Symbolic Dependency Graphs for PCTL$^{>}_{\leq}$ Model-Checking

Anders Mariegaard(✉) and Kim Guldstrand Larsen

Department of Computer Science, Aalborg University,
Selma Lagerlöfs Vej 300, 9220 Aalborg, Denmark
{am,kgl}@cs.aau.dk

Abstract. We consider the problem of model-checking a subset of probabilistic CTL, interpreted over (discrete-time) Markov reward models, allowing the specification of lower bounds on the probability of the set of paths satisfying a cost-bounded path formula. We first consider a reduction to fixed-point computations on a graph structure that encodes a division of the problem into smaller sub-problems by explicit unfolding of the given formula into sub-formulae. Although correct, the size of the graph constructed is highly dependent on the size of the cost bound. To this end, we provide a symbolic extension, effectively ensuring independence between the size of the graph and the cost-bound.

Keywords: Model-checking · Probabilistic CTL · Dependency graphs

1 Introduction

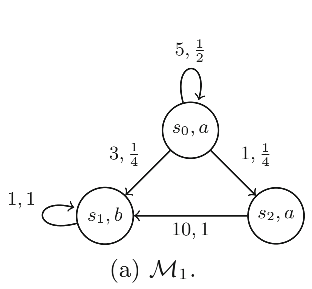
Addressing non-functional properties of embedded and distributed systems has been studied intensely in recent years. This has called for extensions of traditional modeling formalisms and specification languages to directly incorporate information such as resource consumption, timing constraints and probabilistic behavior. For real-time systems, various extensions of the popular Timed Automata [1] formalism have been studied and successfully implemented in model-checking tools such as Uppaal[17] and PRISM [16]. These extensions include variations and combinations of Priced Timed Automata [4], where costs are associated to both locations and transitions, and Probabilistic Timed Automata [19] where edges have associated probability distributions. Various extensions of Markov chains such as Markov Reward Models [12] assigning cost expressions to states and transitions, have been studied and successfully incorporated in tools such as MRMC [14], PRISM and recently STORM [10]. The underlying semantics of all these models can be given in terms of traditional transition systems where the transition relation is endowed with costs and probabilities. To reason about the costs and probabilities of the underlying (discrete) quantitative models, probabilistic CTL (PCTL) [11] extends the classical logic CTL [7] with probabilistic quantification over path formulae. Various extensions of PCTL have been developed, notably PRCTL [2] for specification of constraints over reward measures.

A. Abate and G. Geeraerts (Eds.): FORMATS 2017, LNCS 10419, pp. 153–169, 2017.
DOI: 10.1007/978-3-319-65765-3_9

Devising efficient techniques for verifying such specifications for complex models is non-trivial as any naïve exploration of the entire state space is many times not possible due to time and memory constraints, even when the state space is finite.

Our contribution. We consider model-checking a subset of PCTL. Our formalism allows for specification of non-trivial properties such as "the probability to reach a goal state through only approved states (indicated by labels), using no more than X amounts of some resource, is strictly greater than 90%". Thus, we consider a cost-bounded logic. This is natural for verification of embedded systems as resources such as time and energy are often sparse and one often wants to ensure that the probability of reaching a goal configuration without running out of resources, is above a certain threshold. We show that the problem can be reduced to fixed-point computations on a probabilistic extension of *dependency graphs* first introduced by Liu and Smolka [18]. They provide linear-time algorithms for computing least fixed-point for a Boolean domain, lending itself to e.g. CTL model checking. They offer both a global and local approach, where the global approach computes the fixed-point value for each node while the local approach in many cases will only explore parts of the entire graph. Our contribution is a lifting of the approach from the Boolean domain to a domain of probabilities for model-checking a subset of PCTL. The first approach is a new type of *probabilistic* dependency graph, constructed by a simple unfolding of the formula. Although correct, this approach is highly dependent on the size of the concrete cost bounds. To this end we provide a *symbolic* extension of probabilistic dependency graphs that effectively ensures independence between the size of the graph and the cost-bound. Although this paper does not describe a concrete implementation, the framework is constructed in such a way that it lends itself to an adaptation of the local algorithm by Liu and Smolka [18].

Related work. The framework of dependency graphs and the local algorithm of Liu and Smolka [18] has recently been extended in various ways. For the Boolean domain, a distributed implementation of the local algorithm has been developed [9] and very recently extended to express negation [8] with promising experimental results. Several extension to different domains have also been proposed. In [5] an extension to time bounds is presented to efficiently analyse Timed Games [5] and in [6,13] the approach was lifted to a (parametric) weighted domain for model-checking a (parametric) weighted variant of CTL. This paper further extends the theory behind this framework by a novel extension to the probabilistic domain. In [2] the notion of a *Path Graph* is used to solve similar model-checking problems. These graphs also express an unfolding of the model, but instead of nodes encoding probabilities for a certain state and formula, as is the case in this paper, the nodes represent possible rewards for path fragments and the associated probabilities. It will be interesting in the future to compare a distributed implementation of our approach to the Path Graph approach.



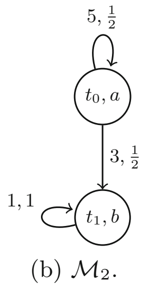


Fig. 1. Two MRMs, $\mathcal{M}_1$ and $\mathcal{M}_2$.

2 Models and Properties

This section introduces the modeling formalism and specification language. Our models will be instances of Markov Reward Models (MRMs) [12]. As we are interested in upper bounds on the non-probabilistic quantities we will from now on refer to them as *costs* instead of rewards, hence the inclusion of a transition cost function.

Definition 1 (Markov Reward Model). *A* Markov Reward Model *(MRM) is a structure $\mathcal{M} = (S, P, c, \mathcal{L})$ where S is a finite set of states, $P\colon S \times S \to [0,1]$ is the transition probability function such that for all $s \in S$, $\sum_{s' \in S} P(s,s') = 1$, $c\colon S \times S \to \mathbb{N}^+$ is the transition cost function and $\mathcal{L}\colon S \to 2^{AP}$ is the labeling function, assigning to each state a set of atomic propositions from a set AP.*

For two states s, s' with s being the current state, $P(s,s')$ is the probability that the next state will be s' and $c(s,s')$ represents the cost of exercising the transition. As our approach requires all paths to diverge w.r.t cost, we simply impose all weights to be strictly positive. Our approach also works for the case where all loops are required to have at least one transition with a strictly positive cost[1]. We denote such a transition from s to s' with probability p and cost w by $s \xrightarrow{w,p} s'$[2]. Thus, any MRM, defines a transition system for which we want to model-check properties. A *path* from a state s_0 is an infinite sequence of states $\pi = s_0, s_1, s_2, s_3, s_4, \ldots$ with $P(s_i, s_{i+1}) > 0$ for any $i \in \mathbb{N}$. We denote by $\pi[j]$ the j'th state of π, s_j.

Example 1. Consider the MRMs in Fig. 1. Each circle represents a state and the set of atomic propositions of that state. Set notation is omitted as each state has exactly one atomic proposition.

[1] Note that using costs from $\mathbb{Q}^+$ does not change the expressivity of the formalism; as any model is finite, one can always multiply all costs by the least common denominator to obtain a model with costs in $\mathbb{N}^+$.

[2] Any such transition could be replaced by a number of unit length transitions with probability 1, transforming the MRM into a (much larger) Markov chain.

