Synthesis of Safety Specifications for Probabilistic Systems

Gaspard Ohlmann*
Independent
Mulhouse, France
gaspard.ohlmann@outlook.com

Edwin Hamel-de le Court*
Imperial College
London, United Kingdom
e.hamel-de-le-court@imperial.ac.uk

Francesco Belardinelli
Imperial College
London, United Kingdom
francesco.belardinelli@imperial.ac.uk

ABSTRACT

Ensuring that agents satisfy safety specifications can be crucial in safety-critical environments. While methods exist for controller synthesis with safe temporal specifications, most existing methods restrict safe temporal specifications to probabilistic-avoidance constraints. Formal methods typically offer more expressive ways to express safety in probabilistic systems, such as Probabilistic Computation Tree Logic (PCTL) formulas. Thus, in this paper, we develop a new approach that supports more general temporal properties expressed in PCTL. Our contribution is twofold. First, we develop a theoretical framework for the Synthesis of safe-PCTL specifications. We show how the reducing global specification satisfaction to local constraints, and define CPCTL, a fragment of safe-PCTL. We demonstrate how the expressiveness of CPCTL makes it a relevant fragment for the Synthesis Problem. Second, we leverage these results and propose a new Value Iteration-based algorithm to solve the synthesis problem for these more general temporal properties, and we prove the soundness and completeness of our method.

KEYWORDS

Controller Synthesis, PCTL, Markov decision process, Value Iteration

ACM Reference Format:
Gaspard Ohlmann*, Edwin Hamel-de le Court*, and Francesco Belardinelli. 2026. Synthesis of Safety Specifications for Probabilistic Systems. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/MGLX3286

∗ *These authors contributed equally to this work.*

1 INTRODUCTION

Synthesizing policies that provably satisfy rich temporal specifications, while optimizing reward, is a long-standing challenge at the interface of Formal Methods and Reinforcement Learning. In probabilistic settings, *Probabilistic Computation-Tree Temporal Logic* (PCTL) [2, 9] provides a natural language to express safety and performance requirements over Markov decision processes (MDPs). However, general PCTL synthesis is computationally hard and, under the assumption of history-dependent strategies, undecidable [6, 7]. Moreover, randomness and memory are necessary for PCTL synthesis, even for restricted fragments [1]. Therefore, further investigations are required to construct strategies that satisfy such specifications.

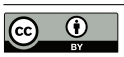
This paper develops a new framework for *Safe PCTL* (PCTL_{safe}) that we apply to the synthesis of specifications in *Continuing* PCTL (CPCTL), a fragment of PCTL_{safe} that we here introduce. Our starting point is a structural insight: the satisfaction of a broad class of PCTL safety properties can be enforced by local, per-transition inequalities in a suitably *augmented MDP*. We introduce two local conditions – *state compatibility* and *path compatibility* – that together imply global satisfaction of the original specification through a coherence theorem. Building on these foundations, we identify a syntactic fragment – CPCTL – for which these local constraints yield constructive algorithms, while still allowing nesting of probabilistic operators and thus providing a new computable class of safety specifications.

On the algorithmic side, we propose CPCTL-VI, a value-iteration type [8] algorithm that monotonically tightens inductive lower bounds on satisfaction probabilities. CPCTL-VI represents a constructive method for the synthesis problem of CPCTL, representing complex nested probabilistic and temporal behaviors.

Contributions. The key contributions of this paper can be summarized as follows:

(1) **An analysis of temporal and probabilistic specifications:** We introduce Continuing PCTL (CPCTL), a fragment of safe PCTL generalizing multi-objective avoidance specifications, allowing nesting of probabilistic operators. We show that there currently exists no decidability result for such specifications. Additionally, we establish structural results for CPCTL such as weak reduction to literal projections.
(2) **A key theoretical result:** A new *augmented MDP* construction for Safe PCTL that encodes global satisfaction as local linear inequalities. We define two conditions, the *state compatibility* and *path compatibility*, and show that their satisfaction guarantees the satisfaction of the corresponding formula.
(3) **An Algorithm for the CPCTL Synthesis Problem:** CPCTL-VI is a value-iteration algorithm that computes lower bounds on satisfaction probabilities and certifies realizability. Moreover, the algorithm is optimal under a generalized version of Slater's assumption.

The paper is organized as follows. In Sec. 2 we review the results on the synthesis problem for PCTL and some of its significant fragments. In Sec. 3 we provide background on MDPs, RL, and PCTL. In Sec. 4 we define the Continuing PCTLfragment and analyse its structural properties, including expressivity In Sec. 5 we introduce the augmented MDP construction, as well as the



Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/MGLX3286

state/path-compatibility conditions, and prove the coherence theorem. In Sec. 6 we present our value iteration algorithm for CPCTL synthesis – including soundness and optimality results – which is then evaluated empirically in Sec. 7. Finally, we conclude in Sec. 8 pointing to future work.

2 RELATED WORK

We present the inclusion relations between PCTL and its significant fragments – including CPCTL– in Figure 1.

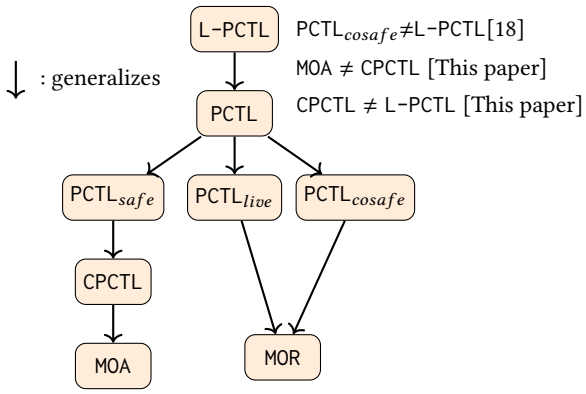


Figure 1: Expressivity of CPCTL in the PCTL hierarchy

Undecidability of PCTL *Synthesis for HR policies.* For history-dependent (perfect-recall) stochastic controllers (**HR**), controller synthesis is undecidable for logics slightly more expressive than PCTL, such as L-PCTL, which allows linear combinations of probabilities [5], or PCTL$_{G\&L}$, which allows finite window operators such as $\mathbf{U}^{\leq n}$ or $\mathbf{W}^{\leq n}$ [6, 7]. While the proof of undecidability, such as reduction from Minsky machines, cannot be directly applied to PCTL, it suggests that the PCTL synthesis problem is either undecidable or highly complex. Moreover, memory and randomization are necessary for PCTL synthesis, even when no nesting is allowed [1]. For the co-safe fragment of PCTL– PCTL$_{cosafe}$– decidability can be recovered, although with very high complexity (3EXPTIME [18]).

Complexity for MR and MD policies. For memoryless randomized strategies (**MR**), [15] provides a construction that yields a **PSPACE** upper bound for the general PCTL synthesis problem. The synthesis problem of a single formula in PCTL$_{safe}$ for **MR** policies is **NP**-hard by reduction from graph three-coloring. In fact, in the easier case of memoryless deterministic strategies (**MD**), the synthesis problem for both PCTL and L-PCTL objectives is already **NP**-complete [6]. Furthermore, the PCTL synthesis problem for **MD** policies is NP-complete [1] and is in EXPTIME for **MR** policies [16].

Multi-Objective Reachability and Avoidance. Multi-objective reachability (MOR) and avoidance (MOA) queries have been shown to be solvable through Linear Programming [10], and thus can be solved in polynomial time. A value iteration scheme for multiple objectives based on computing Pareto frontiers appears in [19], and a practical algorithm that generates successive approximations of the Pareto frontier for multi-objective reachability and avoidance appears in [11]. These results are summarized in Table 2.

Summary. We summarize these results in Table 1. The multi-objective avoidance (resp. reach) fragment MOA (resp. MOR) is composed of all the formulas of the form $\bigwedge_{j=1}^{n} \mathbb{P}_{\geq p_j}(\mathbf{G}\neg a_j)$ (respectively $\bigwedge_{j=1}^{n} \mathbb{P}_{\geq p_j}(\mathbf{F}a_j)$) for $a_j \in AP$, where **G** is the globally operator and **F** is the reach operator. These two fragments have been studied individually, as they are of theoretical and practical relevance. The definition of the other fragments along with a study of their expressivity and/or complexity can be found in [5] for L–PCTL, [3, 6] for PCTL$_{G\&L}$, [14, 18] for PCTL$_{cosafe}$and PCTL$_{safe}$, and [10] for MOA.

Class	Strategies	Complexity
L–PCTL	**HR**	Σ_1^1 hard [5]
	MD	**NP**-complete [1]
PCTL	**HR**	Unknown
	MD	**NP**-complete [1]
PCTL$_{G\&L}$	**HR**	Σ_1^1 hard [6]
PCTL$_{safe}$	**HR**	Unknown
	MR	**NP**-hard [This paper]
PCTL$_{cosafe}$	**HR**	Δ_1^0 [18]
MOA	**HR**	Linear Pro. [10]

Table 1: Complexity of PCTL Synthesis for PCTL fragments under different strategy assumptions.

