Strategy Synthesis for Global Window PCTL

Benjamin Bordais
Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF, 91190 Gif-sur-Yvette, France

Damien Busatto-Gaston
Université libre de Bruxelles, Brussels, Belgium

Shibashis Guha
Tata Institute of Fundamental Research, Mumbai, India

Jean-François Raskin
Université libre de Bruxelles, Brussels, Belgium

Abstract

Given a Markov decision process (MDP) M and a formula Φ, the strategy synthesis problem asks if there exists a strategy σ s.t. the resulting Markov chain $M[\sigma]$ satisfies Φ. This problem is known to be undecidable for the probabilistic temporal logic PCTL. We study a class of formulae that can be seen as a fragment of PCTL where a local, bounded horizon property is enforced all along an execution. Moreover, we allow for linear expressions in the probabilistic inequalities. This logic is at the frontier of decidability, depending on the type of strategies considered. In particular, strategy synthesis is decidable when strategies are deterministic while the general problem is undecidable.

2012 ACM Subject Classification Mathematics of computing → Stochastic processes

Keywords and phrases Markov decision processes, synthesis, PCTL

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.115

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: http://arxiv.org/abs/2204.14107

Funding This work is partially supported by the ARC project Non-Zero Sum Game Graphs: Applications to Reactive Synthesis and Beyond (Fédération Wallonie-Bruxelles), the EOS project Verifying Learning Artificial Intelligence Systems (F.R.S.-FNRS & FWO), the COST Action 16228 GAMENET (European Cooperation in Science and Technology), by the PDR project Subgame perfection in graph games (F.R.S.- FNRS), and the DST-SERB SRG/2021/000466 "Zero-sum and Nonzero-sum Games for Controller Synthesis of Reactive Systems" project.

1 Introduction

Given an MDP M and a probabilistic temporal logic formula Φ, the strategy synthesis problem is to determine if there exists a strategy σ to resolve the nondeterminism in M such that the resulting Markov chain (MC) $M[\sigma]$ satisfies Φ, and if so, to construct one such strategy. The probabilistic temporal logic that we study in this paper allows us to express rich probabilistic global temporal constraints over a *bounded* horizon that must be enforced along all computations. Let us illustrate our logic with a few examples. The formula $\mathrm{A}\,\mathrm{G}(\mathbb{P}(\mathrm{F}^5\,\mathsf{Good}) \geq 0.95)$ expresses that it must always be the case, under the strategy σ, that along all computations, the probability to reach a good state within 5 steps is at least 0.95. This is a quantitative bounded horizon Büchi property. In addition, our logic allows for comparing the probability of different events: $\mathrm{A}\,\mathrm{G}\,\mathbb{P}(\mathrm{F}^5\,\mathsf{Good}) \geq 2 \times \mathbb{P}(\mathrm{F}^{10}\,\mathsf{Bad}))$ expresses that under the strategy σ, along all computations, it is always the case that the probability to reach a good state within 5 steps is at least twice the probability of reaching a bad state within 10 steps. The ability to compare probabilities of different events, while not present in classical

49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).
Editors: Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff;
Article No. 115; pp. 115:1–115:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

logics like PCTL, is necessary to express properties like probabilistic noninterference [13]. This feature has been introduced and studied in probabilistic hyperlogics, e.g. [2], where the ability to compare probabilities plays a central role in describing applications. While hyperlogics are very expressive and highly undecidable, we study here the ability to compare probabilities in a weaker logical setting in order to understand more finely the decidability border that probabilistic comparisons, and more generally linear expressions, induce.

While the model-checking problem for PCTL and MDPs is decidable [6], the synthesis problem is in general undecidable [15].[1] Synthesis for PCTL [15], HyperPCTL [2], and as well as for our logic (as shown in Theorem 37) is undecidable. To recover decidability, we explore two options. First, we consider subclasses of strategies: memoryless deterministic strategies (MD), memoryless randomized strategies (MR), and history-dependant deterministic strategies (HD) are important classes to be considered. Second, we identify syntactically defined sublogics with better decidability properties. For instance, while for PCTL objectives the synthesis problem for HD strategies is highly undecidable (Σ_1^1-complete) [15], it has been shown that the problem is decidable for the cases of MD and MR strategies [15]. The synthesis problem for the qualitative fragment of PCTL, where probabilistic operator can only be compared to constant 0 and 1, is decidable for HD strategies. An important contribution of this paper is to show that the synthesis problem for our sublogics is decidable for HD strategies. To the best of our knowledge, this is the first decidability result for a class of unbounded memory strategies (here HD) and quantitative probabilistic temporal properties.

Main technical contributions. We introduce the logic L-PCTL and two sublogics. L-PCTL extends PCTL with linear constraints over probability subformulae. We first study the window L-PCTL fragment that only allows bounded until or bounded weak until operators in the path formulae. The results for this fragment are presented in Table 1(a) where columns distinguish between memoryless (M) and history-dependent strategies (H), and rows between deterministic (D) and randomized strategies (R). Second, we study the *global* window L-PCTL extension of this logic in which window formulae appear in the scope of an A G operator that imposes the window formula to hold on every state of every computation. The results for this fragment are presented in Table 1(b). Third, we adapt results from the literature to the full logic as summarized in Table 1(c). An L-PCTL formula is *flat* if it does not have nested probabilistic operators, while it is *non-strict* if it does not contain strict comparison operators ($>$ or $<$) for comparing probability expressions.

Our two main technical contributions are focused on the synthesis problem for the global window L-PCTL logic. First, we introduce a fixpoint characterization of the set of strategies that enforces an L-PCTL window property globally. This characterization is effective for HD strategies, leads to a 2EXPTIME algorithm, and we provide an EXPTIME lower bound. Furthermore, the fixpoint characterization allows us to prove that the synthesis problem is in coRE for the class of history-dependent randomized (HR) strategies for the flat and non-strict fragment of global window L-PCTL. Second, we prove that the synthesis problem for HR strategies is undecidable with an original technique that reduces the halting problem of 2-counter Minsky machines (2CM) to our synthesis problem. We believe that the fixpoint characterization and the 2CM encoding are of independent interest.

[1] The difference between the two problems is essentially as follows: in the model-checking problem, each probabilistic operator in the formula is associated with one strategy (or scheduler) while in the synthesis problem, a *unique* strategy is fixed and used for all the probabilistic operators.

Table 1 A summary of our results for the synthesis problem on MDPs for L-PCTL formulae.

(a) synthesis for window L-PCTL.

	M	H
D	NP-complete [5]	PSPACE-complete Prop. 27
R	PSPACE SQRT-SUM-hard Prop. 29, [15]	EXPSPACE PSPACE-hard Thm. 17

(b) synthesis for global window L-PCTL.

	M	H
D	NP-complete	2EXPTIME EXPTIME-hard Prop. 28
R	PSPACE SQRT-SUM-hard Prop. 30	coRE-complete[a] Σ_1^1-hard Thm. 24, 37

[a] if the formula is flat and non-strict

(c) synthesis for L-PCTL.

strategies:	Memoryless	History-dependent
Deterministic	NP-complete [5]	Σ_1^1-complete [15]
Randomized	EXPTIME SQRT-SUM-hard [16], [15]	Σ_1^1-hard [15]

Finally, the satisfiability problem [7] for PCTL (and its variants) can be reduced to the synthesis problem. The decidability of the satisfiability problem for PCTL is a long standing open problem. Our decidability result for the synthesis problem for HD strategies and global window PCTL formulae can be transferred to the following version of the satisfiability problem: given a granularity g for the probabilities, and a global window PCTL formula Φ, does there exist an MC with granularity g that satisfies Φ? (Theorem 31). This gives a new positive decidability result for the satisfiability problem with an unbounded horizon fragment of PCTL and unbounded MCs.

Related work. The model-checking problem for PCTL is decidable [6] and should not be confused with the synthesis problem. In [15], the authors study the synthesis problem for PCTL on MDPs and stochastic games. In [5] it is shown that both randomization and memory in strategies are necessary even for flat window PCTL formulae. Further, [5] shows that the synthesis of MD strategies for PCTL objectives is NP-complete, and [16] shows that MR synthesis is in EXPTIME.[2] For the qualitative fragment of PCTL, deciding the existence of MR and HD strategies have been shown to be NP-complete and EXPTIME-complete, respectively [15]. As previously mentioned, the synthesis problems for HD and HR strategies in the general case of (quantitative) PCTL objectives are highly undecidable [15].

In [11], a probabilistic hyperlogic (PHL) has been introduced to study hyperproperties of MDPs. PHL allows quantification over strategies, and includes PCTL* and temporal logics for hyperproperties such as HYPERCTL* [10]. Hence the model-checking problem in PHL can ask for the existence of a strategy for hyperproperties, and has been shown to be

[2] For the existence of MR strategies for PCTL objectives, in the introduction of [15], it is claimed that the problem is in PSPACE, with a reference to [16]. However, in [16] only an EXPTIME upper bound is proven, for the more general problem of stochastic games. The proof encodes the problem as a polynomial-size formula in the first-order theory of the reals with a fixed alternation of quantifiers so that deciding it is in EXPTIME. The claim seems to be that the complexity of their approach drops to PSPACE when all states are controllable. There is no convincing argument there for that claim, in particular their formula still contains universal quantifiers.

undecidable [11]. Another related work is [1], where HyperPCTL [2] has been extended with strategy quantifiers, and studies hyperproperties over MDPs. The model-checking problem for this logic is also undecidable. In both of these undecidability proofs, the constructed formula contains unbounded finally (F) properties that cannot be expressed in the global window fragment of PCTL that we study here. In both [11] and [1], the model-checking problem is decidable when restricted to MD strategies but is undecidable for HD strategies.

The PCTL satisfiability problem is open for decades. In [14], decidability of finite and infinite satisfiability has been considered for several fragments of PCTL using unbounded finally (F) and unbounded always (G) operators. In [9], satisfiability for bounded PCTL has been considered where the number of steps or horizon used in the operators is restricted by a bound. In [4], a problem related to the satisfiability problem called the feasibility problem has been studied. Given a PCTL formula φ, and a family of Markov chains defined using a set of parameters and with a fixed number of states, the feasibility problem is to identify a valuation for the parameters such that the realized Markov chain satisfies φ. In the satisfiability problem that we study here, the number of states is however not fixed a priori and can be arbitrarily large.

2 Preliminaries

A *probability distribution* on a finite set S is a function $d : S \to [0, 1]$ such that $\sum_{s\in S} d(s) = 1$. We denote the set of all probability distributions on set S by $\mathsf{Dist}(S)$.

▶ **Definition 1.** *A* Markov chain *(MC) is a tuple $M = \langle S, s_{\mathsf{init}}, \mathbb{P}, \mathsf{AP}, L\rangle$ where S is a countable set of states, $s_{\mathsf{init}} \in S$ is an initial state, $\mathbb{P} : S \to \mathsf{Dist}(S)$ is a transition function, AP is a non-empty finite set of atomic propositions, and $L : S \to 2^{\mathsf{AP}}$ is a labelling function.*

If $\mathbb{P}$ maps a state s to a distribution d so that $d(s') > 0$, we write $s \xrightarrow{d(s')} s'$ or simply $s \to s'$, and we denote $\mathbb{P}(s, s')$ the probability $d(s')$. We say that the atomic proposition p holds on a state s if $p \in L(s)$.

A *finite path* $\rho = s_0s_1 \cdots s_i$ in an MC M is a sequence of consecutive states, so that for all $j \in [0, i-1]$, $s_j \to s_{j+1}$. We denote $|\rho| = i$ the length of ρ, $\mathsf{last}(\rho) = s_i$ and $\mathsf{first}(\rho) = s_0$. We also consider states to be paths of length 0. Similarly, an *infinite path* is an infinite sequence $\rho = s_0s_1 \cdots$ so that for all $j \in \mathbb{N}$, $s_j \to s_{j+1}$. If ρ is a finite (resp. infinite) path $s_0s_1 \cdots$, we let $\rho[i]$ denote s_i, $\rho[:i]$ denote the finite prefix $s_0 \cdots s_i$, and $\rho[i:]$ denote the finite (resp. infinite) suffix $s_is_{i+1} \cdots$.