As specification language, we consider PCTL restricted to strict lower bounds on the probabilistic modality and upper bounds on path formulae. The combination of a lower and upper bound induces a monotonic fixed point operator while the strict lower bound is needed for the operator to be chain-continuous, which implies the existence of the fixed point in the symbolic case (see Lemma 2).

Definition 2 (PCTL$^{\geq}_{\leq}$). *The syntax of PCTL$^{\geq}_{\leq}$* state formulae *is as follows:*

$$\Phi ::= a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \mathcal{P}_{>\lambda}(\varphi)$$

where $a \in AP$ and $\lambda \in [0,1)$. The path formulae *are then constructed according to the following grammar, with $k \in \mathbb{N}$:*

$$\varphi ::= X_{\leq k}\Phi \mid \Phi_1 U_{\leq k} \Phi_2.$$

Informally, a state s satisfies $\mathcal{P}_{>\lambda}(\Phi_1 U_{\leq k} \Phi_2)$ if the probability of the set of paths from s satisfying $\Phi_1 U_{\leq k} \Phi_2$ is greater than λ. A path satisfies $\Phi_1 U_{\leq k} \Phi_2$ if, from the beginning of the path, all states satisfy Φ_1 until a state satisfying Φ_2 is reached while the sum of the costs between the start of the path and the state satisfying Φ_2 is less than or equal to k.

The probability associated with a given path-formula is well defined based on the σ-algebra generated from the standard cylinder-set construction (see [3, Chap. 10]) assigning probabilities to sets of infinite paths sharing a finite prefix. This construction ensures that the following PCTL$^{\geq}_{\leq}$ semantics is well defined. $\mathbb{P}$ will be used to denote the (unique) probability measure.

For a state formula Φ and an MRM $\mathcal{M}$ with state s, we denote by $\mathcal{M}, s \models \Phi$ the satisfiability of Φ in s. Similarly $\mathcal{M}, \pi \models \varphi$ denotes the satisfaction of path formula φ by the path π of $\mathcal{M}$.

Definition 3 (PCTL$^{\geq}_{\leq}$ Semantics). *For MRM $\mathcal{M} = (S, P, c, \mathcal{L})$ with state s, the* satisfiability relation $\models$ *is defined inductively on PCTL$^{\geq}_{\leq}$ formulae:*

$$\begin{array}{lll}
\mathcal{M}, s \models a & \textit{iff} & a \in \mathcal{L}(s) \\
\mathcal{M}, s \models \neg a & \textit{iff} & a \notin \mathcal{L}(s) \\
\mathcal{M}, s \models \Phi_1 \wedge \Phi_2 & \textit{iff} & \mathcal{M}, s \models \Phi_1 \textit{ and } \mathcal{M}, s \models \Phi_2 \\
\mathcal{M}, s \models \Phi_1 \vee \Phi_2 & \textit{iff} & \mathcal{M}, s \models \Phi_1 \textit{ or } \mathcal{M}, s \models \Phi_2 \\
\mathcal{M}, s \models \mathcal{P}_{>\lambda}(\varphi) & \textit{iff} & \mathbb{P}(\pi \mid \pi[0] = s, \mathcal{M}, \pi \models \varphi) > \lambda \\
\mathcal{M}, \pi \models X_{\leq k}\Phi & \textit{iff} & \mathcal{M}, \pi[1] \models \Phi \textit{ and } c(\pi[0], \pi[1]) \leq k \\
\mathcal{M}, \pi \models \Phi_1 U_{\leq k} \Phi_2 & \textit{iff} & \textit{there exists a } j \textit{ such that } \mathcal{M}, \pi[j] \models \Phi_2, \\
 & & \mathcal{M}, \pi[i] \models \Phi_1 \textit{ for all } i < j \textit{ and} \\
 & & \displaystyle\sum_{l=0}^{j-1} c(\pi[l], \pi[l+1]) \leq k.
\end{array}$$

The satisfiability of a formula of the form $\mathcal{P}_{>\lambda}(\varphi)$ for a state s can be model-checked by deciding satisfiability of certain sub-formulae in the successor states of s.

Proposition 1. *If $\mathcal{M}, s \models \mathbb{P}_{>\lambda}(\Phi_1 U_{\leq k} \Phi_2)$ then at least one of the two following properties must hold:*

1. *there exists transitions $s \xrightarrow{w_i, p_i} s_i$ with $w_i \leq k$ such that $\mathcal{M}, s_i \models \mathbb{P}_{>\lambda_i}(\Phi_1 U_{\leq k - w_i} \Phi_2)$ with $\sum p_i \cdot \lambda_i > \lambda$.*
2. *$\mathcal{M}, s \models \Phi_2$.*

Proposition 1 suggests a procedure for generation of *dependencies* for PCTL$^{\gtrless}_{\leq}$ model checking as a disjunction between the dependencies represented by property (1) and the dependency of property (2). In Sect. 3 we will explicitly encode these dependencies as a probabilistic *dependency graph*. This involves a recursive application of property (1) At some point, a cost bound of less than or equal to 0 is reached. At this point, the generation of dependencies stops, as we do not allow 0-weights in a MRM.

Finally, the probability measure associated with path formulae is monotonically increasing w.r.t cost bounds.

Proposition 2. *For any MRM $\mathcal{M}$ and state s,*

$$\mathcal{M}, s \models \mathbb{P}_{>\lambda}(\Phi_1 U_{\leq k} \Phi_2) \implies \mathcal{M}, s \models \mathbb{P}_{>\lambda'}(\Phi_1 U_{\leq k'} \Phi_2)$$

whenever $\lambda' \leq \lambda$ and $k' \geq k$.

Example 2. Consider the formula $\Phi = \mathcal{P}_{>\frac{1}{2}}(\varphi)$ with $\varphi = a\, U_{\leq k}\, b$, $k \in \mathbb{N}$ and the MRM $\mathcal{M}_1$ in Fig. 1a. For $k = 10$, $\mathbb{P}(\pi \mid \pi[0] = s_0, \mathcal{M}_1, \pi \models \varphi) = P(s_0, s_1) + P(s_0, s_0) \cdot P(s_0, s_1) = \frac{1}{4} + \frac{1}{8} = \frac{3}{8} < \frac{1}{2}$ If instead $k = 11$ the direct path through s_2 affects the total probability i.e. $\mathbb{P}(\pi \mid \pi[0] = s_0, \mathcal{M}_1, \pi \models \varphi) = \frac{3}{8} + P(s_0, s_2) \cdot P(s_2, s_1) = \frac{5}{8} > \frac{1}{2}$. Thus, $\mathcal{M}_1, s_0 \models \mathcal{P}_{>\frac{1}{2}}(a\, U_{\leq 11}\, b)$. By Proposition 2 we conclude $\mathcal{M}_1, s_0 \models \mathcal{P}_{>\frac{1}{2}}(a\, U_{\leq k} b)$ for any $k \geq 11$. Similarly for $\mathcal{M}_2$ of Fig. 1b we conclude $\mathcal{M}_2, t_0 \models \mathcal{P}_{>\frac{5}{8}}(a\, U_{\leq k}\, b)$ for any $k \geq 8$.

3 Probabilistic Dependency Graphs

This section introduces *probabilistic dependency graphs* to explicitly represent the dependencies implied by Proposition 1. We show that PCTL$^{\gtrless}_{\leq}$ model checking can be solved by computing the *least* fixed-point on a complete lattice of probability *assignments* to nodes of the graph. These assignments will be concrete probabilities.