Continuing PCTL.. We identify the fragment CPCTL of PCTL$_{safe}$ that strictly extends multi-objective avoidance specifications, and we provide a synthesis algorithm for this logic that solves the decidability problem and outputs a safe policy whenever one exists under a generalized Slater's Assumption. This assumption intuitively means that the formula can be satisfied when $\geq$ are replaced by strict $>$. Precise definitions are provided in Sec. 3. In addition, CPCTL contains formulas that do not belong to any of the known computable classes. Common classes for which an algorithm has been described are summarized in Table 2. Although the decidability of the synthesis problem for the whole PCTL$_{safe}$remains open, our contribution is a step towards understanding the subtle boundaries of decidability in PCTL synthesis. As no existing algorithm solves the synthesis problem for CPCTL or any larger class of properties to the best of our knowledge, performance comparisons are not available.

Class	Safe & Complete Algorithm
Avoid	Yes [10, 12]
MOA	Yes [10]
CPCTL	Yes [This paper]
PCTL$_{safe}$	No
PCTL$_{cosafe}$	Yes [18]
MOR	Yes [10]

Table 2: Currently computable classes, HR strategies.

3 PRELIMINARIES

In this section we provide preliminaries on Markov decision chains and processes, and Probabilistic CTL. We refer the interested reader to [2] for an in-depth presentation.

3.1 Stochastic Models

Markov Decision Processes. A (labeled) *Markov Decision Process* [17] is a tuple $\mathcal{M} = \langle S, A, P, s_{init}, AP, L, R\rangle$, where S is a set of *states*; A is a mapping that associates every state $s \in S$ to a nonempty finite set $A(s)$ of *actions*; $P : S \times A \to \mathcal{D}(S)$ is a *transition probability function* that maps every state-action pair (s, a) to a probability measure $P(s, a)$ over S; $s_{init} \in S$ is the *initial state*[1]; AP is a set of *atomic propositions* (or atoms); $L : S \mapsto 2^{AP}$ is a *labeling function*; and $R : S \mapsto \mathbb{R}$ is the *reward function*.

In the following, all MDPs are labeled so we will simply call them Markov Decision Processes (MDP). For the sake of simplicity, we will write $P(s'|s, a)$ instead of $P(s, a)(s')$. An MDP is finite if the sets of states and actions are finite.

Paths. A finite (resp. infinite) *path* (or history) in an MDP $\mathcal{M}$ is a finite (resp. infinite) word $\zeta = s_0 a_0 \cdots s_{n-1} a_{n-1} s_n$ (resp. $\cdots$) such that for any $i \leq n$ (resp. for any $i \in \mathbb{N}$), s_i is a state of $\mathcal{M}$, a_i is an action in $A(s_i)$, and s_{i+1} is in the support of $P(s_i, a_i)$. In addition, paths $(\mathcal{M})$ denotes the set of *infinite* paths of $\mathcal{M}$. We denote as $\mathbb{P}_{\mathcal{M},\pi}$ the usual probability measure on infinite paths $s_0 s_1 \cdots \in S^\omega$ induced by a policy π on a $\mathcal{M}$ (c.f. [2] for full details). Finally, for a path $\xi = s_0 s_1 \cdots$ and integer $i \geq 0$, we let $\xi_i = \xi[i] = s_i$.

Policies. For any measurable space E [13], we denote by $\mathcal{D}(E)$ the set of probability measures over E. A *policy* (also called *controller* or *strategy*) π of $\mathcal{M}$ is a mapping that associates any finite path ξ of $\mathcal{M}$ to a probability measure over $A(\text{last}\,(\zeta))$. It is memoryless if $\pi(\zeta)$ only depends on last (ζ), in which case we denote $\pi(\zeta) = \pi(s)$, where $s = \text{last}\,(\zeta)$. It is deterministic if for any finite path ζ, $\pi(\zeta)$ is a Dirac measure. We let **HR** denote the *history-dependent and randomized* policies.

Markov Chain Induced by a Policy on a MDP. For any policy π of $\mathcal{M}$, we let $\mathcal{M}_\pi = (S_\pi, P_\pi, s_{init}, AP, L)$ denote the (labelled) *Markov chain* induced by π in $\mathcal{M}$ such that S_π is the set of finite histories of $\mathcal{M}$, AP is the same as in $\mathcal{M}$, $L_\pi(\xi) = L(\text{last}\,(\xi))$, and

$$P_\pi(s|\xi) = \sum_{a \in A(s)} \pi(a|\xi) P(s|\text{last}\,(\xi), a).$$

When the policy is memoryless, the states of $\mathcal{M}_\pi$ correspond to the states of $\mathcal{M}$ and will be denoted the same. When the policy is history-dependent, the states of $\mathcal{M}_\pi$ correspond to all possible histories. For more details on MDPs and induced Markov chains, see [2, 4].

3.2 Probabilistic and Temporal Specifications

In this section, we define the probabilistic temporal Logic PCTL[9] and its safe fragment PCTL_{safe}[14]. These two logics distinguish between two kind of formulas: path and the state formulas, and allow for probabilistic nesting of temporal specifications.

3.2.1 Probabilistic Temporal Logic (PCTL). Let AP be a finite set of atomic propositions and $q \in [0, 1]$. A formula of PCTL is generated by the nonterminal Φ in the following grammar:

$$\Phi \quad ::= \quad a \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \neg\Phi \mid \mathbb{P}_{\geq q}(\varphi), \quad a \in AP \cup \{\bot, \top\},$$
$$\varphi \quad ::= \quad \mathbf{X}\,\Phi_1 \mid \Phi_1\,\mathbf{W}\,\Phi_2 \mid \Phi_1\,\mathbf{U}\,\Phi_2.$$

We call the formulas generated by Φ the *state formulas*, and the formulas generated by φ the *path formulas*. The satisfaction relation $\models$ is defined on both sets by induction as follows.

Definition 1 (Semantics). *Given a Markv chain $\mathcal{M}$, state s, and path ξ, the satisfaction relation $\models$ is defined inductively as follows:*

***State formulas**:*

$(\mathcal{M}, s) \models a$ *iff* $a \in L(s)$
$(\mathcal{M}, s) \models \neg\Phi_1$ *iff* $(\mathcal{M}, s) \not\models \Phi_1$
$(\mathcal{M}, s) \models \Phi_1 \wedge \Phi_2$ *iff* $(\mathcal{M}, s) \models \Phi_1$ *and* $(\mathcal{M}, s) \models \Phi_2$
$(\mathcal{M}, s) \models \Phi_1 \vee \Phi_2$ *iff* $(\mathcal{M}, s) \models \Phi_1$ *or* $(\mathcal{M}, s) \models \Phi_2$
$(\mathcal{M}, s) \models \mathbb{P}_{\geq q}(\varphi)$ *iff* $\mathbb{P}_{\mathcal{M}^s}\big(\{\xi \in Paths(s),\ (\mathcal{M}, \xi) \models \varphi\}\big) \geq q$.

***Path formulas**:*

$(\mathcal{M}, \xi) \models \mathbf{X}\Phi_1$ *iff* $\mathcal{M}^{\xi[1]} \models \Phi_1$
$(\mathcal{M}, \xi) \models \Phi_1\,\mathbf{U}\,\Phi_2$ *iff* $\exists\, j \in \mathbb{N}$, $(\mathcal{M}, \xi[j]) \models \Phi_2$ *and* $\forall\, i \leq j$, $(\mathcal{M}, \xi[i]) \models \Phi_1$,
$(\mathcal{M}, \xi) \models \Phi_1\,\mathbf{W}\,\Phi_2$ *iff* $(\mathcal{M}, \xi) \models (\Phi_1\,\mathbf{U}\,\Phi_2) \vee (\mathbf{G}\,\Phi_1)$,
where $(\mathcal{M}, \xi) \models \mathbf{G}\,\Phi_1$ *iff* $\forall i \in \mathbb{N}$, $\xi[j] \models \Phi_1$.

Satisfaction, Markov Chain. $(\mathcal{M}, s) \models \Phi$ is alternatively denoted $\mathcal{M}^s \models \Phi$. When clear from the context, we write for state formulas (resp. path formulas) $s \models \Phi ::= (\mathcal{M}, s) \models \Phi$ (resp. $\xi \models \phi ::= (\mathcal{M}, \xi) \models \phi$). Moreover, for a Markov chain $\mathcal{M}$, we define its *semantics* of a state formula Φ as $[\![\Phi]\!]_\mathcal{M} = \{s,\ (\mathcal{M}, s) \models \Phi\}$ and its *semantics along a path* ξ as $[\![\Phi]\!]_{\xi,\mathcal{M}} = \{i \in \mathbb{N},\ (\mathcal{M}, \xi[i]) \models \Phi\}$. For readability, we may write $[\![\Phi]\!] ::= [\![\Phi]\!]_\mathcal{M}$ and $[\![\Phi]\!]_\xi ::= [\![\Phi]\!]_{\xi,\mathcal{M}}$.