We denote the set of all finite paths in M by FPaths_M. We introduce notations for the subsets FPaths^i_M (resp. $\mathsf{FPaths}^{\leq i}_M$, $\mathsf{FPaths}^{<i}_M$) of paths of length i (resp. of length at most or less than i). Let $\mathsf{FPaths}_M(s)$ denote the set of paths ρ in FPaths_M such that $\mathsf{first}(\rho) = s$. More generally, $\mathsf{FPaths}_M(\rho)$ denotes the set of paths which admit ρ as a prefix. Similarly, we let Paths_M be the set of infinite paths of M, and extend the previous notations for fixing an initial state or a shared prefix. In particular, $\mathsf{Paths}_M(\rho)$ is called the cylinder of ρ.

If $\rho = s_0 \ldots s_i$ is a finite path and $\rho' = s_is_{i+1} \ldots$ is a finite or infinite path so that $\mathsf{first}(\rho') = \mathsf{last}(\rho)$, let $\rho \cdot \rho' = s_0 \ldots s_is_{i+1} \ldots$ denote their concatenation.

▶ **Definition 2.** *Let s be a state of an MC M. The MC M naturally defines a* probability measure *μ^s_M on $(\mathsf{Paths}_M(s), \Omega^s_M)$, where Ω^s_M is the σ-algebra of cylinders, i.e. the sets $\mathsf{Paths}_M(\rho)$ with $\rho \in \mathsf{FPaths}_M(s)$, their complements and countable unions.*

The measure of a cylinder $\mathsf{Paths}_M(\rho)$ is the product of the probabilities of each transition in the finite path ρ, and by Carathéodory's extension theorem we get a measure μ^s_M over Ω^s_M. As s is always obvious from context (first state of the paths being considered), we omit it

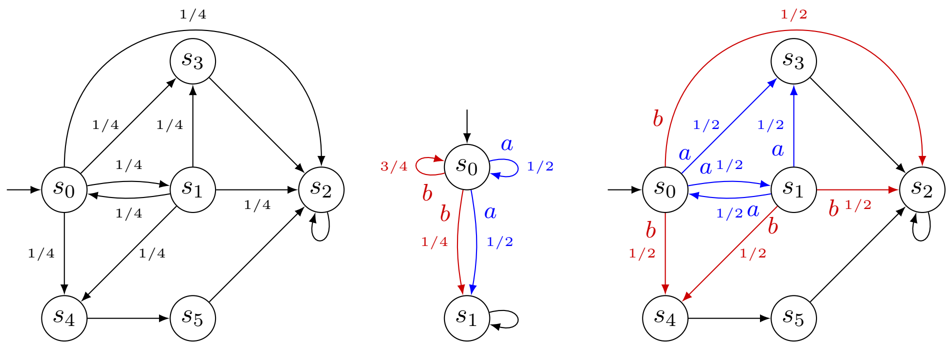


Figure 1 An MC and two MDPs, with states s_i and actions $\{a, b\}$. In MCs, transitions are labelled by their probability, and the probability is 1 if unspecified. In MDPs, transitions are labelled by their action and probability. If the action is unspecified (black transitions), then every action allows this transition. The initial state is s_0. In our examples, the set of atomic propositions is the set of states, so that the proposition s_i holds on state s_i only.

from the measure notation, in favour of μ_M. We note that $\mathsf{FPaths}_M(s)$ is a set of finite words over a countable alphabet, and as such is countable. In particular, if $\mathcal{C} \subseteq \mathsf{FPaths}_M(s)$ is a set of prefixes forming disjoint cylinders,[3] then $\mu_M(\bigcup_{\rho\in\mathcal{C}} \mathsf{Paths}_M(\rho)) = \sum_{\rho\in\mathcal{C}} \mu_M(\mathsf{Paths}_M(\rho))$. Moreover, if $\Pi \subseteq \mathsf{Paths}_M(s)$ is the complement of a measurable set Π', $\mu_M(\Pi) = 1 - \mu_M(\Pi')$.

▶ **Definition 3.** *A* Markov decision process *(MDP) is a tuple* $\mathcal{M} = \langle S, A, s_{\mathsf{init}}, \mathbb{P}, \mathsf{AP}, L\rangle$*, where S is a finite set of states, A is a finite set of actions, $s_{\mathsf{init}} \in S$ is an initial state, $\mathbb{P} : S \times A \to \mathsf{Dist}(S)$ is a transition function*[4]*, AP is a non-empty finite set of atomic propositions, and $L : S \to 2^{\mathsf{AP}}$ is a labelling function.*

If $\mathbb{P}$ maps a state s and an action a to a distribution d so that $d(s') > 0$, we write $s \xrightarrow{a,d(s')} s'$ or simply $s \xrightarrow{a} s'$, and we denote $\mathbb{P}(s, a, s')$ the probability $d(s')$. We extend from MCs to MDPs the definitions and notations of finite and infinite paths, now labelled by actions and denoted $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots$. Moreover, for a finite path ρ, we denote by $\rho \cdot as$ (resp. $sa \cdot \rho$) the concatenation of ρ with $\mathsf{last}(\rho) \xrightarrow{a} s$ (resp. of $s \xrightarrow{a} \mathsf{first}(\rho)$ with ρ).

We say that $\mathcal{M}$ is stored in size $|\mathcal{M}|$ if the number of states $|S|$, the number of actions $|A|$ and the number of transitions $s \xrightarrow{a} s'$ in $\mathcal{M}$ are bounded by $|\mathcal{M}|$. Then, $|\mathsf{FPaths}^{\leq i}_{\mathcal{M}}|$, the number of paths of horizon at most i, is in $|\mathcal{M}|^{\mathcal{O}(i)}$. Moreover, the probabilities in $\mathbb{P}$ are assumed to be rational numbers stored as pairs of integers $\frac{a}{b}$ in binary, so that $a, b \leq 2^{|\mathcal{M}|}$.

A (probabilistic) *strategy* is a function $\sigma : \mathsf{FPaths}_{\mathcal{M}} \to \mathsf{Dist}(A)$ that maps finite paths ρ to distributions on actions. A strategy σ is *deterministic* if the support of the distribution $\sigma(\rho)$ has size 1 for every ρ, it is *memoryless* if $\sigma(\rho)$ depends only on the last state of ρ, *i.e.* if σ satisfies that for all $\rho, \rho' \in \mathsf{FPaths}_{\mathcal{M}}$, $\mathsf{last}(\rho) = \mathsf{last}(\rho')$ implies $\sigma(\rho) = \sigma(\rho')$. We denote by $\sigma(\rho, a)$ the probability of the action a in the distribution $\sigma(\rho)$.

An MDP $\mathcal{M} = \langle S, A, s_{\mathsf{init}}, \mathbb{P}, \mathsf{AP}, L\rangle$ equipped with a strategy σ defines an MC, denoted $\mathcal{M}[\sigma]$, obtained intuitively by unfolding $\mathcal{M}$ and using σ to define the transition probabilities. Formally $\mathcal{M}[\sigma] = \langle \mathsf{FPaths}_{\mathcal{M}}, s_{\mathsf{init}}, \mathbb{P}_\sigma, \mathsf{AP}, L'\rangle$, with finite paths of $\mathcal{M}$ as states, transitions defined for all $\rho \in \mathsf{FPaths}_{\mathcal{M}}$, $a \in A$ and $s \in S$ by $\mathbb{P}_\sigma(\rho, \rho \cdot as) = \sigma(\rho, a)\mathbb{P}(\mathsf{last}(\rho), a, s')$, and

[3] $\mathsf{Paths}_M(\rho)$ and $\mathsf{Paths}_M(\rho')$ share a path if and only if either ρ is a prefix of ρ' or ρ' is a prefix of ρ.
[4] This formalism implies that every action is available from every state. This is w.l.o.g., as one can model illegal actions by sending them to a special state.

atomic propositions assigned by $L'(\rho) = L(\mathsf{last}(\rho))$. In particular, note that since S is finite $\mathsf{FPaths}_{\mathcal{M}}$ is infinite but countable. We say that a finite path ρ in $\mathcal{M}$ *matches* a finite path ρ' in $\mathcal{M}[\sigma]$ if $\mathsf{last}(\rho') = \rho$, so that they follow the same sequence of states and actions. We say that a path $\rho \in \mathsf{FPaths}_{\mathcal{M}}$ has probability m in $\mathcal{M}[\sigma]$ if ρ matches $\rho' \in \mathsf{FPaths}_{\mathcal{M}[\sigma]}$ and m is the measure of $\mathsf{Paths}_{\mathcal{M}[\sigma]}(\rho')$. It corresponds to the likelihood of having ρ as a prefix when following σ and starting from $\mathsf{first}(\rho)$.

We may omit M or $\mathcal{M}$ from all previous notations when they are clear from the context. MC notations may use σ as shorthand for $\mathcal{M}[\sigma]$, *e.g.* μ_σ is the probability measure induced by $\mathcal{M}[\sigma]$, and FPaths_σ refers to finite paths of non-zero probability under σ.

▶ **Example 4.** Consider the MC on the left of Figure 1, and the property asking to reach the state s_2 in at most two steps. Consider the set of paths of length at most two from s_0 to s_2. Let $\Pi = \mathsf{Paths}(s_0 s_2) \uplus \mathsf{Paths}(s_0 s_1 s_2) \uplus \mathsf{Paths}(s_0 s_3 s_2)$ be the infinite paths obtained from their cylinders. Then, the probability of reaching s_2 in two steps when starting from s_0 is $\mu(\Pi) = \frac{1}{4} + \frac{1}{16} + \frac{1}{4} = \frac{9}{16}$. Note that the probability of reaching s_2 in two steps when starting from s_1 is also $\frac{9}{16}$. Every other state reaches s_2 with probability 1 in two steps. Consider now the MDP in the middle of Figure 1, and the property asking that the state reached after the first transition is s_1. For every strategy σ, the probability that this property holds in $\mathcal{M}[\sigma]$ is equal to $\sigma(s_0, a)\frac{1}{2} + \sigma(s_0, b)\frac{1}{4} = \sigma(s_0, a)\frac{1}{2} + (1 - \sigma(s_0, a))\frac{1}{4} = \sigma(s_0, a)\frac{1}{4} + \frac{1}{4}$.

Probabilistic CTL with Linear expressions. A formula of L-PCTL is generated by the nonterminal Φ in the following grammar:

▶ **Definition 5** (L-PCTL in normal form, syntax)**.**

$$\Phi := p \mid \neg p \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \sum_{i=1}^{n} c_i \mathbb{P}\left[\varphi_i\right] \succcurlyeq c_0$$
$$\varphi := \mathrm{X}^\ell\, \Phi \mid \Phi_1 \,\mathrm{U}^\ell\, \Phi_2 \mid \Phi_1 \,\mathrm{W}^\ell\, \Phi_2 \mid \Phi_1 \,\mathrm{U}^\infty\, \Phi_2 \mid \Phi_1 \,\mathrm{W}^\infty\, \Phi_2$$

where p ranges over the atomic propositions in AP*, ℓ ranges over $\mathbb{N}$, and $n \in \mathbb{N}_{>0}$, $(c_0, \cdots, c_n) \in \mathbb{Z}^n$, $\succcurlyeq \in \{\geq, >\}$ define linear inequalities.*

We call a formula generated by Φ a *state formula*, and a formula generated by φ a *path formula*. The *horizon label* of a path formula is the label of its root operator, *i.e.* either ℓ or ∞. Intuitively, the Next operator $\mathrm{X}^\ell\, \Phi$ means that Φ holds in exactly ℓ steps, the (unbounded) Until and Weak until operators U^∞ and W^∞ are defined as usual in CTL, and their bounded version $\Phi_1 \,\mathrm{U}^\ell\, \Phi_2$ and $\Phi_1 \,\mathrm{W}^\ell\, \Phi_2$ impose a horizon on the reachability of Φ_2. We will use the standard notations X, U and W, defined by X^1, U^∞ and W^∞, respectively.

