differs from non-stochastic resolution, as such derived clauses c^p need not be implications of the given formula, but are just entailed with some probability. Informally speaking, the derivation of a clause c^p means that under SSAT formula $\mathcal{Q} : \varphi$, the clause c is violated with a maximum probability at most p, i.e. the satisfaction probability of $\mathcal{Q} : (\varphi \wedge \neg c)$ is at most p. More intuitively, the minimum probability that clause c is implied by φ is at least $1 - p$.[1] Once an annotated empty clause $\emptyset^p$ is derived, it follows that the probability of the given SSAT formula is at most p, i.e. $Pr(\mathcal{Q} : (\varphi \wedge \neg\mathtt{false})) = Pr(\mathcal{Q} : \varphi) \leq p$.

The following presentation of S-resolution differs slightly from [17] in order to avoid overhead in interpolant generation incurred when employing the original definition, like the necessity of enforcing particular resolution sequences. For readers familiar with [17], the particular modifications are: 1) derived clauses c^p may also carry value $p = 1$, 2) former rules R.2 and R.5 are joined into the new rule R.2, and 3) former rules R.3 and R.4 are collapsed into rule R.3. These modifications do not

[1] We remark that $Pr(\mathcal{Q} : \psi) = 1 - Pr(\mathcal{Q}' : \neg\psi)$, where $\mathcal{Q}'$ arises from $\mathcal{Q}$ by replacing existential quantifiers by universal ones, where universal quantifiers call for *minimizing* the satisfaction probability.

affect soundness and completeness of S-resolution (cf. Corollary 1 and Theorem 1). The advantage of the modification is that derivable 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 *exactly* is p (cf. Lemma 1). This fact confirms the conjecture from [17, p. 14] about the existence of such clauses $(c \vee \ell)^p$ and allows for a generalized clause learning scheme to be integrated into DPLL-SSAT solvers: the idea is that under a partial assignment falsifying c, one may directly propagate literal ℓ as the satisfaction probability of the other branch, for which the negation of ℓ holds, is known to be p already.

In the sequel, let $\Phi = \mathcal{Q} : \varphi$ be an SSAT formula with φ in CNF. W.l.o.g., φ contains only non-tautological clauses[2], i.e. $\forall c \in \varphi : \not\models c$. Let $\mathcal{Q} = Q_1 x_1 \dots Q_n x_n$ be the quantifier prefix and φ be some propositional formula with $Var(\varphi) \subseteq \{x_1, \dots, x_n\}$. The quantifier prefix $\mathcal{Q}(\varphi)$ is defined to be shortest prefix of $\mathcal{Q}$ that contains all variables from φ, i.e. $\mathcal{Q}(\varphi) = Q_1 x_1 \dots Q_i x_i$ where $x_i \in Var(\varphi)$ and for each $j > i : x_j \notin Var(\varphi)$. Let further be $Var(\varphi) \downarrow_k := \{x_1, \dots, x_k\}$ for each integer $0 \leq k \leq n$. For a non-tautological clause c, i.e. if $\not\models c$, we define the unique assignment ff_c that falsifies c as the mapping

$$\mathit{ff}_c : Var(c) \to \mathbb{B} \text{ such that } \forall x \in Var(c) : \mathit{ff}_c(x) = \begin{cases} \texttt{true} & ; \neg x \in c, \\ \texttt{false} & ; \quad x \in c. \end{cases}$$

Consequently, c evaluates to `false` under assignment ff_c.

Starting with clauses in φ, *S-resolution* is given by the consecutive application of rules R.1 to R.3 to derive new clauses c^p with $0 \leq p \leq 1$. Rule R.1 derives a clause c^0 from an original clause c in φ. Referring to the definition of $Pr(\Phi)$ in Section 2, R.1 corresponds to the quantifier-free base case where φ is equivalent to `false` under any assignment that falsifies c.

$$\text{(R.1)} \qquad \frac{c \in \varphi}{c^0}$$

Similarly, R.2 reflects the quantifier-free base case in which φ is equivalent to `true` under any assignment τ' that is conform to the partial assignment τ since $\models \varphi[\tau(x_1)/x_1] \dots [\tau(x_i)/x_i]$. The constructed clause c^1 then encodes the opposite of this satisfying (partial) assignment τ. We remark that finding such a τ in the premise of R.2 is NP-hard (equivalent to finding a solution of a propositional formula in CNF). This strong condition on τ is not essential for soundness and completeness and could be removed[3] but, as mentioned above, facilitates a less technical presentation of generalized interpolation in Section 4. Another argument justifying the strong premise of R.2 is a potential integration of S-resolution into DPLL-based SSAT solvers since whenever a satisfying (partial) assignment τ of φ is found by an SSAT solver then τ meets the requirements of R.2.

$$\text{(R.2)} \qquad \frac{\begin{array}{c} c \subseteq \{x, \neg x | x \in Var(\varphi)\}, \not\models c, \mathcal{Q}(c) = Q_1 x_1 \dots Q_i x_i, \\ \text{for each } \tau : Var(\varphi) \downarrow_i \to \mathbb{B} \text{ with } \forall x \in Var(c) : \tau(x) = \mathit{ff}_c(x) : \\ \models \varphi[\tau(x_1)/x_1] \dots [\tau(x_i)/x_i] \end{array}}{c^1}$$

Rule R.3 finally constitutes the actual resolution rule as known from the non-stochastic case. Depending on whether an existential or a randomized variable is resolved upon, the probability value of the resolvent clause is computed according

[2] Tautological clauses c, i.e. $\models c$, are redundant, i.e. $Pr(\mathcal{Q} : (\varphi \wedge c)) = Pr(\mathcal{Q} : \varphi)$.

[3] Then, Lemma 1 must be weakened (as for original S-resolution [17]) to $Pr(Q_{i+1} x_{i+1} \dots Q_n x_n : \varphi[\tau(x_1)/x_1] \dots [\tau(x_i)/x_i]) \leq p$.

to the semantics $Pr(\Phi)$ defined in Section 2.

$$\text{(R.3)}\quad \frac{(c_1 \vee \neg x)^{p_1}, (c_2 \vee x)^{p_2}, Qx \in \mathcal{Q}, Qx \notin \mathcal{Q}(c_1 \vee c_2), \not\models (c_1 \vee c_2), \quad p = \begin{cases} \max(p_1, p_2) & ;\ Q = \exists \\ p_x \cdot p_1 + (1 - p_x) \cdot p_2 & ;\ Q = \mathbb{R}^{p_x} \end{cases}}{(c_1 \vee c_2)^p}$$

The derivation of a clause c^p by R.1 from c, by R.2, and by R.3 from $c_1^{p_1}, c_2^{p_2}$ is denoted by $c \vdash_{\mathsf{R.1}} c^p$, by $\vdash_{\mathsf{R.2}} c^p$, and by $(c_1^{p_1}, c_2^{p_2}) \vdash_{\mathsf{R.3}} c^p$, respectively. Given rules R.1 to R.3, S-resolution is sound and complete in the following sense.

Lemma 1. *Let clause c^p be derivable by S-resolution and let $\mathcal{Q}(c) = Q_1 x_1 \dots Q_i x_i$. For each $\tau : Var(\varphi) \downarrow_i \to \mathbb{B}$ with $\forall x \in Var(c) : \tau(x) = \mathit{ff}_c(x)$ it holds that $Pr(Q_{i+1}x_{i+1} \dots Q_n x_n : \varphi[\tau(x_1)/x_1] \dots [\tau(x_i)/x_i]) = p$.*

Proof. We show the lemma by induction over the application of rules R.1, R.2, and R.3.[4] The base case is given by rules R.1 and R.2. By construction of τ, $\varphi[\tau(x_1)/x_1] \dots [\tau(x_i)/x_i]$ is unsatisfiable for R.1 and tautological for R.2 which immediately establishes the result for the base case. Now assume that the assumption holds for all clauses in the premises of R.3, i.e.

$$\begin{aligned} Pr(Q_{j+1}x_{j+1} \dots Q_n x_n : \varphi[\tau_1(x_1)/x_1] \dots [\tau_1(x_{j-1})/x_{j-1}][\mathtt{true}/x_j]) &= p_1, \\ Pr(Q_{j+1}x_{j+1} \dots Q_n x_n : \varphi[\tau_2(x_1)/x_1] \dots [\tau_2(x_{j-1})/x_{j-1}][\mathtt{false}/x_j]) &= p_2, \end{aligned}$$

where $x_j = x$ with $j \geq i+1$. By definition of Pr, for each τ with $\tau(x) = \tau_1(x)$ if $x \in Var(c_1)$ and $\tau(x) = \tau_2(x)$ if $x \in Var(c_2)$ we then have

$$Pr(Q_j x_j\ Q_{j+1}x_{j+1} \dots Q_n x_n : \varphi[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}]) = p.$$

The result is obvious for $j = i+1$. For $j > i+1$, note that variables $x_{i+1}, \dots, x_{j-1}$ do not occur in the derived clause $(c_1 \vee c_2)$. Hence, for $k = j-1$ to $i+1$ we successively conclude that

$$\begin{aligned} Pr(Q_{k+1}x_{k+1} \dots Q_n x_n : \varphi[\tau(x_1)/x_1] \dots [\tau(x_{k-1})/x_{k-1}][\mathtt{true}/x_k]) &= p, \\ Pr(Q_{k+1}x_{k+1} \dots Q_n x_n : \varphi[\tau(x_1)/x_1] \dots [\tau(x_{k-1})/x_{k-1}][\mathtt{false}/x_k]) &= p. \end{aligned}$$

From case $k = i+1$ the lemma follows. □

Corollary 1 (Soundness of S-resolution). *If the empty clause $\emptyset^p$ is derivable by S-resolution from a given* SSAT *formula $\mathcal{Q} : \varphi$ then $Pr(\mathcal{Q} : \varphi) = p$.*

Corollary 1 follows directly from Lemma 1, namely for the special case $c^p = \emptyset^p$. Theorem 1 shows completeness of S-resolution.

Theorem 1 (Completeness of S-resolution). *If $Pr(\mathcal{Q} : \varphi) = p$ for some* SSAT *formula $\mathcal{Q} : \varphi$ then the empty clause $\emptyset^p$ is derivable by S-resolution.*

Proof. If $\emptyset \in \varphi$, i.e. φ contains the empty clause, then $p = 0$ and the empty clause $\emptyset^0$ is derivable by rule R.1. In the remaining proof, we assume that $\emptyset \notin \varphi$. We prove the theorem by induction over the number of quantifiers in the quantifier prefix $\mathcal{Q}$. For the base case $\mathcal{Q} = Qx$ we distinguish three cases: 1) $\varphi = (\neg x) \wedge (x)$. Then $p = 0$, and $(\neg x)^0$, $(x)^0$ are derivable by R.1, and R.3 finally yields $\emptyset^0$. 2) $\varphi = (\neg x)$. Clauses $(\neg x)^0$ and $(x)^1$ are derivable by R.1 and R.2, resp., the latter since $\models \varphi[\mathtt{false}/x]$. If $Q = \exists$ or $Q = \mathbb{R}^{p_x}$ then $p = 1$ or $p = (1 - p_x)$, and $\emptyset^1$ or $\emptyset^{(1-p_x)}$ can be derived by R.3, respectively. 3) $\varphi = (x)$. Analogously to 2), if $Q = \exists$ or $Q = \mathbb{R}^{p_x}$ then $p = 1$ or $p = p_x$, and $\emptyset^1$ or $\emptyset^{p_x}$ can be derived by R.3, respectively.

In the induction step, we will show that $\emptyset^p$ is derivable for $Pr(Qx\,\mathcal{Q} : \varphi) = p$. Let $p_1 = Pr(\mathcal{Q} : \varphi[\mathtt{true}/x])$ and $p_2 = Pr(\mathcal{Q} : \varphi[\mathtt{false}/x])$. Induction hypothesis assumes that $\emptyset^{p_1}$ and $\emptyset^{p_2}$ are derivable from $\mathcal{Q} : \varphi[\mathtt{true}/x]$ and $\mathcal{Q} : \varphi[\mathtt{false}/x]$.

[4] The proof is similar to [17] but is slightly adapted due to the modified version of S-resolution.