Consider the model-checking problem $\mathcal{M}, t_0 \models \mathcal{P}_{>\frac{1}{3}}(a\, U_{\leq 8}\, b)$ for t_0 of Fig. 1b. To encode this problem, a node representing the entire problem is constructed. This can be seen in Fig. 2. Now, the probability mass for the set of paths that satisfy the path formula $a\, U_{\leq 8}\, b$ must be strictly greater than $\frac{1}{3}$. This dependency is encoded by a special *cover edge* (dashed edge), labeled by the probability, to a successor node encoding the sub-problem. The semantics is that the *assignment* (probability) of the successor node must be strictly greater than $\frac{1}{3}$ for the node to have value 1. At this point we can directly apply Proposition 1 to construct new nodes. If $\mathcal{M}, t_0 \models b$, then the problem is trivial and must have associated probability 1, hence the labeling of max on the outgoing edge, indicating that a maximum will be computed. If $\mathcal{M}, t_0 \not\models b$, then $\mathcal{M}, t_0 \models a$ must be the case (value 1) and at the same time we have to apply Proposition 1 (1) to reason about successors of t_0 in the MRM, hence the minimum. A *hyper edge* is created, labeled by the transition probabilities out of t_0 with target nodes encoding the sub-problems by Proposition 1 (1). The rest of the graph can be generated in a similar manner, but one can also apply a *local* approach in lieu of Liu and Smolka [18]. If we choose to locally expand the tree at node $\langle t_1, a\,U_{\leq 5}\,b\rangle$ and first construct the node $\langle t_1, b\rangle$ it is trivial that the value of $\langle t_1, a\,U_{\leq 5}\,b\rangle$ should be 1 as $\langle t_1, b\rangle$ would have value 1. The value for the node $\langle \Sigma\rangle$ would then be $p \cdot \frac{1}{2} + \frac{1}{2} \cdot 1$ which, no matter the value for p, is strictly greater than $\frac{1}{3}$. Hence the root gets the value 1 and we can stop, even though p is completely unknown.

Our aim is in the future to use this approach to implement an extension of the local fixed-point algorithm by Liu and Smolka [18].

$$\begin{tikzcd}
\langle t_0, \mathcal{P}_{>\frac{1}{3}}(a\,U_{\leq 8}\,b)\rangle \arrow[rr, dashed, "\frac{1}{3}"] & & \langle t_0, a\,U_{\leq 8}\,b\rangle \arrow[d, "\max"'] \arrow[r, bend left, "0"] & \langle t_0, b\rangle & \\
& & \langle\ \rangle \arrow[d, "\min"] \arrow[dl, bend right] & & \\
& 1\ \ \langle t_0, a\rangle & \langle \Sigma\rangle \arrow[dl, bend right, "\frac{1}{2}"'] \arrow[dr, bend left, "\frac{1}{2}"] & & \\
& \mathrm{p}\ \ \langle t_0, a\,U_{\leq 3}\,b\rangle & & \langle t_1, a\,U_{\leq 5}\,b\rangle \arrow[r] & \langle t_1, b\rangle\ \ 1
\end{tikzcd}$$

Fig. 2. On-the-fly unfolding of dependencies.

Definition 4 (Probabilistic dependency graph). *A probabilistic dependency graph (PDG) is a structure* $G = (N, C, E_\Sigma, E_{\min}, E_{\max})$ *where*

- N *is a finite set of* nodes*.*
- $C \subseteq N \times [0,1] \times N$ *is a finite set of* cover edges*.*
- $E_\Sigma \subseteq N \times 2^{[0,1]\times N}$ *is a finite set of (probabilistic weighted)* sum-edges *where*
  - *for any* $E \in E_\Sigma$ *with* $E = (n, T)$, $\sum_{(w_i, p_i, n_i)\in T} p_i = 1$.
- $E_{\min}, E_{\max} \subseteq N \times 2^N$ *are finite sets of* minimum/maximum-*edges.*

All nodes are restricted to have at most one outgoing edge.

PDGs for PCTL$^{\gtrless}_{\leq}$ model checking will mostly have nodes of the types $\langle s, \Phi\rangle$ or $\langle s, \varphi\rangle$ where Φ is a state-formula and φ is a path-formula. Figure 3 shows the concrete construction rules. Given an MRM state and a PCTL$^{\gtrless}_{\leq}$ formula, one can apply the rules in a recursive manner to obtain a PDG. The rules for $\neg a$ are omitted as they are simply the inverse of the rules for a. Maximum/minimum edges are labeled with max/min, cover edges are represented by dashed lines and sum edges by solid lines. Note that any PDG will be finite and without cycles as MRMs are finite and 0-costs are not allowed. Cover edges abstract away the probability bound before unfolding the until formula by the until rule (Fig. 3h) according to Proposition 1. The semantic value of a node $\langle s, \varphi\rangle$ is a value in the interval $[0, 1]$ corresponding to the probability of the set of paths out of s that satisfy φ. Nodes $\langle s, \Phi\rangle$ will be assigned either 1 or 0, depending on whether Φ is satisfied in s or not.

Example 3. Consider MRM $\mathcal{M}_2$ in Fig. 1b with formula $\Phi = \mathcal{P}_{>\frac{5}{8}}(a\, U_{\leq 8}\, b)$. After applying the construction rules we get the PDG in Fig. 3. As the entire PDG is quite large, a few nodes have been omitted, indicated by dots. These nodes all represent the unfolding of $a\, U_{\leq k}\, b$ in state t_1 for various k. The size of the PDG is therefore highly dependent on the cost bound.

The formal semantics of a node is given by an *assignment.*

Definition 5 (Assignments). *Given a PDG $G = (N, C, E_{\Sigma}, E_{\min}, E_{\max})$, an* assignment, *$A\colon N \to [0, 1]$ on G is a mapping from each node to a probability.*

An assignment represents the probability associated with the satisfiability of a PCTL$^{\gtrless}_{\leq}$ formula in an MRM state. We denote by $\mathcal{A}^G$, the set of all assignments for a PDG G and order assignments by the partial order $\sqsubseteq$: for two assignments $A_1, A_2, A_1 \sqsubseteq A_2$ iff $\forall n \in N. A_1(n) \leq A_2(n)$. $(\mathcal{A}^G, \sqsubseteq)$ then constitutes a complete lattice as the meet and join of any (possibly infinite) subset $D = \{A_1, A_2, \ldots\}$ is given by the well defined supremum and infimum defined on elements of the unit interval $[0, 1]$. The join is given by $\bigvee D = A_{\bigvee}$ where $\forall n \in N. A_{\bigvee}(n) = \sup_{A_i \in D} A_i(n)$. The meet can be defined similarly, using infimum.

As we are interested in the least fixed point of $(\mathcal{A}^G, \sqsubseteq)$, we define a monotone function that iteratively refines assignments. In the following we let $\max\emptyset = 1$.