▶ **Definition 6** (L-PCTL in normal form, semantics)**.** *For a fixed MC M of states S, we inductively define $\llbracket \Phi \rrbracket_M$ as a set of states, and for each state s we define $\llbracket \varphi \rrbracket_M^s$ as a measurable set of infinite paths starting from s:*

$$\llbracket p \rrbracket_M = \{s \in S \mid p \in L(s)\} \qquad \llbracket \neg p \rrbracket_M = \{s \in S \mid p \notin L(s)\}$$
$$\llbracket \Phi_1 \wedge \Phi_2 \rrbracket_M = \llbracket \Phi_1 \rrbracket_M \cap \llbracket \Phi_2 \rrbracket_M \qquad \llbracket \Phi_1 \vee \Phi_2 \rrbracket_M = \llbracket \Phi_1 \rrbracket_M \cup \llbracket \Phi_2 \rrbracket_M$$
$$\left\llbracket \sum_{i=1}^{n} c_i \mathbb{P}\left[\varphi_i\right] \succcurlyeq c_0 \right\rrbracket_M = \{s \in S \mid \sum_{i=1}^{n} c_i \ \mu_M(\llbracket \varphi_i \rrbracket_M^s) \succcurlyeq c_0\}$$
$$\left\llbracket \mathrm{X}^\ell\, \Phi \right\rrbracket_M^s = \{\rho \in \mathsf{Paths}(s) \mid \rho[\ell] \in \llbracket \Phi \rrbracket_M\}$$
$$\left\llbracket \Phi_1 \,\mathrm{U}^\ell\, \Phi_2 \right\rrbracket_M^s = \{\rho \in \mathsf{Paths}(s) \mid \exists j \leq \ell, \rho[j] \in \llbracket \Phi_2 \rrbracket_M \wedge \forall i < j, \rho[i] \in \llbracket \Phi_1 \rrbracket_M\}$$

$$\left[\!\left[\Phi_1 \, \mathrm{W}^\ell \, \Phi_2\right]\!\right]^s_M = \{\rho \in \mathsf{Paths}(s) \mid \forall j \leq \ell, (\rho[j] \in [\![\Phi_1]\!]_M \vee \rho[j] \in [\![\Phi_2]\!]_M) \vee \exists i < j, \rho[i] \in [\![\Phi_2]\!]_M\}$$

$$[\![\Phi_1 \, \mathrm{U} \, \Phi_2]\!]^s_M = \{\rho \in \mathsf{Paths}(s) \mid \exists j \in \mathbb{N}, \rho[j] \in [\![\Phi_2]\!]_M \wedge \forall i < j, \rho[i] \in [\![\Phi_1]\!]_M\}$$

$$[\![\Phi_1 \, \mathrm{W} \, \Phi_2]\!]^s_M = \{\rho \in \mathsf{Paths}(s) \mid \forall j \in \mathbb{N}, (\rho[j] \in [\![\Phi_1]\!]_M \vee \rho[j] \in [\![\Phi_2]\!]_M) \vee \exists i < j, \rho[i] \in [\![\Phi_2]\!]_M\}$$

Then, we write $s \models_M \Phi$ (resp. $\rho \models_M \varphi$) if $s \in [\![\Phi]\!]_M$ (resp. $\rho \in [\![\varphi]\!]_M^{\mathsf{first}(\rho)}$), and say that s satisfies Φ (resp. ρ satisfies φ). We denote by $\equiv$ the semantic equivalence of state or path formulae (that holds on all MCs). Finally, we write $M \models \Phi$ if $s_{\mathsf{init}} \models_M \Phi$. Note that by restricting the linear inequalities to $n = 1$ and $\ell = 1$ in X^ℓ, we recover the standard definition of PCTL (see *e.g.* [6]).

We define usual notions as syntactic sugar, so that state formulae allow for $\bot := p \wedge \neg p$ and $\top := p \vee \neg p$ (for any $p \in \mathsf{AP}$). We allow rational constants c_1 and all comparison symbols in $\{\leq, <, =, \neq, >, \geq\}$ in linear expressions, with $\sum_{i=1}^n c_i \mathbb{P}[\varphi_i] \preccurlyeq c_0 := \sum_{i=1}^n (-c_i)\mathbb{P}[\varphi_i] \succcurlyeq -c_0$ and $=, \neq$ defined as conjunctions or disjunctions. Moreover, path formulae allow for $\mathrm{F}^\ell \, \Phi := \top \, \mathrm{U}^\ell \, \Phi$ and $\mathrm{G}^\ell \, \Phi := \Phi \, \mathrm{W}^\ell \, \bot$. We allow the negation operation $\neg$ in state and path formulae, and recover a formula in normal form using De Morgan's laws, the negation of inequalities ($\geq$ becomes $<$ and $>$ becomes $\leq$), and the duality rule $\neg(\Phi_1 \, \mathrm{W}^\ell \, \Phi_2) \equiv (\neg\Phi_1 \wedge \neg\Phi_2) \, \mathrm{U}^\ell \, \neg\Phi_1$. Finally, boolean implication and equivalence are defined as usual. A notable property is $\Phi_1 \, \mathrm{W}^\ell \, \Phi_2 \equiv (\Phi_1 \, \mathrm{U}^\ell \, \Phi_2) \vee \mathrm{G}^\ell \, \Phi_1$.

We encode L-PCTL formulae as trees, whose internal nodes are labelled by state or path operators and whose leaves are labelled by atomic propositions. Let $\ell_{\max} \geq 1$ denote an upper bound on horizon labels ℓ of subformula of Φ where ℓ is finite. The constants c_i in linear expressions are encoded in binary, and *the horizon labels ℓ are encoded in unary*, so that if the overall encoding of Φ is of size $|\Phi|$, we shall have $\ell_{\max} \leq |\Phi|$. We argue that this choice is justified from a larger point of view that extends PCTL to PCTL* by allowing boolean operations in path formulae, as the bounded horizon operators $\mathrm{X}^\ell, \mathrm{U}^\ell, \mathrm{W}^\ell$ can be seen as syntactic sugar for a disjunction of nested X operators of size $\mathcal{O}(\ell)$.

▶ **Definition 7.** *An L-PCTL formula Φ (in normal form) is a* window formula *if the horizon label ℓ of every path operator in Φ is finite, so that the unbounded* U *and* W *are not used. It is a* non-strict *formula if $\succcurlyeq$ is always $\geq$ in its linear inequalities. It is a* flat *formula if the measure operator $\mathbb{P}$ is never nested, so that if Φ is seen as a tree, every branch has at most one node labelled by a linear inequality $\sum_{i=1}^n c_i \mathbb{P}[\varphi_i] \succcurlyeq c_0$.*

▶ **Definition 8.** *A* global window *formula is a formula of the shape $\mathrm{A}\,\mathrm{G}\,\Phi$, with Φ a window L-PCTL formula. It is satisfied by a state s of M if every infinite path in $\mathsf{Paths}_M(s)$ satisfies the path formula $\mathrm{G}\,\Phi$, or equivalently if every state reachable from s satisfies Φ.*

▶ **Lemma 9.** *The* global window *formula $\mathrm{A}\,\mathrm{G}\,\Phi$ is satisfied on a state s of M if and only if s satisfies the L-PCTL formula $\mathbb{P}[\mathrm{G}\,\Phi] = 1$.*

Proof. If $\mathrm{A}\,\mathrm{G}\,\Phi$ holds on s, then $\mu_M([\![\mathrm{G}\,\Phi]\!]^s_M) = \mu_M(\mathsf{Paths}_M(s)) = 1$. If $\mathrm{A}\,\mathrm{G}\,\Phi$ is not satisfied on s, then there exists a finite path ρ leading to a state that violates Φ, so that the entire cylinder $\mathsf{Paths}_M(\rho)$ satisfies the path formula $\mathrm{F}\,\neg\Phi$. It follows that $\mu_M([\![\mathrm{G}\,\Phi]\!]^s_M) = 1 - \mu_M([\![\mathrm{F}\,\neg\Phi]\!]^s_M) \leq 1 - \mu_M(\mathsf{Paths}_M(\rho)) < 1$. ◀

▶ **Example 10.** Consider the MC M to the left of Figure 1. Let Φ be the L-PCTL formula $\mathbb{P}\left[\mathrm{F}^2\, s_2\right] \geq \frac{9}{16}$. It is a window formula, that is flat and non-strict. As detailed in Example 4, every state of M satisfies Φ. Therefore, M satisfies the global window formula $\mathrm{A}\,\mathrm{G}\,\Phi$.

Consider now the MDP $\mathcal{M}$ to the right of Figure 1. Let σ denote the memoryless strategy that chooses, in s_0 and s_1, action a with probability $\frac{1}{2}$ and action b with probability $\frac{1}{2}$. While $\mathcal{M}[\sigma]$ is an infinite MC by definition, it is bisimilar to the MC on the left of Figure 1 and must satisfy the same PCTL formulae, so that $\mathcal{M}[\sigma] \models \mathrm{A\,G}\,\Phi$. In Section 3, we will show that σ is the only strategy on $\mathcal{M}$ that satisfies $\mathrm{A\,G}\,\Phi$.

Model checking and synthesis problems. The *model-checking problem* of an L-PCTL formula Φ and of a finite MC M is the decision problem asking if $M \models \Phi$. The *synthesis problem* of an L-PCTL formula Φ and of an MDP $\mathcal{M}$ asks if there exists a strategy σ so that $\mathcal{M}[\sigma] \models \Phi$. We also consider the sub-problems that restrict the set of strategies to subsets defined by constraints on the memory or on determinism. For example, the memoryless (resp. deterministic) synthesis problem asks for a memoryless (resp. deterministic) strategy satisfying the formula. They are indeed distinct problems:

▶ **Example 11.** Consider the MDP in the middle of Figure 1. Let Φ be the window formula $(\mathbb{P}\left[\mathrm{F}^2\, s_1\right] = \frac{5}{8}) \wedge (\mathbb{P}\left[\mathrm{X}\, s_1\right] \geq \frac{1}{2} \vee \mathbb{P}\left[\mathrm{X}\, s_1\right] \leq \frac{1}{4})$. First, $s_0 \models \mathbb{P}\left[\mathrm{X}\, s_1\right] \geq \frac{1}{2} \Leftrightarrow \sigma(s_0, a) = 1$ and $s_0 \models \mathbb{P}\left[\mathrm{X}\, s_1\right] \leq \frac{1}{4} \Leftrightarrow \sigma(s_0, a) = 0$, so that the first move must be deterministic. If the first action is a, and the transition $s_0 \xrightarrow{a} s_0$ is chosen, then the next choice must be b to ensure $\mathbb{P}\left[\mathrm{F}^2\, s_1\right] = \frac{5}{8}$. Similarly, if the first action is b, the next choice on s_0 must be a. Moreover, $s_1 \models \Phi$ under any strategy. Thus, the only strategies that satisfy $\mathrm{A\,G}\,\Phi$ are the strategies that alternate between a and b as long as we are in s_0, while no memoryless strategy satisfies $\mathrm{A\,G}\,\Phi$. On the other hand, in Example 10 randomisation is needed. An example that require both randomisation and memory can be constructed by combining both examples.

▶ **Proposition 12.** *The* model-checking *problem for L-PCTL formulae and finite MCs can be solved in* PTIME. *This comes at no extra cost when compared to standard PCTL [6].*

Proof. This problem is detailed in [6, Thm. 10.40] for a PCTL definition that only allows expressions of the shape $c_1\mathbb{P}\left[\varphi\right] \succcurlyeq c_0$ to quantify over path formulae. Extending to $\sum_{i=1}^{n} c_i\mathbb{P}\left[\varphi_i\right] \succcurlyeq c_0$ is straight-forward, as their algorithm computes the measure of $\mathbb{P}\left[\varphi\right]$, and then checks if the comparison holds. The intuition is that the measure of bounded operators X^ℓ, U^ℓ and W^ℓ are obtained by $\mathcal{O}(\ell)$ vector-matrix multiplications, while unbounded U and W are seen as linear equation systems. Overall, the complexity is in $|M|^{\mathcal{O}(1)}|\Phi|\ell_{\max}$. ◀

3 Synthesis for global window PCTL

In this section, we detail complexity results on the synthesis problem for global window L-PCTL formulae. We fix a Markov decision process $\mathcal{M}$, a formula $\mathrm{A\,G}\,\Phi$ where Φ is a window L-PCTL formula, and ask if there exists a strategy σ so that $\mathcal{M}[\sigma] \models \mathrm{A\,G}\,\Phi$. We also address the sub-problems concerning deterministic or memoryless strategies.