Applying the resolution sequence deriving $\emptyset^{p_1}$ from $\mathcal{Q} : \varphi[\mathtt{true}/x]$ on $Qx\ \mathcal{Q} : \varphi$ yields either $\emptyset^{p_1}$ or $(\neg x)^{p_1}$. Analogously, either $\emptyset^{p_2}$ or $(x)^{p_2}$ is derivable from $Qx\ \mathcal{Q} : \varphi$. If $\emptyset^{p_1}$ (resp. $\emptyset^{p_2}$) was derived then $p = p_1$ (resp. $p = p_2$) by Corollary 1. (Note that if both $\emptyset^{p_1}$ and $\emptyset^{p_2}$ are derivable then $p_1 = p_2$.) Otherwise, i.e. $(\neg x)^{p_1}$ and $(x)^{p_2}$ are derived, application of R.3 gives $\emptyset^{p}$. □

Example. Consider the SSAT formula $\Phi = \text{ꓤ}^{0.8} x_1\ \exists x_2\ \text{ꓤ}^{0.3} x_3 : ((x_1 \vee x_2) \wedge (\neg x_2) \wedge (x_2 \vee x_3))$ with $Pr(\Phi) = 0.24$. Clauses $(x_1 \vee x_2)^0$, $(\neg x_2)^0$, $(x_2 \vee x_3)^0$ are then derivable by R.1. As $x_1 = \mathtt{true}, x_2 = \mathtt{false}, x_3 = \mathtt{true}$ is a satisfying assignment, $\vdash_{\mathsf{R.2}} (\neg x_1 \vee x_2 \vee \neg x_3)^1$. Then, $((\neg x_1 \vee x_2 \vee \neg x_3)^1, (x_2 \vee x_3)^0) \vdash_{\mathsf{R.3}} (\neg x_1 \vee x_2)^{0.3}$, $((\neg x_2)^0, (\neg x_1 \vee x_2)^{0.3}) \vdash_{\mathsf{R.3}} (\neg x_1)^{0.3}$, $((\neg x_2)^0, (x_1 \vee x_2)^0) \vdash_{\mathsf{R.3}} (x_1)^0$, and finally $((\neg x_1)^{0.3}, (x_1)^0) \vdash_{\mathsf{R.3}} \emptyset^{0.24}$.

4 Interpolation for SSAT

Craig interpolation [18] is a well-studied notion in formal logics which has several applications in Computer Science, among them model checking [19, 20]. Given two formulae φ and ψ such that $\varphi \Rightarrow \psi$ is valid, a *Craig interpolant* for (φ, ψ) is a formula $\mathcal{I}$ which refers only to common variables of φ and ψ, and $\mathcal{I}$ is "intermediate" in the sense that $\varphi \Rightarrow \mathcal{I}$ and $\mathcal{I} \Rightarrow \psi$. Such interpolants do trivially exist in all logics permitting quantifier elimination, e.g. in propositional logic. Using the observation that $\varphi \Rightarrow \psi$ holds iff $\varphi \wedge \neg\psi$ is unsatisfiable, this gives rise to an equivalent definition which we refer to in the rest of the paper:[5] given an unsatisfiable formula $\varphi \wedge \neg\psi$, formula $\mathcal{I}$ is an interpolant for (φ, ψ) iff both $\varphi \wedge \neg\mathcal{I}$ and $\mathcal{I} \wedge \neg\psi$ are unsatisfiable and $\mathcal{I}$ mentions only common variables.

In the remainder of this section, we investigate the issue of interpolation for stochastic SAT. We propose a generalization of Craig interpolants suitable for SSAT and show the general existence of such interpolants, alongside with an automatic method for computing them based on S-resolution.

4.1 Generalized Craig interpolants

When approaching a reasonable definition of interpolants for SSAT, the semantics of the non-classical quantifier prefix poses problems: Let $\Phi = \mathcal{Q} : (A \wedge B)$ be an SSAT formula. Each variable in $A \wedge B$ is bound by $\mathcal{Q}$, which provides the probabilistic interpretation of the variables that is lacking without the quantifier prefix. This issue can be addressed by considering the quantifier prefix $\mathcal{Q}$ as the global setting that serves to interpret the quantifier-free part, and consequently interpreting the interpolant also within the scope of $\mathcal{Q}$, thus reasoning about $\mathcal{Q} : (A \wedge \neg\mathcal{I})$ and $\mathcal{Q} : (\mathcal{I} \wedge B)$. A more fundamental problem is that a classical Craig interpolant for Φ only exists if $Pr(\Phi) = 0$, since $A \wedge B$ has to be unsatisfiable by definition of a Craig interpolant which applies iff $Pr(\mathcal{Q} : (A \wedge B)) = 0$. The precondition that $Pr(\mathcal{Q} : (A \wedge B)) = 0$ would be far too restrictive for application of interpolation, as the notion of unsatisfiability of $A \wedge B$ is naturally generalized to satisfiability with insufficient probability, i.e. $Pr(\mathcal{Q} : (A \wedge B))$ being "sufficiently small", in the stochastic setting. Such relaxed requirements actually appear in practice, e.g. in probabilistic verification where safety properties like "a fatal system error is never reachable" are frequently replaced by probabilistic ones like "a fatal system error is

[5] This is of technical nature as SSAT formulae are interpreted by the maximum probability of satisfaction. As the *maximum* probability that an implication $\varphi \Rightarrow \psi$ holds is inappropriate for our purpose, we reason about the maximum satisfaction probability p of the negated implication, i.e. of $\varphi \wedge \neg\psi$, instead. The latter coincides with the *minimum* probability $1 - p$ that $\varphi \Rightarrow \psi$ holds, which is the desired notion.

reachable only with (sufficiently small) probability of at most 0.1‰". Motivated by above facts, interpolants for SSAT should also exist when $A \wedge B$ is satisfiable with reasonably low probability.

The resulting notion of interpolation, which is to be made precise in Definition 1, matches the following intuition. In classical Craig interpolation, when performed in logics permitting quantifier elimination, the Craig interpolants of $(A, \neg B)$ form a lattice with implication as its ordering, $A^{\exists} = \exists a_1, \dots a_\alpha : A$ as its bottom element and $\overline{B}^{\forall} = \neg\exists b_1, \dots b_\beta : B$ as its top element, where the a_i and b_i are the local variables of A and of B, respectively. In the generalized setting required for SSAT[6], $A \Rightarrow \neg B$ and thus $A^{\exists} \Rightarrow \overline{B}^{\forall}$ may no longer hold such that the above lattice can collapse to the empty set. To preserve the overall structure, it is however natural to use the lattice of propositional formulae "in between" $A^{\exists} \wedge \overline{B}^{\forall}$ as bottom element and $A^{\exists} \vee \overline{B}^{\forall}$ as top element instead. This lattice is non-empty and coincides with the classical one whenever $A \wedge B$ is unsatisfiable.

Definition 1 (Generalized Craig interpolant). *Let A, B be propositional formulae and $V_A := Var(A) \setminus Var(B) = \{a_1, \dots, a_\alpha\}$, $V_B := Var(B) \setminus Var(A) = \{b_1, \dots, b_\beta\}$, $V_{A,B} := Var(A) \cap Var(B)$, $A^{\exists} = \exists a_1, \dots, a_\alpha : A$, and $\overline{B}^{\forall} = \neg\exists b_1, \dots, b_\beta : B$. A propositional formula $\mathcal{I}$ is called* generalized Craig interpolant for (A, B) *iff $Var(\mathcal{I}) \subseteq V_{A,B}$, $\left(A^{\exists} \wedge \overline{B}^{\forall}\right) \Rightarrow \mathcal{I}$, and $\mathcal{I} \Rightarrow \left(A^{\exists} \vee \overline{B}^{\forall}\right)$.*

Given any two propositional formulae A and B, the four quantifier-free propositional formulae equivalent to $A^{\exists} \wedge \overline{B}^{\forall}$, to $A^{\exists}$, to $\overline{B}^{\forall}$, and to $A^{\exists} \vee \overline{B}^{\forall}$, are generalized Craig interpolants for (A, B). These generalized interpolants always exist since propositional logic has quantifier elimination.

While Definition 1 motivates the generalized notion of Craig interpolant from a model-theoretic perspective, we will state an equivalent definition of generalized Craig interpolants in Lemma 2 that substantiates the intuition of generalized interpolants and allows for an illustration of their geometric shape. Given two formulae A and B, the idea of generalized Craig interpolant is depicted in Fig. 2. The set of solutions of A is defined by the rectangle on the $V_A, V_{A,B}$-plane with a cylindrical extension in V_B-direction as A does not contain variables in V_B. Similarly, the solution set of B is given by the triangle on the $V_B, V_{A,B}$-plane and its cylinder in V_A-direction. The solution set of $A \wedge B$ is then determined by the intersection of both cylinders. Since $A \wedge B \wedge \neg(A \wedge B)$ is unsatisfiable, the sets $A \wedge \neg(A \wedge B)$ and $B \wedge \neg(A \wedge B)$ are disjoint. This gives us the possibility to talk about interpolants wrt. these sets. However, a formula $\mathcal{I}$ over only common variables in $V_{A,B}$ may not exist when demanding $A \wedge \neg(A \wedge B) \wedge \neg\mathcal{I}$ and $\mathcal{I} \wedge B \wedge \neg(A \wedge B)$ to be unsatisfiable. This is indicated by Fig. 2 and proven by the simple example $A = (a)$, $B = (b)$. As $V_{A,B} = \emptyset$, $\mathcal{I}$ is either `true` or `false`. In first case, $\texttt{true} \wedge (b) \wedge \neg(a \wedge b)$ is satisfiable, while $(a) \wedge \neg(a \wedge b) \wedge \neg\texttt{false}$ is in second case. If we however project the solution set of $A \wedge B$ onto the $V_{A,B}$-axis and subtract the resulting hyperplane $\mathcal{S}_{A,B}$ from A and B then such a formula $\mathcal{I}$ over $V_{A,B}$-variables exists. The next lemma formalizes such generalized interpolants $\mathcal{I}$ and shows their equivalence to the ones from Definition 1.

Lemma 2 (Generalized Craig interpolant for SSAT). *Let $\Phi = \mathcal{Q} : (A \wedge B)$ be some SSAT formula, V_A, V_B, $V_{A,B}$ be defined as in Definition 1, and $\mathcal{S}_{A,B}$ be a propositional formula with $Var(\mathcal{S}_{A,B}) \subseteq V_{A,B}$ s.t. $\mathcal{S}_{A,B} \equiv \exists a_1, \dots, a_\alpha, b_1, \dots, b_\beta : (A \wedge B)$. Then, a propositional formula $\mathcal{I}$ is a generalized Craig interpolant for (A, B) iff the following properties are satisfied.*

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

[6] Though the concept seems to be more general, this paper addresses SSAT only.

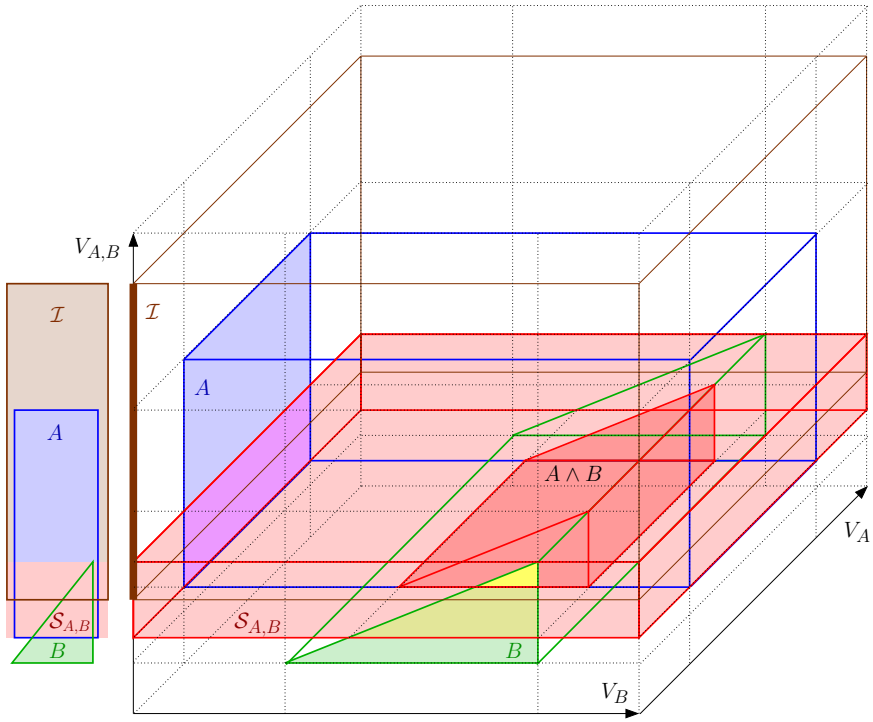

Fig. 2. Geometric interpretation of a generalized Craig interpolant $\mathcal{I}$. V_A-, V_B-, and $V_{A,B}$-axes denote assignments of variables occurring only in A, only in B, and in both A and B, respectively.