Definition 6 (Iterator). *For a PDG $G = (N, C, E_{\Sigma}, E_{\min}, E_{\max})$, $F\colon \mathcal{A}^G \to \mathcal{A}^G$ is a function that, given an assignment A on G, produces a new updated assignment, $F(A)$, defined for any node $n \in N$:*

$$F(A)(n) = \begin{cases} \begin{cases} 1 & \textit{if } A(n') > \lambda \\ 0 & \textit{otherwise} \end{cases} & \textit{if } (n, \lambda, n') \in C \\ \sum_{(p_i, n_i) \in T} (A(n_i) \cdot p_i) & \textit{if } (n, T) \in E_{\Sigma} \\ \max_{n' \in T}\{A(n')\} & \textit{if } (n, T) \in E_{\max} \\ \min_{n' \in T}\{A(n')\} & \textit{if } (n, T) \in E_{\min} \\ 0 & \textit{otherwise} \end{cases}$$

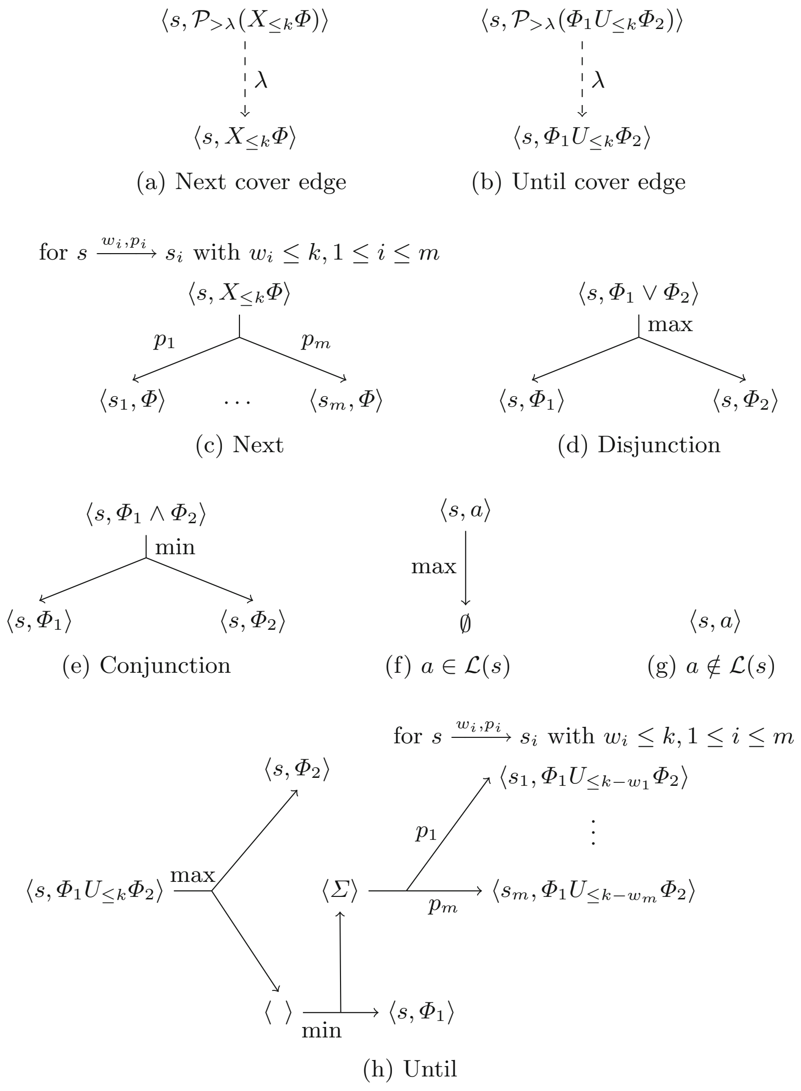

$\langle s, \mathcal{P}_{>\lambda}(X_{\leq k}\Phi)\rangle$
λ
$\langle s, X_{\leq k}\Phi\rangle$
(a) Next cover edge
$\langle s, \mathcal{P}_{>\lambda}(\Phi_1 U_{\leq k}\Phi_2)\rangle$
λ
$\langle s, \Phi_1 U_{\leq k}\Phi_2\rangle$
(b) Until cover edge
for $s \xrightarrow{w_i,p_i} s_i$ with $w_i \leq k, 1 \leq i \leq m$
$\langle s, X_{\leq k}\Phi\rangle$
p_1
p_m
$\langle s_1, \Phi\rangle$ $\cdots$ $\langle s_m, \Phi\rangle$
(c) Next
$\langle s, \Phi_1 \vee \Phi_2\rangle$
max
$\langle s, \Phi_1\rangle$ $\langle s, \Phi_2\rangle$
(d) Disjunction
$\langle s, \Phi_1 \wedge \Phi_2\rangle$
min
$\langle s, \Phi_1\rangle$ $\langle s, \Phi_2\rangle$
(e) Conjunction
$\langle s, a\rangle$
max
$\emptyset$
(f) $a \in \mathcal{L}(s)$
$\langle s, a\rangle$
(g) $a \notin \mathcal{L}(s)$
for $s \xrightarrow{w_i,p_i} s_i$ with $w_i \leq k, 1 \leq i \leq m$
$\langle s, \Phi_2\rangle$
$\langle s_1, \Phi_1 U_{\leq k-w_1}\Phi_2\rangle$
p_1
$\langle s, \Phi_1 U_{\leq k}\Phi_2\rangle$
max
$\langle \Sigma\rangle$
p_m
$\langle s_m, \Phi_1 U_{\leq k-w_m}\Phi_2\rangle$
$\langle\ \rangle$
min
$\langle s, \Phi_1\rangle$
(h) Until

Fig. 3. PDG construction rules

For a formula $\varphi = \Phi_1 U_{\leq k}\Phi_2$ and a state s, a cover edge is created to abstract away the cost bound. The target node of this edge will compute the probability for the set of paths from s that satisfy φ which is compared to λ in the cover edge case of F. By Proposition 1, this probability can be split into a weighted sum of probabilities for similar formula in successor states. This is implemented in the sum-edge case of F. To argue that F is well defined we note that assignments are closed under maximum and minimum. Furthermore, for the case $(n, T) \in E_\Sigma$ we have, by definition of E_Σ, that $\sum_{(p_i,n_i)\in T} p_i = 1$ for any $(n, T) \in E_\Sigma$. Thus,

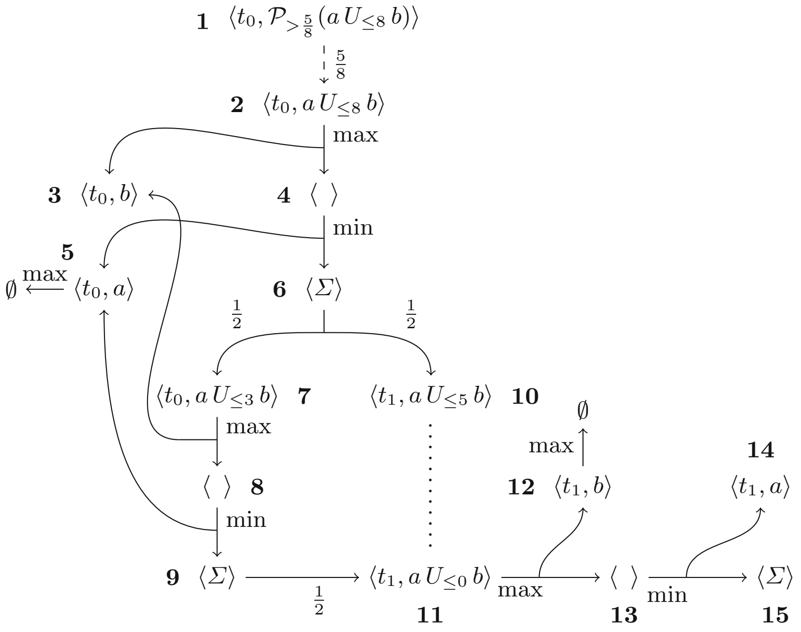

1 $\langle t_0, \mathcal{P}_{>\frac{5}{8}}(a\,U_{\leq 8}\,b)\rangle$
$\frac{5}{8}$
2 $\langle t_0, a\,U_{\leq 8}\,b\rangle$
max
3 $\langle t_0, b\rangle$
4 $\langle\ \rangle$
min
5 $\langle t_0, a\rangle$
$\emptyset$ max
6 $\langle \Sigma\rangle$
$\frac{1}{2}$
$\frac{1}{2}$
$\langle t_0, a\,U_{\leq 3}\,b\rangle$ 7
$\langle t_1, a\,U_{\leq 5}\,b\rangle$ 10
max
$\langle\ \rangle$ 8
min
$\emptyset$
max
12 $\langle t_1, b\rangle$
14 $\langle t_1, a\rangle$
9 $\langle \Sigma\rangle$
$\frac{1}{2}$
$\langle t_1, a\,U_{\leq 0}\,b\rangle$ 11
max
$\langle\ \rangle$ 13
min
$\langle \Sigma\rangle$ 15