Satisfaction, Probabilities. For a Markov Chain $\mathcal{M}$ and a path formula ϕ, we write $\mathbb{P}(\phi|s, \mathcal{M})$ to denote the probability of the set of paths starting from s and satisfying ϕ in $\mathcal{M}$.

Satisfaction, MDP. When the Markov chain is of the form $\mathcal{M}_\pi$ and is induced by policy π on the MDP $\mathcal{M}$, the states of $\mathcal{M}_\pi$ are histories $\rho \in \mathcal{H}(\mathcal{S})$. Instead of $(\mathcal{M}_\pi, \xi) \models \phi$, where ξ is a path of histories $\rho_i \in \mathcal{H}(\mathcal{M})$, for $i \in \mathbb{N}$, we may write $\xi \models_\pi \phi$ or simply $\xi \models \phi$. Similarly, instead of $\mathcal{M}^\rho_\pi \models \Phi$, we may write $\rho \models \Phi$. When the policy π is memoryless, we write $s \models \Phi$ for $\mathcal{M}^\rho_\pi \models \Phi$, when $s = \text{last}\,(\rho)$.

3.2.2 Safe PCTL.. The safe fragment of PCTL, denote PCTL_{safe}, is the subset of PCTLformulas, whose syntax is given by

$$\Phi \quad ::= \quad a \mid \neg a \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \mathbb{P}_{\geq q}(\varphi), \quad a \in AP \cup \{\bot, \top\},$$
$$\varphi \quad ::= \quad \mathbf{X}\,\Phi_1 \mid \Phi_1\,\mathbf{W}\,\Phi_2.$$

The semantic of the relevant operators remain the same. As opposed to PCTL, the negation is only allowed on atomic propositions, and the until operator $\mathbf{U}$ is not allowed. Note that the operator $\mathbf{G}$ is in the safety fragment as $\mathbf{G}\,\Phi = \Phi\,\mathbf{W}\,\bot$.

The safety fragment of PCTL extends the multi-objective avoidance case, and corresponds to formulas whose violation can be witnessed in finite time, but not their satisfaction. For example, the PCTL_{safe}formula $\mathbb{P}_{\geq 1/2}(\mathbf{G}\neg a)$ – stating that at least 50% of the paths should always avoid a – satisfies this property, as any path can always *a priori* reach a later. For more in-depth intuition and analysis of the safety fragment, we refer the reader to [14].

[1] This can be assumed wlog compared to a model with an *initial probability distribution* since it is always possible to add a new initial state to such a model with an action from this initial state whose associated probability distribution is the aforementioned initial probability distribution.

When a policy satisfies a safety formula, we will say that the policy is *safe* (for this formula). Finally, the term *safety specification* is synonymous of *safety formula.*

4 CONTINUING PCTL

4.1 Definition and Problem Statement

Continuing PCTL (CPCTL) is the fragment of PCTLbuilt on *state* formulas Φ and *path* formulas φ, according to the following BNF:

$$\Phi ::= a \mid \neg a \mid \Phi \wedge \Phi, \mid \mathbb{P}_{\geq q}(\varphi), \quad a \in AP,$$
$$\varphi ::= \Phi_1 \, \mathbf{W} \, (\Phi_1 \wedge \Phi_2).$$

Continuing PCTL , as opposed to PCTL_{safe}, does not allow disjunctions or the next operator **X**. Moreover, the syntax of the weak until **W** is restricted such that condition Φ_1 appears as a conjunct in goal $\Phi_1 \wedge \Phi_2$.

Hereafter let $\mathcal{M}$ be an MDP and Φ a CPCTL state formula, and denote the set of state subformulas $\mathcal{SF}(\Phi) = \{\Phi_1, \ldots, \Phi_{sf(\Phi)}\}$ and path formulas $\mathcal{PF}(\Phi) = \{\phi_1, \ldots, \phi_{pf(\Phi)}\}$, with the added convention that for all $j \leq pf(\Phi)$, $\Phi_j = \mathbb{P}_{\geq p_j}(\phi_j)$.

The Synthesis Problem. In this contribution we design an algorithm for the following problem: Let $\mathcal{M}$ be an MDP and $\Phi \in$ CPCTL. Then, find a **HR** policy π such that $\mathcal{M}_\pi \models \Phi$.

We emphasize first that we adopt the standard semantics of PCTL operators for the synthesis problem: a single policy is used throughout the formula evaluation. This differs from the semantics often used in the model-checking problem, where each probabilistic operator in the formula might be associated with a different strategy in principle. On the other hand, in the synthesis problem as stated above, a unique strategy is used to evaluate all the probabilistic operators.

Slater's Generalized Assumption. We provide an analogue of Slater's assumption in the case where nesting are considered, that we call Slater's generalized assumption. Intuitively, a Markov Chain $\mathcal{M}$ and a formula Φ satisfy Slater's generalized assumption if $\mathcal{M} \models \Psi$ for any Ψ, where Ψ is obtained by replacing all p_i in the formula Φ by $q_i > p_i$.

Definition 2. *(Slater's generalized assumption) We consider $\mathcal{M}$ a MDP and Φ a* CPCTL *formula. For $\boldsymbol{d} = (d_1, \ldots, d_{pf(\Phi)})$, we define $\mathsf{T}(\Phi, \boldsymbol{d}) \in$ CPTCL by induction as $\mathsf{T}(\Phi, \boldsymbol{d}) = \mathsf{T}(\Phi_{j_1}, \boldsymbol{d}) \wedge \mathsf{T}(\Phi_{j_2}, \boldsymbol{d})$ for $\Phi = \Phi_{j_1} \wedge \Phi_{j_2}$, $\mathbb{P}_{\geq d_j}[\mathsf{T}(\Phi_{j_1}, \boldsymbol{d}) \mathbf{W} \mathsf{T}(\Phi_{j_1} \wedge \Phi_{j_2}, \boldsymbol{d})]$ for $\Phi = \Phi_{j_1} \mathbf{W} \Phi_{j_2}$, a for $\Phi = a \in AP$ and $\neg a$ for $\Phi = \neg a$, $a \in AP$.*

We say that $\mathcal{M}, \Phi$ satisfy the generalized Slater's Assumption (gSA) if there exists $\boldsymbol{p} = (p'_1, \ldots, p'_{N_1})$ satisfying $p'_i > p_i$ for all i, such that there exists π satisfying $\mathcal{M}_\pi \models \mathsf{T}(\Phi, \boldsymbol{p})$.

4.2 Structural CPCTL Properties

The results presented hereafter are essential for the initialization of the value iteration algorithm to solve the synthesis problem, introduced in Sec. 6. They describe the set $\mathbb{P}_{=1}(\phi)$, for path formula ϕ, which corresponds to the starting point of the algorithm.

The intuition for the name "Continuing PCTL" comes from the following lemma. It states that for $\Phi \in$CPCTL, the satisfaction of Φ can be postponed, as long as we stay on the literal projection of Φ, a boolean logic formula that we introduce now, and Φ is satisfied later on.

Definition 3 (Literal projection). *We define the literal projection $\mathsf{L}(\phi)$ of a* CPCTL *formula ϕ by induction on ϕ.*

- *For any literal a, $\mathsf{L}(a) = a$,*
- *for any* CPCTL *state formulas Φ_1 and Φ_2, we have $\mathsf{L}(\Phi_1 \wedge \Phi_2) = \mathsf{L}(\Phi_1) \wedge \mathsf{L}(\Phi_2)$ and*

$$\mathsf{L}\left(\mathbb{P}_{\geq p}\left[\Phi_1 \mathbf{W} (\Phi_1 \wedge \Phi_2)\right]\right) = \begin{cases} \top & \textbf{\textit{if } } p = 0 \\ \mathsf{L}(\Phi_1) & \textbf{\textit{otherwise.}} \end{cases}$$

The Literal Projection is the strongest necessary condition in boolean logic that a state has to satisfy to satisfy a state CPCTL formula. For example, for the following formulas

$$\Phi_1 = \mathbb{P}_{\geq 0.7}\left((a \wedge \neg b) \mathbf{W} ((a \wedge \neg b) \mathbf{W} c) \Rightarrow \mathsf{L}(\Phi_1)\right) = a \wedge \neg b,$$
$$\Phi_2 = \mathbb{P}_{\geq 0.5}(\phi_2) \wedge \neg b \Rightarrow \mathsf{L}(\Phi_2) = a \wedge \neg b.$$

Define ϕ_1, ϕ_2 such that $\Phi_1 = \mathbb{P}_{\geq p}(\phi_1)$ and $\Phi_2 = \mathbb{P}_{\geq 0.5}(\phi_2)$. For any state s in a Markov Chain $\mathcal{M}$, we have $s \models \Phi_1 \Rightarrow a \wedge \neg b$, otherwise we would have $\mathbb{P}(\phi_1 | s, \mathcal{M}) = 0$ since for every path ξ, we would have $\xi \not\models \phi_1$. Similarly, $s \models \Phi_2 \Rightarrow s \models \neg b$, and $s \models \mathbb{P}_{\geq 0.5}(\phi_2)$. Conversely, a quick induction shows that satisfying $\mathbb{P}_{\geq 1}(\mathbf{G}\mathsf{L}(\Phi))$ implies the satisfaction of Φ. For example, $\mathbb{P}_{=1}(\mathbf{G}(a \wedge \neg b))$ implies Φ_1. The following lemma precisely formalizes this intuition.