Solving window formulae. We start by constructing a strategy σ so that $\mathcal{M}[\sigma] \models \Phi$. The formula Φ can be seen syntactically as a tree with state or path operators on internal nodes and atomic propositions on leaves. The window length of a branch of this tree is the sum of the horizon labels ℓ of path operators in the branch. The *window length* of the formula Φ is an integer $\mathcal{L}$ obtained as the maximum over every branch of Φ of their respective window lengths. In particular, $\mathcal{L} \leq |\Phi|\ell_{\max}$. For example, if $\Phi = \mathbb{P}\left[\mathrm{X}\,\mathbb{P}\left[\mathrm{X}^2\, p_1\right] \geq \frac{1}{2}\right] \leq \frac{1}{2} \vee \mathbb{P}\left[\mathrm{F}^2\, p_3\right] > 0$, then $\ell_{\max} = 2$ and $\mathcal{L} = \max(1 + 2, 2) = 3$.

▶ **Definition 13.** *Let s be a state of $\mathcal{M}$ and Φ be a window L-PCTL formula of window length $\mathcal{L}$. A* window strategy *for s of horizon $\mathcal{L}$ is a mapping $\partial : \mathsf{FPaths}_{\mathcal{M}}^{<\mathcal{L}}(s) \to \mathsf{Dist}(A)$.*

A window strategy ∂ for state s can be seen as a partial strategy, only defined on paths of length under $\mathcal{L}$ that start from s. Formally, ∂ defines a set of strategies $\sigma : \mathsf{FPaths}_{\mathcal{M}} \to \mathsf{Dist}(A)$, where the first $\mathcal{L}$ steps from s are specified by ∂, and the subsequent steps are not. This set of strategies is called the *cylinder* of the window strategy ∂. In particular, if two strategies σ and σ' are in the cylinder of the window strategy ∂, then the MCs $\mathcal{M}[\sigma]$ and $\mathcal{M}[\sigma']$ coincide for the first $\mathcal{L}$ steps from s, in the sense that every path $\rho \in \mathsf{FPaths}^{<\mathcal{L}}_{\mathcal{M}}(s)$ has the same probability m in $\mathcal{M}[\sigma]$ and in $\mathcal{M}[\sigma']$. In this case, we say that m is the probability of ρ under ∂.

We may conflate a window strategy ∂ with an arbitrary strategy σ in its cylinder, so that $\mathsf{FPaths}^{<\mathcal{L}}_{\partial}(s)$ is a set of paths in $\mathcal{M}[\sigma]$. Then, we say that the window strategy ∂ for state s satisfies Φ, noted $s \models_\partial \Phi$, if $s \models_\sigma \Phi$ for all σ in the cylinder of ∂. Conversely, a strategy $\sigma : \mathsf{FPaths}_{\mathcal{M}} \to \mathsf{Dist}(A)$ naturally defines a window strategy ∂_ρ for every fixed prefix ρ, so that for all $\rho' \in \mathsf{FPaths}^{<\mathcal{L}}_{\mathcal{M}}(\mathsf{last}(\rho))$, $\partial_\rho(\rho') = \sigma(\rho \cdot \rho')$.

▶ **Lemma 14.** *Let Φ be a window L-PCTL formula of window length $\mathcal{L}$, σ be a strategy for $\mathcal{M}$, and let ∂_s be the window strategy defined by σ on state s and horizon $\mathcal{L}$ (the fixed prefix is s). Then, it holds that $s \models_\sigma \Phi \Leftrightarrow s \models_{\partial_s} \Phi$.*

Thus, the synthesis problem on window formulae reduces to finding a window strategy ∂ for s_{init} so that $s_{\mathsf{init}} \models_\partial \Phi$. Let ∂ be a window strategy for state s and horizon $\mathcal{L}$. Let $\mathcal{X}_s$ denote a finite set of variables $x_{\rho,a}$, with $\rho \in \mathsf{FPaths}^{<\mathcal{L}}_{\mathcal{M}}(s)$, and $a \in A$. The window strategy ∂ can be seen as a point in the real number space $\mathbb{R}^{\mathcal{X}_s}$, where $x_{\rho,a}$ encodes $\partial(\rho, a)$. Conversely, every point in $\mathbb{R}^{\mathcal{X}_s}$ so that $\forall x \in \mathcal{X}_s$, we have $x \in [0,1]$, and $\forall \rho \in \mathsf{FPaths}^{<\mathcal{L}}_{\mathcal{M}}(s)$, we have $\sum_{a \in A} x_{\rho,a} = 1$ represents a window strategy. Therefore, the points of $\mathbb{R}^{\mathcal{X}_s}$ that encode a window strategy can be described by a finite conjunction of linear inequalities $x \geq 0$, $x \leq 1$ and $x_{\rho,a_1} + \cdots + x_{\rho,a_k} = 1$ over the variables $\mathcal{X}_s$.

We want to similarly characterise the set of window strategies satisfying a given window L-PCTL formula. As will become apparent later on, we will need polynomial inequalities.

▶ **Definition 15.** *The* first-order theory of the reals *(FO-$\mathbb{R}$) is the set of all well-formed sentences of first-order logic that involve universal and existential quantifiers and logical combinations of equalities and inequalities of real polynomials.*[5]

We allow the use of strict comparison operators $\{<, \neq, >\}$ as the negation of their non-strict versions. We also assume that the formula is written in *prenex normal form* (PNF), *i.e.* as a sequence of alternating blocks of quantifiers followed by a quantifier-free formula. Finally, if $\mathcal{S} = \{x_1, \cdots, x_k\}$ is a finite set of variables, we use the notation $\exists \mathcal{S}$ as shorthand for the quantifier sequence $\exists x_1 \cdots \exists x_k$.

This theory is decidable, and admits a doubly-exponential quantifier elimination procedure [18]. Of particular interest is the existential fragment of FO-$\mathbb{R}$, denoted $\exists$-$\mathbb{R}$, where only $\exists$ is allowed. It is decidable in PSPACE [8].

We say that an FO-$\mathbb{R}$ formula of free variables $\mathcal{X}$ is *non-strict* if it is satisfied by a closed set of points in $\mathbb{R}^{\mathcal{X}}$. In particular, an FO-$\mathbb{R}$ formula that only uses non-strict comparison symbols $\{\leq, =, \geq\}$ and that is negation-free[6] is non-strict.

▶ **Proposition 16.** *Let s be a state of $\mathcal{M}$ and Φ be a window L-PCTL formula. The set of window strategies ∂ such that $s \models_\partial \Phi$ can be represented in $\exists$-$\mathbb{R}$ as a PNF formula of free variables $\mathcal{X}_s$. This formula is of size $|\Phi||\mathcal{M}|^{\mathcal{O}(\mathcal{L})}$, and can be computed in* $\mathsf{EXPTIME}$. *If Φ is flat and non-strict then the $\exists$-$\mathbb{R}$ formula is non-strict.*

---

[5] The primitives operations are multiplication and addition, the comparison symbols are $\{\leq, =, \geq\}$.
[6] A formula is negation-free if the Boolean negation operator $\neg$ is not used.

Proof sketch. We encode the problem in the theory of the reals, by using free variables $x_{\rho,a} \in [0,1]$ that have the value of $\partial(\rho,a)$, existential variables $y_{\rho,\Phi'} \in \{0,1\}$ that are true if the state subformula Φ' is satisfied when one follows ∂ after a fixed history of ρ, and existential variables $z_{\rho,\varphi} \in [0,1]$ having the probability that the path subformula φ is satisfied when one follows ∂ after a fixed history of ρ. The z variables use "local consistency equations" that equate the probability of a path formula on the current state as a linear combination of its probability on the successor states. For the $\mathrm{X}^\ell\,\Phi'$ formula this translates into $z_{\rho,\mathrm{X}^\ell\,\Phi'} = \sum_{s\xrightarrow{a}s'} x_{\rho,a}\mathbb{P}(s,a,s')z_{\rho\cdot as',\mathrm{X}^{\ell-1}\,\Phi'}$ for example. The y variables can then be defined, so that $y_{\rho,(\sum_{i=1}^n c_i\mathbb{P}[\varphi_i]\succcurlyeq c_0)} = 1$ if and only if $\sum_{i=1}^n c_i z_{\rho,\varphi_i} \succcurlyeq c_0$. Lastly we ask that $y_{s,\Phi} = 1$. To maintain the non-strict property, some subtlety is needed in the way nested probabilistic operators are dealt with. ◀

If we use a PSPACE decision procedure for $\exists$-$\mathbb{R}$ on the formula of Proposition 16, we get:

▶ **Theorem 17.** *The synthesis problem for* window *L-PCTL formulae is in* EXPSPACE.

▶ **Example 18.** Consider the MDP $\mathcal{M}$ to the right of Figure 1, and $\Phi = \mathbb{P}\left[\mathrm{F}^2\,s_2\right] \geq \frac{9}{16}$. Let us describe the formula obtained by Proposition 16. $s_0 \models_\partial \Phi$ can be encoded schematically as the formula $\exists z_{s_0,\mathrm{F}^2\,s_2}\exists z_{s_0as_1,\mathrm{F}^1\,s_2}$, s.t. $z_{s_0,\mathrm{F}^2\,s_2} = \frac{x_{s_0,a}}{2} z_{s_0as_1,\mathrm{F}^1\,s_2} + \frac{x_{s_0,a}}{2} + \frac{x_{s_0,b}}{2} \wedge z_{s_0as_1,\mathrm{F}^1\,s_2} = \frac{x_{s_0as_1,b}}{2} \wedge z_{s_0,\mathrm{F}^2\,s_2} \geq \frac{9}{16}$. For readability reasons, we simplified boolean expressions involving $\top$ or $\perp$ when appropriate, and we omitted the variables that can be simplified out immediately, as well as the constraints making sure that the variables x encode probabilities.

After quantifier elimination, and using $x_{s_0,a}+x_{s_0,b} = 1$, we get $x_{s_0,a}\ x_{s_0as_1,b} \geq \frac{1}{4}$. Observe that, as mentioned in Example 10, a window strategy ∂ that sets $x_{s_0,a} = x_{s_0as_1,b} = \frac{1}{2}$ satisfies Φ. Similarly, $s_1 \models_\partial \Phi$ can be encoded as $x_{s_1,a}\ x_{s_1as_0,b} \geq \frac{1}{4}$.

Fixed point characterisation of global window formulae. Let Φ be a window L-PCTL formula of window length $\mathcal{L}$. In this subsection, we describe a fixed point characterisation of the synthesis problem for the global window formula $\mathrm{A}\,\mathrm{G}\,\Phi$.

A *window strategy portfolio* Π of horizon $\mathcal{L}$ (in short, a portfolio Π) maps each state s to a set Π_s of window strategies for s of horizon $\mathcal{L}$. A window strategy portfolio can be seen as a set of points in $\mathbb{R}^{\mathcal{X}_s}$ for every state s. Given two window strategy portfolios Π and Π' of horizon $\mathcal{L}$, we write $\Pi \subseteq \Pi'$ if for all $s \in S$, it holds that $\Pi_s \subseteq \Pi'_s$. Then, the set of all window strategy portfolios of horizon $\mathcal{L}$ is a complete lattice w.r.t. $\subseteq$, where for a set $\mathcal{S}$ of portfolios, the meet $\bigsqcap \mathcal{S}$ (resp. the join $\bigsqcup \mathcal{S}$) maps s to $\bigcap_{\Pi\in\mathcal{S}} \Pi_s$ (resp. $\bigcup_{\Pi\in\mathcal{S}} \Pi_s$).

