Conditional Probabilities over Probabilistic and Nondeterministic Systems

Miguel E. Andrés and Peter van Rossum

Institute for Computing and Information Sciences,
Radboud University Nijmegen, The Netherlands.
Email:{mandres,petervr}@cs.ru.nl

Abstract. This paper introduces the logic cpCTL, which extends the probabilistic temporal logic pCTL with conditional probability, allowing one to express that the probability that φ is true given that ψ is true is at least a. We interpret cpCTL over Markov Chain and Markov Decision Processes. While model checking cpCTL over Markov Chains can be done with existing techniques, those techniques do not carry over to Markov Decision Processes. We present a model checking algorithm for Markov Decision Processes. We also study the class of schedulers that suffice to find the maximum and minimum probability that φ is true given that ψ is true. Finally, we present the notion of counterexamples for cpCTL model checking and provide a method for counterexample generation.

1 Introduction

Conditional probabilities are a fundamental concept in probability theory. In system validation these appear for instance in anonymity, risk assessment, and diagnosability. Typical probabilities here are the probability that a certain message was sent by Alice, given that an intruder observes a certain traffic pattern; the probability that the dykes break, given that it rains heavily; the probability that component A has failed, given error message E.

This paper introduces the logic cpCTL extending the probabilistic temporal logic pCTL [HJ89] with new probabilistic operators of the form $\mathbf{P}_{\leq a}[\varphi|\psi]$, which expresses that the probability that φ is true given that ψ is true is at most a. We interpret cpCTL formulas over Markov Chains (MCs) and Markov Decision Processes (MDPs). Model checking cpCTL over MCs can be done with model checking techniques for pCTL*, using the equality $\mathbf{P}[\varphi|\psi] = \mathbf{P}[\varphi \wedge \psi]/\mathbf{P}[\psi]$.

For MDPs, cpCTL model checking is significantly more complex. Writing $\mathbf{P}_\eta[\varphi|\psi]$ for the probability $\mathbf{P}[\varphi|\psi]$ under scheduler η, model checking $\mathbf{P}_{\leq a}[\varphi|\psi]$ boils down to computing $\mathbf{P}^+[\varphi|\psi] = \max_\eta \mathbf{P}_\eta[\varphi|\psi] = \max_\eta \mathbf{P}_\eta[\varphi \wedge \psi]/\mathbf{P}_\eta[\psi]$. Thus, we have to maximize a non-linear function. (Note that in general it is not true that $\mathbf{P}^+[\varphi|\psi] = \mathbf{P}^+[\varphi \wedge \psi]/\mathbf{P}^+[\psi]$). Therefore, we cannot reuse the efficient machinery for pCTL model checking, which heavily relies on linear optimization techniques [BA95].

In particular we show that, unlike for pCTL [BA95], memoryless schedulers are not sufficient for optimizing reachability properties. We introduce the class of semi history-independent schedulers and show that these suffice to attain the optimal conditional probability. We also show that in cpCTL optimizing schedulers are not determined by the local structure of the system. That is, the choices made by the scheduler in one branch may influence the optimal choices in other branches. Surprisingly, deterministic schedulers still suffice to find the optimal conditional probability. This is remarkable indeed, since many non-linear optimization problems attain their optimal value in the interior of a convex polytope (which correspond to randomized schedulers in our setting).

Based on these properties, we present an exponential algorithm for checking if a given system satisfies a formula in the logic. We also present two heuristic optimizations of this algorithm: one trades time for space by exploiting the semi-history-independendentness of optimizing schedulers; the other uses the fact that in certain cases optimal decisions can be decided locally. Finally, we present the notion of counterexamples for cpCTL model checking as pairs of sets of paths and provide a method for counterexample generation.

1.1 Applications

Complex Systems. One application of the techniques in this paper can be found in the area of complex system behavior. Modeling naturally occurring events as probabilistic choices and operator actions as non-deterministic choices, computing maximum and minimum conditional probabilities can help optimize run-time behavior. For instance, suppose that the desired behavior of the system is expressed as a pCTL formula φ and that during run-time we are making an observation about the system, expressed as a pCTL formula ψ. The techniques in this paper allow us to compute the maximum probability of obtaining φ given that ψ is true and compute the corresponding actions (non-deterministic choices) that have to be taken to achieve this probability.

Anonymizing Protocols. Another application can be found in anonymizing protocols. These protocols such as Onion Routing [CL05], Dining Cryptographers [Cha88], voting protocols [FOO92] try to hide the originator of a message rather than the content. Strong anonymity is commonly formulated [Cha88,BP05] in terms of conditional probability: A protocol is considered strongly anonymous if no information about the sender of a message can be derived from observations of the network traffic. Formally, this is expressed by saying that the (random variable representing) sender of a specific message is independent of the (random variable representing) the observations the adversary makes. That is, for all users u and all observations of the adversary o:

$$\mathbf{P}[\text{sender} = u \mid \text{observations} = o] = \mathbf{P}[\text{sender} = u].$$

It is customary to give the adversary full control over the network [DY83] and model the capabilities of the adversary as nondeterministic choices in the system; probabilistic choices model user behavior and random choices in the protocol. Since anonymity should be guaranteed for all possible attacks of the adversary, equality should hold for all schedulers. That is: for all schedulers η, all users u and all adversarial observations o:

$$\mathbf{P}_\eta[\text{sender} = u \mid \text{observations} = o] = \mathbf{P}_\eta[\text{sender} = u]$$

In practice, $\mathbf{P}_\eta[\text{sender} = u]$ does not depend on the adversary. Since the techniques in this paper allow us to compute the maximal and minimal conditional probabilities, we can use them to prove strong anonymity.

Similarly, probable innocence is often formulated as saying that a user is (at worst) as likely to have not sent a message as to have sent it. In cpCTL this can immediately be expressed as $\mathbf{P}_{\leq 1/2}[\text{sender} = u \mid \text{observations} = o]$.

1.2 Organization of the paper

In Section 2 we present the necessary background on MDPs. In Section 3 we introduce conditional probabilities over MDPs and cpCTL is introduced in Section 4. Section 5 introduces the class of semi history-independent schedulers and Section 6 explains how to compute maximum and minimum conditional probabilities. In Section 7, we investigate the notion of counterexamples. Finally, in Section 8 we give directions for future research.

2 Markov Decision Processes

Markov Decision Processes constitute a formalism that combines nondeterministic and probabilistic choices. They are a dominant model in corporate finance, supply chain optimization and system verification and optimization. While there are many slightly different variants of this formalism (e.g., action-labeled MDPs [Bel57,FV97], probabilistic automata [SL95,SV04]), we work with the state-labeled MDPs from [BA95].

The set of all discrete probability distributions on a set S is denoted by $\mathrm{Distr}(S)$. The Dirac distribution on an element $s \in S$ is written as 1_s. We also fix a set $\mathcal{P}$ of propositions.

Definition 2.1. A *Markov Decision Process* (MDP) is a four-tuple $\Pi = (S, s_0, \tau, L)$, where S is the finite state space of the system; $s_0 \in S$ is the initial state; $L\colon S \to \wp(\mathcal{P})$ is a labeling function that associates to each state $s \in S$ a subset of $\mathcal{P}$; $\tau\colon S \to \wp(\mathrm{Distr}(S))$ is a function that associates to each $s \in S$ a non-empty and finite subset of $\mathrm{Distr}(S)$ of successor distributions.

We define a *successor* relation $\rho \subseteq S \times S$ by $\rho \triangleq \{(s,t) | \exists \pi \in \tau(s)\ .\ \pi(t) > 0\}$ and for each state $s \in S$ we define the sets $\Omega_s \triangleq \{s_0 s_1 s_2 \ldots \in S^\omega | s_0 = s \wedge \forall n \in \mathbb{N}\ .\ \rho(s_n, s_{n+1})\}$, and $\Omega_s^* \triangleq \{s_0 s_1 \ldots s_n \in S^\star | s_0 = s \wedge \forall 0 \leq i < n\ .\ \rho(s_n, s_{n+1})\}$. of paths and finite paths resp. beginning at s. For $\omega \in \Omega_s$, we write the n-th state of ω as ω_n. As usual, we let $\mathcal{B}_s \subseteq \wp(\Omega_s)$ be the Borel σ-algebra on the basic cylinders $\langle s_0 \ldots s_n \rangle \triangleq \{\omega \in \Omega_s | \omega_0 = s_0 \wedge \ldots \wedge \omega_n = s_n\}$.