Fig. 4. PDG constructed from $\mathcal{M}_2$ (Fig. 1) and $\Phi = \mathcal{P}_{>\frac{5}{8}}(a\,U_{\leq 8}\,b)$

each term of the sum can be at most p_i implying that the sum is within [0,1]. We now argue for the existence of a least fixed point of F. To this end we show that F is monotone on the complete lattice of assignments.

Lemma 1 (Monotonicity). *F is monotone on the complete lattice $(\mathcal{A}^G, \sqsubseteq)$.*

Let $F^i(A)$ denote i repeated applications of F on assignment A i.e. $F^i(A) = F(F^{i-1}(A))$ for $i > 0$ and $F^0(A) = A$. As F is monotone on the complete lattice $(\mathcal{A}^G, \sqsubseteq)$, Tarski's fixed point theorem [21] guarantees the existence of a least (pre-) fixed point assignment $A_{\min}$. As the PDG is finite and has no cycles, the least fixed point is computable by a repeated application of the monotone function F on the bottom element of the complete lattice of assignments. The following theorem states the correctness of our approach.

Theorem 1 (Correctness). *Given a state s of an MRM $\mathcal{M}$, a PCTL$^{\geq}_{\leq}$ state-formula Φ and the generated PDG $\mathcal{G}$ with root node $\langle s, \Phi\rangle$,*

- $\mathcal{M}, s \models \Phi \iff A_{\min}(\langle s, \Phi\rangle) = 1$
- *For any node $n = \langle s', \varphi'\rangle$ where φ' is a path-formula,*

$$A_{\min}(n) = \mathbb{P}(\pi \mid \pi[0] = s', \mathcal{M}, s' \models \varphi').$$

Example 4. We now apply F on the PDG in Fig. 4. Our starting assumption is that the assignment to all nodes is 0. The node indices in bold will be used as shorthand for a given node and $F^i(\mathbf{j})$ denotes $F^i(\mathbf{0})(n)$ whenever $\mathbf{j}$ is the index of node n. First note that $F^1(\mathbf{12}) = 1$ and therefore $F^2(\mathbf{11}) = 1$, which will never change as $F^i(\mathbf{13}) = F^i(\mathbf{14}) = F^i(\mathbf{15}) = 0$ for any i. Now, there are a set of nodes $\langle s_1, a\,U_{\leq k}\,b\rangle$ for $0 \leq k \leq 5$, two of them shown ($k \in \{5, 0\}$). According to F, the assignment to such a node for iteration i will be

$$F^i(\langle t_1, aU_{\leq k}b\rangle) = \max\left\{\begin{array}{l} F^{i-1}(\langle t_1, b\rangle), \\ \min\{F^{i-1}(\langle t_1, a\rangle), F^{i-1}(\langle t_1\, aU_{\leq k-1}\, b\} \end{array}\right\}.$$

As $F^1(\mathbf{12}) = 1$, $F^2(\langle t_1, aU_{\leq k}b\rangle) = 1$. These values can never change by the maximum and the fact that **12** is fixed after just 1 iteration of F. Table 1 shows the first 10 iterations for nodes that at some point have their value increased above 0. '–' denotes that the assignment did not change from previous the iteration; hence a fixed point is reached after 9 iterations. The fixed-point assignment to node **1** is 1, correctly implying satisfiability of formula $\Phi = \mathcal{P}_{>\frac{5}{8}}(a\,U_{\leq 8}\,b)$ in state t_0 of the MRM $\mathcal{M}_2$ in Fig. 1b.

Table 1. Iterations of F on PDG in Fig. 3

Iter#/Node	1	2	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0
1	-	-	-	1	-	-	-	-	-	-	1
2	-	-	-	-	-	-	-	-	1	1	-
3	-	-	-	-	$\frac{1}{2}$	-	-	$\frac{1}{2}$	-	-	-
4	-	-	$\frac{1}{2}$	-	-	-	$\frac{1}{2}$	-	-	-	-
5	-	$\frac{1}{2}$	-	-	-	$\frac{1}{2}$	-	-	-	-	-
6	-	-	-	-	$\frac{3}{4}$	-	-	-	-	-	-
7	-	-	$\frac{3}{4}$	-	-	-	-	-	-	-	-
8	-	$\frac{3}{4}$	-	-	-	-	-	-	-	-	-
9	1	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-

4 Probabilistic Symbolic Dependency Graphs

As witnessed by the previous section, a simple unfolding of the dependencies arising from a probabilistic formula can be used for PCTL$^{\geq}_{\leq}$ model-checking. Although correct, the approach implies that larger cost bounds on path formula results in larger PDGs as illustrated in Example 4. In this section we introduce a symbolic version of PDGs that abstracts away the cost bound, effectively collapsing many concrete nodes into *symbolic* nodes of the form $\langle s, \Phi_1 U_{\leq ?} \Phi_2\rangle$. This reduces the size of the graph significantly but may introduce cycles.

Definition 7 (Probabilistic symbolic dependency graph). *A probabilistic symbolic dependency graph (PSDG) is a structure* $G = (N, C, E_{\Sigma}, E_{\min}, E_{\max})$ *where*

– $N, E_{\min}, E_{\max}$ *are defined as for PDGs.*
– $E_{\Sigma} \subseteq N \times 2^{\mathbb{N}^+ \times [0,1] \times N}$ *is a finite set of sum-edges.*
– $C \subseteq N \times \mathbb{N}^+ \times [0,1] \times N$ *is a finite set of cover edges.*

All nodes are restricted to have at most one outgoing edge.

The new construction rules for cover edges and symbolic nodes are shown in Fig. 5. From the rules, we see that symbolic nodes imply independence between the size of the PSDG and the cost-bound.