Lemma 1. *For every MDP $\mathcal{M}$, path formula $\phi = \Phi_1 \mathbf{W} (\Phi_1 \wedge \Phi_2)$, and policy π, we have $\mathcal{M}_\pi \models \mathbb{P}_{\geq 1}[\phi]$ iff $\mathcal{M}_\pi \models \mathbb{P}_{\geq 1}[\mathsf{L}(\Phi_1) \mathbf{W} (\Phi_1 \wedge \Phi_2)]$.*

Corollary 1. *For every MDP $\mathcal{M}$, $\Phi \in$CPCTL and policy π, we have $\mathcal{M}_\pi \models \mathbb{P}_{\geq 1}(\mathbf{G}(\mathsf{L}(\Phi)) \Rightarrow \mathcal{M}_\pi \models \Phi$.*

The intuition of the results **cannot** be extended further.

- (Equivalence) For $\Phi = \mathbb{P}_{\geq p}[\Psi_1 \mathbf{W} \Psi_2]$, one cannot write $\Phi = \mathbb{P}_{\geq 1}(\mathbf{G}\mathsf{L}(\Phi))$ even assuming that $\Psi_2 = \bot$ so that Φ never "stops" and always continues.
- (Paths) The result can not be extended to paths. For instance, if we consider the Markov Chain $\mathcal{M}$ with $\mathcal{S} = \{s_1, s_2\}$, $L(s_1) = \emptyset$, $L(s_2) = \{a\}$, and $\mathcal{P}(s_1|s_1) = \mathcal{P}(s_2|s_1) = 1/2$, $\mathcal{P}(s_2|s_2) = 1$ and the path formula $\phi = (\mathbb{P}_{\geq 2/3}\mathbf{G}a)\mathbf{W}\bot$, then the path $\xi = s_1, s_1, s_1, \ldots$ satisfies $\left(\mathsf{L}\left(\mathbb{P}_{\geq 1/2}\mathbf{G}a\right)\right)\mathbf{W}\bot$ but does not satisfy ϕ.
- (PCTL_{safe}) This key structural property is not true for general safe formulas. For example, no boolean logic formula on a state s can be deduced provided that $s \models \mathbb{P}_{\geq 1}(\mathbf{X}a)$.

4.3 Expressivity of CPCTL

The restrictions imposed on CPCTL formulas do not imply a one dimensional monotonicity, and maximizing several nested properties does not reduce to maximizing the last one, as shown in example 1.

Example 1. ***(No collapse of nested objectives)***
We consider the formula

$$\Phi = \mathbb{P}_{\geq p_1}[\mathbf{G}\Phi_a], \quad \Phi_a = \mathbb{P}_{\geq p_1}[\mathbf{G}\neg a], \quad p_1 = \frac{7}{12}.$$

The path formula $\mathbf{G}[\Phi_a]$ means that we must stay on states such that the probability of reaching a from those states is lower than p_1. We look for π that maximizes the probability of those paths. One intuition may be that there is never a reason to visit a more often. However, the safest policy that minimizes the probability of reaching

a from any state does not maximize Φ. *We consider the* $\mathcal{M}$ *presented in Figure 2. The actions are represented by arrows labelled by the associated probabilities. The states without an outgoing arrow have only one action looping on themselves with probability* 1. *Finally,* $[\![a]\!] = \{s_5, s_7\}$.

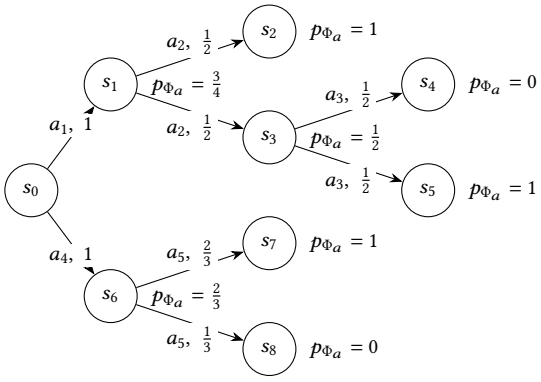


Figure 2: The Markov Decision Process $\mathcal{M}_1$

The policy that maximizes the probability of avoiding a π_1 *chooses* a_1 *with probability* 1. *In this case,* $\mathbb{P}(\mathbf{G}\neg a|\pi, s_0) = 3/4$. *Now, the states that satisfy* $\Phi_a = \mathbb{P}_{\geq 7/12}[\mathbf{G}\neg a]$ *are* $[\![\Phi_a]\!] = \{s_0, s_1, s_2, s_5, s_6\}$. *Hence, the probability of never reaching a state in* $\mathcal{S} \setminus [\![\Phi_a]\!]$ *is equal to* $1/2$. *If we consider* π_2 *however, the policy taking* a_1 *with probability* 0 *and* a_4 *with probability* 1, *since* s_6 *and* s_7 *are in* $[\![\Phi_a]\!]$ *and* $s_8 \notin [\![\Phi_a]\!]$, *we obtain* $\mathbb{P}(\Phi|\pi_1, s_0) = \frac{1}{2} < \frac{2}{3} = \mathbb{P}(\Phi|\pi_2, s_0)$. *Thus, in this case, we have* $s_0, \pi_1 \not\models \Phi$ *while* $s_0, \pi_2 \models \Phi$.

4.4 A New Computable Class

The following theorem shows that CPCTL specifications cannot be reduced to flat PCTL formulas and form a new class of computable specifications.

Theorem 1. *There exists formulas* $\Phi \in$ CPCTL *such that for no flat formula (nesting depth one)* $\Psi \in$ PCTL, *we have that for any Markov chain* $\mathcal{M}$, $\mathcal{M} \models \Phi \Leftrightarrow \mathcal{M} \models \Psi$.

Equivalently, for such formulas, there exists no flat formula $\Psi \in$ PCTL such that for any MDP $\mathcal{M}$ and any policy π, we have $\mathcal{M}_\pi \models \Phi \Leftrightarrow \mathcal{M}_\pi \models \Psi$. In particular, currently there exists no algorithm for the synthesis of CPCTL specifications.

Proof. We provide an example of a CPCTL formula that is not equivalent to any flat formula. We consider the nested formula $\Phi = \mathbb{P}_{\geq 1}\left(c\mathbf{W}\mathbb{P}_{\geq 1/2}(c\mathbf{W}(c \wedge a))\right) \in$ CPCTL, and assume the existence of a flat formula Ψ of the form $\Psi = \mathcal{B}(\Psi_1, \dots, \Psi_l), \quad \Psi_k = \mathbb{P}_{\geq p_k}(\psi_k)$ where $\mathcal{B}$ is an operator that involves only a finite number of boolean operations and atomic propositions, and ψ_k are formulas of the form $\psi_k = \Psi_{k,1}\mathbf{O}_k\Psi_{k,2}$ where $\mathbf{O}_k \in \{\mathbf{U}, \mathbf{W}\}$ and $\Psi_{k,1}, \Psi_{k,2}$ are boolean formulas. We introduce the Markov Chain $\mathcal{M}_{\alpha,\varepsilon}$ as defined in Figure 3.

For any $\varepsilon > 0$, we have $\mathbb{P}(\phi|\mathcal{M}_{\alpha,\varepsilon}) \leq 1 - \varepsilon < 1$, so $\mathcal{M} \not\models \Phi$. For $\varepsilon = 0$, we denote ξ the path obtained by going to the left once and to the right once. ξ has label $\{c\}\{c\}\{b\}^*$ and does not satisfy the path formula $c\mathbf{W}[c\mathbf{W}(c \wedge a)]_{\geq 1/2}$ because $c \notin \{b\}$, except if there

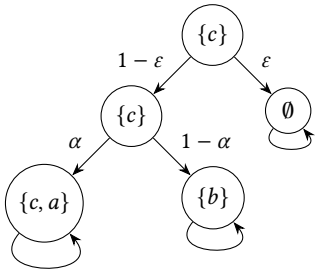


Figure 3: The Markov Chain $\mathcal{M}_{\alpha,\varepsilon}$.

exists i such that $\xi_i \models [c\mathbf{W}(c \wedge a)]_{\geq 1/2}$. $i < 2$ which means $i = 0$ or $i = 1$. We have $\mathbb{P}(c\mathbf{W}(c \wedge a)|\xi_i) = \alpha$ for $i = 1, 2$, so $\xi \models \phi$ if and only if $\alpha \geq 1/2$.