Example 2.2. Figure 1 shows a MDP. Absorbing states (i.e., states s with $\tau(s) = \{1_s\}$)

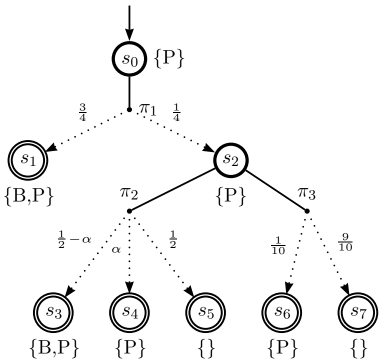


Fig. 1: Markov Decision Process

are represented by double lines. This MDP features a single nondeterministic decision, to be made in state s_2.

Schedulers (also called strategies, adversaries, or policies) resolve the nondeterministic choices in a MDP [PZ93,Var85,BA95].

Definition 2.3. Let $\Pi = (S, s_0, \tau, L)$ be a MDP and $s \in S$. An *s-scheduler* η on Π is a function from Ω_s^* to $\mathrm{Distr}(\wp(\mathrm{Distr}(S)))$ such that for all $\sigma \in \Omega_s^*$ we have $\eta(\sigma) \in \mathrm{Distr}(\tau(\mathrm{last}(\sigma)))$. We denote the set of all s-schedulers on Π by $\mathrm{Sch}_s(\Pi)$. When $s = s_0$ we omit it.

Note that our schedulers are randomized, i.e., in a finite path σ a scheduler chooses an element of $\tau(\mathrm{last}(\sigma))$ probabilistically. Under a scheduler η, the probability that the next state reached after the path σ is t, equals $\sum_{\pi\in\tau(\mathrm{last}(\sigma))} \eta(\sigma)(\pi)\cdot \pi(t)$. In this way, a scheduler induces a probability measure on $\mathcal{B}_s$ as usual.

Definition 2.4. Let Π be a MDP, $s \in S$, and η an s-scheduler on Π. We define the probability measure $\mu_{s,\eta}$ as the unique measure on $\mathcal{B}_s$ such that for all $s_0 s_1 \dots s_n \in \Omega_s^*$

$$\mu_{s,\eta}(\langle s_0 s_1 \dots s_n\rangle) = \prod_{i=0}^{n-1} \sum_{\pi\in\tau(s_i)} \eta(s_0 s_1 \dots s_i)(\pi)\cdot \pi(s_{i+1}).$$

We recall the notions of deterministic and history independent schedulers.

Definition 2.5. Let Π be a MDP, $s \in S$, and η an s-scheduler of Π. We say that η is *deterministic* if $\eta(\sigma)(\pi_i)$ is either 0 or 1 for all $\pi_i \in \tau(\mathrm{last}(\sigma))$ and all $\sigma \in \Omega_s^*$. We say that a scheduler is *history independent* (HI) if for all finite paths σ_1, σ_2 of Π with $\mathrm{last}(\sigma_1) = \mathrm{last}(\sigma_2)$ we have $\eta(\sigma_1) = \eta(\sigma_2)$ The set of all deterministic and HI s-schedulers will be denoted by $\mathrm{Sch}_s^{\mathrm{HI}}(\Pi)$.

Definition 2.6. Let Π be a MDP, $s \in S$, and $\Delta \in \mathcal{B}_s$. Then the *maximal and minimal probabilities of* Δ, $\mu_s^+(\Delta), \mu_s^-(\Delta)$, are defined by

$$\mu_s^+(\Delta) \triangleq \sup_{\eta\in\mathrm{Sch}_s(\Pi)} \mu_{s,\eta}(\Delta) \quad \text{and} \quad \mu_s^-(\Delta) \triangleq \inf_{\eta\in\mathrm{Sch}_s(\Pi)} \mu_{s,\eta}(\Delta).$$

A scheduler that attains $\mu_s^+(\Delta)$ or $\mu_s^-(\Delta)$ is called a *maximizing* or *minimizing* scheduler respectively.

We define the notion of (finite) convex combination of schedulers.

Definition 2.7. Let Π be a MDP, $s \in S$. An s-scheduler η is a *convex combination* of the s-schedulers $\eta_1, \dots, \eta_n$ if there are $\alpha_1, \dots, \alpha_n \in [0, 1]$ with $\alpha_1 + \dots + \alpha_n = 1$ such that for all $\Delta \in \mathcal{B}_s$, $\mu_{s,\eta}(\Delta) = \alpha_1\mu_{s,\eta_1}(\Delta) + \dots + \alpha_n\mu_{s,\eta_n}(\Delta)$.

Note that taking the convex combination η of η_1 and η_2 as functions, i.e., $\eta(\sigma)(\pi) = \alpha\eta_1(\sigma)(\pi) + (1-\alpha)\eta_2(\sigma)(\pi)$, does not imply that η is a convex combination of η_1 and η_2 in the sense above.

3 Conditional Probabilities over MDPs

The conditional probability $P(A \mid B)$ is the probability of an event A, given the occurrence of another event B. Recall that given a probability space (Ω, F, P) and two events $A, B \in F$ with $P(B) > 0$, $P(A \mid B)$ is defined as $P(A \cap B)/P(B)$.

If $P(B) = 0$, then $P(A \mid B)$ is undefined. In particular, given a MDP Π, a scheduler η and a state s, $(\Omega_s, \mathcal{B}_s, \mu_{s,\eta})$ is a probability space. So, for two sets of paths $\Delta_1, \Delta_2 \in \mathcal{B}_s$ with $\mu_{s,\eta}(\Delta_2) > 0$, the conditional probability of Δ_1 given Δ_2 is $\mu_{s,\eta}(\Delta_1 \mid \Delta_2) = \mu_{s,\eta}(\Delta_1 \cap \Delta_2)/\mu_{s,\eta}(\Delta_2)$. If $\mu_{s,\eta}(\Delta_2) = 0$, then $\mu_{\eta,s}(\Delta_1 \mid \Delta_2)$ is undefined. For technical reasons, we define the maximum and minimum conditional probabilities for all $\Delta_2 \in \mathcal{B}_s$.

Definition 3.1. Let Π be a MDP. The *maximal and minimal conditional probabilities* $\mu_s^+(\Delta_1|\Delta_2)$, $\mu_s^-(\Delta_1|\Delta_2)$ of sets of paths $\Delta_1, \Delta_2 \in \mathcal{B}_s$ are defined by

$$\mu_s^+(\Delta_1|\Delta_2) \triangleq \begin{cases} \sup_{\eta \in \mathrm{Sch}_{\Delta_2}^{>0}} \mu_{s,\eta}(\Delta_1|\Delta_2) & \text{if } \mathrm{Sch}_{\Delta_2}^{>0} \neq \emptyset, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mu_s^-(\Delta_1|\Delta_2) \triangleq \begin{cases} \inf_{\eta \in \mathrm{Sch}_{\Delta_2}^{>0}} \mu_{s,\eta}(\Delta_1|\Delta_2) & \text{if } \mathrm{Sch}_{\Delta_2}^{>0} \neq \emptyset, \\ 1 & \text{otherwise,} \end{cases}$$

where $\mathrm{Sch}_{\Delta_2}^{>0} = \{\eta \in \mathrm{Sch}_s(\Pi) \mid \mu_{s,\eta}(\Delta_2) > 0\}$.

The following lemma generalizes Lemma 6 of [BA95] to conditional probabilities.

Lemma 3.2. *Given $\Delta_1, \Delta_2 \in \mathcal{B}_s$, its maximal and minimal conditional probabilities are related by: $\mu_s^+(\Delta_1|\Delta_2) = 1 - \mu_s^-(\Omega_s - \Delta_1|\Delta_2)$.*

4 Conditional Probabilistic Temporal Logic

The logic cpCTL extends pCTL with formulas of the form $\mathbf{P}_{\bowtie a}[\varphi|\psi]$. Intuitively, $\mathbf{P}_{\leq a}[\varphi|\psi]$ holds if the probability that φ holds given that ψ holds is at most a.