Let $s \xrightarrow{a} s'$ be a transition in $\mathcal{M}$, and let ∂, ∂' be window strategies for s and s', respectively, of horizon $\mathcal{L}$. We say that ∂ and ∂' are *compatible* w.r.t. $s \xrightarrow{a} s'$ if they make the same decisions on shared paths, *i.e.* for all $\rho \in \mathsf{FPaths}^{<\mathcal{L}}_{\mathcal{M}}(s')$ the probability of $sa\cdot\rho$ under ∂ equals the probability of ρ under ∂' multiplied by $\partial(s,a)\mathbb{P}(s,a,s')$. In particular, whenever $sa\cdot\rho$ has non-zero probability under ∂ and $|\rho| < \mathcal{L}-1$, we have $\partial(sa\cdot\rho) = \partial'(\rho)$. Similarly, we say that ∂ and a set $\Pi_{s'}$ of window strategies for s' are compatible w.r.t. $s \xrightarrow{a} s'$ if either $\partial(s,a) = 0$ or there exists a window strategy ∂' in $\Pi_{s'}$ so that ∂ and ∂' are compatible w.r.t. $s \xrightarrow{a} s'$.

Let f map portfolios to portfolios, so that $\mathsf{f}(\Pi)$ maps $s \in S$ to the set $\mathsf{f}(\Pi)_s$ of window strategies $\partial \in \Pi_s$ so that for each $s \xrightarrow{a} s'$ in $\mathcal{M}$, we have that ∂ and $\Pi_{s'}$ are compatible w.r.t. $s \xrightarrow{a} s'$. Intuitively, f removes from Π_s the window strategies ∂ that are not compatible with any continuation after a transition $s \xrightarrow{a} s'$ for some action a. Then, f is expressible in the theory of the reals:

▶ **Lemma 19.** *Let Π be a portfolio, encoded as an $\exists$-$\mathbb{R}$ formula R^s, of free variables $\mathcal{X}_s$, for every state s. Assume that each R^s is a PNF formula of size F. Then, $\mathsf{f}(\Pi)_s$ can also be encoded as a PNF formula, of size in $\mathcal{O}(|\mathcal{M}|F) + |\mathcal{M}|^{\mathcal{O}(\mathcal{L})}$. Moreover, if the formulae associated with Π are non-strict, so are the formulae of $\mathsf{f}(\Pi)$.*

Proof. Let $s \xrightarrow{a} s'$ be a transition in $\mathcal{M}$, and let ∂, ∂' be window strategies for s and s', encoded as points in $\mathbb{R}^{\mathcal{X}_s}$ and $\mathbb{R}^{\mathcal{X}_{s'}}$, respectively. If $\rho \in \mathsf{FPaths}^{<\mathcal{L}}_{\mathcal{M}}(s')$, then let $\mathrm{POLY}(\rho)$ denote the polynomial $\prod_{0 \leq i < |\rho|} x_{\rho[:i],a_i} \mathbb{P}(s_i, a_i, s_{i+1})$. Then, the strategies ∂ and ∂' are compatible w.r.t. $s \xrightarrow{a} s'$ if for all $\rho \in \mathsf{FPaths}^{<\mathcal{L}-1}_{\mathcal{M}}(s')$ so that $\mathrm{POLY}(\rho) > 0$, for all $a' \in A$ we have $x_{sa\cdot\rho,a'} = x_{\rho,a'}$. Then, if $\Pi_{s'}$ is encoded as the formula $R^{s'}$, we get that ∂ and $\Pi_{s'}$ are compatible w.r.t. $s \xrightarrow{a} s'$ if there exists a valuation of $\mathcal{X}_{s'}$ that encodes a window strategy ∂' so that ∂ and ∂' are compatible w.r.t. $s \xrightarrow{a} s'$. This property corresponds to the formula defined by

$$\mathcal{F}(s,a,s') := \exists \mathcal{X}_{s'}, R^{s'} \wedge \bigwedge_{\rho \in \mathsf{FPaths}^{<\mathcal{L}-1}_{\mathcal{M}}(s')} \mathrm{POLY}(\rho) = 0 \vee \bigwedge_{a' \in A} x_{sa\cdot\rho,a'} = x_{\rho,a'}$$

Therefore, if Π_s is encoded as R^s then the formula $R^s \wedge \bigwedge_{s \xrightarrow{a} s'} x_{s,a} = 0 \vee \mathcal{F}(s,a,s')$ encodes $\mathsf{f}(\Pi)_s$. Observe that it introduces non-strict comparisons, existential quantifiers, and no negation operations, and is of size in $F + |\mathcal{M}|(F + |\mathcal{M}|^{\mathcal{O}(\mathcal{L})})$. ◀

▶ **Example 20.** Consider again the MDP $\mathcal{M}$ to the right of Figure 1. Let Π be the portfolio where Π_{s_0} is defined by $x_{s_0,a}\ x_{s_0as_1,b} \geq \frac{1}{4} \wedge x_{s_0as_1,b} \leq c$ with $c \in \left[\frac{1}{2}, 1\right]$, Π_{s_1} is defined by $x_{s_1,a}\ x_{s_1as_0,b} \geq \frac{1}{4} \wedge x_{s_1as_0,b} \leq c$, and Π is $\top$ on every other state of $\mathcal{M}$.[7] Then, using Lemma 19 yields that a formula equivalent to $x_{s_0,a}\ x_{s_0as_1,b} \geq \frac{1}{4} \wedge x_{s_0as_1,b} \leq 1 - \frac{1}{4c}$ encodes $\mathsf{f}(\Pi)_{s_0}$. Symmetrically, $\mathsf{f}(\Pi)_{s_1}$ can be encoded as $x_{s_1,a}\ x_{s_1as_0,b} \geq \frac{1}{4} \wedge x_{s_1as_0,b} \leq 1 - \frac{1}{4c}$.

▶ **Lemma 21** (Knaster-Tarski, Kleene). *The operator f is Scott-continuous (upwards and downwards), and is thus monotone. Let Q be a set of window strategy portfolios of horizon $\mathcal{L}$ that forms a complete lattice w.r.t. $\subseteq$. Then, the set of fixed points of f in Q forms a complete lattice w.r.t. $\subseteq$. Moreover, f has a greatest fixed point in Q equal to $\bigsqcap\{\mathsf{f}^n(\bigsqcup Q) \mid n \in \mathbb{N}\}$.*

Let Φ be a window L-PCTL formula of window length $\mathcal{L}$. Let $Q^\Phi = \{\Pi \mid \forall s \in S, \forall \partial \in \Pi_s, s \models_\partial \Phi\}$ be the set of portfolios containing window strategies of horizon $\mathcal{L}$ that ensure Φ. It is closed by $\bigsqcap$ and $\bigsqcup$, and therefore forms a complete lattice. The greatest element $\bigsqcup Q^\Phi$ is the full portfolio mapping every s to all window strategies ∂ so that $s \models_\partial \Phi$. We denote Π^Φ the greatest fixed point of f in Q^Φ, that must exist by Lemma 21.

▶ **Proposition 22.** *Let s_0 be a state, and let Φ be a window L-PCTL formula. Then, $\Pi^\Phi_{s_0} \neq \emptyset$ if and only if there exists a strategy σ so that $s_0 \models_\sigma \mathrm{A\,G}\,\Phi$.*

Proof sketch. On the one hand, we show that if σ is a strategy so that $s_0 \models_\sigma \mathrm{A\,G}\,\Phi$, and if $\Pi^{s_0,\sigma}$ is the set of window strategies obtained for state s and horizon $\mathcal{L}$ from fixed prefixes of non-zero probability in σ, then $\Pi^{s_0,\sigma}$ is a fixed point of f in Q^Φ so that $\Pi^{s_0,\sigma}_{s_0} \neq \emptyset$. On the other hand, we show that from every fixed point Π of f in Q^Φ that is non-empty on a state s_0, we can inductively construct a strategy σ so that $s_0 \models_\sigma \mathrm{A\,G}\,\Phi$, that intuitively consists in picking successive window strategies from Π that are compatible with each other. ◀

[7] We omitted the constraints that ensure that all variables encode probabilities.

Therefore, computing Π^Φ solves the synthesis problem for global window L-PCTL formulae. By Lemma 21, we have that Π^Φ is the limit of the non-increasing sequence $(\mathsf{f}^i(\bigsqcup Q^\Phi))_{i\in\mathbb{N}}$, with $\bigsqcup Q^\Phi$ being the full portfolio that can be obtained as an $\exists$-$\mathbb{R}$ formula by Proposition 16, so that $\mathsf{f}^i(\bigsqcup Q^\Phi)$ is computable by Lemma 19 as an $\exists$-$\mathbb{R}$ formula of size in $|\Phi||\mathcal{M}|^{\mathcal{O}(\mathcal{L}+i)}$.

▶ **Example 23.** Let $\mathcal{M}$ be the MDP to the right of Figure 1, and $\Phi = \mathbb{P}\left[\mathrm{F}^2\, s_2\right] \geq \frac{9}{16}$. As detailed in Example 18, the set of strategies $(\bigsqcup Q^\Phi)_{s_0}$ is described by $x_{s_0,a}x_{s_0as_1,b} \geq \frac{1}{4}$, the set of strategies $(\bigsqcup Q^\Phi)_{s_1}$ is described by $x_{s_1,a}x_{s_1as_0,b} \geq \frac{1}{4}$, and the set $\bigsqcup Q^\Phi$ is described by $\top$ on all other states.[8] By Example 20, $\mathsf{f}^i(\bigsqcup Q^\Phi)_{s_0}$ is described by $x_{s_0,a}x_{s_0as_1,b} \geq \frac{1}{4} \wedge x_{s_0as_1,b} \leq c_i$, where the constant c_i is defined by $c_0 = 1$ and $c_{i+1} = 1 - \frac{1}{4c_i}$. Similarly, $\mathsf{f}^i(\bigsqcup Q^\Phi)_{s_1}$ is described by $x_{s_1,a}x_{s_1as_0,b} \geq \frac{1}{4} \wedge x_{s_1as_0,b} \leq c_i$, and $\mathsf{f}^i(\bigsqcup Q^\Phi)$ is $\top$ on all other states. The sequence $(c_i)_{i\in\mathbb{N}}$ is a decreasing sequence that converges towards $\frac{1}{2}$ (but never reaches it).

The limit of this sequence is the greatest fixpoint $\Pi^\Phi_{s_0}$, described by $x_{s_0,a}x_{s_0as_1,b} \geq \frac{1}{4} \wedge x_{s_0as_1,b} \leq \frac{1}{2}$ on s_0, $x_{s_1,a}x_{s_1as_0,b} \geq \frac{1}{4} \wedge x_{s_1as_0,b} \leq \frac{1}{2}$ on s_1, $\top$ everywhere else. If we follow the proof of Proposition 22, we can recover the only choice on s_0 and s_1 that ensures $\mathrm{A\,G}\,\Phi$: play a and b with probability $\frac{1}{2}$.

We note that this fixed point computation is not an algorithm: as we have seen in Example 23 the fixed point may not be reachable in finitely many steps. In this case, we do not know if the limit will be empty or not. Nonetheless, this characterisation yields multiple corollary results, that we detail in the remainder of this section.

Flat, non-strict formulae. If Φ is flat and non-strict then $\mathsf{f}^i(\bigsqcup Q^\Phi)$ maps every state to a compact set.[9] The limit of an infinite decreasing sequence of non-empty compact sets in $\mathbb{R}^{\mathcal{X}_s}$ is non-empty. Therefore, if the limit of a decreasing sequence of compact sets is the empty set, it must be reached after finitely many steps. Thus, if $\Pi^\Phi_s = \emptyset$, then there exists $i \in \mathbb{N}$ so that $\mathsf{f}^i(\bigsqcup Q^\Phi)_s = \emptyset$.

▶ **Theorem 24.** *The synthesis problem for flat, non-strict* global window *L-PCTL formulae is in* coRE.

As we will detail in Section 4, the synthesis problem for flat, non-strict global window formulae is undecidable (coRE-hard), and therefore coRE-complete.

▶ Remark 25. From the proof of Proposition 16, it follows that if Φ is non-flat, that is, it contains nested probabilistic operators, then the set of window strategies ∂ such that $s \models_\partial \Phi$ may not be closed and hence $\bigsqcup Q^\Phi$ is not necessarily a compact set.