2. $Pr(\mathcal{Q} : (A \wedge \neg \mathcal{S}_{A,B} \wedge \neg \mathcal{I})) = 0$
3. $Pr(\mathcal{Q} : (\mathcal{I} \wedge B \wedge \neg \mathcal{S}_{A,B})) = 0$

Proof. As $Var(\mathcal{I}) \subseteq V_{A,B}$ holds for generalized Craig interpolants $\mathcal{I}$, it remains to show that $(A^{\exists} \wedge \overline{B}^{\forall}) \Rightarrow \mathcal{I}$ and $\mathcal{I} \Rightarrow (A^{\exists} \vee \overline{B}^{\forall})$ iff $Pr(\mathcal{Q} : (A \wedge \neg \mathcal{S}_{A,B} \wedge \neg \mathcal{I})) = 0$ and $Pr(\mathcal{Q} : (\mathcal{I} \wedge B \wedge \neg \mathcal{S}_{A,B})) = 0$. Observe that $\models (A^{\exists} \wedge \overline{B}^{\forall}) \Rightarrow \mathcal{I}$ iff $\models \forall a_1, \ldots, a_\alpha : (A \wedge \overline{B}^{\forall}) \Rightarrow \mathcal{I}$ iff $\models (A \wedge \overline{B}^{\forall}) \Rightarrow \mathcal{I}$ iff $\models (A \wedge (\neg A^{\exists} \vee \overline{B}^{\forall})) \Rightarrow \mathcal{I}$ iff $\models (A \wedge \neg \mathcal{S}_{A,B}) \Rightarrow \mathcal{I}$ iff $A \wedge \neg \mathcal{S}_{A,B} \wedge \neg \mathcal{I}$ is unsatisfiable iff $Pr(\mathcal{Q} : (A \wedge \neg \mathcal{S}_{A,B} \wedge \neg \mathcal{I})) = 0$. Analogously, $\models \mathcal{I} \Rightarrow (A^{\exists} \vee \overline{B}^{\forall})$ iff $\models \forall b_1, \ldots, b_\beta : \mathcal{I} \Rightarrow (A^{\exists} \vee \neg B)$ iff $\models \mathcal{I} \Rightarrow (A^{\exists} \vee \neg B)$ iff $\models \mathcal{I} \Rightarrow ((A^{\exists} \wedge \neg \overline{B}^{\forall}) \vee \neg B)$ iff $\models \mathcal{I} \Rightarrow (\mathcal{S}_{A,B} \vee \neg B)$ iff $\mathcal{I} \wedge \neg \mathcal{S}_{A,B} \wedge B$ is unsatisfiable iff $Pr(\mathcal{Q} : (\mathcal{I} \wedge B \wedge \neg \mathcal{S}_{A,B})) = 0$. □

We remark that the concept of generalized Craig interpolants is a generalization of Craig interpolants in the sense that whenever $A \wedge B$ is unsatisfiable, i.e. when $Pr(\mathcal{Q} : (A \wedge B)) = 0$, then each generalized Craig interpolant $\mathcal{I}$ for (A, B) actually is a Craig interpolant for A and B since $\mathcal{S}_{A,B} \equiv \texttt{false}$.

4.2 Computation of generalized Craig interpolants

In this subsection, we proceed to the efficient computation of generalized Craig interpolants. The remark following Definition 1 shows that generalized interpolants can in principle be computed by *explicit* quantifier elimination methods, like Shannon's expansion or BDDs. We aim at a more efficient method based on S-resolution, akin to resolution-based Craig interpolation for propositional SAT by Pudlák [21], as has been integrated into DPLL-based SAT solvers featuring conflict analysis and successfully applied to symbolic model checking [19, 20].

Observe that on SSAT formulae $\mathcal{Q} : (A \wedge B)$, Pudlák's algorithm, which has unsatisfiability of $A \wedge B$ as precondition, will not work in general. When instead considering the unsatisfiable formula $A \wedge B \wedge \neg\mathcal{S}_{A,B}$ with $\neg\mathcal{S}_{A,B}$ in CNF then Pudlák's method would be applicable and would actually produce a generalized interpolant. The main drawback of this approach, however, is the explicit construction of $\neg\mathcal{S}_{A,B}$, again calling for explicit quantifier elimination.

We now propose an algorithm based on S-resolution for computing generalized Craig interpolants which operates directly on $A \wedge B$ without adding $\neg\mathcal{S}_{A,B}$, and thus does not comprise any preprocessing involving quantifier elimination. For this purpose, rules of S-resolution are enhanced to deal with pairs (c^p, I) of annotated clauses c^p and propositional formulae I. Such formulae I are in a certain sense *intermediate* generalized interpolants, i.e. generalized interpolants for subformulae arising from instantiating some variables by partial assignments that falsify c (cf. Lemma 3). Once a pair $(\emptyset^p, I)$ comprising the empty clause is derived, I thus is a generalized Craig interpolant for the given SSAT formula. This augmented S-resolution, which we call *interpolating S-resolution*, is defined by rules R'.1, R'.2, and R'.3. The construction of intermediate interpolants I in R'.1 and R'.3 coincides with the classical rules by Pudlák [21], while R'.2 misses a corresponding counterpart. The rationale is that R'.2 (or rather R.2) refers to satisfying valuations τ of $A \wedge B$, which do not exist in classical interpolation. As $A \wedge B$ becomes a tautology after substituting the partial assignment τ from R.2 into it, its quantified variant $\mathcal{S}_{A,B} = \exists a_1, \ldots, b_1, \ldots : A \wedge B$ also becomes tautological under the same substitution $\mathcal{S}_{A,B}[\tau(x_1)/x_1, \ldots, \tau(x_i)/x_i]$. Consequently, $\neg\mathcal{S}_{A,B}[\tau(x_1)/x_1, \ldots, \tau(x_i)/x_i]$ is unsatisfiable, and so are $(A \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1, \ldots, \tau(x_i)/x_i]$ and $(B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1, \ldots, \tau(x_i)/x_i]$. This implies that the actual intermediate interpolant in R'.2 can be chosen arbitrarily over variables in $V_{A,B}$. This freedom will allow us to control the geometric extent of generalized interpolants within the "don't care"-region provided by the models of $\mathcal{S}_{A,B}$ (cf. Corollary 3).

$$\text{(R'.1)} \qquad \frac{c \vdash_{\mathsf{R.1}} c^p, I = \begin{cases} \texttt{false} & ;\ c \in A \\ \texttt{true} & ;\ c \in B \end{cases}}{(c^p, I)}$$

$$\text{(R'.2)} \qquad \frac{\vdash_{\mathsf{R.2}} c^p, I \text{ is any formula over } V_{A,B}}{(c^p, I)}$$

$$\text{(R'.3)} \qquad \frac{\begin{array}{c}((c_1 \vee \neg x)^{p_1}, I_1), ((c_2 \vee x)^{p_2}, I_2), \\ ((c_1 \vee \neg x)^{p_1}, (c_2 \vee x)^{p_2}) \vdash_{\mathsf{R.3}} (c_1 \vee c_2)^p, \\ I = \begin{cases} I_1 \vee I_2 & ;\ x \in V_A \\ I_1 \wedge I_2 & ;\ x \in V_B \\ (\neg x \vee I_1) \wedge (x \vee I_2) & ;\ x \in V_{A,B} \end{cases}\end{array}}{((c_1 \vee c_2)^p, I)}$$

The following lemma establishes the theoretical foundation of computing generalized Craig interpolants by interpreting the derived pairs (c^p, I).

Lemma 3. *Let $\Phi = \mathcal{Q} : (A \wedge B)$ with $\mathcal{Q} = Q_1 x_1 \ldots Q_n x_n$ be some SSAT formula, and the pair (c^p, I) be derivable from Φ by interpolating S-resolution, where $\mathcal{Q}(c) = Q_1 x_1 \ldots Q_i x_i$. Then, for each $\tau : Var(A \wedge B) \downarrow_i \to \mathbb{B}$ with $\forall x \in Var(c) : \tau(x) = \mathit{ff}_c(x)$ it holds that*

1. *$Var(I) \subseteq V_{A,B}$,*
2. *$Pr(Q_{i+1} x_{i+1} \ldots Q_n x_n : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1] \ldots [\tau(x_i)/x_i]) = 0$, and*

3. $Pr(Q_{i+1}x_{i+1}\dots Q_n x_n : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) = 0.$

Proof. We prove the lemma by induction over application of the interpolating S-resolution rules R'.1-R'.3. In the base case, we can just apply R'.1 and R'.2. Item 1 clearly holds for both rules since I contains only variables in $V_{A,B}$. Let us consider R'.1 first. If $c \in A$ then $I = \texttt{false}$. By construction of τ, i.e. c evaluates to `false` under τ, it follows that $A[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]$ is unsatisfiable and thus

$$Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) = 0.$$

As $I = \texttt{false}$, immediately

$$Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) = 0.$$

If $c \in B$ then $I = \texttt{true}$. Obviously,

$$Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) = 0$$

and by construction of τ,

$$Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) = 0.$$

For rule R'.2, we have $\models (A \wedge B)[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]$ which immediately implies that $\models (\exists a_1,\dots,a_\alpha, b_1,\dots,b_\beta : (A \wedge B))[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]$, i.e. $\models \mathcal{S}_{A,B}[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]$ by definition of $\mathcal{S}_{A,B}$. Rephrasing the latter, $\neg\mathcal{S}_{A,B}[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]$ is unsatisfiable. Consequently, for any propositional formula I

$$\begin{aligned} Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) &= 0,\\ Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) &= 0. \end{aligned}$$

We now assume that the lemma holds for all clauses in the premises of rule R'.3. Then, by construction of I, item 1 clearly holds for I, i.e. $Var(I) \subseteq V_{A,B}$. Induction hypothesis assumes that

$$\begin{aligned} Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_1)[\tau_1(x_1)/x_1]\dots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) &= 0,\\ Pr(\mathcal{Q}' : (I_1 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_1(x_1)/x_1]\dots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) &= 0 \end{aligned}$$

holds for $((c_1 \vee \neg x)^{p_1}, I_1)$, and

$$\begin{aligned} Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_2)[\tau_2(x_1)/x_1]\dots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) &= 0,\\ Pr(\mathcal{Q}' : (I_2 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_2(x_1)/x_1]\dots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) &= 0 \end{aligned}$$

for $((c_2 \vee x)^{p_2}, I_2)$ where $x_j = x$, $j \geq i+1$, and $\mathcal{Q}' = Q_{j+1}x_{j+1}\dots Q_n x_n$. Let $\tau : Var(A \wedge B) \downarrow_{j-1} \to \mathbb{B}$ be any assignment with $\tau(x) = \tau_1(x)$ if $x \in Var(c_1)$ and $\tau(x) = \tau_2(x)$ if $x \in Var(c_2)$. Note that τ is well-defined as $\not\models (c_1 \vee c_2)$, i.e. for each $x \in Var(c_1) \cap Var(c_2) : \tau_1(x) = \tau_2(x)$. We will now show that

$$\begin{aligned} Pr_A &:= Pr(Q_j x_j \mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_{j-1})/x_{j-1}]) &&= 0,\\ Pr_B &:= Pr(Q_j x_j \mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_{j-1})/x_{j-1}]) &&= 0 \end{aligned}$$

by showing that

$$\begin{aligned} Pr_{A,x} &:= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) &&= 0,\\ Pr_{A,\neg x} &:= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) &&= 0,\\ Pr_{B,x} &:= Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) &&= 0,\\ Pr_{B,\neg x} &:= Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) &&= 0. \end{aligned}$$

We therefore distinguish the three cases $x_j \in V_A$, $x_j \in V_B$, and $x_j \in V_{A,B}$.

First, let be $x_j \in V_A$. Then, $I = I_1 \vee I_2$. By construction of τ and I and by induction hypothesis,

$$\begin{aligned}
Pr_{A,x} &\leq Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau_1(x_1)/x_1] \dots [\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_1 \wedge \neg I_2)[\tau_1(x_1)/x_1] \dots [\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&\leq Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_1)[\tau_1(x_1)/x_1] \dots [\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= 0, \\
Pr_{A,\neg x} &\leq Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau_2(x_1)/x_1] \dots [\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_1 \wedge \neg I_2)[\tau_2(x_1)/x_1] \dots [\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&\leq Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_2)[\tau_2(x_1)/x_1] \dots [\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= 0.
\end{aligned}$$

As $x_j \notin \mathit{Var}(I) \cup \mathit{Var}(B) \cup \mathit{Var}(\neg\mathcal{S}_{A,B})$, for each $v \in \mathbb{B}$ it holds that

$$\begin{aligned}
&Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}][v/x_j]) \\
&= Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}])
\end{aligned}$$

which implies $Pr_{B,x} = Pr_{B,\neg x}$. Thus, by construction of I and τ, and by induction hypothesis,

$$\begin{aligned}
Pr_{B,x} &= Pr_{B,\neg x} \\
&= Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}]) \\
&= Pr(\mathcal{Q}' : ((I_1 \vee I_2) \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}]) \\
&= Pr(\mathcal{Q}' : \begin{pmatrix} (I_1 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}] \\ \vee\ (I_2 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}] \end{pmatrix} \\
&\leq Pr(\mathcal{Q}' : \begin{pmatrix} (I_1 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_1(x_1)/x_1] \dots [\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j] \\ \vee\ (I_2 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_2(x_1)/x_1] \dots [\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j] \end{pmatrix} \\
&= 0.
\end{aligned}$$

For the latter step, note that if $Pr(\mathcal{Q} : \varphi_1) = 0$ and $Pr(\mathcal{Q} : \varphi_2) = 0$ then $Pr(\mathcal{Q} : (\varphi_1 \vee \varphi_2) = 0$ since $Pr(\mathcal{Q} : \varphi) = 0$ iff φ is unsatisfiable.[7]

Second, let be $x_j \in V_B$. Then, $I = I_1 \wedge I_2$. As $x_j \notin \mathit{Var}(A) \cup \mathit{Var}(\neg\mathcal{S}_{A,B}) \cup \mathit{Var}(\neg I)$, with the same argument as above,

$$\begin{aligned}
Pr_{A,x} &= Pr_{A,\neg x} \\
&= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}]) \\
&= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge (\neg I_1 \vee \neg I_2))[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}]) \\
&= Pr(\mathcal{Q}' : \begin{pmatrix} (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_1)[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}] \\ \vee\ (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_2)[\tau(x_1)/x_1] \dots [\tau(x_{j-1})/x_{j-1}] \end{pmatrix} \\
&\leq Pr(\mathcal{Q}' : \begin{pmatrix} (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_1)[\tau_1(x_1)/x_1] \dots [\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j] \\ \vee\ (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_2)[\tau_2(x_1)/x_1] \dots [\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j] \end{pmatrix} \\
&= 0.
\end{aligned}$$

[7] This statement is not true in general if $\mathcal{Q}$ also contains *universal* quantifiers which is not the case in this paper. However, extentions of SSAT involving universal quantifiers have been also considered in the literature, cf. [22].