Definition 4.1. The cpCTL logic is defined as the set of state and path formulas, i.e., cpCTL $\triangleq$ Stat $\cup$ Path, where Stat and Path are defined inductively:

$$\begin{aligned} \mathcal{P} &\subseteq \mathrm{Stat}, \\ \varphi, \psi \in \mathrm{Stat} &\Rightarrow \varphi \wedge \psi, \neg\varphi \in \mathrm{Stat}, \\ \varphi, \psi \in \mathrm{Path} &\Rightarrow A\varphi, E\varphi, \mathbf{P}_{\bowtie a}[\varphi], \mathbf{P}_{\bowtie a}[\varphi|\psi] \in \mathrm{Stat}, \\ \varphi, \psi \in \mathrm{Stat} &\Rightarrow \varphi\, \mathcal{U} \psi, \Diamond\varphi, \Box\, \varphi \in \mathrm{Path}\,. \end{aligned}$$

Here $\bowtie \in \{<, \leq, >, \geq\}$ and $a \in [0, 1]$.

Semantics. Satisfiability of state-formulas ($s \models \varphi$ for a state s) and path-formulas ($\omega \models \psi$ for a path ω) is defined as an extension of satisfiability for pCTL. Satisfiability of the logical, temporal, and pCTL operators is defined in the usual way. For the conditional probabilistic operators we define

$$\begin{aligned} s \models \mathbf{P}_{\leq a}[\varphi|\psi] &\Leftrightarrow \mu_s^+(\{\omega \in \Omega_s \mid \omega \models \varphi\}|\{\omega \in \Omega_s \mid \omega \models \psi\}) \leq a, \\ s \models \mathbf{P}_{\geq a}[\varphi|\psi] &\Leftrightarrow \mu_s^-(\{\omega \in \Omega_s \mid \omega \models \varphi\}|\{\omega \in \Omega_s \mid \omega \models \psi\}) \geq a, \end{aligned}$$

and similarly for $s \models \mathbf{P}_{<a}[\varphi|\psi]$ and $s \models \mathbf{P}_{>a}[\varphi|\psi]$. Following [BA95] we define

$$\begin{aligned} \mathbf{P}_s^+[\varphi] &\triangleq \mu_s^+(\{\omega \in \Omega_s \mid \omega \models \varphi\}), \\ \mathbf{P}_s^+[\varphi|\psi] &\triangleq \mu_s^+(\{\omega \in \Omega_s \mid \omega \models \varphi\}|\{\omega \in \Omega_s \mid \omega \models \psi\}), \\ \mathbf{P}_{s,\eta}[\varphi|\psi] &\triangleq \mu_{s,\eta}(\{\omega \in \Omega_s \mid \omega \models \varphi\}|\{\omega \in \Omega_s \mid \omega \models \psi\}) \end{aligned}$$

and we define $\mathbf{P}_s^-[\varphi|\psi]$ and $\mathbf{P}_s^-[\varphi]$ analogously.

Observation 4.2. As usual, for checking if $s \models \mathbf{P}_{\bowtie a}[\varphi|\psi]$, we only need to consider the cases where $\varphi = \varphi_1 \mathcal{U} \varphi_2$ and where ψ is either $\psi_1 \mathcal{U} \psi_2$ or $\Box\psi_1$. This follows using $\Box\varphi \leftrightarrow \neg\Diamond\neg\varphi$, $\Diamond\varphi \leftrightarrow \mathbf{true}\mathcal{U}\varphi$, and the relations

$$\mathbf{P}_s^+[\neg\varphi|\psi] = 1 - \mathbf{P}_s^-[\varphi|\psi] \qquad \mathbf{P}_s^-[\neg\varphi|\psi] = 1 - \mathbf{P}_s^+[\varphi|\psi]$$

derived from Lemma 3.2. Because there is no way to relate $\mathbf{P}^+[\varphi|\psi]$ and $\mathbf{P}^+[\varphi|\neg\psi]$, we have to provide two algorithms, one to compute $\mathbf{P}^+[\varphi|\psi_1 \mathcal{U} \psi_2]$ and one to compute $\mathbf{P}^+[\varphi|\Box\psi_1]$

5 Deterministic and Semi History-Independent Schedulers

Recall that there exist maximizing and minimizing schedulers on pCTL that are deterministic and HI [BA95]. We show that for cpCTL deterministic schedulers still suffice to reach optimal conditional probability. Because we now have to solve a non-linear optimization problem, the proof differs from the pCTL case in an essential way. We also show that HI schedulers do not suffice and we introduce semi history-independent schedulers that do attain optimal conditional probability.

To simplify notation, for a deterministic scheduler η, we use $\eta(\sigma)$ to denote the unique distribution $\pi \in \tau(\mathrm{last}(\sigma))$ such that $\eta(\sigma)(\pi) = 1$.

5.1 Semi History-Independent Schedulers

The following example shows that maximizing schedulers are not necessarily HI.

Example 5.1. Let Π be the MDP of Figure 2 and the conditional probability $\mathbf{P}_{s_0,\eta}[\Diamond B|\Diamond P]$. There are only three deterministic history independent schedulers, choosing π_1, π_2, or π_3 in s_0. For the first one, the conditional probability is undefined and for the second and third it is 0. The scheduler η that maximizes $\mathbf{P}_{s_0,\eta}[\Diamond B|\Diamond P]$ satisfies $\eta(s_0) = \pi_3$, $\eta(s_0 s_3) = \pi_5$, and $\eta(s_0 s_3 s_0) = \pi_1$. Since η chooses on s_0 first π_2 and later π_1, η is not history independent.

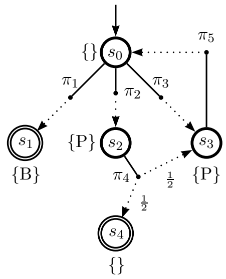


Fig. 2: MDP

However, there exists a maximizing scheduler that is "nearly HI" in the sense that it always takes the same decision *before* the system reaches a certain condition φ and also always takes the same decision *after* φ. This family of schedulers is called φ-semi history independent (φ-sHI for short).

Definition 5.2. Let $\Pi = (S, s_0, \tau, L)$ be a MDP, $s \in S$, η a scheduler of Π, and $\varphi \in \mathrm{Stat}$. We say that η is a φ-sHI *s-scheduler* if it satisfies

1. for all $\sigma_1, \sigma_2 \in \Omega_s^*$, if $\mathrm{last}(\sigma_1) = \mathrm{last}(\sigma_2)$ and $\sigma_1, \sigma_2 \not\models \Diamond\varphi$, then $\eta(\sigma_1) = \eta(\sigma_2)$;
2. for all $\sigma \in \Omega_s^*$, if $\sigma \models \Diamond\varphi$, then for all $\sigma', \sigma'' \in \Omega^*_{\mathrm{last}(\sigma)}$ such that $\sigma \sqsubseteq \sigma'$, $\sigma \sqsubseteq \sigma''$, and $\mathrm{last}(\sigma') = \mathrm{last}(\sigma'')$ we have $\eta(\sigma') = \eta(\sigma'')$.

Here $\mathrm{last}(s_0 s_1 \ldots s_n) = s_n$, $\mathrm{tail}(s_0 s_1 \ldots s_n) = s_1 \ldots s_n$, and $\sqsubseteq$ denotes the prefix order over finite paths, i.e. $\sigma' \sqsubseteq \sigma \Leftrightarrow \sigma = \sigma'\sigma''$ for some σ''.

Theorem 5.3. *Let Π be a MDP, $s \in S$, and $\varphi_1 \mathcal{U} \varphi_2$, $\psi_1 \mathcal{U} \psi_2$, $\Box \psi_1 \in$ cpCTL. There exists a $(\neg\varphi_1 \vee \varphi_2 \vee \neg\psi_1 \vee \psi_2)$-sHI s-scheduler η' such that*

$$\mathbf{P}_{s,\eta'}[\varphi_1 \mathcal{U} \varphi_2 | \psi_1 \mathcal{U} \psi_2] = \mathbf{P}^+_s[\varphi_1 \mathcal{U} \varphi_2 | \psi_1 \mathcal{U} \psi_2]$$

and a $(\neg\varphi_1 \vee \varphi_2 \vee \neg\psi_1)$-sHI s-scheduler η'' such that

$$\mathbf{P}_{s,\eta''}[\varphi_1 \mathcal{U} \varphi_2 | \Box \psi_1] = \mathbf{P}^+_s[\varphi_1 \mathcal{U} \varphi_2 | \Box \psi_1].$$

We define $\varphi_U \triangleq \neg\varphi_1 \vee \varphi_2 \vee \neg\psi_1 \vee \psi_2$ and $\varphi_\Box \triangleq \neg\varphi_1 \vee \varphi_2 \vee \neg\psi_1$. We refer to φ_U (resp. $\varphi_\Box$) as the until (resp. globally) *stopping condition*.