▶ Remark 26. Note that in Example 23 we were able to compute by hand the limit of the sequence of $\exists$-$\mathbb{R}$ formulae describing $\mathsf{f}^i(\bigsqcup Q^\Phi)$, and obtained an $\exists$-$\mathbb{R}$ formula for the greatest fixed point Π^Φ_s. This is not always possible: there exists an MDP $\mathcal{M}$ and a flat, non-strict global window formula $\mathrm{A\,G}\,\Phi$ so that Π^Φ_s cannot be expressed in FO-$\mathbb{R}$. Indeed, FO-$\mathbb{R}$ formulae can be seen as finite words over a countable alphabet, so that there are countably many of them. If by contradiction Π^Φ_s was always expressible in FO-$\mathbb{R}$, we could enumerate all FO-$\mathbb{R}$ formulae and check for each of them if it describes a fixed-point of f where s_{init} is mapped to a non-empty set, two properties also expressible in FO-$\mathbb{R}$ by using Lemma 19. This would show that the synthesis problem is recursively enumerable, therefore in RE$\cap$coRE *i.e.* decidable, which is absurd as it is coRE-complete as we will see in Section 4.

[8] Once again, we omit the constraints that ensure that all variables encode probabilities.
[9] Non-strict formulae describe closed sets, and all variables are in $[0, 1]$ as they encode probabilities.

Deterministic strategies. In this paragraph, we study the synthesis problem for deterministic strategies. First, note that the window strategy defined by a deterministic strategy for a given prefix and horizon is also deterministic. Conversely, if ∂ is a deterministic window strategy then there exists a deterministic strategy in its cylinder. Therefore, Lemma 14 carries over, and finding a deterministic strategy satisfying a window formula reduces to finding a deterministic window strategy for it. Then, note that for a fixed state s, each deterministic window strategy can be seen as a boolean assignment over the set $\mathcal{X}_s$ of variables, and we have that $|\mathcal{X}_s| = |\mathcal{M}|^{\mathcal{O}(\mathcal{L})}$. Therefore, the set of deterministic window strategies is finite, of doubly-exponential size $2^{|\mathcal{M}|^{\mathcal{O}(\mathcal{L})}}$. We denote by W the number of deterministic window strategies. By guessing a window strategy and verifying it in EXPTIME, we get a NEXPTIME upper bound on the synthesis problem for window formulae. By guessing a strategy in an online manner we can lower this complexity, and show that the problem is in fact PSPACE-complete.

▶ **Proposition 27.** *The synthesis problem for* window *L-PCTL formulae is* PSPACE*-complete when restricted to* deterministic *strategies.*

Proof. We present a non-deterministic algorithm, running in polynomial space, that accepts all positive instances of the synthesis problem with deterministic strategies. We will guess a deterministic window strategy ∂ and check that Φ holds on the resulting MC. In order to avoid guessing an exponential certificate (the entire strategy ∂), we will perform a depth-first search (DFS) traversal of the MDP, starting from s_{init} and of horizon $\mathcal{L}$, where we guess every decision of ∂ in an online manner ($\partial(\rho)$ is guessed when the search path, that is the path from s_{init} to the current state, is ρ for the first time). We will compute along the way information that ultimately lets us evaluate if Φ holds on the root node of the search. At any point in the DFS, when the current path traversed from s_{init} is ρ, this information represents partial evaluations of subformulae of Φ on states along ρ, according to the strategy ∂. Formally, we equip each state in the DFS with a set of formulae to be evaluated. For each path formula φ in the set, we store the probability of satisfying the formula according to paths previously visited by the DFS. Once all of the subtree below a state has been seen by the search, this value matches the probability $\mathbb{P}[\varphi]$, and we can then use this value to evaluate the state formulae that needs to know $\mathbb{P}[\varphi]$ on the current state. In order to define the sets of subformulae to evaluate, we can use the same induction rules as in the proof of Proposition 16, that reduce the evaluation of a formula such as $\mathrm{X}^{\ell}\,\Phi_1$ or $\Phi_1\,\mathrm{U}^{\ell}\,\Phi_2$ on a given state to the evaluation of Φ_1, Φ_2, $\mathrm{X}^{\ell-1}\,\Phi_1$ or $\Phi_1\,\mathrm{U}^{\ell-1}\,\Phi_2$ on the current or the next state. Overall, we need to remember a path of length at most $\mathcal{L}$, a set of subformulae of Φ on each state in this path, and a probability for each such path formula. Assuming that the probabilities can be stored in polynomial space, this is indeed a PSPACE algorithm.

We argue that these probabilities can always be stored in polynomial space. Indeed, they correspond to the measure, in the MC defined by ∂, of a finite union of cylinders defined by prefixes of length at most $\mathcal{L}$. Since ∂ is deterministic, these measures are finite sums of real numbers obtained as the product of at most $\mathcal{L}$ constants appearing as transition probabilities on $\mathcal{M}$. Using a standard binary representation of rational numbers as irreducible fractions, we get that since $\mathcal{L}$ is polynomial in $|\Phi|$ these probabilities are always rational and of polynomial size.

We show a reduction from the synthesis problem with a generalized reachability objective in a two-player game. Given an arena with a set V of vertices that are partitioned into vertices belonging to Player 1 and Player 2, given an initial vertex v_0, and reachability sets $F_1, \ldots, F_k$, the problem asks for a (deterministic) Player 1 strategy that ensures reaching each of the sets against any Player 2 strategies. The generalized reachability problem is PSPACE-complete [12].

We construct an MDP $\mathcal{M}$ with $Q = V$ set of states and transitions which are the same as the edges of the two-player game arena. A Player 1 vertex corresponds to a state in the MDP such that for every outgoing edge (v, v_i) from v, we have an action a_i labelling the transition (v, v_i) in $\mathcal{M}$. For a Player 2 vertex v, all the outgoing edges (v, v_i) correspond to transitions for the same action to vertices v_i with equal probability. Also a state $v \in Q$ in $\mathcal{M}$ is labelled x_i for $i \leq i \leq k$ if and only if the corresponding vertex $v \in F_i$ in the two-player game. Further, if Player 1 has a wining strategy in the generalized reachability game, then she can visit all the reachability sets within a total of nk steps with a deterministic strategy.

Now consider the property Φ defined as $\bigwedge_{i=1}^{k} \mathbb{P}\left[\mathrm{F}^{nk}\, x_i\right] = 1$. There exists a deterministic strategy from v_0 in $\mathcal{M}$ satisfying Φ if and only if Player 1 has a winning strategy for the generalized reachability objective. ◀

As there are finitely many deterministic window strategies of horizon $\mathcal{L}$, the fixed point computation always terminates and thus provides decidability. We also reduce the problem asking if an alternating Turing machine running in polynomial space accepts a given word to deterministic strategy synthesis.

▶ **Proposition 28.** *The synthesis problem for* global window *L-PCTL formulae is in* 2EXPTIME *when restricted to* deterministic *strategies. Moreover, it is* EXPTIME*-hard.*

Proof. For global window formulae, we need to change the set Q^Φ, defined in Section 3, to only contain deterministic window strategies. It is still a complete lattice, and Proposition 22 carries over for deterministic strategies. Moreover, for every portfolio Π, there are finitely many strictly smaller portfolios, at most $|\mathcal{M}|W$. As the sequence $(\mathsf{f}^i(\bigsqcup Q^\Phi))_{i\in\mathbb{N}}$ is non-increasing, the fixed point is reached in at most $|\mathcal{M}|W$ steps. Each step can be performed without relying on the theory of the reals, by representing the window strategies explicitly as trees of depth $\mathcal{L}$. Applying the operator f on a portfolio amounts to checking if a tree is a prefix of another. Overall, the fixed point computation is doubly-exponential.

Note that if we rely on computing $\exists$-$\mathbb{R}$ formulae for $(\mathsf{f}^i(\bigsqcup Q^\Phi))_{i\in\mathbb{N}}$ instead of these explicit sets of strategies, the formulae could a priori grow to sizes in $|\Phi||\mathcal{M}|^{\mathcal{O}(\mathcal{L}+|\mathcal{M}|W)}$, so that we end up with a triply-exponential upper bound.

For the EXPTIME-hardness, we use APSPACE = EXPTIME and present a polynomial reduction from alternating polynomial-space Turing machines. We consider a Turing machine of states Q and tape alphabet Σ, so that each state q is equipped with a label $L(q) \in \{\forall, \exists\}$, except for the accepting and rejecting states $q_\top$ and $q_\bot$, where $L(q) = q$. Let $w \in \Sigma^*$ be an input word, and let $n \in \mathbb{N}$ be a bound on the length of the tape used when running w on the machine. Since we considered a polynomial space machine, n is polynomial. W.l.o.g., we assume that for every input, the Turing machine we consider above halts, and the input is accepted if and only if it halts in $q_\top$.

Let $\mathcal{M}$ be the MDP from Figure 2, where each named state is assigned an identical label. We describe a window formula Φ that ensures that controller only makes choices that faithfully represent an execution of the alternating Turing machine. Let $N = 2n + 5$ be the number of steps needed to follow a cycle from s to s in $\mathcal{M}$. If l is a label in $\mathcal{M}$, we shorten the L-PCTL formula $\mathbb{P}\left[\mathrm{F}^N\, l\right] = 1$ as $\mathrm{F}_1^N\, l$. It means that every path of length N must reach l. Intuitively, a run of the Turing machine can be described as a path in this MDP, where one full loop around s describes a configuration of the Turing machine: visiting a_i means that cell number i contains a, visiting a_i^H means that additionally the reading head is in position i, and entering the gadget q^a means that we are in state q and will read an a. Either controller or the environment gets to pick the next transition, and then we go back to s.

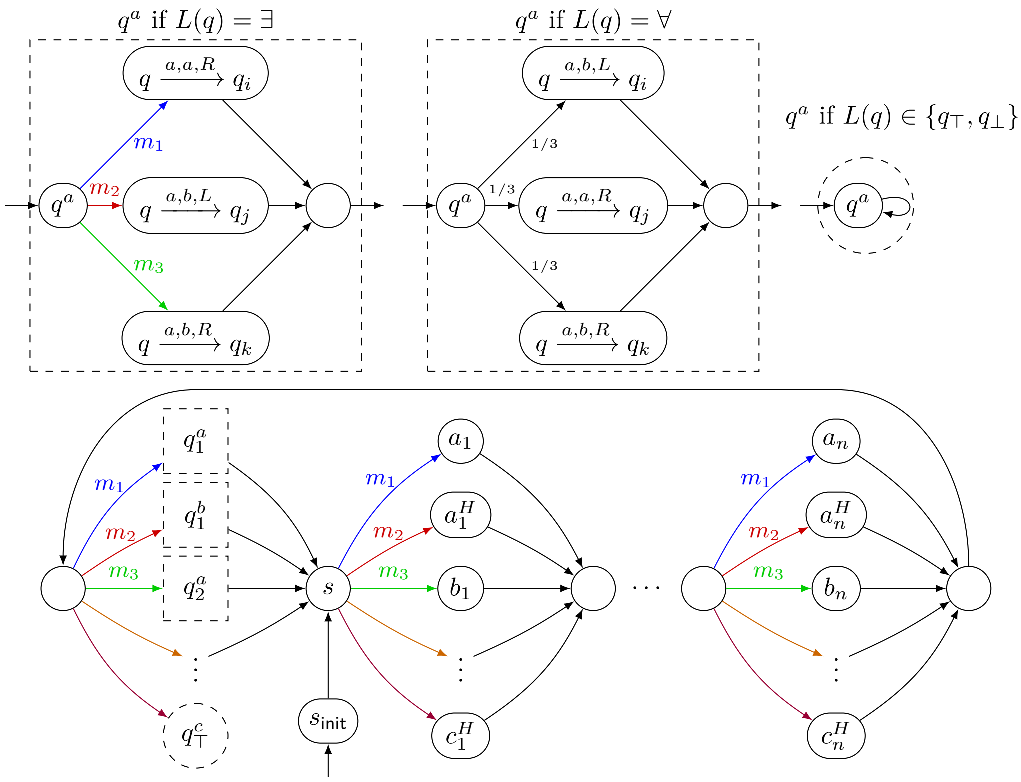


Figure 2 The MDP used in the reduction from APSPACE to deterministic synthesis for global window formulae. The colored transitions m_1, m_2, ... represent different actions, black transitions are available for every action, and the probability of a transition is 1 if unspecified. The letters a, b, c enumerate the tape alphabet Σ, while q_1, q_2, ... enumerate the states in Q, with $q_\top$ the accepting state. The only randomized transition is in the universal gadget q^a, and uses a uniform distribution over reachable states.