Hence, $\mathcal{E}_\Phi = \{(\alpha, \varepsilon),\ \mathcal{M}_{\alpha,\varepsilon} \models \Phi\} = \left[\frac{1}{2}, 1\right] \times \{0\}$.

The following conditions are incompatible for a flat formula Ψ:

$$(i)\ \forall \varepsilon > 0,\ \forall \alpha \in [0,1],\ (\alpha, \varepsilon) \notin \mathcal{E}_\Psi,\ (ii)\ \left(\frac{1}{2}, 0\right) \in \mathcal{E}_\Psi \wedge \left(\frac{1}{3}, 0\right) \notin \mathcal{E}_\Psi.$$

To see it, note that the evaluation of path formulas ψ_k on a path $\xi = (\xi_0\xi_1\dots)$ only depend on the sequence of labels $L(\xi_0)L(\xi_1)\dots$. With $\xi_{ll}, \xi_{lr}, \xi_r$, the paths respectively going to the left then left, to the left then right, and to the right, we have with $\mathbf{B}(True) = 1$ and $\mathbf{B}(False) = 0$,

$$\begin{aligned}\mathbb{P}(\psi_j|\mathcal{M}_{\alpha,\varepsilon}) =& (1-\varepsilon)\alpha\mathbf{B}(\xi_{ll} \models \psi_j) \\ &+ (1-\varepsilon)(1-\alpha)\mathbf{B}(\xi_{lr} \models \psi_j) + \varepsilon\mathbf{B}(\xi_r \models \psi_j).\end{aligned}$$

This condition restricts expressivity and prevents the whole formula Ψ from satisfying the previous condition.

□

5 FROM SAFE-PCTL SATISFACTION TO LOCAL CONSTRAINTS

In this section we construct, for a safe-PCTL formula Φ and an MDP $\mathcal{M}$, an augmented MDP $\mathcal{M} \times [\Phi]$, such that satisfying Φ in $\mathcal{M}$ is equivalent to satisfying local constraints provided in Def. 7 in $\mathcal{M} \times [\Phi]$.

Definition 4 (Augmented MDP). *We define* $\mathcal{M} \times [\Phi]$ – the MDP augmented by formula Φ – *as follows:*

- *Set of States: Let* $\hat{\mathcal{S}}_{\geq 0}$ *be the set of all triples* (s, μ, ν) *in* $\mathcal{S} \times [0,1]^{sf(\Phi)} \times \{0,1\}^{pf(\Phi)}$. *Each (augmented) state* $\hat{s} \in \hat{\mathcal{S}}_{\geq 0}$ *is of the form* (s, μ, ν), *with*

$$\nu = (\nu_1, \dots, \nu_{sf(\Phi)}),\ \mu = (\mu_1, \dots, \mu_{pf(\Phi)}),$$

  where $\nu_i \in [0,1]$ *(resp.* $\mu_i \in \{0,1\}$*) are the counters of the path subformulas* $\phi_i \in \mathcal{PF}(\Phi)$ *of* Φ, *(resp. the valuations of the state subformulas* $\psi_i \in \mathcal{SF}(\Phi)$ *of* Φ*). Finally, we define* $\hat{\mathcal{S}} = \hat{\mathcal{S}}_{\geq 0} \cup \{\hat{s}_{-1}\}$, *with initial state* $\hat{s}_{-1}$.
- *Set of Actions: Let* $\hat{s} = (s, \mu, \nu) \in \hat{S}$, $\mathcal{A}(s)$ *the set of actions available at* s *in* $\mathcal{M}$, *and* $\{s_1, \dots, s_m\}$ *the successors of* s *in* $\mathcal{M}$. *Then, for any* $\hat{s} \in \hat{\mathcal{S}} \setminus \{\hat{s}_{-1}\}$ *we define the (augmented) set of actions* $\hat{\mathcal{A}}(\hat{s})$ *as the set of* $(a, \nu^1, \dots, \nu^m, \mu^1, \dots, \mu^m)$ *such that* $a \in \mathcal{A}(s)$, *each* ν^i *(resp.* μ^i*) is a set of valuations (resp. counters) for the successor* s_i, *and* $(s_i, \nu^i, \mu^i) \in \hat{\mathcal{S}}$.

Initial State: *The set* $\hat{\mathcal{A}}(\hat{s}_{-1})$ *is the set of all* (ν, μ) *such that* $(s_0, \nu, \mu) \in \hat{\mathcal{S}}$, *where* s_0 *is the initial state of* $\mathcal{M}$.

- *Set of Transitions: For each* $\hat{a} = (a, \mu, \nu) \in \hat{\mathcal{A}}(\hat{s})$ *with* $\mu = (\mu_1, \dots, \mu_{st(\Phi)})$, $\nu = (\nu_1, \dots, \nu_{pf(\Phi)})$, *for* $\hat{s}' = (s_i, \nu', \mu')$

$$P(\hat{s}'|\hat{a}) = P(s_i|a), \text{ if } \mu' = \mu \text{ and } \nu' = \nu;$$
$$P(\hat{s}'|\hat{a}) = 0, \text{ otherwise}$$

That is, action $\hat{a}$ *performs similarly to action* a *in* $\mathcal{M}$, *but the agent can choose the valuations and counters for the successors and ends up in the augmented state* (s_i, ν_i, μ_i) *rather than* s_i.
Initial State: *For any* $\hat{a} = (\nu, \mu)$, *for any* $\hat{s}' = (s', \nu', \mu') \in \hat{\mathcal{S}}$, $P(\hat{s}' \mid \hat{a}, \hat{s}_{-1}) = 1$ *if* $s' = s_0$, $\nu' = \nu$ *and* $\mu' = \mu$, *and* $P(\hat{s}' \mid \hat{a}, \hat{s}_{-1}) = 0$ *otherwise.*

The augmented MDP $\mathcal{M} \times [\Phi]$ is infinite, as the set of states is the entire $\hat{\mathcal{S}} = \mathcal{S} \times \{0,1\}^{sf(\Phi)} \times [0,1]^{pf(\Phi)}$. However, in Section 6 we show how to navigate a finite portion of states and actions to obtain a policy that satisfies Φ in $\mathcal{M}$.

We next define calued policies. Intuitively, these corresponds to policies $\hat{\pi}$ on $\mathcal{M} \times [\Phi]$ that are deterministic on the counters and valuations of the augmented actions.

Definition 5. ***(Valued Policy)*** *A policy* $\hat{\pi}$ *is* valued *iff there exist functions* θ *and* Δ *s.t. for* $m(\hat{s}) = \#succ_{\mathcal{M}}(s)$ *and* $\hat{\rho}$ *a history in* $\mathcal{M} \times [\Phi]$,

$$\theta : \hat{\rho} \mapsto ((\mu^1, \nu^1), \dots, (\mu^m, \nu^m)),\ \Delta : \hat{\rho} \mapsto \delta \in \mathcal{D}(\mathcal{A}(s)),$$

For $last(\hat{\rho}) = (s, \mu, \nu)$. *Such that for any* $a \in \mathcal{A}(s)$, $\delta = \Delta(\hat{\rho})$,

$$\hat{\pi}(\hat{a}|\hat{\rho}) = \delta(a), \quad \hat{a} = (a, \theta(\hat{\rho})), \tag{1}$$

and $\hat{\pi}(\hat{a}|\hat{\rho}) = 0$ *for any other* a. *With an abuse of notation, we denote* $\theta(\hat{\rho})(s_i) ::= (\mu^i, \nu^i)$.

Additionally, we denote by $\hat{s}_{0,\hat{\pi}}$ the state reached from $\hat{s}_0$ after one step, and we write $\hat{s}_0$ when $\hat{\pi}$ is clear from the context. Since the policy is valued it chooses a unique set of valuations and counters for each state $s \in \mathcal{S}$, it is in particular deterministic at $\hat{s}_{-1}$ and $\hat{s}_0$ is well defined.

Finally, we denote by $\hat{S}_{\hat{\pi}}$ the set of states that are reachable from $\hat{s}_{0,\hat{\pi}}$.

We say that a state $\hat{s}$ of $\mathcal{M} \times [\Phi]$ is *realizable* if there exists a policy such that starting from this state, all constraints are satisfied.

We now introduce the compatibility conditions.