5.2 Deterministic Schedulers

Lemma 5.4. *Let $v_1, v_2 \in [0, \infty)$ and $w_1, w_2 \in (0, \infty)$. Then the function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) \triangleq \frac{xv_1 + (1-x)v_2}{xw_1 + (1-x)w_2}$ is monotonous.*

Proof. $f'(x) = \frac{v_1 w_2 - v_2 w_1}{(xw_1 - (1-x)w_2)^2}$ which is always ≥ 0 or always ≤ 0. $\square$

The following result states that taking the convex combination of schedulers does not increase the conditional probability $\mathbf{P}[\varphi|\psi]$.

Lemma 5.5. *Let Π be a MDP, s a state, and φ, ψ path formulas. Suppose that the s-scheduler η is a convex combination of η_1 and η_2. Then $\mathbf{P}_{s,\eta}[\varphi|\psi] \leq \max(\mathbf{P}_{s,\eta_1}[\varphi|\psi], \mathbf{P}_{s,\eta_2}[\varphi|\psi])$.*

Proof. Applying the above lemma to

$$[0,1] \ni \alpha \mapsto \frac{\alpha \mathbf{P}_{s,\eta_1}[\varphi \wedge \psi] + (1-\alpha)\mathbf{P}_{s,\eta_2}[\varphi \wedge \psi]}{\alpha \mathbf{P}_{s,\eta_1}[\psi] + (1-\alpha)\mathbf{P}_{s,\eta_2}[\psi]}$$

we get that the maximum is reached at $\alpha = 0$ or $\alpha = 1$. Because η is a convex combination of η_1 and η_2, $\mathbf{P}_{s,\eta}[\varphi|\psi] \leq \mathbf{P}_{s,\eta_2}[\varphi|\psi]$ (in the first case) or $\mathbf{P}_{s,\eta}[\varphi|\psi] \leq \mathbf{P}_{s,\eta_1}[\varphi|\psi]$ (in the second case). $\square$

Theorem 5.6. *Let Π be a MDP, s a state, and φ a path formula. Then every s-scheduler on Π is a convex combination of deterministic φ-sHI s-schedulers.*

Theorem 5.7. *Let Π be a MDP, $s \in S$, and $\varphi_1 \mathcal{U} \varphi_2$, $\psi_1 \mathcal{U} \psi_2$, $\Box \psi_1 \in$ cpCTL. There exists a deterministic φ_U-sHI s-scheduler η' such that*

$$\mathbf{P}_{s,\eta'}[\varphi_1 \mathcal{U} \varphi_2 | \psi_1 \mathcal{U} \psi_2] = \mathbf{P}^+_s[\varphi_1 \mathcal{U} \varphi_2 | \psi_1 \mathcal{U} \psi_2]$$

and a deterministic $\varphi_\Box$-sHI s-scheduler η'' such that

$$\mathbf{P}_{s,\eta''}[\varphi_1 \mathcal{U} \varphi_2 | \Box \psi_1] = \mathbf{P}^+_s[\varphi_1 \mathcal{U} \varphi_2 | \Box \psi_1],$$

where φ_U and $\varphi_\Box$ are the stopping conditions.

Example 5.8. Consider the MDP and cpCTL formula of Example 5.1. According to Theorem 5.7 there exists a deterministic and $(B \vee P)$-sHI scheduler that maximizes $\mathbf{P}_{s_0,\eta}[\Diamond B | \Diamond P]$. In this case, a maximizing scheduler will take always the same decision (π_3) before the system reaches s_3 (a state satisfying the until stopping condition $(B \vee P)$) and always the same decision (π_1) after the system reaches s_3.

6 Model Checking cpCTL

Model checking cpCTL means checking if a state s satisfies a certain state formula φ. We focus on formulas of the form $\mathbf{P}_{\leq a}[\varphi|\psi]$ and show how to compute $\mathbf{P}_s^+[\varphi|\psi]$ given $\varphi, \psi \in$ Path. The case $\mathbf{P}_s^-[\varphi|\psi]$ is similar.

Recall that model checking pCTL is based on the Bellman-equations. For instance, $\mathbf{P}_s^+[\Diamond B] = \max_{\pi\in\tau(s)} \sum_{t\in\mathrm{succ}(s)} \pi(t)\mathbf{P}_t^+[\Diamond B]$ whenever $s \not\models B$. So a scheduler η that maximizes $\mathbf{P}_s[\Diamond B]$ chooses $\pi \in \tau(s)$ maximizing $\sum_{t\in\mathrm{succ}(s)}\pi(t)\cdot \mathbf{P}_t^+[\Diamond B]$. In a successor state t, η still behaves as a scheduler that maximizes $\mathbf{P}_t[\Diamond B]$. As shown below, such a local Bellman-equation is not true for conditional probabilities: a scheduler that maximizes a conditional probability such as $\mathbf{P}_s[\Diamond B|\Box P]$ does not necessarily maximize $\mathbf{P}_t[\Diamond B|\Box P]$ for successors t of s.

Example 6.1. Again, consider the MDP and cpCTL formula $\mathbf{P}_{\leq a}[\Diamond B|\Box P]$ of Figure 1. There are only two deterministic schedulers. The first one, η_1, chooses π_2 when the system reaches the state s_2 and the second one, η_2, chooses π_3 when the system reaches s_2. For the first one $\mathbf{P}_{s_0,\eta_1}[\Diamond B|\Box P] = 1 - \frac{2\alpha}{7}$, and for the second one $\mathbf{P}_{s_0,\eta_2}[\Diamond B|\Box P] = \frac{30}{31}$. So $\mathbf{P}_{s_0}^+[\Diamond B|\Box P] = \max(1 - \frac{2\alpha}{7}, \frac{30}{31})$. Therefore, if $\alpha \geq \frac{7}{62}$ the scheduler that maximizes $\mathbf{P}_{s_0}[\Diamond B|\Box P]$ is η_2 ($\mathbf{P}_{s_0,\eta_2}[\Diamond B|\Box P] = \mathbf{P}_{s_0}^+[\Diamond B|\Box P]$) and otherwise it is η_1 ($\mathbf{P}_{s_0,\eta_1}[\Diamond B|\Box P] = \mathbf{P}_{s_0}^+[\Diamond B|\Box P]$).

Furthermore, $\mathbf{P}_{s_1}^+[\Diamond B|\Box P] = 1$ and $\mathbf{P}_{s_2}^+[\Diamond B|\Box P] = 1-2\alpha$; the scheduler that obtains this last maximum is the one that chooses π_2 in s_2.

So, if $\alpha \geq \frac{7}{62}$ the scheduler that maximizes the conditional probability from s_0 is taking a different decision than the one that maximize the conditional probability from s_2. Furthermore, for all α, $\max(1 - \frac{2\alpha}{7}, \frac{30}{31}) = \mathbf{P}_{s_0}^+[\Diamond B|\Box P] \neq \frac{3}{4}\mathbf{P}_{s_1}^+[\Diamond B|\Box P] + \frac{1}{4}\mathbf{P}_{s_2}^+[\Diamond B|\Box P] = 1 - \frac{1}{2}\alpha$, showing that the Bellman-equation from above does not generalize to cpCTL.

An obvious way to compute $\mathbf{P}_s^+[\varphi|\psi]$ is by computing the pairs ($\mathbf{P}_{s,\eta}[\varphi\wedge\psi]$, $\mathbf{P}_{s,\eta}[\psi]$) for all sHI schedulers η, and taking the maximum quotient $\mathbf{P}_{s,\eta}[\varphi\wedge\psi]/\mathbf{P}_{s,\eta}[\psi]$. We present two methods to avoid the computation of certain pairs of acyclic MDPs. We can use these for a MDP with cycles by first transforming it to an equivalent acyclic one using the strongly connected component structure.