The formula Φ is obtained as the conjunction of the following constraints:

In order to ensure that the tape is initialized appropriately, we ask for every letter $a \in \Sigma$ in position $i > 1$ in the input word w that $s_{\mathsf{init}} \Rightarrow \mathrm{F}_1^N\, a_i$. If the first letter in w is a and the initial state of the Turing machine is q_1, we also ask $s_{\mathsf{init}} \Rightarrow \mathrm{F}_1^N\, a_1^H \wedge \mathrm{F}_1^N\, q_1^a$.

In order to simulate the transitions on the tape cells correctly, we ask for every $1 \leq i < n$, $a, c \in \Sigma$ and transition $q \xrightarrow{a,b,L} q'$ that

$$(s \wedge \mathrm{F}_1^N\, c_i \wedge \mathrm{F}_1^N\, a_{i+1}^H) \Rightarrow \mathbb{P}\Big[\mathrm{X}^{N-2}((q \xrightarrow{a,b,L} q') \Rightarrow \mathrm{F}_1^N\, c_i^H \wedge \mathrm{F}_1^N\, b_{i+1})\Big] = 1\,.$$

We similarly ask for every $1 \leq i < n$, $a, c \in \Sigma$ and transition $q \xrightarrow{a,b,R} q'$ that

$$(s \wedge \mathrm{F}_1^N\, a_i^H \wedge \mathrm{F}_1^N\, c_{i+1}) \Rightarrow \mathbb{P}\Big[\mathrm{X}^{N-2}((q \xrightarrow{a,b,R} q') \Rightarrow \mathrm{F}_1^N\, b_i \wedge \mathrm{F}_1^N\, c_{i+1}^H)\Big] = 1\,.$$

The other tape cells should be left untouched, so that for every $1 \leq i < n$, $a, b \in \Sigma$ and transition $q \xrightarrow{c,d,L} q'$, we ask that

$$(s \wedge \mathrm{F}_1^N\, a_i \wedge \mathrm{F}_1^N\, b_{i+1}) \Rightarrow \mathbb{P}\Big[\mathrm{X}^{N-2}((q \xrightarrow{c,d,L} q') \Rightarrow \mathrm{F}_1^N\, a_i)\Big] = 1\,.$$

Similarly for every transition transition $q \xrightarrow{c,d,R} q'$, we ask that

$$(s \wedge \mathrm{F}_1^N a_i \wedge \mathrm{F}_1^N b_{i+1}) \Rightarrow \mathbb{P}\Big[\mathrm{X}^{N-2}((q \xrightarrow{c,d,R} q') \Rightarrow \mathrm{F}_1^N b_{i+1})\Big] = 1\,.$$

Finally, in order to update the state, we ask for every transition $q \xrightarrow{a,b,D} q'$ with $D \in \{L, R\}$ and every $1 \leq i \leq n$, $c \in \Sigma$ that $(q \xrightarrow{a,b,D} q') \wedge \mathrm{F}_1^N c_i^H \Rightarrow \mathrm{F}_1^N q'^c$.

Then, we let $q_\top$ and $q_\bot$ be the labels that hold on all states $q_\top^a$ and $q_\bot^a$, respectively, for all $a \in \Sigma$. We consider the global window formula $\mathrm{A}\,\mathrm{G}[(\Phi \vee q_\top) \wedge \neg q_\bot]$, and show that there is a winning strategy for this formula in $\mathcal{M}$ if and only if the alternating Turing machine accepts the input word w. Indeed, an alternating Turing machine equipped with an initial word can be seen as a turn-based two-player zero-sum reachability game played on the execution tree of the machine, where we ask if there exists a strategy for player $\exists$ that ensures against every strategy of $\forall$ the state $q_\top$ is reached. ◀

Memoryless strategies. We study the synthesis of memoryless strategies. The window strategy defined by a memoryless strategy for a given prefix and a horizon is also memoryless. Conversely, a memoryless window strategy has a memoryless strategy in its cylinders. By Lemma 14, finding a memoryless strategy satisfying a window formula reduces to finding a memoryless window strategy for it. Let s be a state. As usual, a window strategy ∂ for state s can be seen as an assignment in $[0, 1]$ for variables $\mathcal{X}_s$. However, the memoryless property asks that $\partial(\rho) = \partial(\rho')$ for all ρ, ρ' that share the same last state s', or equivalently $x_{\rho,a} = x_{\mathsf{last}(\rho),a}$ for all ρ. Thus, we can replace every instance of $x_{\rho,a}$ by $x_{\mathsf{last}(\rho),a}$ in the $\exists$-$\mathbb{R}$ formula of Proposition 16, so that the set of free variables used to represent a memoryless window strategy for s is $\mathcal{X}_s = \{x_{s',a} \mid \exists\rho \in \mathsf{FPaths}_{\mathcal{M}}^{<\mathcal{L}}(s), s' = \mathsf{last}(\rho)\}$. Similarly, the variables $y_{\rho,\Phi}$ and $z_{\rho,\varphi}$ can be replaced by by $y_{\mathsf{last}(\rho),\Phi}$ and $z_{\mathsf{last}(\rho),\varphi}$ respectively, as the satisfaction of a state formula, or the probability of satisfying a path formula, only depend on the current state. The formula is now of polynomial size, so that we obtain as a corollary:

▶ **Proposition 29.** *The synthesis problem for* window *L-PCTL formulae is in* PSPACE *when restricted to* memoryless *strategies.*

Further, following a reduction in [15], it can be shown that the MR synthesis problem for window L-PCTL objectives is at least as hard as the Square-Root-Sum problem which is known to be in PSPACE, but whose exact complexity is a longstanding open problem.

We now study the memoryless synthesis problem for global window formulae. For each state s, let R^s denote the $\exists$-$\mathbb{R}$ formula encoding the window formula Φ for state s, as per Proposition 29. The free variables are the variables in $\mathcal{X}_s \subseteq \mathcal{X} = \{x_{s',a} \mid s' \in S, a \in A\}$. A memoryless strategy σ can be seen as a point in $\mathbb{R}^{\mathcal{X}}$, so that $\sigma(s, a)$ is assigned to $x_{s,a}$. For all states s and s', we define a variable $r_{s,s'} \in \{0, 1\}$ quantified existentially, and construct a formula ensuring that if $r_{s,s'} = 0$ then s' is not reachable from s under the strategy σ. This formula states $r_{s,s} = 1$ for all states s, and asks that the variables r are a solution to the system of equations asking, for all s, s', s'' and a, that if $r_{s,s'} = 1$ and $x_{s',a}\mathbb{P}(s', a, s'') > 0$ then $r_{s,s''} = 1$. Therefore, the set of states s' so that $r_{s,s'} = 1$ is an over-approximation.[10]

Then, the formula asking that there exists a value for each variable r so that $R^{s'}$ holds whenever $r_{s,s'} = 1$ represents the memoryless strategies that satisfy Φ on an over-approximation of the states reachable from s, which is equivalent to satisfying $\mathrm{A}\,\mathrm{G}\,\Phi$ when

[10] For example, the formula is satisfied if $r_{s,s'}$ is 1 for all s, s', which represents an over-approximation of the set of states reachable from s where every state is reachable.

starting from state s. Note that since the variables $r_{s,s'}$ are existentially quantified, and Φ is only required to be satisfied on states reachable from s, then there always exists a valuation for these r variables that sets $r_{s,s'}$ to 1 if and only if s' is reachable from s. It follows that:

▶ **Proposition 30.** *The synthesis problem for* global window *L-PCTL formulae is in* PSPACE *when restricted to* memoryless *strategies.*

PCTL satisfiability. We now consider the satisfiability problem, that asks, given a formula Φ, if there a exists an MC M so that $M \models \Phi$. This is a longstanding open problem for PCTL formulae. One can also consider variants of the problem, that either restrict Φ to a sublogic of PCTL or limit M to MCs that belong to a particular set, such as finite MCs or MCs where all probabilities are rational numbers. The decidability of these variants is also open and, as noted in [7], some PCTL formulae are only satisfiable by infinite MCs. In particular, we say that an MC M has granularity bounded by $N \in \mathbb{N}$ if every probability in the transition function $\mathbb{P}$ is equal to a rational $\frac{a}{b}$ with $b \leq N$. The *bounded granularity satisfiability problem* asks, given Φ and N, if there exists an MC of granularity bounded by N that satisfies Φ. The bounded granularity satisfiability problem for global window L-PCTL formulae can be reduced to the HD strategy synthesis problem for global window L-PCTL formulae. Therefore, we obtain the following result as a corollary of Proposition 28:

▶ **Theorem 31.** *The bounded granularity satisfiability problem for global window L-PCTL formulae is decidable in complexity doubly-exponential in $|\Phi|$ and N. Moreover, finite MCs are sufficient, in the sense that for every formula $\mathrm{A}\,\mathrm{G}\,\Phi$ that admits a model M of granularity bounded by N, there exists a finite MC M' of granularity bounded by N so that $M' \models \mathrm{A}\,\mathrm{G}\,\Phi$.*

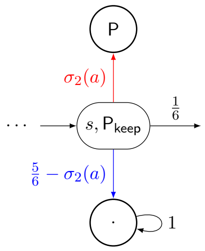
Proof sketch. Given a formula with atomic propositions AP, and a granularity bound N, we intuitively consider an MC of states 2^{AP} with an action for every distribution over 2^{AP} whose granularity is bounded by N, so that this action describes the next states and their probabilities. Then, every MC of granularity bounded by N can be seen as a deterministic strategy in this MDP, so that strategy synthesis and MC satisfiability are equivalent. We can then apply Proposition 28. Moreover, finite MCs are sufficient as finite-memory strategies are sufficient for global window PCTL when restricted to deterministic strategies. ◀

4 Undecidability

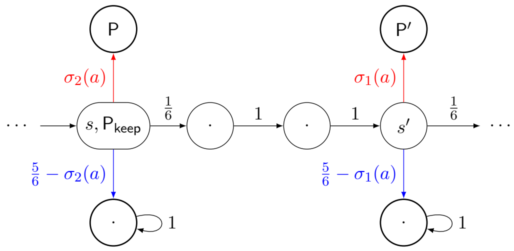
In Section 3, we have shown that the synthesis problem for flat, non-strict global window L-PCTL formulae is in coRE. In this section, we argue that it is coRE-hard and that it becomes Σ_1^1-hard when relaxing the hypothesis that the formulae considered are non-strict.

When considering flat non-strict formulae, we proceed via a reduction from the non-halting problem of a two-counter Minsky machine. A two-counter Minsky machine consists of a list of instructions $l_1 : \mathsf{ins}_1, \ldots, l_n : \mathsf{ins}_n$ and two counters c_2 and c_3 (the indices 2 and 3 are chosen to ease the notations) where, for all $i \leq n$, we have ins_i an instruction in one the following types, for $j \in \{2, 3\}$ and $1 \leq k, m \leq n$: $\mathsf{Inc}_j(k)$: $c_j := c_j + 1$; goto k; $\mathsf{Branch}_j(k, m)$: if $c_j = 0$ then goto k; else $c_j := c_j - 1$, goto m; H: halt. The semantics of these instructions is straightforward. The non-halting problem for Minsky machine, denoted MinskyNotStop, is to decide, given a machine Msk, if its execution is infinite. This problem is undecidable, as stated in the theorem below.

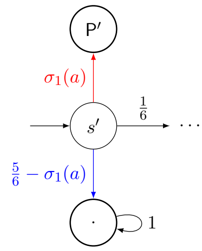
▶ **Theorem 32** ([17]). *MinskyNotStop is* coRE*-complete.*



■ **Figure 3** The end of a gadget.



■ **Figure 4** The end of a gadget on the left, and the beginning of another one on the right.



■ **Figure 5** The beginning of a gadget.