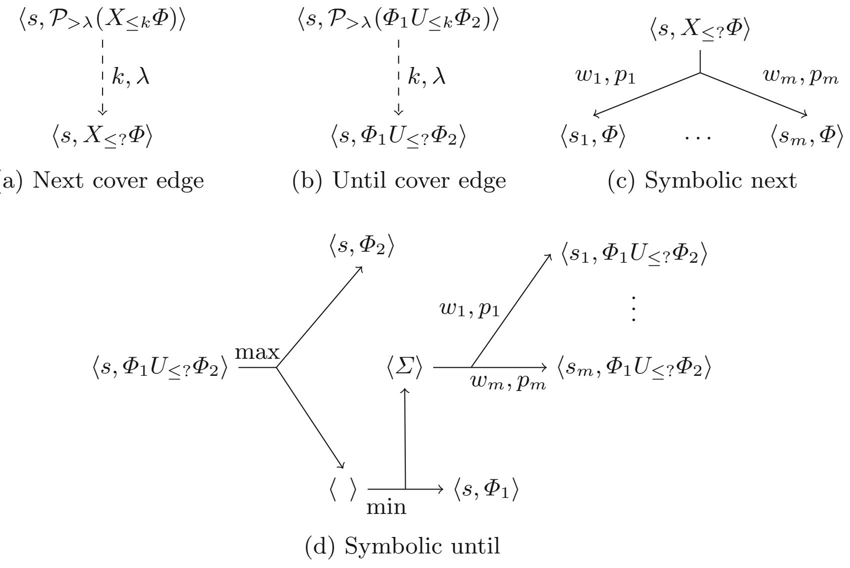

$\langle s, \mathcal{P}_{>\lambda}(X_{\leq k}\Phi)\rangle$
k, λ
$\langle s, X_{\leq ?}\Phi\rangle$
(a) Next cover edge
$\langle s, \mathcal{P}_{>\lambda}(\Phi_1 U_{\leq k}\Phi_2)\rangle$
k, λ
$\langle s, \Phi_1 U_{\leq ?}\Phi_2\rangle$
(b) Until cover edge
$\langle s, X_{\leq ?}\Phi\rangle$
w_1, p_1
w_m, p_m
$\langle s_1, \Phi\rangle$ $\cdots$ $\langle s_m, \Phi\rangle$
(c) Symbolic next
$\langle s, \Phi_2\rangle$
$\langle s, \Phi_1 U_{\leq ?}\Phi_2\rangle$
max
$\langle\ \rangle$
min
$\langle s, \Phi_1\rangle$
$\langle \Sigma\rangle$
w_1, p_1
w_m, p_m
$\langle s_1, \Phi_1 U_{\leq ?}\Phi_2\rangle$
$\vdots$
$\langle s_m, \Phi_1 U_{\leq ?}\Phi_2\rangle$
(d) Symbolic until

Fig. 5. PSDG construction rules for state s where $s \xrightarrow{w_i, p_i} s_i$ for all i with $1 \leq i \leq m$.

Example 5. Consider again the MRM $\mathcal{M}_2$ in Fig. 1. Figure 6 shows the constructed PSDG for $\mathcal{M}_2, t_0 \models \mathcal{P}_{>\frac{5}{8}}(a\, U_{\leq 8}\, b)$ which is much smaller than the corresponding PDG in Fig. 3.

As for PDGs, the semantics of each node is given by an *assignment*. Now that the upper bound on path formulae is abstracted away, each node represents a function from strictly positive naturals to concrete probabilities. Thus, an assignment to a node $\langle s, \Phi_1 U_{\leq ?}\Phi_2\rangle$ is a function f from cost bounds to probabilities such that $f(k)$ is the probability of the set of paths from s that satisfy $\Phi_1 U_{\leq k}\Phi_2$.

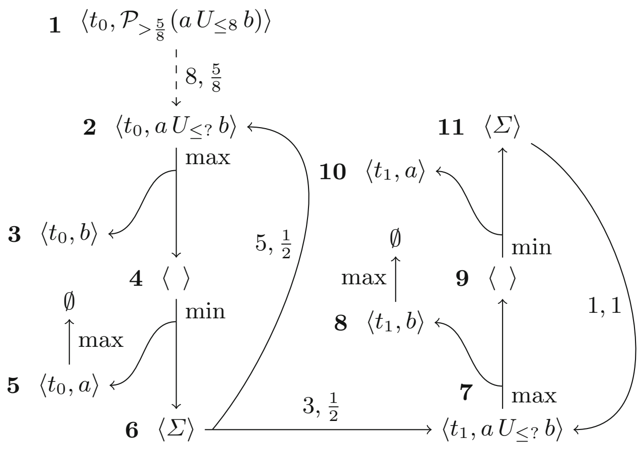

1 $\langle t_0, \mathcal{P}_{>\frac{5}{8}}(a\,U_{\leq 8}\,b)\rangle$
$8, \frac{5}{8}$
2 $\langle t_0, a\,U_{\leq ?}\,b\rangle$
max
3 $\langle t_0, b\rangle$
4 $\langle\ \rangle$
min
$\emptyset$
max
5 $\langle t_0, a\rangle$
6 $\langle \Sigma\rangle$
$5, \frac{1}{2}$
$3, \frac{1}{2}$
11 $\langle \Sigma\rangle$
10 $\langle t_1, a\rangle$
min
9 $\langle\ \rangle$
$\emptyset$
max
8 $\langle t_1, b\rangle$
7 $\langle t_1, a\,U_{\leq ?}\,b\rangle$
max
$1, 1$

Fig. 6. PSDG constructed from $\mathcal{M}_2$ (Fig. 1b) and $\Phi = \mathcal{P}_{>\frac{5}{8}}(a\,U_{\leq 8}\,b)$

Definition 8 (Assignments). *Given a PSDG $G = (N, C, E_\Sigma, E_{\min}, E_{\max})$, an* assignment*, $A\colon N \to (\mathbb{N} \to [0,1])$ on G is a mapping from each node to a function that, given a natural number, yields a probability.*

As for PDGs, we assume a component-wise partial ordering, $\sqsubseteq$ on assignments; $A_1 \sqsubseteq A_2$ iff $\forall n \in N, w \in \mathbb{N}. A_1(n)(w) \leq A_2(n)(w)$. The set of assignment $\mathcal{A}^G$ for a PSDG G ordered by $\sqsubseteq$ constitutes a complete lattice $(\mathcal{A}^G, \sqsubseteq)$.

In practice, a (finite) representation of the assignments is needed. For this, we introduce *probabilistic step-functions.* We will show (Lemma 3) that these are the only types of assignments of interest.

Definition 9 (Probabilistic Step Function). *A (finite discrete)* probabilistic step-function $f\colon \mathbb{N} \to [0,1]$ *is a function*

$$f(k) = \sum_{i=0}^{n} p_i \chi_{B_i}(k)$$

where $n \in \mathbb{N}$ is the number of steps*, $p_i \in [0,1]$ denotes the probability associated with step i, B_i is the interval of step i and χ_{B_i} is the* indicator *function for the interval B_i. The intervals partition $\mathbb{N}$ and all intervals B_i are on the form $[l, u)$ with $l < u$, $l \in \mathbb{N}$, $u \in \mathbb{N} \cup \{\infty\}$.*

Note that our definition of (probabilistic) step-function requires a finite number of steps, implying that any step-function is bounded. We will represent a step-function f as a set $\mathbf{C}_f = \{(k_i, p_i - p_{i-1}) \mid k_i = \texttt{low}(B_i), 0 < i \leq n\}$ where $\texttt{low}(B_i)$ is the lower end of the interval B_i. Thus, for each step, $\mathbf{C}_f$ includes a pair describing the position and size of the step. As we will show, all assignments of interest are probabilistic step-functions (cf. Lemma 3) i.e. any assignment A of interest is *weight-monotonic*; for any node n, $A(n)(w) \geq A(n)(w')$ whenever

$w \geq w'$, capturing that the probability measure associated with properties increases with an increased cost bound (see Proposition 1). An assignment will be referred to as a step-function assignment if it assigns a probabilistic step-function to each node in the PSDG.