6.1 Acyclic MDP

Note that every MDP has cycles associated to absorbing states. We call a MDP acyclic if it the only if the only cycles are selfloops taken with probability one.

Definition 6.2. A MDP Π is called *acyclic* if for all states $s \in S$ and all $\pi \in \tau(s)$ we have $\pi(s) = 0$ or $\pi(s) = 1$ and for all paths ω and all $i < j$ such that $\omega_i = \omega_j$ we have $\omega_i = \omega_{i+1} = \cdots = \omega_j$.

The idea behind the algorithm for acyclic MDPs is as follows. We label each state s by a sequence $(p_1, q_1), \ldots, (p_n, q_n)$ of pairs of probabilities, where $p_i = \mathbf{P}_{s,\eta_i}[\varphi\wedge\psi]$ and $q_i = \mathbf{P}_{s,\eta_i}[\psi]$ for a certain sHI s-scheduler η_i. The algorithm starts by labeling each leaf s with a single pair $(\mathbf{P}_{s,\eta}[\varphi\wedge\psi], \mathbf{P}_{s,\eta}[\psi])$ for the unique deterministic sHI s-scheduler η. The labeling is propagated towards the root node s_0. We obtain the maximum conditional probability $\mathbf{P}_{s_0}^+[\varphi|\psi]$ as the maximum quotient p/q for all (p,q) in the labeling of s_0. Section 6.3 shows that certain pairs can be discarded when propagating the labeling.

Definition 6.3. Let L be the set of expressions of the form $(p_1,q_1)\vee\cdots\vee(p_n,q_n)$ where $p_i,q_i\in[0,\infty)$ and $q_i\geq p_i$, for all $n\in\mathbb{N}^\star$. On L we consider the smallest congruence relation $\equiv_1$ satisfying (Idempotence) $(p_1,q_1)\vee(p_1,q_1)\equiv_1(p_1,q_1)$, (Associativity) $((p_1,q_1)\vee(p_2,q_2))\vee(p_3,q_3)\equiv_1(p_1,q_1)\vee((p_2,q_2)\vee(p_3,q_3))$, (Commutatitivity) $(p_1,q_1)\vee(p_2,q_2)\equiv_1(p_2,q_2)\vee(p_1,q_1)$. Note that $(p_1,q_1)\vee\cdots\vee(p_n,q_n)\equiv_1(p'_1,q'_1)\ldots(p'_{n'},q'_{n'})$ if and only if $\{(p_1,q_1),\ldots,(p_n,q_n)\}=\{(p'_1,q'_1),\ldots,(p'_{n'},q'_{n'})\}$.

We let L_1 be the set of equivalence classes and denote the projection map $L\to L_1$ that maps each expression to its equivalence class by f_1. On L we also define *maximum quotient* $\top:L\to[0,\infty)$, and *minimum quotient* $\bot:L\to[0,\infty)$ by $\top(\bigvee_{i=1}^{n}(p_i,q_i))\triangleq\max(\{\frac{p_i}{q_i}|q_i\neq 0,i=1,\ldots,n\}\cup\{0\})$ and $\bot(\bigvee_{i=1}^{n}(p_i,q_i))\triangleq\min(\{\frac{p_i}{q_i}|q_i\neq 0,i=1,\ldots,n\}\cup\{1\})$.

Note that $\top$ and $\bot$ induce maps $\top_1\colon L_1\to[0,\infty)$ and $\bot_1\colon L_1\to[0,\infty)$ such that $\top_1\circ f_1=\top$ and $\bot_1\circ f_1=\bot$.

Definition 6.4. Let Π be a MDP. We define the function $\delta:S\times\mathrm{Stat}\times\mathrm{Path}\times\mathrm{Path}\to L$ by $\delta(s,\varphi,\varphi,\psi)\triangleq\bigvee_{\eta\in\mathrm{Sch}_s^\varphi(\Pi)}\big(\mathbf{P}_{s,\eta}[\varphi\wedge\psi],\mathbf{P}_{s,\eta}[\psi]\big)$ and we define $\delta_1\colon S\times\mathrm{Stat}\times\mathrm{Path}\times\mathrm{Path}\to L_1$ by $\delta_1\triangleq f_1\circ\delta$.

When no confusion arises, we omit the subscripts 1 and omit the projection map f_1, writing $(p_1,q_1)\vee\cdots\vee(p_n,q_n)$ for the equivalence class it generates.

Example 6.5. In Figure 3 we show the value $\delta(s,B\vee\neg P,\Diamond B,\Box P)$ associated to each state s of the MDP previously presented in Figure 1.

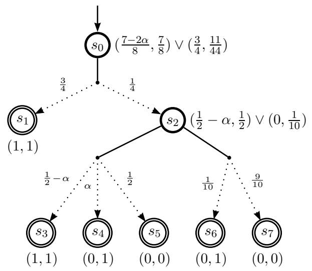


Fig. 3: δ-values

The following result says that we can compute maximum conditional probability from $\delta_s^{\mathcal{U}}$ or $\delta_s^{\Box}$.

Theorem 6.6. *Given $\Pi=(S,s_0,L,\tau)$ an acyclic MDP, and $\varphi_1,\varphi_2,\psi_1,\psi_2\in\mathrm{Stat}$. Then*

$$\mathbf{P}_s^+[\varphi_1\mathcal{U}\varphi_2|\psi_1\mathcal{U}\psi_2]=\top(\overbrace{\delta(s,\varphi_U,\varphi_1\mathcal{U}\varphi_2,\psi_1\mathcal{U}\psi_2)}^{\triangleq\delta_s^{\mathcal{U}}(\varphi_1,\varphi_2,\psi_1,\psi_2)})$$

and

$$\mathbf{P}_s^+[\varphi_1\mathcal{U}\varphi_2|\Box\psi_1]=\top(\overbrace{\delta(s,\varphi_\Box,\varphi_1\mathcal{U}\varphi_2,\Box\psi_1)}^{\triangleq\delta_s^{\Box}(\varphi_1,\varphi_2,\psi_1)}).$$

6.2 Extension to general MDP

Now, we extend our results to general, not necessarily acyclic, MDPs. We first reduce all cycles in Π and create a new acyclic reduced MDP $[\Pi]$ such that the probabilities involved in the computation of $\mathbf{P}^+[-|-]$ are preserved. We do so by removing every strongly connected component (SCC) c of (the graph of) a MDP Π, keeping only input states and transitions to output states. We show that $\mathbf{P}^+[-|-]$ on $[\Pi]$ is equal to the corresponding value on Π. For this, we have to make sure that states satisfying the stopping condition are ignored when we are removing the SCCs.

Identifying SCCs. Our first step is to make stopping condition states absorbing.

Definition 6.7. Let $\Pi = (S, s_0, \tau, L)$ be a MDP and $\varphi \in$ Stat a state formula. We define a new MDP $\langle\Pi\rangle_\varphi = (S, s_0, \langle\tau\rangle_\varphi, L)$ where $\langle\tau\rangle_\varphi(s)$ is equal to $\tau(s)$ if $s \not\models \varphi$ and to 1_s otherwise.

Typically φ will be either the until stopping condition (φ_U) or the globally stopping condition ($\varphi_\Box$).

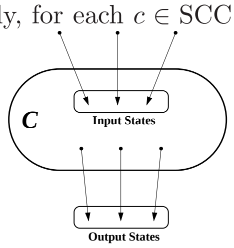
To recognize cycles in the MDP we define a graph associated to it.

Definition 6.8. Let $\Pi = (S, s_0, \tau, L)$ be MDP and $\varphi \in$ Stat. We define the digraph $G = G_{\Pi,\varphi} = (S, \rightarrow)$ associated to $\langle\Pi\rangle_\varphi = (S, s_0, \langle\tau\rangle_\varphi, L)$ where $\rightarrow$ satisfies $u \rightarrow v \Leftrightarrow \exists\pi \in \langle\tau\rangle_\varphi(u).\pi(v) > 0$.