Again following the reasoning above, we have

$$\begin{aligned}
Pr_{B,x} &\leq Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= Pr(\mathcal{Q}' : (I_1 \wedge I_2 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&\leq Pr(\mathcal{Q}' : (I_1 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= 0, \\
Pr_{B,\neg x} &\leq Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= Pr(\mathcal{Q}' : (I_1 \wedge I_2 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&\leq Pr(\mathcal{Q}' : (I_2 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= 0.
\end{aligned}$$

Third, let be $x_j \in V_{A,B}$. Then, $I = (\neg x_j \vee I_1) \wedge (x_j \vee I_2)$, and we deduce

$$\begin{aligned}
Pr_{A,x} &\leq Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \\
&\qquad \wedge ((x_j \wedge \neg I_1) \vee (\neg x_j \wedge \neg I_2)))[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_1)[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= 0, \\
Pr_{A,\neg x} &\leq Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \\
&\qquad \wedge ((x_j \wedge \neg I_1) \vee (\neg x_j \wedge \neg I_2)))[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= Pr(\mathcal{Q}' : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I_2)[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= 0, \\
Pr_{B,x} &\leq Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= Pr(\mathcal{Q}' : ((\neg x_j \vee I_1) \wedge (x_j \vee I_2) \wedge B \\
&\qquad \wedge \neg\mathcal{S}_{A,B})[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= Pr(\mathcal{Q}' : (I_1 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_1(x_1)/x_1]\ldots[\tau_1(x_{j-1})/x_{j-1}][\texttt{true}/x_j]) \\
&= 0, \\
Pr_{B,\neg x} &\leq Pr(\mathcal{Q}' : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= Pr(\mathcal{Q}' : ((\neg x_j \vee I_1) \wedge (x_j \vee I_2) \wedge B \\
&\qquad \wedge \neg\mathcal{S}_{A,B})[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= Pr(\mathcal{Q}' : (I_2 \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau_2(x_1)/x_1]\ldots[\tau_2(x_{j-1})/x_{j-1}][\texttt{false}/x_j]) \\
&= 0.
\end{aligned}$$

Having shown that $Pr_{A,x} = Pr_{A,\neg x} = Pr_{B,x} = Pr_{B,\neg x} = 0$, we can now prove the intermediate result above, i.e. $Pr_A = Pr_B = 0$. If $Q_j = \exists$ then $Pr_A = \max(Pr_{A,x}, Pr_{A,\neg x}) = 0$ and $Pr_B = \max(Pr_{B,x}, Pr_{B,\neg x}) = 0$, and if $Q_j = \text{Ƃ}^{p_x}$ then $Pr_A = p_x \cdot Pr_{A,x} + (1-p_x) \cdot Pr_{A,\neg x} = 0$ and $Pr_B = p_x \cdot Pr_{B,x} + (1-p_x) \cdot Pr_{B,\neg x} = 0$.

To finish the proof, we finally need to show that items 2 and 3, i.e.

$$\begin{aligned}
Pr(Q_{i+1}x_{i+1}\ldots Q_n x_n : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\ldots[\tau(x_i)/x_i]) &= 0, \\
Pr(Q_{i+1}x_{i+1}\ldots Q_n x_n : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\ldots[\tau(x_i)/x_i]) &= 0
\end{aligned}$$

follow from $Pr_A = Pr_B = 0$, i.e. from

$$\begin{aligned}
Pr(Q_j x_j \ldots Q_n x_n : (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\ldots[\tau(x_{j-1})/x_{j-1}]) &= 0, \\
Pr(Q_j x_j \ldots Q_n x_n : (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\ldots[\tau(x_{j-1})/x_{j-1}]) &= 0.
\end{aligned}$$

If $j = i+1$ then the result is obvious. Otherwise, i.e. if $j > i+1$, the variables $x_{i+1}, \ldots, x_{j-1}$ do not occur in the derived clause $(c_1 \vee c_2)$ since $\mathcal{Q}(c_1 \vee c_2) = Q_1 x_1 \ldots Q_i x_i$. By definition of assignment τ, for $k = j-1$ to $i+1$ we may therefore

successively conclude that

$$\begin{aligned}
&Pr(Q_{k+1}x_{k+1}\dots Q_n x_n : \\
&\qquad (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_{k-1})/x_{k-1}][\mathtt{true}/x_k]) &&= 0,\\
&Pr(Q_{k+1}x_{k+1}\dots Q_n x_n : \\
&\qquad (A \wedge \neg\mathcal{S}_{A,B} \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_{k-1})/x_{k-1}][\mathtt{false}/x_k]) &&= 0,\\
&Pr(Q_{k+1}x_{k+1}\dots Q_n x_n : \\
&\qquad (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_{k-1})/x_{k-1}][\mathtt{true}/x_k]) &&= 0,\\
&Pr(Q_{k+1}x_{k+1}\dots Q_n x_n : \\
&\qquad (I \wedge B \wedge \neg\mathcal{S}_{A,B})[\tau(x_1)/x_1]\dots[\tau(x_{k-1})/x_{k-1}][\mathtt{false}/x_k]) &&= 0.
\end{aligned}$$

From case $k = i + 1$ the result immediately follows. □

Completeness of S-resolution, as stated in Theorem 1, together with the above Lemma 3, applied to the derived pair $(\emptyset^p, I)$, yields

Corollary 2 (Generalized Craig interpolants computation). *If $\Phi = \mathcal{Q} : (A \wedge B)$ is an SSAT formula then a generalized Craig interpolant for (A, B) can be computed by interpolating S-resolution.*

Note that computation of generalized interpolants does not depend on the actual truth state of $A \wedge B$. The next observation facilitates to effectively control the geometric extent of generalized Craig interpolants within the "don't care"-region $\mathcal{S}_{A,B}$. This result will be useful in probabilistic model checking in Section 5.

Corollary 3 (Controlling generalized Craig interpolants computation). *If $I = \mathtt{true}$ is used within each application of rule* R'.2 *then $Pr(\mathcal{Q} : (A \wedge \neg\mathcal{I})) = 0$. Likewise, if $I = \mathtt{false}$ is used in rule* R'.2 *then $Pr(\mathcal{Q} : (\mathcal{I} \wedge B)) = 0$.*

Proof. The proof works analogously to the one of Lemma 3. For the base case, it is clear that the desired property for R'.1 is independent of $\neg\mathcal{S}_{A,B}$. For R'.2, if $I = \mathtt{true}$ then clearly $Pr(\mathcal{Q}' : (A \wedge \neg I)[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) = 0$, and if $I = \mathtt{false}$ then $Pr(\mathcal{Q}' : (I \wedge B)[\tau(x_1)/x_1]\dots[\tau(x_i)/x_i]) = 0$. Then, we can modify the induction hypothesis: for case "$I = \mathtt{true}$ in R'.2", we assume that $Pr(\mathcal{Q}' : (A \wedge \neg I_1)[\tau_1(x_1)/x_1]\dots[\mathtt{true}/x_j]) = 0$, $Pr(\mathcal{Q}' : (A \wedge \neg I_2)[\tau_2(x_1)/x_1]\dots[\mathtt{false}/x_j]) = 0$, and for "$I = \mathtt{false}$ in R'.2" that $Pr(\mathcal{Q}' : (I_1 \wedge B)[\tau_1(x_1)/x_1]\dots[\mathtt{true}/x_j]) = 0$, $Pr(\mathcal{Q}' : (I_2 \wedge B)[\tau_2(x_1)/x_1]\dots[\mathtt{false}/x_j]) = 0$. The induction step then follows the same reasoning as in the remaining proof of Lemma 3. □

Observe that the special interpolants $\mathcal{I}$ from Corollary 3 relate to the classical strongest and weakest Craig interpolants $A^{\exists}$ and $\overline{B}^{\forall}$, resp., in the following sense: $Pr(\mathcal{Q} : (A \wedge \neg\mathcal{I})) = 0$ iff $\models A \Rightarrow \mathcal{I}$ iff $\models \forall a_1, \dots, a_\alpha : (A \Rightarrow \mathcal{I})$ iff $\models (A^{\exists} \Rightarrow \mathcal{I})$, as $a_1, \dots, a_\alpha$ do not occur in $\mathcal{I}$. Analogously, $Pr(\mathcal{Q} : (\mathcal{I} \wedge B)) = 0$ iff $\models \mathcal{I} \Rightarrow \neg B$ iff $\models \forall b_1, \dots, b_\beta : (\mathcal{I} \Rightarrow \neg B)$ iff $\models \mathcal{I} \Rightarrow \overline{B}^{\forall}$.

5 Interpolation-based probabilistic model checking

In this section, we investigate the application of generalized Craig interpolation in probabilistic model checking. As a model we consider symbolically represented finite-state Markov decision processes (MDPs), which we check wrt. to probabilistic state reachability properties. That is, given a set of target states T in an MDP $\mathcal{M}$, the goal is to maximize the probability $P^{\pi}_{\mathcal{M}}(T)$ of reaching T over all policies π resolving the non-determinism in $\mathcal{M}$. When considering T as *bad* states of $\mathcal{M}$, e.g. as fatal system errors, this maximum probability $P^{\max}_{\mathcal{M}}(T) := \max_\pi P^{\pi}_{\mathcal{M}}(T)$ reveals the *worst-case* probability of bad system behavior. A *safety property* for $\mathcal{M}$

requires that this worst-case probability does not exceed some given threshold value $\theta \in [0,1)$, i.e. $P^{\max}_{\mathcal{M}}(T) \leq \theta$.

In [8], we proposed a symbolic *falsification* procedure for such safety properties. Though the approach in [8] is based on SSMT (arithmetic extension of SSAT) and considers the more general class of discrete-time probabilistic hybrid systems, which roughly are MDPs with arithmetic-logical transition guards and actions, the same procedure restricted to SSAT is applicable for finite-state MDPs. The key idea here is to adapt *bounded model checking* (BMC) [23] to the probabilistic case by encoding step-bounded reachability as an SSAT problem: like in classical BMC, the initial states, the transition relation, and the target states of an MDP are symbolically encoded by propositional formulae in CNF, namely by $\mathit{Init}(\boldsymbol{s})$, $\mathit{Trans}(\boldsymbol{s}, \boldsymbol{nt}, \boldsymbol{pt}, \boldsymbol{s'})$, and $\mathit{Target}(\boldsymbol{s})$, resp., where the propositional variable vector $\boldsymbol{s}$ represents the system state before and $\boldsymbol{s'}$ after a transition step. To keep track of the *non-deterministic* and *probabilistic* selections of transitions in $\mathit{Trans}(\boldsymbol{s}, \boldsymbol{nt}, \boldsymbol{pt}, \boldsymbol{s'})$, we further introduce propositional variables $\boldsymbol{nt}$ and $\boldsymbol{pt}$ to encode non-deterministic and probabilistic transition choices, respectively. Assignments to these variables determine which of possibly multiple available transitions departing from $\boldsymbol{s}$ is taken. In contrast to traditional BMC, all variables are quantified: all state variables $\boldsymbol{s}$ and $\boldsymbol{s'}$ are existentially quantified in the prefixes $\mathcal{Q}_{\boldsymbol{s}}$ and $\mathcal{Q}_{\boldsymbol{s'}}$. The transition-selection variables $\boldsymbol{nt}$ encoding non-deterministic choice are *existentially quantified* by $\mathcal{Q}_{\boldsymbol{nt}}$, while the probabilistic selector variables $\boldsymbol{pt}$ are bound by *randomized quantifiers* in $\mathcal{Q}_{\boldsymbol{pt}}$.[8] Let be $\boldsymbol{t} := \boldsymbol{nt} \cup \boldsymbol{pt}$ and $\mathcal{Q}_{\boldsymbol{t}} := \mathcal{Q}_{\boldsymbol{nt}} \mathcal{Q}_{\boldsymbol{pt}}$. Due to [8, Proposition 1], the maximum probability of reaching the target sates in a given MDP from the initial states within k transition steps is equivalent to the satisfaction probability

$$lb_k := Pr\Big(\mathcal{Q}(k) : \Big(\overbrace{\mathit{Init}(\boldsymbol{s}_0) \wedge \textstyle\bigwedge_{i=1}^{k} \mathit{Trans}(\boldsymbol{s}_{i-1}, \boldsymbol{t}_i, \boldsymbol{s}_i)}^{\text{states reachable within } k \text{ steps}} \wedge \overbrace{\Big(\textstyle\bigvee_{i=0}^{k} \mathit{Target}(\boldsymbol{s}_i)\Big)}^{\text{hit target states}}\Big)\Big) \tag{1}$$

with $\mathcal{Q}(k) := \mathcal{Q}_{\boldsymbol{s}_0} \mathcal{Q}_{\boldsymbol{t}_1} \mathcal{Q}_{\boldsymbol{s}_1} \ldots \mathcal{Q}_{\boldsymbol{s}_{k-1}} \mathcal{Q}_{\boldsymbol{t}_k} \mathcal{Q}_{\boldsymbol{s}_k}$. The probability lb_k can be determined by SSAT solvers. This symbolic approach, called *probabilistic bounded model checking* (PBMC), produces valid *lower* bounds lb_k for $P^{\max}_{\mathcal{M}}(T)$. Thus, PBMC is able to *falsify* a safety property once $lb_k > \theta$ for some k. However, the development of a corresponding counterpart based on SSAT that is able to compute safe *upper* bounds ub_k for $P^{\max}_{\mathcal{M}}(T)$ was left as an open question. Such an approach would permit to *verify* a safety property once $ub_k \leq \theta$ for some k.