Example 6. Consider again the MRM $\mathcal{M}_2$ in Fig. 1b and the path formula $a\, U_{\leq k}\, b$. For $k \leq 13$, the step-function depicted in Fig. 7 correctly computes the probability of the set of paths outgoing of t_0 that satisfy $a\, U_{\leq k}\, b$. This function is represented by the set $\{(3, \frac{1}{2}), (8, \frac{1}{4}), (13, \frac{1}{8})\}$.

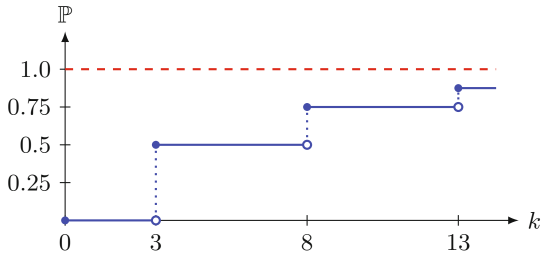


Fig. 7. Step-function for probability of set of paths outgoing of t_0 (MRM $\mathcal{M}_2$, Fig. 1b) satisfying $a\, U_{\leq k}\, b$ for $k \leq 13$.

We now define the fixed-point iterator for a PSDG $G = (N, C, E_\Sigma, E_{\min}, E_{\max})$, to iteratively refine assignments. In the following we let $\boldsymbol{x}$ be an assignment such that for any node $n \in N$ and natural number w, $\boldsymbol{x}(n)(w) = x$.

Definition 10 (Iterator). *For a PSDG $G = (N, C, E_\Sigma, E_{\min}, E_{\max})$, $F\colon \mathcal{A}^G \to \mathcal{A}^G$ is a function that, given an assignment A on G, produces a new updated assignment, $F(A)$. For any node $n \in N$ and weight $w \in \mathbb{N}$:*

$$F(A)(n)(w) = \begin{cases} \begin{cases} 1 \text{ if } A(n')(k) > \lambda \\ 0 \text{ otherwise} \end{cases} & \text{if } (n, k, \lambda, n') \in C \\ \sum_{\substack{(w_i, p_i, n_i) \in T \\ w_i \leq w}} (A(n_i)(w - w_i) \cdot p_i) & \text{if } (n, T) \in E_\Sigma \\ \max_{n' \in T}\{A(n')(w)\} & \text{if } (n, T) \in E_{\max} \\ \min_{n' \in T}\{A(n')(w)\} & \text{if } (n, T) \in E_{\min} \\ 0 & \text{otherwise} \end{cases}$$

Monotonicity of F is straightforward. Thus, by Tarski's fixed point theorem [21], a least fixed point, $A_{\min}$, exists.

As the complete lattice $(\mathcal{A}^G, \sqsubseteq)$ of assignments is of arbitrary size, in addition to the possibility of cycles in the PSDG, applying only the fixed-point theorem by Tarski does not imply a way of constructing $A_{\min}$. To this end, we prove that F is chain (Scott [20])-continuous and apply the Kleene fixed point theorem [15] to show that $A_{\min} = \sup_{n\in\mathbb{N}} F^n(\mathbf{0})$. A function $f : U \to U$ on a partially ordered set U with order $\sqsubseteq$ is *chain-continuous* iff, for any subset $D \subseteq U$ totally ordered by $\sqsubseteq$ (*a chain*), $f(\sup D) = \sup_{u_i \in D} f(u_i)$. By the Kleene fixed point theorem [15], we have that if f is chain-continuous on a complete lattice $(U, \sqsubseteq)$ with bottom element $\perp$, then $\mathit{lfp}(f) = \sup_n f^n(\perp)$ where $\mathit{lfp}(f)$ denotes the least fixed-point of f. As F is monotone on $(\mathcal{A}^G, \sqsubseteq)$ the least fixed-point $A_{\min}$ exists and a repeated application of F on the bottom element $\mathbf{0}$ produces the chain $F^0(\mathbf{0}) \sqsubseteq F^1(\mathbf{0}) \ldots$. Thus, if F is chain-continuous, $A_{\min} = \sup_n F^n(\mathbf{0})$. Chain-continuity of F thus implies an iterative procedure to approximate the least fixed-point $A_{\min}$. The following lemma shows that F is chain-continuous.

Lemma 2 (F chain-continuity). *The iterator F defined for a PSDG $G = (N, C, E_\Sigma, E_{\min}, E_{\max})$ is chain-continuous.*

Note that, if instead $\geq \lambda$ was the cover-condition for a cover edge $(n, k, \lambda, n') \in C$, there would be cases where F computes a chain D' of assignments A_i converging to λ. In this case, $(\sup D)(n') = \lambda$ while $A_i(n') < \lambda$ for all i. Thus $F(\sup D')(n) = 1$ and the lemma would not hold as $F(A_i)(n) = 0$ for any A_i, implying $\sup_{A_i \in D'} F(A_i)(n) = 0$. This is the exact reason for our choice of a strict lower bound on the probabilistic modality.

We have now established that $A_{\min}$ can be approximated by repeated application of F on $\mathbf{0}$. We now argue that all assignments of interest are step-function assignments.

Lemma 3. *For a PSDG $G = (N, C, E_\Sigma, E_{\min}, E_{\max})$, node $n \in N$ and assignment $A \in \mathcal{A}^G$, if A is a step-function assignment then $F(A)$ is a step-function assignment.*

The following lemma states that for any i, the *$i'th$* repeated application of F on the top (bottom) element gives an over(under)-approximation of the least fixed-point. This follows directly from the definition of the least fixed-point.

Lemma 4. *For an arbitrary PSDG $\mathcal{G}$, node n, iteration m and weight w,*

$$A_{\min}(n)(w) \in [F^m(\mathbf{0})(n)(w), F^m(\mathbf{1})(n)(w)].$$

Similarly, the approximations provide upper and lower bounds on the probability associated with the set of paths satisfying a given path formula.

Lemma 5. *For any symbolic node $n = \langle s, \Phi_1 U_{\leq ?} \Phi_2 \rangle$, iteration m and weight w, $\mathbb{P}(\pi \mid \pi[0] = s, \mathcal{M}, s \models \Phi_1 U_{\leq w} \Phi_2) \in [F^m(\mathbf{0})(n)(w), F^m(\mathbf{1})(n)(w)]$.*

Finally, the next theorem states that we only need a finite number of iterations to guarantee these approximations to be equal to the least fixed-point, up to a given cost. Combining this result with Lemma 5 implies correctness of our approach; for any given concrete cost bound, we can compute the exact probability of the set of paths outgoing from a state that satisfy the formula, in a finite number of steps.

Theorem 2. *For an arbitrary PSDG $\mathcal{G}$, node n and weight w, there exists an iteration i such that for any $w' \leq w$, $F^i(\mathbf{0})(n)(w') = F^i(\mathbf{1})(n)(w')$.*

Example 7. We now apply the iterator F to the PSDG in Fig. 6 to correctly verify $\mathcal{M}_2, t_0 \models \mathcal{P}_{>\frac{5}{8}}(a\, U_{\leq 8}\, b)$. We will use $\mathbf{C}_{\boldsymbol{x}} = \{(0, x)\}$ to represent the constant step-function with value x. Table 2 shows the first 9 iterations of F for the nodes that change value, starting from $\mathbf{C_0}$. After 9 iterations, node **1** is assigned $\mathbf{C_1}$ as **2** was assigned $\{(3, \frac{1}{2}), (8, \frac{1}{4})\}$ in iteration 8. This represents the fact that $t_0 \xrightarrow{3,\frac{1}{2}} t_1$ adds $\frac{1}{2}$ to the probability if the cost bound is equal to or greater than 3 and $t_0 \xrightarrow{5,\frac{1}{2}} t_0 \xrightarrow{3,\frac{1}{2}} t_1$ adds $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ if the bound is 8 or greater. Thus, for a bound of exactly 8 the two steps are added and we get the probability $\frac{6}{8} > \frac{5}{8}$. Note that the fixed-point is not reached as there is a cycle between nodes **2,4,6**. Hence the set $\{(3, \frac{1}{2}), (8, \frac{1}{4}), (13, \frac{1}{8})\}$ will be propagated to **2**, but this will not change the assignment to **1**. Thus, at iteration 9 we can stop.