Given $\mathsf{Msk} = l_1 : \mathsf{ins}_1, \ldots, l_n : \mathsf{ins}_n$ on two counters c_2 and c_3, we build an MDP $\mathcal{M}$ and an L-PCTL formula Φ such that there exists a strategy σ for $\mathcal{M}$ s.t. $\mathcal{M}[\sigma] \models \Phi$ if and only if $\mathsf{Msk} \in \mathsf{MinskyNotStop}$. The crucial point of the reduction is to encode the values of the counters that may take unbounded values. It is done in $\mathcal{M}$ by encoding these values in the probability (chosen by the strategy σ) to see a given predicate in the next few steps. More specifically, in the situation where the counters are such that $\{c_2 \mapsto x_2; c_3 \mapsto x_3\}$, we consider the probability $p(x_2, x_3) = \frac{5}{6} \times \frac{1}{2^{x_2}} \times \frac{1}{3^{x_3}}$. We then associate to each different instruction a gadget, i.e. an MDP, and a formula encoding the update of probability $p(x_2, x_3)$ according to how the counters are changed by the corresponding instruction. Inside a gadget, one can find predicates of the shape $(\mathsf{P}.)$. They are used to define the formulae specifying the expected behavior of the strategy. Furthermore, there is also an entering and an exiting probability which correspond to the encoding of the counters respectively before and after the effect of the instructions. We define below formally the notion of well-placed gadgets.

▶ **Definition 33** (Gadgets)**.** *A gadget* Gd *is an MDP with an* entering probability *and an* exiting probability*. Consider Figure 3 that represents how every gadget* Gd *ends. The exiting probability* $p^{\mathsf{ex}}_{\mathsf{Gd}}$ *is the probability* $\sigma_2(a)$ *to visit the state on the top. It is equal to* $p^{\mathsf{ex}}_{\mathsf{Gd}} = \mathbb{P}_s(\mathrm{F}^1\,\mathsf{P})$, i.e. *the probability that* $\mathrm{F}^1\,\mathsf{P}$ *holds on state* s*. Consider Figure 5. All gadgets begin as in this figure: a state* s' *with a successor satisfying the predicate* P'*. The* entering probability $p^{\mathsf{en}}_{\mathsf{Gd}}$ *is the probability* $\sigma_1(a)$ *to see* P'*, that is:* $p^{\mathsf{en}}_{\mathsf{Gd}} = \mathbb{P}_{s'}(\mathrm{F}^1\,\mathsf{P}')$*. A gadget is* well-placed *if, as for the gadget on the right of Figure 4, it is preceded by two dummy states, themselves immediately preceded by a gadget.*

Before looking at how specific instructions are encoded in the counters and the formula, we have to ensure that the exiting probability of a gadget is equal to the entering probability of the following well-placed gadget. This is done with the formula: $\Phi_{\mathsf{keep}} := \mathsf{P}_{\mathsf{keep}} \Rightarrow (\mathbb{P}(\mathrm{F}^1\,\mathsf{P}) = 6 \cdot \mathbb{P}(\mathrm{F}^4\,\mathsf{P}'))$. These definitions ensure the following proposition:

▶ **Proposition 34** (Entering probability of a well-placed gadget)**.** *Assume that a well-placed gadget* Gd' *follows a gadget* Gd*. Then, for a strategy* σ *s.t. the formula* $\mathrm{A}\,\mathrm{G}\,\Phi_{\mathsf{keep}}$ *is satisfied, the exiting probability of gadget* Gd *is equal to the entering probability of gadget* Gd'*:* $p^{\mathsf{ex}}_{\mathsf{Gd}} = p^{\mathsf{en}}_{\mathsf{Gd}'}$*.*

Due to lack of space, we only exhibit the gadgets encoding the increment of a counter and for testing if a counter value is 0. Consider the increment of counter c_2. By definition of $p(x_2, x_3)$, incrementing that counter is simulated by multiplying the probability by $\frac{1}{2}$.We define the gadget Gd_{c_2++} and the formula Φ_{c_2++} ensuring that the probability is indeed multiplied by $\frac{1}{2}$. The gadget Gd_{c_2++} is depicted in Figure 6.In addition, we define the L-PCTL formula Φ_{c_2++} such that $\Phi_{c_2++} := \mathsf{P}_{c_2++} \Rightarrow (\mathbb{P}(\mathrm{F}^2\,\mathsf{P}_{c_2}) = 6 \cdot 2 \cdot \mathbb{P}(\mathrm{F}^3\,\mathsf{P}_{+}))$. The interest of these definitions lies in the proposition below.

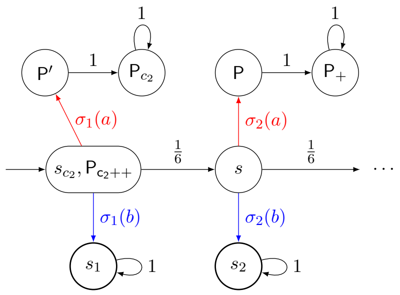


Figure 6 The gadget Gd_{c_2++} for the operation $c_2 := c_2 + 1$.

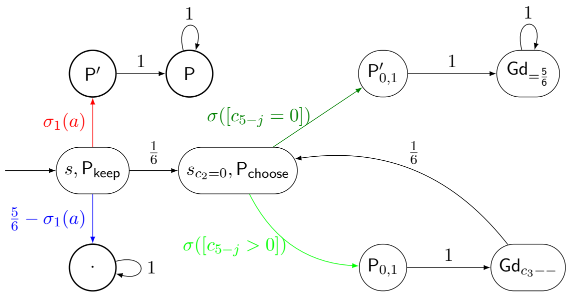


Figure 7 The gadget $\mathsf{Gd}_{c_2=0}$ for testing that the counter $c_2 = 0$.

▶ **Proposition 35** (Incrementing Gadget Specification). *If the entering probability $p^{\mathsf{en}}_{\mathsf{Gd}_{c_2++}}$ of the gadget Gd_{c_2++} is equal to $p(x_2, x_3)$ with $x_2, x_3 \in \mathbb{N}$, then whenever the formula $\mathrm{A}\,\mathrm{G}\,\Phi_{c_2++}$ is satisfied, the exiting probability $p^{\mathsf{ex}}_{\mathsf{Gd}_{c_2++}}$ of this gadget is equal to $p^{\mathsf{ex}}_{\mathsf{Gd}_{c_2++}} = p(x_2 + 1, x_3)$.*

We now consider the gadget that tests if a counter value is 0, let us exemplify it with counter c_2 in gadget $\mathsf{Gd}_{c_2=0}$ depicted in Figure 7. The gadget $\mathsf{Gd}_{=5/6}$ used on the right tests that both counters have value 0 (i.e. the entering probability is equal to 5/6).Then, the idea is as follows: as long as counter c_3 has a positive value, the strategy σ has to take the bottom branch to decrement it and once this counter has reached 0, it can take the top branch to check that the probability is indeed equal to 5/6. Note that one cannot decrement counter c_2 in this gadget, hence if its value is positive, there is no way to pass the test of the comparison to 5/6. To ensure that the choice at state $s_{c_2=0}$ is deterministic, we consider the formula $\Phi_{0,1} := \mathsf{P}_{\mathsf{choose}} \Rightarrow (\mathbb{P}(\mathrm{X}\,\mathsf{P}_{0,1}) = 1 \vee \mathbb{P}(\mathrm{X}\,\mathsf{P}'_{0,1}) = 1)$. We have the proposition below:

▶ **Proposition 36** (Testing Gadget Specification). *Assume that the entering probability $p^{\mathsf{en}}_{\mathsf{Gd}_{c_2=0}}$ of the gadget $\mathsf{Gd}_{c_2=0}$ is equal to $p(x_2, x_3)$ for some $x_2, x_3 \in \mathbb{N}$. Then, there is a strategy σ such that the formula $\mathrm{A}\,\mathrm{G}[\Phi_{\mathsf{keep}} \wedge \Phi_{0,1} \wedge \Phi_{c_3--} \wedge \Phi_{=5/6}]$ is satisfied if and only if $x_2 = 0$.*

We can similarly test that a counter is different from 0. A gadget corresponding to a branching instruction can then be defined by using these gadgets.Finally, a gadget corresponding to the Halt instruction only consists of a gadget where no strategy σ can satisfy the formula considered. Overall, we can combine all these gadgets to encode all the instructions of the Minsky machine Msk.We obtain the theorem below.

▶ **Theorem 37.** *The synthesis problem for flat, non-strict global window L-PCTL formulae is* coRE*-hard.*

When considering arbitrary flat formulae without the non-strict constraint, we can adapt the proof to reduce from the problem asking if there is an execution of a Minsky Machine that visits infinitely often the first instruction [3], so that the strategy synthesis problem becomes highly undecidable.

▶ **Theorem 38.** *The synthesis problem for flat global window L-PCTL formulae is Σ^1_1-hard.*

The construction here is similar to the case with the non-strict constraint, except that whenever the first instruction is seen, a choice is given to the strategy which can set in how many number of steps n the first instruction will be seen again (note that this number may be arbitrarily large). This choice is encoded by resetting a new counter c_5 to value n, which is then decremented each time the first instruction is not seen, and a problem arises if this counter ever reaches 0. In terms of probability, the value $p(x_2, x_3)$ is initially multiplied by $\frac{1}{5^n}$ and then multiplied by 5 each time the first instruction is not seen. Hence, the probabilities chose by σ may be arbitrarily close to 0, but cannot be equal to 0. This is where we need the non-strict comparison with 0.

References

1 E. Ábrahám, E. Bartocci, B. Bonakdarpour, and O. Dobe. Probabilistic hyperproperties with nondeterminism. In *ATVA*, pages 518–534. Springer, 2020.

2 E. Ábrahám and B. Bonakdarpour. Hyperpctl: A temporal logic for probabilistic hyperproperties. In *QEST*, pages 20–35. Springer, 2018.

3 Rajeev Alur and Thomas A. Henzinger. A really temporal logic. *J. ACM*, 41(1):181–204, 1994.

4 R. Andriushchenko, M. Ceska, S. Junges, and J-P Katoen. Inductive synthesis for probabilistic programs reaches new horizons. In *TACAS, Part I*, pages 191–209. Springer, 2021.

5 C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski. Controller synthesis for probabilistic systems. In *Exploring New Frontiers of Theoretical Informatics*, pages 493–506. Springer US, 2004.

6 C. Baier and J-P. Katoen. *Principles of model checking.* MIT Press, 2008.

7 T. Brázdil, V. Forejt, J. Kretínský, and A. Kucera. The satisfiability problem for probabilistic CTL. In *LICS*, pages 391–402. IEEE Computer Society, 2008.

8 John Canny. Some algebraic and geometric computations in pspace. In *STOC*, pages 460–467, 1988.

9 S. Chakraborty and J-P Katoen. On the satisfiability of some simple probabilistic logics. In *LICS*, pages 56–65. Association for Computing Machinery, 2016.

10 M. R. Clarkson, B. Finkbeiner, M. Koleini, K. K. Micinski, M. N. Rabe, and C. Sánchez. Temporal logics for hyperproperties. In *POST*, pages 265–284. Springer, 2014.

11 R. Dimitrova, B. Finkbeiner, and H. Torfah. Probabilistic hyperproperties of markov decision processes. In *ATVA*, pages 484–500. Springer, 2020.

12 N. Fijalkow and F. Horn. The surprizing complexity of reachability games. *CoRR*, abs/1010.2420, 2010. `arXiv:1010.2420`.

13 J. W. Gray III. Toward a mathematical foundation for information flow security. *J. Comput. Secur.*, 1(3-4):255–294, 1992.

14 Jan Kretínský and Alexej Rotar. The satisfiability problem for unbounded fragments of probabilistic CTL. In *CONCUR*, pages 32:1–32:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

15 A. Kucera, V. Forejt, V. Brozek, and T. Brazdil. Stochastic games with branching-time winning objectives. In *LICS*, pages 349–358. IEEE Computer Society, 2006.

16 A. Kučera and O. Stražovský. On the controller synthesis for finite-state markov decision processes. In *FSTTCS*, pages 541–552. Springer Berlin Heidelberg, 2005.

17 Marvin L. Minsky. *Computation: Finite and Infinite Machines.* Prentice-Hall, 1967.

18 J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. part iii: Quantifier elimination. *Journal of Symbolic Computation*, 13(3):329–352, 1992.