We now propose such a verification procedure based on generalized Craig interpolation. The algorithm proceeds in two phases. Phase 1 computes a symbolic representation of an *overapproximation of the backward reachable state set*. This can be integrated into PBMC, as used to falsify the probabilistic safety property. Whenever this falsification fails for a given step depth k, we apply generalized Craig interpolation to the (just failed) PBMC proof to compute a *symbolic overapproximation of the backward reachable state set* at depth k and then proceed to PBMC at some higher depth $k' > k$. As an alternative to the integration into PBMC, interpolants describing the backward reachable state sets can be successively extended by "stepping" them by prepending another transition, as explained below. In either case, phase 1 ends when the backward reachable state set becomes stable, in which case we have computed a symbolic overapproximation of the whole backward-reachable state set. In the second phase, we construct an SSAT formula with parameter k that forces the system to *stay within the backward reachable state set* for k steps. The maximum satisfaction probability of that SSAT formula then gives an upper bound on the maximum probability of reaching the target states. The rationale is that system executions leaving the backward reachable states will never reach the target states.

[8] Non-deterministic branching of n alternatives can be represented by a binary tree of depth $\lceil \log_2 n \rceil$ and probabilistic branching by a sequence of at most $n-1$ binary branches, yielding $\lceil \log_2 n \rceil$ existential and $n-1$ randomized quantifiers, respectively.

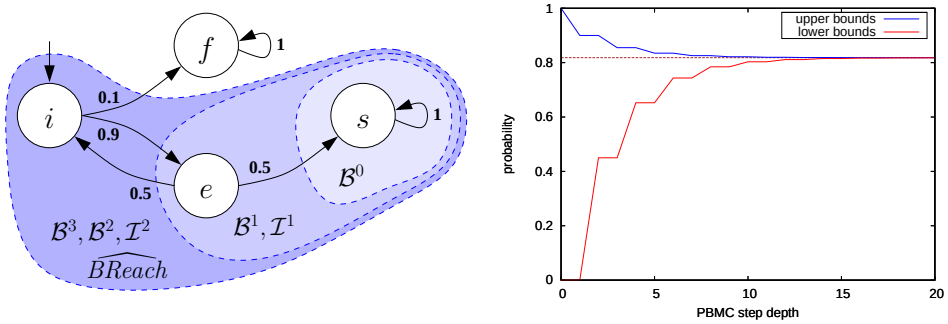

Fig. 3. A simple DTMC $\mathcal{M}$ (left) and lower and upper bounds on probability of reaching s over the PBMC step depth (right).

Phase 1. Given an SSAT encoding of an MDP $\mathcal{M}$ as above, the state-set predicate $\mathcal{B}^k(\boldsymbol{s})$ for $k \in \mathbb{N}_{\geq 0}$ over state variables $\boldsymbol{s}$ is inductively defined as $\mathcal{B}^0(\boldsymbol{s}) := \mathit{Target}(\boldsymbol{s})$, and $\mathcal{B}^{k+1}(\boldsymbol{s}) := \mathcal{B}^k(\boldsymbol{s}) \vee \mathcal{I}^{k+1}(\boldsymbol{s})$ where $\mathcal{I}^{k+1}(\boldsymbol{s}_{j-1})$ is a generalized Craig interpolant for $(\mathit{Trans}(\boldsymbol{s}_{j-1}, \boldsymbol{t}_j, \boldsymbol{s}_j) \wedge \mathcal{B}^k(\boldsymbol{s}_j), \mathit{Init}(\boldsymbol{s}_0) \wedge \bigwedge_{i=1}^{j-1} \mathit{Trans}(\boldsymbol{s}_{i-1}, \boldsymbol{t}_i, \boldsymbol{s}_i))$ with $j \geq 1$ wrt. to SSAT formula

$$\mathcal{Q}(j) : \Big(\overbrace{\mathit{Init}(\boldsymbol{s}_0) \wedge \bigwedge_{i=1}^{j-1} \mathit{Trans}(\boldsymbol{s}_{i-1}, \boldsymbol{t}_i, \boldsymbol{s}_i)}^{j-1 \text{ steps ``forward'' } (=B)} \wedge \overbrace{\mathit{Trans}(\boldsymbol{s}_{j-1}, \boldsymbol{t}_j, \boldsymbol{s}_j) \wedge \mathcal{B}^k(\boldsymbol{s}_j)}^{\text{one step ``backward'' } (=A)} \Big) \tag{2}$$

Observe that each $\mathcal{I}^{k+1}(\boldsymbol{s})$ can be computed by interpolating S-resolution if we rewrite $\mathcal{B}^k(\boldsymbol{s})$ into CNF (which is always possible in linear time by adding auxiliary V_A-variables). When doing so, we take $I = \texttt{true}$ in each application of R'.2 such that $\mathcal{B}^k(\boldsymbol{s})$ overapproximates all system states backward reachable from target states within k steps due to Corollary 3. Whenever $\mathcal{B}^k(\boldsymbol{s})$ stabilizes, i.e. $\mathcal{B}^{k+1}(\boldsymbol{s}) \Rightarrow \mathcal{B}^k(\boldsymbol{s})$, we can be sure that $\widehat{\mathit{BReach}}(\boldsymbol{s}) := \mathcal{B}^k(\boldsymbol{s})$ overapproximates all backward reachable states.

Note that parameter $j \geq 1$ can be chosen arbitrarily, i.e. the system may execute any number of transitions until state $\boldsymbol{s}_{j-1}$ is reached since this does not destroy the "backward-overapproximating" property of $\mathcal{B}^k(\boldsymbol{s})$. The rationale of having parameter j is the additional freedom in constructing $\mathcal{I}^k(\boldsymbol{s})$ as j may influence the shape of $\mathcal{I}^k(\boldsymbol{s})$ (cf. example below).

Phase 2. Having symbolically described all backward reachable states by $\widehat{\mathit{BReach}}(\boldsymbol{s})$, we are able to compute *upper* bounds ub_k of the maximum probability $P^{\max}_{\mathcal{M}}(T)$ of reaching target states T by SSAT solving applied to

$$ub_k := Pr\Big(\mathcal{Q}(k) : \Big(\overbrace{\mathit{Init}(\boldsymbol{s}_0) \wedge \bigwedge_{i=1}^{k} \mathit{Trans}(\boldsymbol{s}_{i-1}, \boldsymbol{t}_i, \boldsymbol{s}_i)}^{\text{states reachable within } k \text{ steps}} \wedge \overbrace{\bigwedge_{i=0}^{k} \widehat{\mathit{BReach}}(\boldsymbol{s}_i)}^{\text{stay in back-reach set}} \Big)\Big) \tag{3}$$

First observe that the formula above excludes all system runs that leave the set of backward reachable states. This is sound since leaving $\widehat{\mathit{BReach}}(\boldsymbol{s})$ means to never reach the $\mathit{Target}(\boldsymbol{s})$ states. Second, the system behavior becomes more and more constrained for increasing k, i.e. the ub_k's are monotonically decreasing. Regarding model checking, a safety property $P^{\max}_{\mathcal{M}}(T) \leq \theta$ is verified by the procedure above once $ub_k \leq \theta$ is computed for some k.

Example. Consider the simple *discrete-time Markov chain* (DTMC)[9] $\mathcal{M}$ on the left of Fig. 3, with s as the only target state. The (maximum) probability of reaching s from the initial state i clearly is $P = 0.9 \cdot (0.5 + 0.5 \cdot P) = \frac{9}{11}$. Applying the generalized interpolation scheme (2) for $k = 0$ with parameter $j = 1$ yields the interpolant

[9] A DTMC is an MDP without non-determinism.

$\mathcal{I}^1(\boldsymbol{s}) = \neg i$. If we proceed then the set $\widehat{BReach}(\boldsymbol{s})$ covers all states though state f is not backward reachable. Thus, $j = 1$ is not a good choice as each ub_k in scheme (3) will then be 1. For $j = 2$, scheme (2) produces: $\mathcal{B}^0(\boldsymbol{s}) = \mathit{Target}(\boldsymbol{s}) = s$, $\mathcal{I}^1(\boldsymbol{s}) = \neg i \wedge \neg f$, $\mathcal{B}^1(\boldsymbol{s}) = \mathcal{B}^0(\boldsymbol{s}) \vee \mathcal{I}^1(\boldsymbol{s}) = (\neg i \vee s) \wedge (\neg f \vee s)$, $\mathcal{I}^2(\boldsymbol{s}) = (\neg i \vee \neg e) \wedge \neg f$, $\mathcal{B}^2(\boldsymbol{s}) = \mathcal{B}^1(\boldsymbol{s}) \vee \mathcal{I}^2(\boldsymbol{s}) = (\neg i \vee \neg e \vee s) \wedge (\neg f \vee s)$, $\mathcal{I}^3(\boldsymbol{s}) = (\neg i \vee \neg e) \wedge \neg f$, $\mathcal{B}^3(\boldsymbol{s}) = \mathcal{B}^2(\boldsymbol{s}) \vee \mathcal{I}^3(\boldsymbol{s}) = \mathcal{B}^2(\boldsymbol{s})$ (cf. left of Fig. 3). Thus, $\widehat{BReach}(\boldsymbol{s}) = \mathcal{B}^2(\boldsymbol{s})$.

We are now able to compute upper bounds of the reachability probability $P = \frac{9}{11}$ by scheme (3). The results are shown on the right of Fig. 3 where the lower bounds are computed according to scheme (1). The figure indicates a fast convergence of the lower and upper bounds: for depth $k = 20$ the difference $ub_k - lb_k$ is below 10^{-3} and for $k = 100$ below 10^{-15}.

All 200 SSAT formulae were solved by the SSMT tool SiSAT [9] in 41.3 sec on a 1.83 GHz Intel Core 2 Duo machine with 1 GByte physical memory running Linux. For the moment, the SSAT encoding of $\mathcal{M}$ and all computations of generalized interpolants from PBMC proofs were performed manually. Details can be found in Appendix A. While the runtime of this first prototype does not compare favorably to value or policy iteration procedures, it should be noted that this is a first step towards a procedure embedding the same interpolation process into SSMT [8] and thereby directly addressing probabilistic hybrid systems, where the iteration procedures are only applicable after finite-state abstraction. It should also be noted that the symbolic procedures provided by SSAT and SSMT support compact representations of concurrent systems, thus alleviating the state explosion problem [9].

6 Conclusion and future work

In this paper, we elaborated on the idea of Craig interpolation for stochastic SAT. In consideration of the difficulties that arise in this stochastic extension of SAT, we first proposed a suitable definition of a generalized Craig interpolant and second presented an algorithm to automatically compute such interpolants. For the latter purpose, we enhanced the SSAT resolution calculus by corresponding rules for the construction of generalized interpolants. We further demonstrated an application of generalized Craig interpolation in probabilistic model checking. The resulting procedure is able to verify probabilistic safety requirements of the form "the worst-case probability of reaching undesirable system states is smaller than a given threshold". This complements the existing SSAT-based bounded model checking approach, which mechanizes falsification of such properties.

An essential issue for future work is the practical evaluation of interpolation-based probabilistic model checking on realistic case studies. This involves, a.o., the integration of interpolating S-resolution into DPLL-based SSAT solvers and a thorough empirical study of the results obtained from the interpolation scheme (2). The latter includes the size and shape of generalized Craig interpolants as well as the computational effort for computing them in practice

Another interesting future direction is the adaptation of generalized Craig interpolation to SSMT [8], i.e. the extension of SSAT with arithmetic theories. Computing Craig interpolants for SSMT would lift schemes (2) and (3) to SSMT problems, thus establishing a symbolic verification procedure for discrete-time probabilistic hybrid systems.