Now we let $\mathrm{SCC} = \mathrm{SCC}_{\Pi,\varphi} \subseteq \wp(S)$ be the set of SCC of G. For each SCC c we define the sets Inp_c of all states in c that have an incoming transition of Π from a state outside of c; we also define the set Out_c of all states outside of c that have an incoming transition from a state of c. Formally, for each $c \in \mathrm{SCC}$ we define

$$Inp_c \triangleq \{u \in c \mid \exists\, s \in S - c.\exists\, \pi \in \tau(s).\pi(u) > 0\},$$

$$Out_c \triangleq \{s \in S - c \mid \exists\, u \in c.\exists\, \pi \in \tau(u).\pi(s) > 0\}.$$



We then associate a MDP Π_c to each SCC c of G. The space of states of Π_c is $c \cup Out_c$ and the transition relation is induced by the transition relation of Π.

Definition 6.9. Let Π be a MDP and $c \in$ SCC be a scc in Π. We pick an arbitrary element s_c of Inp_c and define the MDP $\Pi_c = (S_c, s_c, \tau_c, L)$ where $S_c = c \cup Out_c$ and $\tau_c(s)$ is equal to $\{1_s\}$ if $s \in Out_c$ and to $\tau(s)$ otherwise.

Defining the new acyclic MDP. To obtain a reduced acyclic MDP from the original one we first define the probability of reaching one state from another according to a given HI scheduler in the following way.

Definition 6.10. Let $\Pi = (S, s_0, \tau, L)$ be a MDP, and η be a HI scheduler on Π. Then for each $s, t \in S$ we define the function R such that $R_\Pi(s \overset{\eta}{\rightsquigarrow} t) \triangleq \mu_{s,\eta}(\{\omega \in \Omega_s \mid \exists\, i.\omega_i = t\})$.

Now we are able to define an acyclic MDP $[\Pi]$ related to Π such that $\mathbf{P}^+_{[\Pi]}[-|-] = \mathbf{P}^+_\Pi[-|-]$.

Definition 6.11. Let $\Pi = (S, s_0, \tau, L)$ be a MDP. Then we define $[\Pi]$ as $([S], s_0, [\tau], L)$ where

$$[S] = \overbrace{S - \bigcup_{c \in \mathrm{SCC}} c}^{S_{com}} \cup \overbrace{\bigcup_{c \in \mathrm{SCC}} Inp_c}^{S_{inp}}$$

and for all $s \in [S]$ the set $[\tau](s)$ of probabilistic distributions on $[S]$ is given by

$$[\tau](s) = \begin{cases} \tau(s) & \text{if } s \in S_{com}, \\ \{\lambda \in [S].R_{\Pi_{c_s}}(s \overset{\eta}{\rightsquigarrow} t)) \mid \eta \in \mathrm{Sch}_s^{\mathrm{HI}}(\Pi_{c_s})\} & \text{if } s \in S_{inp}. \end{cases}$$

Here c_s is the SCC associated to s.

Theorem 6.12. *Let $\Pi = (S, s_0, \tau, L)$ be a MDP, and $\mathbf{P}_{\leq a}[\varphi|\psi] \in$ cpCTL. Then $[\Pi]$ is an acyclic MDP and $\mathbf{P}^{+}_{s_0,\Pi}[\varphi|\psi] = \mathbf{P}^{+}_{s_0,[\Pi]}[\varphi|\psi]$, where $\mathbf{P}^{+}_{s,\Pi'}[-|-]$ represents $\mathbf{P}^{+}_{s}[-|-]$ on the MDP Π'.*

Finally we can use the technique for acyclic MDPs on the reduced MDP in order to obtain $\mathbf{P}^{+}_{s_0}[-|-]$. Note that to compute $\mathbf{P}^{+}_{s_0}[-|-]$ it is not necessary to compute reachability properties on SCC that are not reachable on G from its initial state, so in model checking we avoid that.

6.3 Optimizations

We have already shown that δ is computable. Now we show two optimizations in order to compute δ in a more efficient way.

Optimization 1: Reusing Information. We now show how to compute $\delta_s^{\mathcal{U}}(\varphi_1, \varphi_2, \psi_1, \psi_2)$ and $\delta_s^{\Box}(\varphi_1, \varphi_2, \psi_1, \psi_2)$ recursively in s. The base cases of the recursion are the states where the stopping condition holds. Because there exists an optimizing scheduler that is sHI, we only need to consider HI (and determininstic) schedulers in such a state. In the recursive case we can express $\delta_s^{\mathcal{U}}$ (resp. $\delta_s^{\Box}$) in terms of the $\delta_t^{\mathcal{U}}$ (resp. $\delta_t^{\Box}$) of the successor states t of s. Therefore, if we encounter the same state t in more than one branch of the recursive computation, we can reuse the previously computed value of $\delta_t^{\mathcal{U}}$ (resp. $\delta_t^{\Box}$).

To do this, we now define a *scalar multiplication operator* $\odot$ and an *addition operator* $\oplus$ on L.

Definition 6.13. We define $\odot : [0, \infty) \times L \to L$ and $\oplus : L \times L \to L$ by $c \odot \bigvee_{i=1}^{n}(p_i, q_i) \triangleq \bigvee_{i=1}^{n}(c \cdot p_i, c \cdot q_i)$ and $\bigvee_{i=1}^{n}(p_i, q_i) \oplus \bigvee_{j=1}^{m}(p'_j, q'_j) \triangleq \bigvee_{i=1}^{n}\bigvee_{j=1}^{m}(p_i + p'_j, q_i + q'_j)$.

Note that $\odot$ and $\oplus$ induce maps $\odot_1 \colon [0, \infty) \times L_1 \to L_1$ and $\oplus_1 : L_1 \times L_1 \to L_1$. As before, we omit the subscript 1 if that will not cause confusion.

The following result gives recursive equations for the values of $\delta_s^{\mathcal{U}}$ and $\delta_s^{\Box}$. If the MDP is acyclic, it can be used to compute these values.

Theorem 6.14. *Let Π be a MDP, $s \in S$, and $\varphi_1 \mathcal{U} \varphi_2, \psi_1 \mathcal{U} \psi_2, \Box\psi_1 \in$ Path. Then $\delta_s^{\mathcal{U}}(\varphi_1, \varphi_2, \psi_1, \psi_2) =$*

$$\begin{cases} \bigvee_{\eta \in \mathrm{Sch}_s^{\mathrm{HI}}(\Pi)}(\mathbf{P}_{s,\eta}[\psi_1 \mathcal{U} \psi_2], \mathbf{P}_{s,\eta}[\psi_1 \mathcal{U} \psi_2]) & \text{if } s \models \varphi_2, \\ \bigvee_{\eta \in \mathrm{Sch}_s^{\mathrm{HI}}(\Pi)}(\mathbf{P}_{s,\eta}[\varphi_1 \mathcal{U} \varphi_2], 1) & \text{if } s \models \neg\varphi_2 \wedge \psi_2, \\ \bigvee_{\eta \in \mathrm{Sch}_s^{\mathrm{HI}}(\Pi)}(0, \mathbf{P}_{s,\eta}[\psi_1 \mathcal{U} \psi_2]) & \text{if } s \models \neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\psi_2, \\ (0, 0) & \text{if } s \models \varphi_1 \wedge \neg\varphi_2 \wedge \neg\psi_1 \wedge \neg\psi_2, \\ \bigvee_{\pi \in \tau(s)} \left(\bigoplus_{t \in \mathrm{succ}(s)} \pi(t) \odot \delta_t^{\mathcal{U}}(\varphi_1, \varphi_2, \psi_1, \psi_2) \right) & \text{if } s \models \varphi_1 \wedge \neg\varphi_2 \wedge \psi_1 \wedge \neg\psi_2, \end{cases}$$

and $\delta_s^{\Box}(\varphi_1, \varphi_2, \psi_1) =$

$$\begin{cases} \bigvee_{\eta\in \mathrm{Sch}_s^{\mathrm{HI}}(\Pi)}(\mathbf{P}_{s,\eta}[\Box\psi_1], \mathbf{P}_{s,\eta}[\Box\psi_1]) & \textit{if } s \models \varphi_2, \\ (0,0) & \textit{if } s \models \neg\varphi_2 \wedge \neg\psi_1, \\ \bigvee_{\eta\in \mathrm{Sch}_s^{\mathrm{HI}}(\Pi)}(0, \mathbf{P}_{s,\eta}[\Box\psi_1]) & \textit{if } s \models \neg\varphi_1 \wedge \neg\varphi_2 \wedge \psi_1, \\ \bigvee_{\pi\in\tau(s)}\left(\bigoplus_{t\in \mathrm{succ}(s)} \pi(t)\odot \delta_t^{\Box}(\varphi_1,\varphi_2,\psi_1)\right) & \textit{if } s \models \varphi_1 \wedge \neg\varphi_2 \wedge \psi_1. \end{cases}$$