Definition 6. *Let* $\hat{\pi}$ *be a valued policy on* $\hat{\mathcal{M}} = \mathcal{M} \times [\Phi]$, *and denote the corresponding* θ *and* Δ *as in Def. 5.*

We say that $\hat{\pi}$ *satisfies the* ***state compatibility*** *if for every* $\hat{s} = (s, \nu, \mu) \in \hat{S}_{\hat{\pi}}$ *with* $\nu = (\nu_1, \dots, \nu_{N_1})$ *and* $\mu = (\mu_1, \dots, \mu_{N_2})$ *the following are satisfied for any state formulas* $\Phi_j, \Phi_{j_1}, \Phi_{j_2}$,

(i) If $\Phi_j = \mathbb{P}_{\geq p_j}\phi_i$, *then* $(\mu_j = 1) \Rightarrow (\nu_j \geq p_j)$;
(ii) If $\Phi_j = \Phi_{j_1} \wedge \Phi_{j_2}$, *then* $(\mu_j = 1) \Rightarrow (\mu_{j_1} = 1)$ *and* $(\mu_{j_2} = 1)$.
(iii) If $\Phi_j = \Phi_{j_1} \vee \Phi_{j_2}$, *then* $(\mu_j = 1) \Rightarrow (\mu_{j_1} = 1)$ *or* $(\mu_{j_2} = 1)$.
(iv) If $\Phi_j = b$ *for some* $b \in AP$, *then* $(\mu_j = 1) \Rightarrow s_j \models b$.

The state compatibility ensures that the valuations of the augmented states that $\hat{\pi}$ can reach are coherent. We introduce compatibility conditions for path formulas too.

Definition 7. *Let* $\hat{\pi}$ *be a valued policy on* $\hat{\mathcal{M}} = \mathcal{M} \times [\Phi]$, *and denote the corresponding* θ *and* Δ *as in Def. 5.*

The policy $\hat{\pi}$ *satisfies the* ***path compatibility*** *iff, for every augmented state* $\hat{s} = (s, \mu, \nu) \in \hat{S}_{\hat{\pi}}$, *with*

$$\theta(\hat{s}) = (\nu^i, \nu^i),\ \nu^i = (\nu^i_1, \dots, \nu^i_{sf(\Phi)}),\ \mu^i = (\mu^i_1, \dots, \mu^i_{pf(\Phi)}),$$
$$\nu = (\nu_1, \dots, \nu_{sf(\Phi)}),\ \mu = (\mu_1, \dots, \mu_{pf(\Phi)}),\ \delta = \Delta(\hat{s}),$$

the following conditions are satisfied with $m = m(s)$:

- *For every* ϕ_j *path formula of the form* $\phi_j = \Phi_{j_1} \mathbf{W} (\Phi_{j_1} \wedge \Phi_{j_2})$ *(the continuing-weak-until), we have*

$$\nu_j \leq \max\left(\mu_{j_1}\left(\sum_{i=1}^{m} P(s_i|a)\delta(a)\nu^i_j\right), \mu_{j_1}\mu_{j_2}\right).$$

- *For every* ϕ_j *path formula of the form* $\phi_j = \Phi_{j_1} \mathbf{W} \Phi_{j_2}$, *we have*

$$\nu_j \leq \max\left(\mu_{j_1}\left(\sum_{i=1}^{m} P(s_i|a)\delta(a)\nu^i_j\right), \mu_{j_2}\right).$$

- *For every* ϕ_j *path formula of the form* $\phi_j = \mathbf{X}\Phi_{j_1}$, *we have*

$$\nu_j \leq \left(\sum_{i=1}^{m} P(s_i|a)\delta(a)\mu^i_{j_1}\right).$$

The path compatibility ensures that the counters are coherent between an augmented state and its successors.

Theorem 2 (Coherence). *Let* $\hat{\pi}$ *be a valued policy that also satisfies the state and path compatibility conditions. Then, for every augmented state* $\hat{s} = (s, \mu, \nu)$ *reachable from* $\hat{s}_0$ *through* $\hat{\pi}$, *we have*

- *The counters are coherent: for every* $j \in \{1, \dots, pf(\Phi)\}$ *and path formula* ϕ_j, $\hat{\mathcal{M}}_{\hat{\pi}}(\hat{s}) \models \mathbb{P}_{\geq \theta_j}[\phi_j]$.
- *The valuations are coherent: for every* $j \in \{1, \dots, sf(\Phi)\}$ *and state formula* Φ_j, $(\mu_j = 1) \Rightarrow \hat{\mathcal{M}}_{\hat{\pi}}(\hat{s}) \models \Phi_j$.

6 VALUE ITERATION FOR CPCTL SYNTHESIS

Using the Coherence Theorem 2, our goal is to construct iteratively augmented states (s, μ, θ) that are *realizable*, meaning that there exists a policy $\hat{\pi}$ on $\mathcal{M}[\Phi]$ such that for all i, j, $\mathcal{M}_{\hat{\pi}}(s) = 1 \Rightarrow \mathcal{M}_{\hat{\pi}}(s, \mu, \theta) \models \Phi_i$, and $\mathcal{M}_{\hat{\pi}}(s, \mu, \theta) \models \mathbb{P}_{\geq \eta_j}[\phi_j]$. Using the state and path conditions as the backbone of the Bellman operator that we will define, we compute augmented states that are provably realizable. For $\Phi_j = b \in AP$ an atomic proposition and $b \in L(s)$, we can immediately deduce that the state (s, μ, θ) where $\mu_k = 1$ for $j = k$, $\mu_k = 0$ otherwise, and $\theta_k = 0$ for all k is realizable. This initialization, however, does not suffice for the Value-Iteration to reach all the realizable states. We know for instance from [12] that formulas involving the temporal operator $\mathbf{G}\Phi$ require the pre-computation of $\mathbb{P}_{=1}[\mathbf{G}\phi]$. We extend this idea to the continuing operator $\Phi_1 \mathbf{W} (\Phi_1 \wedge \Phi_2)$ and construct a suitable initialization. We then combine these elements and design an algorithm, CPCTL − VI, which solves the synthesis problem. In Theorem 4, we prove its soundness and optimality under a generalized Slater's assumption.

6.1 A Value Iteration Algorithm for CPCTL

For tuples $(\boldsymbol{\mu}, \boldsymbol{\nu})$ in $\{0,1\}^{sf(\Phi)} \times [0,1]^{pf(\Phi)}$, we define the partial order $\leq$ as

$$(\boldsymbol{\mu}, \boldsymbol{\nu}) \leq (\boldsymbol{\mu}', \boldsymbol{\nu}') \quad \text{iff} \quad \text{for all } i, j, \mu_i \leq \mu'_i \text{ and } \nu_j \leq \nu'_j.$$

For any closed subset $E \subseteq \{0,1\}^{sf(\Phi)} \times [0,1]^{pf(\Phi)}$, we let $\uparrow(E)$ be the set of maximal elements of E w.r.t. $\leq$.

Definition 8 (Bellman Operator for CPCTL). *The* extended Bellman operator $\mathcal{B}_\Phi$ *acts on elements of*

$$\mathcal{D}\left(\{0,1\}^{sf(\Phi)} \times [0,1]^{pf(\Phi)}\right)^{\#\mathcal{S}},$$

and is such that, for any $s \in S$, $\mathcal{B}_\Phi[V](s) = \uparrow(E)$, *where* E *is the set of all couple of tuples* $(\boldsymbol{\mu}, \boldsymbol{\nu}) = ((\mu_i)_{i\in\{1,\dots,sf(\Phi)\}}, (\nu_j)_{j\in\{1,\dots,pf(\Phi)\}})$ *such that there exists an action distribution* δ *over the actions* $A(s)$ *and there exists* $(\boldsymbol{\mu}^{s'}, \boldsymbol{\nu}^{s'})_{s'\in\mathcal{S}}$ *with* $(\boldsymbol{\mu}^{s'}, \boldsymbol{\nu}^{s'}) \in V(s')$ *for all* $s' \in S$ *such that all the following are satisfied for all* i, j.