References

1. Papadimitriou, C.H.: Games against nature. J. Comput. Syst. Sci. **31**(2) (1985) 288–301

2. Littman, M.L., Majercik, S.M., Pitassi, T.: Stochastic Boolean Satisfiability. Journal of Automated Reasoning **27**(3) (2001) 251–296
3. Majercik, S.M., Littman, M.L.: MAXPLAN: A New Approach to Probabilistic Planning. In: Artificial Intelligence Planning Systems. (1998) 86–93
4. Majercik, S.M., Littman, M.L.: Contingent Planning Under Uncertainty via Stochastic Satisfiability. Artificial Intelligence Special Issue on Planning With Uncertainty and Incomplete Information **147**(1-2) (2003) 119–162
5. Walsh, T.: Stochastic constraint programming. In: Proc. of the 15th European Conference on Artificial Intelligence (ECAI'02), IOS Press (2002)
6. Balafoutis, T., Stergiou, K.: Algorithms for Stochastic CSPs. In Benhamou, F., ed.: CP. Volume 4204 of LNCS., Springer (2006) 44–58
7. Bordeaux, L., Samulowitz, H.: On the stochastic constraint satisfaction framework. In: SAC, ACM (2007) 316–320
8. 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 LNCS., Springer (2008) 172–186
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 (2011) to appear.
10. Barrett, C., Sebastiani, R., Seshia, S.A., Tinelli, C.: Satisfiability modulo theories. [24] chapter 26 825–885
11. Littman, M.L.: Initial Experiments in Stochastic Satisfiability. In: Proc. of the 16th National Conference on Artificial Intelligence. (1999) 667–672
12. Davis, M., Putnam, H.: A Computing Procedure for Quantification Theory. Journal of the ACM **7**(3) (1960) 201–215
13. Davis, M., Logemann, G., Loveland, D.: A Machine Program for Theorem Proving. Communications of the ACM **5** (1962) 394–397
14. Majercik, S.M.: Nonchronological backtracking in stochastic Boolean satisfiability. In: ICTAI, IEEE Computer Society (2004) 498–507
15. Robinson, J.A.: A machine-oriented logic based on the resolution principle. J. ACM **12**(1) (1965) 23–41
16. Büning, H.K., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. **117**(1) (1995) 12–18
17. Teige, T., Fränzle, M.: Resolution for stochastic Boolean satisfiability. In Fermüller, C.G., Voronkov, A., eds.: Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR-17. Volume 6397 of LNCS., Springer (2010) 625–639
18. Craig, W.: Linear Reasoning. A New Form of the Herbrand-Gentzen Theorem. J. Symb. Log. **22**(3) (1957) 250–268
19. McMillan, K.L.: Interpolation and SAT-based model checking. In Jr., W.A.H., Somenzi, F., eds.: CAV. Volume 2725 of Lecture Notes in Computer Science., Springer (2003) 1–13
20. McMillan, K.L.: Applications of Craig interpolants in model checking. In Halbwachs, N., Zuck, L.D., eds.: TACAS. Volume 3440 of Lecture Notes in Computer Science., Springer (2005) 1–12
21. Pudlák, P.: Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations. Journal of Symbolic Logic **62**(3) (September 1997) 981–998
22. Majercik, S.M.: Stochastic Boolean satisfiability. [24] chapter 27 887–925
23. Biere, A., Cimatti, A., Clarke, E.M., Zhu, Y.: Symbolic model checking without BDDs. In Cleaveland, R., ed.: TACAS. Volume 1579 of Lecture Notes in Computer Science., Springer (1999) 193–207
24. Biere, A., Heule, M.J.H., van Maaren, H., Walsh, T., eds.: Handbook of Satisfiability. Volume 185 of Frontiers in Artificial Intelligence and Applications. IOS Press (February 2009)

Appendix

A Derivations of generalized Craig interpolants

A.1 SSAT encoding of DTMC

The state space of $\mathcal{M}$ from Fig. 3 is encoded by four Boolean variables i, f, e, s. As there are two probabilistic choices in $\mathcal{M}$ we take two Boolean variables pi (for the choice from i) and pe (for the choice from e). The initial state predicate then is $\mathit{Init}(\boldsymbol{s}) = (i) \wedge (\neg f) \wedge (\neg e) \wedge (\neg s)$. The transition relation predicate $\mathit{Trans}(\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{s}')$ is given by clauses $(i' \vee f' \vee e' \vee s')$, $(\neg i' \vee \neg f')$, $(\neg i' \vee \neg e')$, $(\neg i' \vee \neg s')$, $(\neg f' \vee \neg e')$, $(\neg f' \vee \neg s')$, $(\neg e' \vee \neg s')$ meaning that $\mathcal{M}$ is in exactly one state after the transition, and by $(i \wedge \neg pi \Rightarrow f') \equiv (\neg i \vee pi \vee f')$, $(i \wedge pi \Rightarrow e') \equiv (\neg i \vee \neg pi \vee e')$, $(e \wedge \neg pe \Rightarrow s') \equiv (\neg e \vee pe \vee s')$, $(e \wedge pe \Rightarrow i') \equiv (\neg e \vee \neg pe \vee i')$, $(f \Rightarrow f') \equiv (\neg f \vee f')$, $(s \Rightarrow s') \equiv (\neg s \vee s')$ describing the possible state changes. The target predicate is $\mathit{Target}(\boldsymbol{s}) = (s)$. Finally, the quantifier prefixes are $\mathcal{Q}_{\boldsymbol{s}} = \exists i, f, e, s$, $\mathcal{Q}_{\boldsymbol{t}} = \mathbb{R}^{0.9} pi\ \mathbb{R}^{0.5} pe$, and $\mathcal{Q}_{\boldsymbol{s}'} = \exists i', f', e', s'$.

A.2 Interpolating S-resolution

Choice $j = 1$: We first apply scheme (2) for $j = 1$, i.e.

$$\mathcal{Q}_{\boldsymbol{s}_0} \mathcal{Q}_{\boldsymbol{t}_1} \mathcal{Q}_{\boldsymbol{s}_1} : \big(\overbrace{\mathit{Init}(\boldsymbol{s}_0)}^{B} \wedge \overbrace{\mathit{Trans}(\boldsymbol{s}_0, \boldsymbol{t}_1, \boldsymbol{s}_1) \wedge \mathcal{B}^0(\boldsymbol{s}_1)}^{A} \big),$$

where $\mathcal{B}^0(\boldsymbol{s}_1) = \mathit{Target}(\boldsymbol{s}_1) = s_1$. The concrete quantifier prefix is

$$\exists i_0, f_0, e_0, s_0\ \mathbb{R}^{0.9} pi_1\ \mathbb{R}^{0.5} pe_1\ \exists i_1, f_1, e_1, s_1 :$$

and the sets of variables are $V_A = \{pi_1, pe_1, i_1, f_1, e_1, s_1\}$, $V_B = \emptyset$, $V_{A,B} = \{i_0, f_0, e_0, s_0\}$. In the following, we present the S-resolution proof together with the construction of the generalized Craig interpolant.

$c_1 = (i_0)^0$	$I_1 = \texttt{true}$	(R'.1)
$c_2 = (\neg f_0)^0$	$I_2 = \texttt{true}$	(R'.1)
$c_3 = (\neg e_0)^0$	$I_3 = \texttt{true}$	(R'.1)
$c_4 = (\neg s_0)^0$	$I_4 = \texttt{true}$	(R'.1)
$c_5 = (i_1 \vee f_1 \vee e_1 \vee s_1)^0$	$I_5 = \texttt{false}$	(R'.1)
$c_6 = (\neg i_1 \vee \neg f_1)^0$	$I_6 = \texttt{false}$	(R'.1)
$c_7 = (\neg i_1 \vee \neg e_1)^0$	$I_7 = \texttt{false}$	(R'.1)
$c_8 = (\neg i_1 \vee \neg s_1)^0$	$I_8 = \texttt{false}$	(R'.1)
$c_9 = (\neg f_1 \vee \neg e_1)^0$	$I_9 = \texttt{false}$	(R'.1)
$c_{10} = (\neg f_1 \vee \neg s_1)^0$	$I_{10} = \texttt{false}$	(R'.1)
$c_{11} = (\neg e_1 \vee \neg s_1)^0$	$I_{11} = \texttt{false}$	(R'.1)
$c_{12} = (\neg i_0 \vee pi_1 \vee f_1)^0$	$I_{12} = \texttt{false}$	(R'.1)
$c_{13} = (\neg i_0 \vee \neg pi_1 \vee e_1)^0$	$I_{13} = \texttt{false}$	(R'.1)
$c_{14} = (\neg e_0 \vee pe_1 \vee s_1)^0$	$I_{14} = \texttt{false}$	(R'.1)
$c_{15} = (\neg e_0 \vee \neg pe_1 \vee i_1)^0$	$I_{15} = \texttt{false}$	(R'.1)
$c_{16} = (\neg f_0 \vee f_1)^0$	$I_{16} = \texttt{false}$	(R'.1)
$c_{17} = (\neg s_0 \vee s_1)^0$	$I_{17} = \texttt{false}$	(R'.1)
$c_{18} = (s_1)^0$	$I_{18} = \texttt{false}$	(R'.1)

$$
\begin{array}{lll}
c_{19} = (\neg f_1)^0 & I_{19} = \texttt{false} & (10, 18, \mathsf{R'}.3) \\
c_{20} = (\neg e_1)^0 & I_{20} = \texttt{false} & (11, 18, \mathsf{R'}.3) \\
c_{21} = (\neg i_0 \vee pi_1)^0 & I_{21} = \texttt{false} & (12, 19, \mathsf{R'}.3) \\
c_{22} = (\neg i_0 \vee \neg pi_1)^0 & I_{22} = \texttt{false} & (13, 20, \mathsf{R'}.3) \\
c_{23} = (\neg i_0)^0 & I_{23} = \texttt{false} & (21, 22, \mathsf{R'}.3) \\
c_{24} = \emptyset^0 & I_{24} = \neg i_0 & (1, 23, \mathsf{R'}.3)
\end{array}
$$

We obtain the generalized Craig interpolant $\mathcal{I}^1(\boldsymbol{s}) = \neg i$.

Choice $j = 2$: We now apply scheme (2) for $j = 2$, i.e.

$$\mathcal{Q}_{\boldsymbol{s}_0}\mathcal{Q}_{\boldsymbol{t}_1}\mathcal{Q}_{\boldsymbol{s}_1}\mathcal{Q}_{\boldsymbol{t}_2}\mathcal{Q}_{\boldsymbol{s}_2} : \big(\overbrace{\mathit{Init}(\boldsymbol{s}_0) \wedge \mathit{Trans}(\boldsymbol{s}_0, \boldsymbol{t}_1, \boldsymbol{s}_1)}^{B} \wedge \overbrace{\mathit{Trans}(\boldsymbol{s}_1, \boldsymbol{t}_2, \boldsymbol{s}_2) \wedge \mathcal{B}^0(\boldsymbol{s}_2)}^{A}\big)$$

where $\mathcal{B}^0(\boldsymbol{s}_2) = \mathit{Target}(\boldsymbol{s}_2) = s_2$. The concrete quantifier prefix is

$$\exists i_0, f_0, e_0, s_0 \ \text{ᛋ}^{0.9} pi_1 \ \text{ᛋ}^{0.5} pe_1 \ \exists i_1, f_1, e_1, s_1 \ \text{ᛋ}^{0.9} pi_2 \ \text{ᛋ}^{0.5} pe_2 \ \exists i_2, f_2, e_2, s_2 :$$

and the sets of variables are $V_A = \{pi_2, pe_2, i_2, f_2, e_2, s_2\}$, $V_B = \{i_0, f_0, e_0, s_0, pi_1, pe_1\}$, $V_{A,B} = \{i_1, f_1, e_1, s_1\}$. In the following, we present the S-resolution proof together with the construction of the generalized Craig interpolant.