Optimization 2: Using pCTL **algorithms after the stopping condition.** Up to now we have computed $(\mathbf{P}_{s_0,\eta}[\varphi\wedge\psi], \mathbf{P}_{s_0,\eta}[\psi])$ for all sHI schedulers. The reason for this is that the (local) Bellman-equations do not hold for cpCTL. Therefore, it is not enough to know the values $\mathbf{P}_t^+[\varphi|\psi]$ for all successors t of s. However, in some cases, we can locally decide that one sHI scheduler is guaranteed to be better than another one. We now give some intuition for this; a formal claim is in Lemma 6.16 below.

For instance, let s be a state that is reachable from s_0. Assume that η' and η'' are sHI s-schedulers such that $\mathbf{P}_{s,\eta'}[\varphi\wedge\psi] = \mathbf{P}_{s,\eta''}[\varphi\wedge\psi]$ and $\mathbf{P}_{s,\eta'}[\psi] \leq \mathbf{P}_{s,\eta''}[\psi]$. Furthermore, assume that η_1 and η_2 are sHI s_0-schedulers that are equal except that η_1 behaves like η' "below" s and η_2 behaves like η'' "below" s. One can easily see that $\mathbf{P}_{s_0,\eta_1}[\varphi|\psi] \geq \mathbf{P}_{s_0,\eta_2}[\varphi|\psi]$. Therefore, when computing $\mathbf{P}_{s_0}^+[\varphi|\psi]$ we do not have to consider all sHI s-schedulers, but, in this case, we can omit η'' from consideration.

Similarly, if $\mathbf{P}_{s,\eta'}[\varphi\wedge\psi] \leq \mathbf{P}_{s,\eta''}[\varphi\wedge\psi]$ and $\mathbf{P}_{s,\eta'}[\psi] = \mathbf{P}_{s,\eta''}[\psi]$, then we do not have to consider the scheduler η'.

Finally, it follows from Lemma 5.4 that we don't have to consider the scheduler η'' if $\mathbf{P}_{s,\eta'}[\varphi\wedge\psi] + a = \mathbf{P}_{s,\eta''}[\varphi\wedge\psi]$ and $\mathbf{P}_{s,\eta'}[\psi] + a = \mathbf{P}_{s,\eta''}[\psi]$. This is used to show that when we reach a state s satisfying the stopping condition, we only have to compute $\mathbf{P}_s^+[\psi]$ and we do not have to consider conditional probabilities anymore.

As a consequence of these facts we do not have to compute $(\mathbf{P}_{s_0,\eta}[\varphi\wedge\psi], \mathbf{P}_{s_0,\eta}[\psi])$ for all sHI schedulers. In particular, if we reach a state satisfying the stopping condition we can always choose a scheduler that maximizes or minimizes one pCTL formula.

Definition 6.15. Consider the set of expressions L defined in Definition 6.3. On L we now consider the smallest congruence relation $\equiv_2$ containing $\equiv_1$ and satisfying (1) $(p_1,q_1)\vee(p_1,q_2) \equiv_2 (p_1, \min(q_1,q_2))$, (2) $(p_1,q_1)\vee(p_2,q_1) \equiv_2 (\max(p_1,p_2), q_1)$, (3) $(p_1+a, q_1+a)\vee(p_1,q_1) \equiv_2 (p_1+a, q_1+a)$, where $a\in[0,\infty)$. We write L_2 for the set of equivalence classes and denote the projection map $L_2 \to L$ by f_2.

Since $\equiv_1 \subseteq \equiv_2$, this projection maps factors through f_1, say $g\colon L_1 \to L_2$ is the unique map such that $g\circ f_1 = f_2$. The following seemingly innocent lemma is readily proven, but it contains the heart of this optimization. The fact that $\top$ and $\perp$ induce operations on L_2 means that it is correct to "simplify" expressions using $\equiv_2$ when we are interested in the maximum or minimum quotient. After that, we show that this implies that we do not have to consider *all* sHI schedulers when computing maximum or minimum conditional probabilities, but can on-the-fly omit some from consideration.

Lemma 6.16. *The operators* $\odot$, $\oplus$, $\top$, *and* $\bot$ *on* L *induce operators* $\odot_2$, $\oplus_2$, $\top_2$, *and* $\bot_2$ *on* L_2.

Definition 6.17. We define $\delta_2 : S \times \mathrm{Stat} \times \mathrm{Path} \times \mathrm{Path} \to L_2$ by $\delta_2 \triangleq f_2 \circ \delta$.

As usual, we omit subscripts 2 when confusion is unlikely. Note that with this convention Theorem 6.6 still holds. Finally, the following theorem allow us to recursively compute $\delta_s^{\mathcal{U}}$ and $\delta_s^{\Box}$ considering these last optimizations.

Theorem 6.18. *Let* Π *be a MDP,* $s \in S$*, and* $\varphi_1 \mathcal{U} \varphi_2, \psi_1 \mathcal{U} \psi_2, \Box\psi_1 \in \mathrm{Path}$*. Then* $\delta_s^{\mathcal{U}}(\varphi_1, \varphi_2, \psi_1, \psi_2) =$

$$\begin{cases} (\mathbf{P}_s^+[\psi_1 \mathcal{U} \psi_2], \mathbf{P}_s^+[\psi_1 \mathcal{U} \psi_2]) & \textit{if } s \models \varphi_2, \\ (\mathbf{P}_s^+[\varphi_1 \mathcal{U} \varphi_2], 1) & \textit{if } s \models \neg\varphi_2 \wedge \psi_2, \\ (0, \mathbf{P}_s^-[\psi_1 \mathcal{U} \psi_2]) & \textit{if } s \models \neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\psi_2, \\ (0,0) & \textit{if } s \models \varphi_1 \wedge \neg\varphi_2 \wedge \neg\psi_1 \wedge \neg\psi_2, \\ \bigvee_{\pi \in \tau(s)} \left(\bigoplus_{t \in \mathrm{succ}(s)} \pi(t) \odot \delta_t^{\mathcal{U}}(\varphi_1, \varphi_2, \psi_1, \psi_2) \right) & \textit{if } s \models \varphi_1 \wedge \neg\varphi_2 \wedge \psi_1 \wedge \neg\psi_2, \end{cases}$$

and $\delta_s^{\Box}(\varphi_1, \varphi_2, \psi_1) =$

$$\begin{cases} (\mathbf{P}_s^+[\Box\psi_1], \mathbf{P}_s^+[\Box\psi_1]) & \textit{if } s \models \varphi_2, \\ (0,0) & \textit{if } s \models \neg\varphi_2 \wedge \neg\psi_1, \\ (0, \mathbf{P}_s^-[\Box\psi_1]) & \textit{if } s \models \neg\varphi_1 \wedge \neg\varphi_2 \wedge \psi_1, \\ \bigvee_{\pi \in \tau(s)} \left(\bigoplus_{t \in \mathrm{succ}(s)} \pi(t) \odot \delta_t^{\Box}(\varphi_1, \varphi_2, \psi_1)) \right) & \textit{if } s \models \varphi_1 \wedge \neg\varphi_2 \wedge \psi_1. \end{cases}$$

7 Counterexamples

Counterexamples in model checking provide important diagnostic information used, among others, for debugging, abstraction-refinement [CGJ+00], and scheduler synthesis [LBB+01]. For systems without probability, a counterexample typically consists of a path violating the property under consideration. Counterexamples in MCs are sets of paths. E.g, a counterexample for the formula $\mathbf{P}_{\leq a}[\varphi]$ is a set Δ of paths, none satisfying φ, and such that the probability mass of Δ is greater than a [HK07,And06,AL06].