Valuations for state formulas:

- *If* $\Phi_i = a$, *then* $\mu_i = 1$ *if* $a \in L(s)$ *and* 0 *otherwise*
- *If* $\Phi_i = \neg a$, *then* $\mu_i = 0$ *if* $a \in L(s)$ *and* 1 *otherwise*
- *If* $\Phi_i = \Phi_{i_1} \wedge \Phi_{i_2}$, *then* $\mu_i = \mu_{i_1}\mu_{i_2}$
- *If* $\Phi_i = \mathbb{P}_{\geq p}(\phi_j)$, *then* $\mu_i = 1$ *if* $\nu_j \geq p$ *and* 0 *otherwise*

Counters for path formulas:

- *If* $\Phi_j = \Phi_{j_1} \mathbf{W} (\Phi_{j_1} \wedge \Phi_{j_2})$, *then*

$$\nu_j = \begin{cases} 0 \text{ if } \mu_{j_1} = 0 \\ 1 \text{ if } \mu_{j_1} = \mu_{j_2} = 1 \\ \sum_{a\in A(s), s'\in S'} \delta(a)\, P(s,a,s')\, \nu_j^{s'} \text{ otherwise.} \end{cases}$$

Definition 9 (Initial Value Vector). *We let* $\mathcal{I}_\mathcal{M}$ *be such that, for any* $s \in S$, $\mathcal{I}_\mathcal{M}(s)$ *is the set of tuples* $(\boldsymbol{\mu}, \boldsymbol{\nu}) \in \{0,1\}^{sf(\Phi)} \times [0,1]^{pf(\Phi)}$ *satisfying: there exists a policy* π *such that for all* $\phi_j = \Phi_{j_1} \mathbf{W} (\Phi_{j_1} \wedge \Phi_{j_2})$

$$\nu_j = 1 \Leftrightarrow \mathcal{M}_\pi^s \models \mathbb{P}_{\geq 1}[\mathbf{G}(\mathsf{L}(\Phi_j))].$$

Additionally, the valuations are maximal for the partial order $\geq$ *among the set of valuations such that* $\mathcal{I}_\mathcal{M}$ *satisfies the state compatibility (as in Definition 6).*

Our Value Iteration algorithm is given in Algorithm 1.

Algorithm 1 CPCTL-VI

```
Input: MDP M, CPCTL formula Φ = ∧_{i=1}^{k} ℙ_{≥p_i}[φ_i]
Initialize V ← I_M
while for any q ∈ V(s_init), there exists i such that q_{Ψ_i} < p_i do
    V ← B_Φ[V]
end while
return V
```

The algorithm constructs a set of tuples $V(s)$ for each state s, starting with the values provided by Definition 9 and iteratively applies the Bellman Operator defined in Definition 8. The algorithm's goal is to maintain the following property: for each state s and each tuple $(\boldsymbol{\mu}, \boldsymbol{\nu})$, the augmented state $(s, \boldsymbol{\mu}, \boldsymbol{\nu})$ is realizable. Moreover, for each such augmented state, the algorithm associates a policy that realizes this augmented state. This construction is described in more detail in the next section.

6.2 CPCTL-VI: Soundness and Optimality

This section is devoted to the formal soundness and optimality of the algorithm. We first introduce the following lemma, showing how policies of the augmented MDP can be "de-augmented" back to the original MDP.

Lemma 2 (Projection onto original MDP). *Let* $\mathcal{M}$ *be a MDP,* Φ *a* CPCTL *formula,* $\hat{\mathcal{M}}$ *the augmented MDP, and* $\hat{\pi}$ *be a valued policy satisfying the state and path compatibilities. We define* $\hat{s}_0 = (s_0, \theta, \nu)$ *the state chosen from* s_0 *by* $\hat{\pi}$. *With* $\nu = (\mu, \mu_1, \dots, \mu_{N_2})$, *if* $\mu = 1$, *there exists* π *a policy on* $\mathcal{M}$, *such that* $\mathcal{M}_\pi(s_0) \models \Phi$. *Moreover, for any path subformula* ϕ_j, $\mathcal{M}_\pi(s_0) \models \mathbb{P}_{\geq \eta_j}\phi_j$.

We now introduce the Soundness theorem which states that each values (μ, ν) that we add to the set $E_n(s_0)$ corresponds to a realizable augmented state $\hat{s}$. In fact, the algorithm outputs for each such added values a policy that realizes this augmented state.

Theorem 3 (Soundness). *Let* $\mathcal{M}$ *be a MDP and* Φ *be a formula in* CPCTL. *Let* $E_n(s)$ *be the realisability set for* s *obtained after* n *steps using CPCTL-VI, then for every* $(\boldsymbol{\eta}, \boldsymbol{\mu}) \in E_n(s)$, *there exists a policy* π *such that*

$$\begin{cases} \forall j \leq pf(\Phi),\ \mathbb{P}_\mathcal{M}(\phi_j | s, \pi) \geq \eta_j, \\ \forall j \leq sf(\Phi),\ \mu_j = 1 \Rightarrow s \models_{\mathcal{M}_\pi} \Phi_j. \end{cases}$$

Moreover, for any $(\mu^0, \nu^0) \in E_n(s_0)$, *with* π_m^0 *the memoryful policy defined as:*

Initial memory state: *Initially,* $m_0 = (\mu^0, \nu^0)$.

Definition of $\pi_{(\mu,\nu)}(s)$***:***

- *If* $\forall j$ *such that* $\mu_j \neq 0$, $\Phi_j = b_j$ *or* $\neg b_j$ *for some* $b_j \in AP$, *then the policy takes any action, and the constructive algorithm stops.*
- *If there exists* $\Phi_j = \mathbb{P}_{\geq p_j}(\phi_j)$ *such that* $\mu_j = \nu_j = 1$ *and for all* k *such that* $\mu_k \neq 0$, Φ_k *is a subformula of* Φ_j *and* $\exists \pi_j$, $\mathcal{M}_{\pi_j}^s \models \mathbb{P}_{\geq 1}(\mathbf{G}\mathsf{L}(\Phi_j))$, *then* π_m^0 *follows the policy* π_j *indefinitely.*
- *Otherwise, with* $m = (\mu, \nu)$ *the current memory state, there exists* δ, *a distribution of actions of* $\mathcal{A}(s)$ *and tuples* (μ^i, ν^i) *for each successor* s_i *such that* $(\delta, \mu, \nu, (\mu^i)_i, (\nu^i)_i)$ *satisfies Bellman's inequality. The policy* π_m^0 *follows the distribution of action* δ. *When reaching the state* s_i, *it modifies its memory to the new value* $m = (\mu^i, \nu^i)$, *and follows* $\pi_{(\mu^i,\nu^i)}(s_i)$ *from there.*

the policy π^0 *is well defined and satisfies for all* $j \leq pf(\Phi)$, $\mathcal{M}_{\pi_{m_0}^0} \models \mathbb{P}_{\geq j}(\phi_j)$, *and for all* $j \leq sf(\Phi)$, $\mu_j = 1 \Rightarrow \mathcal{M}_{\pi_{m_0}} \models \Phi_j$.

Finally, we show in the next theorem that under the generalized Slater's Assumption, the algorithm is optimal, meaning that it finds a policy satisfying the Synthesis Problem when such a policy exists.

Theorem 4 (Optimality of CPCTL − VI). *We consider* $\mathcal{M}$ *a MDP and* Φ *a* CPCTL *formula. Denote by* V_n *the value frontier obtained after* n *steps of* CPCTL − VI, *i.e.* $V_n = (\mathcal{B}_\Phi)^n[\mathcal{I}_\mathcal{M}]$. *If* $\mathcal{M}, \Phi$ *satisfy the generalized Slater's Assumption, there exists* $n_0 \in \mathbb{N}, \exists \boldsymbol{p} \in V_{n_0}(s_0),\ \boldsymbol{p} \geq (p_1, \dots, p_{N_1})$.

7 NUMERICAL EXPERIMENTS

Presentation of the problem: A robot moves over a surface where the ground becomes increasingly slippery toward the unsafe borders. On the right side, the slipperiness remains constant and moderate, while near the left edge the surface is highly polished, making the

robot prone to sliding unpredictably in any direction. The robot must reach the goal region at the top while avoiding the unsafe edges and can not cross the central wall that divides the terrain.

We consider the formula $\mathbb{P}_{\geq p_1}\left(\mathbf{G}(\mathbb{P}_{\geq p_2}(\neg d \mathbf{W} G))\right)$, where $p_2 = 0.6$ and our goal is to maximize $\phi = \mathbb{P}\left(\mathbf{G}(\mathbb{P}_{\geq p_2}(\neg d \mathbf{W} G))|S\right)$. Compared to flat formulas, this reduces the bias from starting at a given state, as the robot is asked to minimize the risk of reaching high risk states later on. We implemented CPCTL-VI for ϕ in Python, and the experiments were carried on Windows 10 with an Intel(R) Core(TM) i7-12650H CPU.

7.1 First Model: a Gridworld with a central Wall

We introduce the first 7×7 Gridworld model with a central wall in figure 4. The Pareto Curve outputted by the Algorithm for this model is presented in Figure 5a. For any a on the curve, there exists a policy such that $\mathcal{M}_\pi \models \mathbb{P}_{\geq a}(\mathbf{G}\mathbb{P}_{\geq p_2}(\neg d \mathbf{W}(\neg d \wedge G)))$, while b is only used internally by the algorithm.