$$
\begin{array}{lll}
c_1 = (i_0)^0 & I_1 = \texttt{true} & (\mathsf{R'}.1) \\
c_2 = (\neg f_0)^0 & I_2 = \texttt{true} & (\mathsf{R'}.1) \\
c_3 = (\neg e_0)^0 & I_3 = \texttt{true} & (\mathsf{R'}.1) \\
c_4 = (\neg s_0)^0 & I_4 = \texttt{true} & (\mathsf{R'}.1) \\
c_5 = (i_1 \vee f_1 \vee e_1 \vee s_1)^0 & I_5 = \texttt{true} & (\mathsf{R'}.1) \\
c_6 = (\neg i_1 \vee \neg f_1)^0 & I_6 = \texttt{true} & (\mathsf{R'}.1) \\
c_7 = (\neg i_1 \vee \neg e_1)^0 & I_7 = \texttt{true} & (\mathsf{R'}.1) \\
c_8 = (\neg i_1 \vee \neg s_1)^0 & I_8 = \texttt{true} & (\mathsf{R'}.1) \\
c_9 = (\neg f_1 \vee \neg e_1)^0 & I_9 = \texttt{true} & (\mathsf{R'}.1) \\
c_{10} = (\neg f_1 \vee \neg s_1)^0 & I_{10} = \texttt{true} & (\mathsf{R'}.1) \\
c_{11} = (\neg e_1 \vee \neg s_1)^0 & I_{11} = \texttt{true} & (\mathsf{R'}.1) \\
c_{12} = (\neg i_0 \vee pi_1 \vee f_1)^0 & I_{12} = \texttt{true} & (\mathsf{R'}.1) \\
c_{13} = (\neg i_0 \vee \neg pi_1 \vee e_1)^0 & I_{13} = \texttt{true} & (\mathsf{R'}.1) \\
c_{14} = (\neg e_0 \vee pe_1 \vee s_1)^0 & I_{14} = \texttt{true} & (\mathsf{R'}.1) \\
c_{15} = (\neg e_0 \vee \neg pe_1 \vee i_1)^0 & I_{15} = \texttt{true} & (\mathsf{R'}.1) \\
c_{16} = (\neg f_0 \vee f_1)^0 & I_{16} = \texttt{true} & (\mathsf{R'}.1) \\
c_{17} = (\neg s_0 \vee s_1)^0 & I_{17} = \texttt{true} & (\mathsf{R'}.1) \\
c_{18} = (i_2 \vee f_2 \vee e_2 \vee s_2)^0 & I_{18} = \texttt{false} & (\mathsf{R'}.1) \\
c_{19} = (\neg i_2 \vee \neg f_2)^0 & I_{19} = \texttt{false} & (\mathsf{R'}.1) \\
c_{20} = (\neg i_2 \vee \neg e_2)^0 & I_{20} = \texttt{false} & (\mathsf{R'}.1) \\
c_{21} = (\neg i_2 \vee \neg s_2)^0 & I_{21} = \texttt{false} & (\mathsf{R'}.1) \\
c_{22} = (\neg f_2 \vee \neg e_2)^0 & I_{22} = \texttt{false} & (\mathsf{R'}.1) \\
c_{23} = (\neg f_2 \vee \neg s_2)^0 & I_{23} = \texttt{false} & (\mathsf{R'}.1) \\
c_{24} = (\neg e_2 \vee \neg s_2)^0 & I_{24} = \texttt{false} & (\mathsf{R'}.1) \\
c_{25} = (\neg i_1 \vee pi_2 \vee f_2)^0 & I_{25} = \texttt{false} & (\mathsf{R'}.1) \\
c_{26} = (\neg i_1 \vee \neg pi_2 \vee e_2)^0 & I_{26} = \texttt{false} & (\mathsf{R'}.1)
\end{array}
$$

$c_{27} = (\neg e_1 \vee pe_2 \vee s_2)^0$ $\quad I_{27} = \texttt{false}$ (R'.1)

$c_{28} = (\neg e_1 \vee \neg pe_2 \vee i_2)^0$ $\quad I_{28} = \texttt{false}$ (R'.1)

$c_{29} = (\neg f_1 \vee f_2)^0$ $\quad I_{29} = \texttt{false}$ (R'.1)

$c_{30} = (\neg s_1 \vee s_2)^0$ $\quad I_{30} = \texttt{false}$ (R'.1)

$c_{31} = (s_2)^0$ $\quad I_{31} = \texttt{false}$ (R'.1)

$c_{32} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2 \vee f_2 \vee e_2 \vee \neg s_2)^1$

$I_{32} = DC$ (R'.2)

$c_{33} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2 \vee f_2 \vee e_2)^1$

$I_{33} = DC$ $(31, 32, \text{R'.3})$

$c_{34} = (\neg e_2)^0$ $\quad I_{34} = \texttt{false}$ $(24, 31, \text{R'.3})$

$c_{35} = (\neg f_2)^0$ $\quad I_{35} = \texttt{false}$ $(23, 31, \text{R'.3})$

$c_{36} = (\neg i_2)^0$ $\quad I_{36} = \texttt{false}$ $(21, 31, \text{R'.3})$

$c_{37} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2)^1$

$I_{37} = DC$ $(33, 34, 35, 36, \text{R'.3})$

$c_{38} = (\neg e_1 \vee \neg pe_2)^0$ $\quad I_{38} = \texttt{false}$ $(28, 36, \text{R'.3})$

$c_{39} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1)^{0.5}$

$I_{39} = DC$ $(37, 38, \text{R'.3})$

$c_{40} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1)^{0.5}$

$$I_{40} = (\neg s_1 \vee \texttt{true}) \wedge (s_1 \vee DC)$$
$$\equiv (s_1 \vee DC) \quad (11, 39, \text{R'.3})$$

$c_{41} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1)^{0.5}$

$$I_{41} = (\neg e_1 \vee s_1 \vee DC) \wedge (e_1 \vee \texttt{true})$$
$$\equiv (\neg e_1 \vee s_1 \vee DC) \quad (13, 40, \text{R'.3})$$

$c_{42} = (\neg f_1)^0$ $\quad I_{42} = \texttt{false}$ $(29, 35, \text{R'.3})$

$c_{43} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1)^{0.5}$

$$I_{43} = (\neg f_1 \vee \texttt{false}) \wedge (f_1 \vee \neg e_1 \vee s_1 \vee DC)$$
$$\equiv \neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC) \quad (41, 42, \text{R'.3})$$

$c_{44} = (\neg i_1 \vee pi_2)^0$ $\quad I_{44} = \texttt{false}$ $(25, 35, \text{R'.3})$

$c_{45} = (\neg i_1 \vee \neg pi_2)^0$ $\quad I_{45} = \texttt{false}$ $(26, 34, \text{R'.3})$

$c_{46} = (\neg i_1)^0$ $\quad I_{46} = \texttt{false}$ $(44, 45, \text{R'.3})$

$c_{47} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1)^{0.5}$

$$\begin{aligned} I_{47} &= (\neg i_1 \vee \texttt{false}) \\ &\quad \wedge (i_1 \vee (\neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC))) \\ &\equiv \neg i_1 \wedge \neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC) && (43, 46, \mathsf{R'}.3) \end{aligned}$$

$c_{48} = (\neg i_0 \vee pi_1)^0$ $\quad I_{48} = (\neg f_1 \vee \texttt{false}) \wedge (f_1 \vee \texttt{true}) \equiv \neg f_1$ $(12, 42, \mathsf{R'}.3)$

$c_{49} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0)^{0.45}$

$$\begin{aligned} I_{49} &= (\neg i_1 \wedge \neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC)) \\ &\quad \wedge (\neg f_1) \\ &\equiv \neg i_1 \wedge \neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC) && (47, 48, \mathsf{R'}.3) \end{aligned}$$

$c_{50} = \emptyset^{0.45}$ $\quad I_{50} = \neg i_1 \wedge \neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC)$ $(4, 3, 2, 1, 49, \mathsf{R'}.3)$

The term DC stands for "don't care" as rule R'.2 allows any I over $V_{A,B}$. We need to choose $DC = \texttt{true}$. Interpolant thus is $\mathcal{I}^1(\boldsymbol{s}) = \neg i \wedge \neg f$. Hence, $\mathcal{B}^1(\boldsymbol{s}) = \mathcal{B}^0(\boldsymbol{s}) \vee \mathcal{I}^1(\boldsymbol{s}) = s \vee (\neg i \wedge \neg f) \equiv (\neg i \vee s) \wedge (\neg f \vee s)$ with $\mathcal{B}^0(\boldsymbol{s}) = \mathit{Target}(\boldsymbol{s}) = s$. It does *not* hold that $\mathcal{B}^1(\boldsymbol{s}) \Rightarrow \mathcal{B}^0(\boldsymbol{s})$ since $\big(s \vee (\neg i \wedge \neg f)\big) \not\Rightarrow s$. We need another iteration and apply scheme (2) again, i.e.

$$\mathcal{Q}_{\boldsymbol{s}_0} \mathcal{Q}_{\boldsymbol{t}_1} \mathcal{Q}_{\boldsymbol{s}_1} \mathcal{Q}_{\boldsymbol{t}_2} \mathcal{Q}_{\boldsymbol{s}_2} : \big(\overbrace{\mathit{Init}(\boldsymbol{s}_0) \wedge \mathit{Trans}(\boldsymbol{s}_0, \boldsymbol{t}_1, \boldsymbol{s}_1)}^{B} \wedge \overbrace{\mathit{Trans}(\boldsymbol{s}_1, \boldsymbol{t}_2, \boldsymbol{s}_2) \wedge \mathcal{B}^1(\boldsymbol{s}_2)}^{A} \big).$$

The variable sets V_A, V_B, and $V_{A,B}$ remain the same as above. The construction of the interpolant is as follows. Please note that clauses c_1 to c_{30} are the same as above since only $\mathcal{B}^0(\boldsymbol{s}_2)$ is replaced by $\mathcal{B}^1(\boldsymbol{s}_2)$. We thus do not write down clauses c_1 to c_{30} again.

$c_{31} = (\neg i_2 \vee s_2)^0$ $\quad I_{31} = \texttt{false}$ $(\mathsf{R'}.1)$

$c_{32} = (\neg f_2 \vee s_2)^0$ $\quad I_{32} = \texttt{false}$ $(\mathsf{R'}.1)$

$c_{33} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2 \vee f_2 \vee e_2 \vee \neg s_2)^1$

$I_{33} = DC$ $(\mathsf{R'}.2)$

$c_{34} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2 \vee f_2 \vee e_2)^1$

$I_{34} = \texttt{false} \vee DC \equiv DC$ $(27, 33, \mathsf{R'}.3)$

$c_{35} = (\neg e_1 \vee pe_2 \vee \neg e_2)^0$ $\quad I_{35} = \texttt{false}$ $(24, 27, \mathsf{R'}.3)$

$c_{36} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2 \vee f_2)^1$

$I_{36} = \texttt{false} \vee DC \equiv DC$ $(34, 35, \mathsf{R'}.3)$

$c_{37} = (\neg e_1 \vee pe_2 \vee \neg f_2)^0$ $\quad I_{37} = \texttt{false}$ $(23, 27, \mathsf{R'}.3)$

$c_{38} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2)^1$

$I_{38} = \texttt{false} \vee DC \equiv DC$ $(36, 37, \mathsf{R'}.3)$

$c_{39} = (\neg e_1 \vee pe_2 \vee \neg i_2)^0$ $\quad I_{39} = \texttt{false}$ $(21, 27, \mathsf{R'}.3)$

$c_{40} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2)^1$

$I_{40} = \texttt{false} \vee DC \equiv DC$ $(38, 39, \mathsf{R'}.3)$

$c_{41} = (\neg i_2)^0$ $\quad I_{41} = \texttt{false}$ $(21, 31, \mathsf{R'}.3)$

$c_{42} = (\neg e_1 \vee \neg pe_2)^0$ $\quad I_{42} = \texttt{false}$ $(28, 41, \mathsf{R'}.3)$

$c_{43} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1)^{0.5}$

$$I_{43} = \texttt{false} \vee DC \equiv DC \qquad (40, 42, \text{R'.3})$$

$c_{44} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1)^{0.5}$

$$\begin{aligned} I_{44} &= (\neg s_1 \vee \texttt{true}) \wedge (s_1 \vee DC) \\ &\equiv (s_1 \vee DC) \end{aligned} \qquad (11, 43, \text{R'.3})$$

$c_{45} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1)^{0.5}$

$$\begin{aligned} I_{45} &= (\neg e_1 \vee s_1 \vee DC) \wedge (e_1 \vee \texttt{true}) \\ &\equiv (\neg e_1 \vee s_1 \vee DC) \end{aligned} \qquad (13, 44, \text{R'.3})$$

$c_{46} = (\neg f_2)^0 \qquad I_{46} = \texttt{false} \qquad (23, 32, \text{R'.3})$

$c_{47} = (\neg f_1)^0 \qquad I_{47} = \texttt{false} \qquad (29, 46, \text{R'.3})$

$c_{48} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1)^{0.5}$

$$\begin{aligned} I_{48} &= (\neg f_1 \vee \texttt{false}) \\ &\quad \wedge (f_1 \vee \neg e_1 \vee s_1 \vee DC) \\ &\equiv \neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC) \end{aligned} \qquad (45, 47, \text{R'.3})$$

$c_{49} = (\neg i_1 \vee pi_2)^0 \qquad I_{49} = \texttt{false} \qquad (25, 46, \text{R'.3})$

$c_{50} = (\neg e_1 \vee pe_2 \vee \neg e_2)^0 \qquad I_{50} = \texttt{false} \qquad (24, 27, \text{R'.3})$

$c_{51} = (\neg i_1 \vee \neg e_1 \vee \neg pi_2 \vee pe_2)^0$

$$I_{51} = \texttt{false} \qquad (26, 50, \text{R'.3})$$

$c_{52} = (\neg i_1 \vee \neg pi_2 \vee \neg i_2)^0 \qquad I_{52} = \texttt{false} \qquad (20, 26, \text{R'.3})$

$c_{53} = (\neg i_1 \vee \neg e_1 \vee \neg pi_2 \vee \neg pe_2)^0$

$$I_{53} = \texttt{false} \qquad (28, 52, \text{R'.3})$$

$c_{54} = (\neg i_1 \vee \neg e_1 \vee \neg pi_2)^0 \qquad I_{54} = \texttt{false} \qquad (51, 53, \text{R'.3})$