In MDPs, we first have to find the scheduler achieving the optimal probability. Both for pCTL and cpCTL, this scheduler can be derived from the algorithms computing the optimal probabilities [And06]. Once the optimal scheduler is fixed, the MDP can be turned into a Markov Chain and the approaches mentioned before can be used to construct counterexamples for pCTL. For cpCTL however, the situation is slightly more complex. It follows directly from the semantics that:

$$s \not\models \mathbf{P}_{\leq a}[\varphi|\psi] \quad \text{iff} \quad \exists \eta \in \mathrm{Sch}_s(\Pi). \frac{\mu_{s,\eta}(\{\omega \in \ \Omega_s | \omega \models \varphi \wedge \psi\})}{\mu_{s,\eta}(\{\omega \in \ \Omega_s | \omega \models \psi\})} > a.$$

Lemma 7.1. *Let* $a \in [0,1]$ *and consider the formula* $\mathbf{P}_{\leq a}[\varphi|\psi]$*. Let* $\Delta_\varphi \triangleq \{\omega \in \Omega \mid \omega \models \varphi\}$*,* $\Delta_1 \subseteq \Delta_{\varphi \wedge \psi}$*, and* $\Delta_2 \subseteq \Delta_{\neg\psi}$*. Then* $a < \mu_\eta(\Delta_1)/(1 - \mu_\eta(\Delta_2))$ *implies* $a < \mathbf{P}_\eta[\varphi|\psi]$.

Proof. The proof follows from $\mu_\eta(\Delta_1) \leq \mu_\eta(\Delta_{\varphi\wedge\psi})$ and $\mu_\eta(\Delta_2) \leq \mu_\eta(\Delta_{\neg\psi})$. Then $a < \frac{\mu_\eta(\Delta_1)}{1-\mu_\eta(\Delta_2)} \leq \frac{\mu_\eta(\Delta_{\varphi\wedge\psi})}{1-\mu_\eta(\Delta_{\neg\psi})} = \frac{\mu_\eta(\Delta_{\varphi\wedge\psi})}{\mu_\eta(\Delta_\psi)} = \mathbf{P}_\eta[\varphi|\psi]$. $\Box$

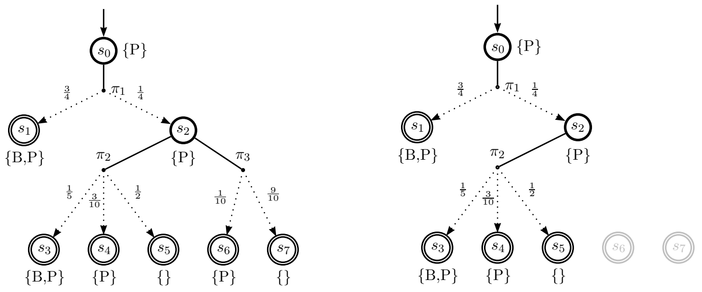
This leads to the following notion of counterexample.

Definition 7.2. A *counterexample* for $\mathbf{P}_{\leq a}[\varphi|\psi]$ is a pair (Δ_1, Δ_2) of measurable sets of paths satisfying $\Delta_1 \subseteq \Delta_{\varphi\wedge\psi}$, $\Delta_2 \subseteq \Delta_{\neg\psi}$, and $a < \mu_\eta(\Delta_1)/(1-\mu_\eta(\Delta_2))$, for some scheduler η.

Note that such sets Δ_1 and Δ_2 can be computed using the techniques on Markov Chains mentioned above.

Example 7.3. Consider the evaluation of $s_0 \models \mathbf{P}_{\leq 3/4}[\Diamond B|\Box P]$ on the MDP obtained by taking $\alpha = \frac{1}{10}$ in Example 2.2 (see Figure 4(a)). In this case the maximizing scheduler, say η, chooses π_2 in s_2. In Figure 4(b) we show the Markov Chain derived from MDP using η. In this setting we have $\mathbf{P}_{s_0,\eta}[\Diamond B|\Box P] = \frac{68}{70}$ and consequently s_0 does not satisfy this formula.

We show this fact with the notion of counterexample of Definition 7.2. Note that $\Delta_{\Diamond B\wedge\Box P} = \langle s_0 s_1\rangle \cup \langle s_0 s_2 s_3\rangle$ and $\Delta_{\neg\Box P} = \langle s_0 s_2 s_5\rangle$. Using Lemma 7.1 with $\Delta_1 = \langle s_0 s_1\rangle$ and $\Delta_2 = \langle s_0 s_2 s_5\rangle$ we have $\frac{3}{4} < \frac{\mu_\eta(\Delta_1)}{1-\mu_\eta(\Delta_2)} = \frac{3/4}{1-1/8} = \frac{6}{7}$. Consequently $\frac{3}{4} < \mathbf{P}_{s_0,\eta}[\Diamond B|\Box P]$, which proves that $s_0 \not\models \mathbf{P}_{\leq 3/4}[\Diamond B|\Box P]$.



4(a) MDP 4(b) Markov Chain

8 Conclusion and Future Work

In this paper we extended the probabilistic temporal logic pCTL to cpCTL, in which it is possible to express conditional probabilities. We showed that optimal scheduling decisions can always be reached by a deterministic and semi history-independent scheduler. Using this we presented an algorithm to check if a MDP satisfies a cpCTL-formula. Our algorithm first reduces the MDP to an acyclic MDP and then computes optimal conditional probabilities in over this reduction. Counterexamples for conditional formulas consist of two sets of paths in the MDP or MC. We have sketched an algorithm for counterexample generation.

A natural direction for future research is to extend pCTL* to cpCTL* and find algorithms for model checking cpCTL*. Furthermore, we plan to investigate ways to find better counterexamples in cpCTL model checking. Finally, we intend to implement our algorithms in a probabilistic model checker and apply cpCTL model checking to verify the correctness of anonymity protocols.

Acknowledgement. The authors thank Mariëlle Stoelinga for helpful comments on an earlier version of this paper.

References

[AL06] H. Aljazzar and S. Leue. Extended directed search for probabilistic timed reachability. In *FORMATS '06*, pages 33–51, 2006.

[And06] M.E. Andrés. Derivation of counterexamples for quantitative model checking. Master's thesis, National University of Córdoba, 2006.

[BA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *FSTTCS '95*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.

[Bel57] R.E. Bellman. A Markovian decision process. *J. Math. Mech.*, 6:679–684, 1957.

[BP05] M. Bhargava and C. Palamidessi. Probabilistic anonymity. *CONCUR '05*, volume 3653 of *LNCS*, pages 171–185. Springer, 2005.

[CGJ+00] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement. In *CAV '00*, pages 154–169, 2000.

[Cha88] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[CL05] J. Camenisch and A. Lysyanskaya. A formal treatment of onion routing. *CRYPTO '05*, volume 3621 of *LNCS*, pages 169–187. Springer, 2005.

[DY83] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In *AUSCRYPT '92*, volume 718 of *LNCS*, pages 244–251. Springer, 1992.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.

[HJ89] H. Hansson and B. Jonsson. A framework for reasoning about time and reliability. In *Proceedings of Real Time Systems Symposium*, pages 102–111. IEEE, 1989.

[HK07] T. Han and J.-P. Katoen. Counterexamples in probabilistic model checking. In *TACAS '07*, volume 4424 of *LNCS*, pages 60–75. Springer, 2007.

[LBB+01] K.G. Larsen, G. Behrmann, E. Brinksma, A. Fehnker, T.S. Hune, P. Pettersson, and J. Romijn. As cheap as possible: Efficient cost-optimal reachability for priced timed automata. In *CAV'01*, 2001.

[PZ93] A. Pnueli and L.D. Zuck. Probabilistic verification. *Information and Computation*, 103(1):1–29, 1993.

[SV04] A. Sokolova and E.P. de Vink. Probabilistic automata: System types, parallel composition and comparison. *Validation of Stochastic Systems: A Guide to Current Research*, volume 2925 of *LNCS*, pages 1–43. Springer, 2004.

[SL95] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.

[Var85] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state systems. In *Proc. 26th IEEE Symp. Found. Comp. Sci.*, pages 327–338, 1985.