Table 2. Iterations of F on PDG in Fig. 6

Iter#/Node	1	2	4	5	6	7	8
0	$\mathbf{C_0}$	$\mathbf{C_0}$	$\mathbf{C_0}$	$\mathbf{C_0}$	$\mathbf{C_0}$	$\mathbf{C_0}$	$\mathbf{C_0}$
1	-	-	-	$\mathbf{C_1}$	-	-	$\mathbf{C_1}$
2	-	-	-	-	-	$\mathbf{C_1}$	-
3	-	-	-	-	$\{(3, \frac{1}{2}\}$	-	-
4	-	-	$\{3, \frac{1}{2})\}$	-	-	-	-
5	-	$\{(3, \frac{1}{2})\}$	-	-	-	-	-
6	-	-	-	-	$\{(3, \frac{1}{2}), (8, \frac{1}{4})\}$		-
7	-	-	$\{(3, \frac{1}{2}), (8, \frac{1}{4})\}$	-	-	-	-
8	-	$\{(3, \frac{1}{2}), (8, \frac{1}{4})\}$	-	-	-	-	-
9	$\mathbf{C_1}$	-	-	-	$\{(3, \frac{1}{2}), (8, \frac{1}{4}), (13, \frac{1}{8})\}$	-	-

5 Conclusion

We presented an approach for model-checking PCTL$^{\geq}_{\leq}$, a subset of PCTL restricted to strict lower bounds on the probabilistic modalities and lower bounds on the path formulae, against weighted probabilistic transition systems by reduction to fixed-point computations on new probabilistic versions of dependency graphs. First, we presented a simple encoding by unfolding of the path formula, leading to graphs highly dependent on the size of the cost-bound. To this end, a symbolic approach was developed to ensure independence between the size of the graph and the cost-bound, by collapsing many concrete nodes into one symbolic node. For the symbolic approach, all assignments are step-functions that, given a cost bound return a probability that corresponds to the probability mass of the set of paths that satisfy the path formula with the specified cost-bound.

Future work includes efficient data-structures for step-functions and operations on step-function in order to develop an efficient implementation based on the polynomial time on-the-fly algorithm presented in [13]. Another direction could be to lift the approach to parametric model-checking of probabilistic weighted systems.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. **126**(2), 183–235 (1994). http://dx.doi.org/10.1016/0304-3975(94)90010-8
2. Andova, S., Hermanns, H., Katoen, J.-P.: Discrete-time rewards model-checked. In: Larsen, K.G., Niebert, P. (eds.) FORMATS 2003. LNCS, vol. 2791, pp. 88–104. Springer, Heidelberg (2004). doi:10.1007/978-3-540-40903-8_8
3. Baier, C., Katoen, J.: Principles of Model Checking. MIT Press, Cambridge (2008)
4. Behrmann, G., Fehnker, A., Hune, T., Larsen, K., Pettersson, P., Romijn, J., Vaandrager, F.: Minimum-cost reachability for priced time automata. In: Benedetto, M.D., Sangiovanni-Vincentelli, A. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 147–161. Springer, Heidelberg (2001). doi:10.1007/3-540-45351-2_15
5. Cassez, F., David, A., Fleury, E., Larsen, K.G., Lime, D.: Efficient on-the-fly algorithms for the analysis of timed games. In: Abadi, M., Alfaro, L. (eds.) CONCUR 2005. LNCS, vol. 3653, pp. 66–80. Springer, Heidelberg (2005). doi:10.1007/11539452_9
6. Christoffersen, P., Hansen, M., Mariegaard, A., Ringsmose, J.T., Larsen, K.G., Mardare, R.: Parametric verification of weighted systems. In: 2nd International Workshop on Synthesis of Complex Parameters, SynCoP 11, 2015, London, UK, pp. 77–90 (2015). http://dx.doi.org/10.4230/OASIcs.SynCoP.2015.77
7. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Trans. Program. Lang. Syst. **8**(2), 244–263 (1986). http://doi.acm.org/10.1145/5397.5399
8. Dalsgaard, A.E., et al.: Extended dependency graphs and efficient distributed fixed-point computation. In: van der Aalst, W., Best, E. (eds.) PETRI NETS 2017. LNCS, vol. 10258, pp. 139–158. Springer, Cham (2017). doi:10.1007/978-3-319-57861-3_10
9. Dalsgaard, A.E., Enevoldsen, S., Larsen, K.G., Srba, J.: Distributed computation of fixed points on dependency graphs. In: Fränzle, M., Kapur, D., Zhan, N. (eds.) SETTA 2016. LNCS, vol. 9984, pp. 197–212. Springer, Cham (2016). doi:10.1007/978-3-319-47677-3_13
10. Dehnert, C., Junges, S., Katoen, J., Volk, M.: A storm is coming: a modern probabilistic model checker. CoRR abs/1702.04311 (2017). http://arxiv.org/abs/1702.04311
11. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Asp. Comput. **6**(5), 512–535 (1994). http://dx.doi.org/10.1007/BF01211866
12. Howard, R.A.: Dynamic Probabilistic Systems, vol. 2. Wiley, New York (1971)
13. Jensen, J.F., Larsen, K.G., Srba, J., Oestergaard, L.K.: Efficient model-checking of weighted CTL with upper-bound constraints. STTT **18**(4), 409–426 (2016). http://dx.doi.org/10.1007/s10009-014-0359-5
14. Katoen, J., Khattri, M., Zapreev, I.S.: A Markov reward model checker. In: Second International Conference on the Quantitative Evaluaiton of Systems (QEST 2005), Torino, Italy, 19–22 September 2005, pp. 243–244 (2005). http://dx.doi.org/10.1109/QEST.2005.2
15. Kleene, S.C.: Introduction to metamathematics. Van Nostrand, Princeton (1952)

16. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22110-1_47
17. Larsen, K.G., Pettersson, P., Yi, W.: UPPAAL in a nutshell. STTT **1**(1–2), 134–152 (1997). http://dx.doi.org/10.1007/s100090050010
18. Liu, X., Smolka, S.A.: Simple linear-time algorithms for minimal fixed points. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 53–66. Springer, Heidelberg (1998). doi:10.1007/BFb0055040
19. Norman, G., Parker, D., Sproston, J.: Model checking for probabilistic timed automata. Formal Methods Syst. Des. **43**(2), 164–190 (2013). http://dx.doi.org/10.1007/s10703-012-0177-x
20. Scott, D.: Continuous lattices. In: Lawvere, F.W. (ed.) Toposes, Algebraic Geometry and Logic. LNM, vol. 274, pp. 97–136. Springer, Heidelberg (1972). doi:10.1007/BFb0073967
21. Tarski, A., et al.: A lattice-theoretical fixpoint theorem and its applications. Pacific J. Math. **5**(2), 285–309 (1955)