$c_{55} = (\neg i_1 \vee \neg e_1)^0 \qquad I_{55} = \texttt{false} \qquad (49, 54, \text{R'.3})$

$c_{56} = (\neg i_0 \vee \neg pi_1 \vee \neg i_1)^0 \qquad I_{56} = (\neg e_1 \vee \texttt{false}) \wedge (e_1 \vee \texttt{true}) \equiv \neg e_1 \qquad (13, 55, \text{R'.3})$

$c_{57} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1)^{0.5}$

$$\begin{aligned} I_{57} &= (\neg i_1 \vee \neg e_1) \\ &\quad \wedge (i_1 \vee (\neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC))) \end{aligned} \qquad (48, 56, \text{R'.3})$$

$c_{58} = (\neg i_0 \vee pi_1)^0 \qquad I_{58} = (\neg f_1 \vee \texttt{false}) \wedge (f_1 \vee \texttt{true}) \equiv \neg f_1 \qquad (12, 47, \text{R'.3})$

$c_{59} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0)^{0.45}$

$$\begin{aligned} I_{59} &= (\neg i_1 \vee \neg e_1) \\ &\quad \wedge (i_1 \vee (\neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC))) \\ &\quad \wedge \neg f_1 \end{aligned} \qquad (57, 58, \text{R'.3})$$

$c_{60} = \emptyset^{0.45} \qquad I_{60} = I_{59} \qquad (4, 3, 2, 1, 49, \text{R'.3})$

We need to choose $DC = \texttt{true}$. We then conclude,

$$\begin{aligned}
\mathcal{I}^2(\boldsymbol{s}) &= (\neg i \vee \neg e) \wedge (i \vee \neg f) \wedge \neg f \equiv (\neg i \vee \neg e) \wedge \neg f,\\
\mathcal{B}^2(\boldsymbol{s}) &= \mathcal{B}^1(\boldsymbol{s}) \vee \mathcal{I}^2(\boldsymbol{s})\\
&\equiv \big((\neg i \vee s) \wedge (\neg f \vee s)\big) \vee \big((\neg i \vee \neg e) \wedge \neg f\big)\\
&\equiv \big(s \vee (\neg i \wedge \neg f)\big) \vee \big((\neg i \wedge \neg f) \vee (\neg e \wedge \neg f)\big)\\
&\equiv s \vee (\neg i \wedge \neg f) \vee (\neg e \wedge \neg f)\\
&\equiv \big((s \vee \neg i) \wedge (s \vee \neg f)\big) \vee (\neg e \wedge \neg f)\\
&\equiv \big(s \vee \neg i \vee (\neg e \wedge \neg f)\big) \wedge \big(s \vee \neg f \vee (\neg e \wedge \neg f)\big)\\
&\equiv (\neg i \vee \neg e \vee s) \wedge (\neg i \vee \neg f \vee s) \wedge (\neg f \vee s)\\
&\equiv (\neg i \vee \neg e \vee s) \wedge (\neg f \vee s).
\end{aligned}$$

It does *not* hold that $\mathcal{B}^2(\boldsymbol{s}) \Rightarrow \mathcal{B}^1(\boldsymbol{s})$ since

$$\big((\neg i \vee \neg e \vee s) \wedge (\neg f \vee s)\big) \not\Rightarrow \big((\neg i \vee s) \wedge (\neg f \vee s)\big).$$

We need another iteration and apply scheme (2) again, i.e.

$$\mathcal{Q}_{\boldsymbol{s}_0}\mathcal{Q}_{\boldsymbol{t}_1}\mathcal{Q}_{\boldsymbol{s}_1}\mathcal{Q}_{\boldsymbol{t}_2}\mathcal{Q}_{\boldsymbol{s}_2} : \big(\overbrace{\mathit{Init}(\boldsymbol{s}_0) \wedge \mathit{Trans}(\boldsymbol{s}_0, \boldsymbol{t}_1, \boldsymbol{s}_1)}^{B} \wedge \overbrace{\mathit{Trans}(\boldsymbol{s}_1, \boldsymbol{t}_2, \boldsymbol{s}_2) \wedge \mathcal{B}^2(\boldsymbol{s}_2)}^{A}\big).$$

The variable sets V_A, V_B, and $V_{A,B}$ remain the same as above. The construction of the interpolant is as follows. Please note that clauses c_1 to c_{30} are the same as above since only $\mathcal{B}^1(\boldsymbol{s}_2)$ is replaced by $\mathcal{B}^2(\boldsymbol{s}_2)$. We thus do not write down clauses c_1 to c_{30} again.

$$c_{31} = (\neg i_2 \vee \neg e_2 \vee s_2)^0 \qquad I_{31} = \texttt{false} \tag{R'.1}$$

$$c_{32} = (\neg f_2 \vee s_2)^0 \qquad I_{32} = \texttt{false} \tag{R'.1}$$

$$c_{33} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2 \vee f_2 \vee e_2 \vee \neg s_2)^1$$

$$I_{33} = DC_1 \tag{R'.2}$$

$$c_{34} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee \neg pe_2 \vee \neg i_2 \vee f_2 \vee e_2 \vee s_2)^1$$

$$I_{34} = DC_2 \tag{R'.2}$$

$$c_{35} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2 \vee f_2 \vee e_2)^1$$

$$I_{35} = \texttt{false} \vee DC_1 \equiv DC_1 \tag{27, 33, R'.3}$$

$$c_{36} = (\neg e_1 \vee pe_2 \vee \neg e_2)^0 \qquad I_{36} = \texttt{false} \tag{24, 27, R'.3}$$

$$c_{37} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2 \vee f_2)^1$$

$$I_{37} = \texttt{false} \vee DC_1 \equiv DC_1 \tag{35, 36, R'.3}$$

$$c_{38} = (\neg e_1 \vee pe_2 \vee \neg f_2)^0 \qquad I_{38} = \texttt{false} \tag{23, 27, R'.3}$$

$$c_{39} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2 \vee i_2)^1$$

$$I_{39} = \texttt{false} \vee DC_1 \equiv DC_1 \tag{37, 38, R'.3}$$

$$c_{40} = (\neg e_1 \vee pe_2 \vee \neg i_2)^0 \qquad I_{40} = \texttt{false} \tag{21, 27, R'.3}$$

$$c_{41} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee pe_2)^1$$

$$I_{41} = \texttt{false} \vee DC_1 \equiv DC_1 \tag{39, 40, R'.3}$$

$c_{42} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee \neg pe_2 \vee \neg i_2 \vee f_2 \vee e_2)^1$

$$I_{42} = \texttt{false} \vee DC_2 \equiv DC_2 \qquad (21, 34, \text{R'.3})$$

$c_{43} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee \neg pe_2 \vee \neg i_2 \vee f_2)^1$

$$I_{43} = \texttt{false} \vee DC_2 \equiv DC_2 \qquad (20, 42, \text{R'.3})$$

$c_{44} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee \neg pe_2 \vee \neg i_2)^1$

$$I_{44} = \texttt{false} \vee DC_2 \equiv DC_2 \qquad (19, 43, \text{R'.3})$$

$c_{45} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1 \vee \neg pe_2)^1$

$$I_{45} = \texttt{false} \vee DC_2 \equiv DC_2 \qquad (28, 44, \text{R'.3})$$

$c_{46} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1 \vee s_1)^1$

$$I_{46} = DC_1 \vee DC_2 =: DC \qquad (41, 45, \text{R'.3})$$

$c_{47} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1 \vee \neg e_1)^1$

$$\begin{aligned} I_{47} &= (\neg s_1 \vee \texttt{true}) \wedge (s_1 \vee DC) \\ &\equiv (s_1 \vee DC) \end{aligned} \qquad (11, 46, \text{R'.3})$$

$c_{48} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1 \vee f_1)^1$

$$\begin{aligned} I_{48} &= (\neg e_1 \vee s_1 \vee DC) \wedge (e_1 \vee \texttt{true}) \\ &\equiv (\neg e_1 \vee s_1 \vee DC) \end{aligned} \qquad (13, 47, \text{R'.3})$$

$$c_{49} = (\neg f_2)^0 \qquad I_{49} = \texttt{false} \qquad (23, 32, \text{R'.3})$$

$$c_{50} = (\neg f_1)^0 \qquad I_{50} = \texttt{false} \qquad (29, 49, \text{R'.3})$$

$c_{51} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1 \vee i_1)^1$

$$\begin{aligned} I_{51} &= (\neg f_1 \vee \texttt{false}) \\ &\quad \wedge (f_1 \vee \neg e_1 \vee s_1 \vee DC) \\ &\equiv \neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC) \end{aligned} \qquad (48, 50, \text{R'.3})$$

$$c_{52} = (\neg i_1 \vee pi_2)^0 \qquad I_{52} = \texttt{false} \qquad (25, 49, \text{R'.3})$$

$$c_{53} = (\neg e_1 \vee pe_2 \vee \neg e_2)^0 \qquad I_{53} = \texttt{false} \qquad (24, 27, \text{R'.3})$$

$c_{54} = (\neg i_1 \vee \neg e_1 \vee \neg pi_2 \vee pe_2)^0$

$$I_{54} = \texttt{false} \qquad (26, 53, \text{R'.3})$$

$$c_{55} = (\neg i_1 \vee \neg pi_2 \vee \neg i_2)^0 \qquad I_{55} = \texttt{false} \qquad (20, 26, \text{R'.3})$$

$c_{56} = (\neg i_1 \vee \neg e_1 \vee \neg pi_2 \vee \neg pe_2)^0$

$$I_{56} = \texttt{false} \qquad (28, 55, \text{R'.3})$$

$$c_{57} = (\neg i_1 \vee \neg e_1 \vee \neg pi_2)^0 \qquad I_{57} = \texttt{false} \qquad (54, 56, \text{R'.3})$$

$$c_{58} = (\neg i_1 \vee \neg e_1)^0 \qquad I_{58} = \texttt{false} \qquad (52, 57, \text{R'.3})$$

$$c_{59} = (\neg i_0 \vee \neg pi_1 \vee \neg i_1)^0 \qquad \begin{aligned} I_{59} &= (\neg e_1 \vee \texttt{false}) \wedge (e_1 \vee \texttt{true}) \\ &\equiv \neg e_1 \end{aligned} \qquad (13, 58, \text{R'.3})$$

$c_{60} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0 \vee \neg pi_1)^1$

$$\begin{aligned} I_{60} &= (\neg i_1 \vee \neg e_1) \\ &\quad \wedge (i_1 \vee (\neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC))) && (51, 59, \mathsf{R'}.3) \end{aligned}$$

$c_{61} = (\neg i_0 \vee pi_1)^0$

$$\begin{aligned} I_{61} &= (\neg f_1 \vee \texttt{false}) \wedge (f_1 \vee \texttt{true}) \\ &\equiv \neg f_1 && (12, 50, \mathsf{R'}.3) \end{aligned}$$

$c_{62} = (\neg i_0 \vee f_0 \vee e_0 \vee s_0)^{0.9}$

$$\begin{aligned} I_{62} &= (\neg i_1 \vee \neg e_1) \\ &\quad \wedge (i_1 \vee (\neg f_1 \wedge (\neg e_1 \vee s_1 \vee DC))) \\ &\quad \wedge \neg f_1 && (60, 61, \mathsf{R'}.3) \end{aligned}$$

$c_{63} = \emptyset^{0.9}$ $\qquad I_{63} = I_{62}$ $\qquad (4, 3, 2, 1, 62, \mathsf{R'}.3)$

We need to choose $DC_1 = DC_2 = \texttt{true}$, thus $DC = \texttt{true}$. We then conclude,

$$\begin{aligned} \mathcal{I}^3(\boldsymbol{s}) &= (\neg i \vee \neg e) \wedge (i \vee \neg f) \wedge \neg f \equiv (\neg i \vee \neg e) \wedge \neg f \\ \mathcal{B}^3(\boldsymbol{s}) &= \mathcal{B}^2(\boldsymbol{s}) \vee \mathcal{I}^3(\boldsymbol{s}) \\ &\equiv \big((\neg i \vee \neg e \vee s) \wedge (\neg f \vee s)\big) \vee \big((\neg i \vee \neg e) \wedge \neg f\big) \\ &\equiv \big(s \vee ((\neg i \vee \neg e) \wedge \neg f)\big) \vee \big((\neg i \vee \neg e) \wedge \neg f\big) \\ &\equiv s \vee ((\neg i \vee \neg e) \wedge \neg f) \\ &\equiv (\neg i \vee \neg e \vee s) \wedge (\neg f \vee s) \\ &\equiv \mathcal{B}^2(\boldsymbol{s}) \end{aligned}$$

We now stop the process since $\mathcal{B}^2(\boldsymbol{s})$ has stabilized, i.e. $\mathcal{B}^3(\boldsymbol{s}) \Rightarrow \mathcal{B}^2(\boldsymbol{s})$ is valid due to $\mathcal{B}^3(\boldsymbol{s}) \equiv \mathcal{B}^2(\boldsymbol{s})$. Thus, $\widehat{BReach}(\boldsymbol{s}) := \mathcal{B}^3(\boldsymbol{s})$ overapproximates the backward reachable states in $\mathcal{M}$.