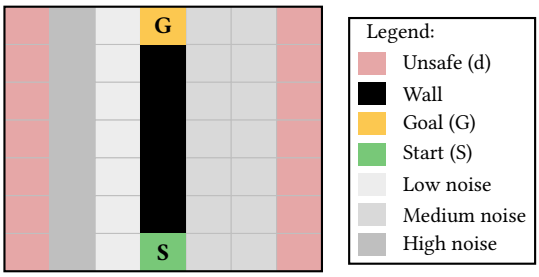
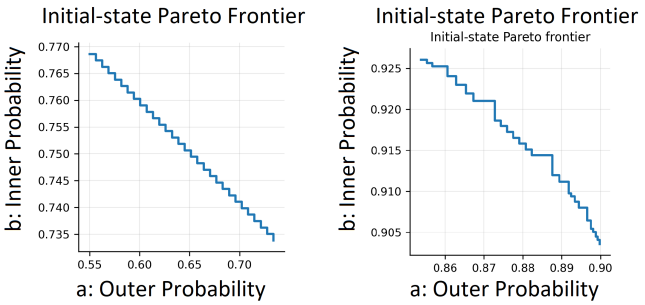


Figure 4: First Grid World: 7×7 grid with a central wall and slip gradient represented as gray shading (darker means higher slip probability). The slip is a uniform stochastic noise in all the available directions. The goal (G) is safe and the start (S) is the initial state.

In this model, the safest strategy, i.e. the strategy π that maximizes $\mathbb{P}(\phi|\pi, \text{Start})$ is to enter the right corridor and to try and stay in the fifth column. This strategy however corresponds to a higher risk to reach unsafe states in average, but decreases the probability of reaching high risk states compared to a strategy that uses the left corridor.

7.2 Second Model: Gridworld with a hole

We present the second model in figure 6. A hole is now present in the wall and the agent can move through it. The optimal strategy is now to follow the right side of the wall (sixth column) or go in the direction of this column when the agent is moved away from it by the stochastic noise. When too far on the left and blocked by the wall, the agent follows the left side of the wall (fourth column). The Pareto Curves obtained by the algorithm are presented in Figure 5b.



(a) First Model. **(b) Second Model.**

Figure 5: Results of the Algorithm CPCTL-VI for the two models. The curves are step functions because the convexity of the Pareto Frontier is not guaranteed. The points are not joined to ensure soundness and guarantee safety. The curve obtained in the second model is an example of non-convex Pareto Curve.

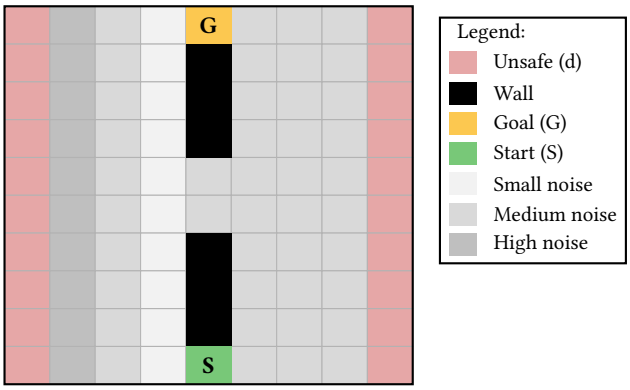


Figure 6: Second Grid World: 9×10. The agent starts at the initial state S and must reach the safe goal G while avoiding the unsafe borders. The central wall is impassable but possesses a hole in the middle that behaves like any right-side state. Darker gray indicates higher slip probability. The slip is a stochastic noise uniform in all the available directions.

8 CONCLUSIONS

In this paper we made the following key contributions to the synthesis problem for probabilistic temporal specifications. First, we introduced Continuing PCTL (CPCTL), an expressive fragment of PCTL, which allows the nesting of probabilistic operators. thus generalizing multi-objective avoidance specifications. Then, we provided a novel *augmented MDP* construction for Safe PCTL that encodes global satisfaction as local linear inequalities. In particular, we proved the coherence result Theorem 2. Further, we presented CPCTL-VI, a value-iteration algorithm for CPCTL synthesis, and proved its optimality under a generalized version of Slater's assumption. Finally, we evaluated CPCTL-VI experimentally, thus showing the feasibility of our method in practice.

Acknowledgments. The research presented in this paper was supported by the EPSRC grant number EP/X015823/1, "An abstraction-based technique for Safe Reinforcement Learning".

REFERENCES

[1] Christel Baier, Marcus Größer, Martin Leucker, Benedikt Bollig, and Frank Ciesinski. 2004. Controller Synthesis for Probabilistic Systems. In *Exploring New Frontiers of Theoretical Informatics, IFIP 18th World Computer Congress, TC1 3rd International Conference on Theoretical Computer Science (TCS2004), 22-27 August 2004, Toulouse, France (IFIP, Vol. 155)*, Jean-Jacques Lévy, Ernst W. Mayr, and John C. Mitchell (Eds.). Kluwer/Springer, 493–506. https://doi.org/10.1007/1-4020-8141-3_38

[2] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking.* MIT press.

[3] Nathalie Bertrand, John Fearnley, and Sven Schewe. 2012. Bounded satisfiability for PCTL. *arXiv preprint arXiv:1204.0469* (2012).

[4] Dimitri P. Bertsekas and Steven E. Shreve. 2007. *Stochastic Optimal Control: The Discrete-Time Case.* Athena Scientific.

[5] Benjamin Bordais, Damien Busatto-Gaston, Shibashis Guha, and Jean-François Raskin. 2022. Strategy Synthesis for Global Window PCTL. *arXiv preprint arXiv:2204.14107* (2022).

[6] Tomás Brázdil, Václav Brozek, Vojtech Forejt, and Antonín Kucera. 2006. Stochastic games with branching-time winning objectives. In *21st Annual IEEE Symposium on Logic in Computer Science (LICS'06)*. IEEE, 349–358.

[7] Tomás Brázdil, Vojtech Forejt, and Antonín Kucera. 2008. Controller Synthesis and Verification for Markov Decision Processes with Qualitative Branching Time Objectives. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations (Lecture Notes in Computer Science, Vol. 5126)*, Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz (Eds.). Springer, 148–159. https://doi.org/10.1007/978-3-540-70583-3_13

[8] Krishnendu Chatterjee and Thomas A. Henzinger. 2008. *Value Iteration.* Springer-Verlag, Berlin, Heidelberg, 107–138. https://doi.org/10.1007/978-3-540-69850-0_7

[9] Frank Ciesinski and Marcus Größer. 2004. *On Probabilistic Computation Tree Logic.* Springer Berlin Heidelberg, Berlin, Heidelberg, 147–188. https://doi.org/10.1007/978-3-540-24611-4_5

[10] Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi, and Mihalis Yannakakis. 2008. Multi-Objective Model Checking of Markov Decision Processes. *Log. Methods Comput. Sci.* 4, 4 (2008). https://doi.org/10.2168/LMCS-4(4:8)2008

[11] Vojtech Forejt, Marta Z. Kwiatkowska, and David Parker. 2012. Pareto Curves for Probabilistic Model Checking. In *Automated Technology for Verification and Analysis - 10th International Symposium, ATVA 2012, Thiruvananthapuram, India, October 3-6, 2012. Proceedings (Lecture Notes in Computer Science, Vol. 7561)*, Supratik Chakraborty and Madhavan Mukund (Eds.). Springer, 317–332. https://doi.org/10.1007/978-3-642-33386-6_25

[12] Serge Haddad and Benjamin Monmege. 2018. Interval iteration algorithm for MDPs and IMDPs. *Theor. Comput. Sci.* 735 (2018), 111–131. https://doi.org/10.1016/J.TCS.2016.12.003

[13] Paul R Halmos. 2013. *Measure theory.* Vol. 18. Springer.

[14] Joost-Pieter Katoen, Lei Song, and Lijun Zhang. 2014. Probably safe or live. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 1–10.

[15] Antonín Kučera and Oldřich Stražovskỳ. 2005. On the controller synthesis for finite-state Markov decision processes. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer, 541–552.

[16] Antonín Kucera and Oldrich Strazovskỳ. 2008. On the Controller Synthesis for Finite-State Markov Decision Processes. *Fundam. Informaticae* 82, 1-2 (2008), 141–153. http://content.iospress.com/articles/fundamenta-informaticae/fi82-1-2-10

[17] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (1st ed.). John Wiley & Sons, Inc., USA.

[18] Lei Song, Yuan Feng, and Lijun Zhang. 2015. Planning for Stochastic Games with Co-Safe Objectives. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Qiang Yang and Michael J. Wooldridge (Eds.). AAAI Press, 1682–1688. http://ijcai.org/Abstract/15/240

[19] D.J White. 1982. Multi-objective infinite-horizon discounted Markov decision processes. *J. Math. Anal. Appl.* 89, 2 (1982), 639–647. https://doi.org/10.1016/0022-247X(82)90122-6