Ranking and Repulsing Supermartingales for Reachability in Randomized Programs

TORU TAKISAKA, National Institute of Informatics, Japan

YUICHIRO OYABU, National Institute of Informatics, Japan and The Graduate University for Advanced Studies (SOKENDAI), Japan

NATSUKI URABE*, National Institute of Informatics, Japan and University of Tokyo, Japan

ICHIRO HASUO, National Institute of Informatics, Japan and The Graduate University for Advanced Studies (SOKENDAI), Japan

Computing reachability probabilities is a fundamental problem in the analysis of randomized programs. This article aims at a comprehensive and comparative account of various *martingale-based methods* for over- and under-approximating reachability probabilities. Based on the existing works that stretch across different communities (formal verification, control theory, etc.), we offer a unifying account. In particular, we emphasize the role of order-theoretic fixed points—a classic topic in computer science—in the analysis of randomized programs. This leads us to two new martingale-based techniques, too. We also make an experimental comparison using our implementation of template-based synthesis algorithms for those martingales.

CCS Concepts: • **Software and its engineering** → **Formal software verification**; • **Theory of computation** → *Probabilistic computation*;

Additional Key Words and Phrases: Randomized program, reachability, fixed point, martingale

ACM Reference format:

Toru Takisaka, Yuichiro Oyabu, Natsuki Urabe, and Ichiro Hasuo. 2021. Ranking and Repulsing Supermartingales for Reachability in Randomized Programs. *ACM Trans. Program. Lang. Syst.* 43, 2, Article 5 (June 2021), 46 pages.

<https://doi.org/10.1145/3450967>

The present article is an extended version of Reference [39], which has been presented at the *International Symposium on Automated Technology for Verification and Analysis (ATVA 2018)*, October 7–10, 2018, Los Angeles, CA.

*Current address: Max Planck Institute for Software Systems, Paul-Ehrlich Strasse G 26, 67663, Kaiserslautern, Germany; email: urabenatsuki@mpi-sws.org.

The authors are supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), the JSPS-INRIA Bilateral Joint Research Project “CRECOGI,” and JSPS KAKENHI Grant Numbers 15KT0012 & 15K11984. Natsuki Urabe is supported by JSPS KAKENHI Grant Number 16J08157.

Authors’ addresses: T. Takisaka, National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan; email: takisaka@nii.ac.jp; Y. Oyabu and I. Hasuo, National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan, The Graduate University for Advanced Studies (SOKENDAI), Shonan Village, Hayama, Kanagawa, 240-0193, Japan; emails: oyabu82@gmail.com, hasuo@nii.ac.jp; N. Urabe, National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan, University of Tokyo, Room 615, Faculty of Science Building 7, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2021 Copyright held by the owner/author(s).

0164-0925/2021/06-ART5

<https://doi.org/10.1145/3450967>

1 INTRODUCTION

Computing *reachability probabilities* is a fundamental problem in the analysis of probabilistic systems. It is known that probabilistic model checking problems can be solved via reachability probabilities [7], much like nondeterministic model checking problems are reduced to emptiness and hence to reachability [42]. While the computation of reachability probabilities for finite-state systems is effectively solved by linear programming, the problem becomes much more challenging for *probabilistic programs*—a paradigm that attracts growing attention as a programming language foundation for machine learning [19]—because their transition graphs are infinite in general.

There are two features that are unique to probabilistic programs: *randomization* and *conditioning*. Our focus is on what we call *randomized programs*, a fragment of probabilistic programs without conditioning. Reachability probabilities of randomized programs with while loops are clearly not computable, because the problem encompasses termination of (non-probabilistic) while programs. Therefore the existing research efforts have focused on sound *approximation methods* for reachability probabilities. An approach that is widely used in the literature is to use *ranking supermartingales*—a probabilistic analogue of *ranking functions*—as a witness for the qualitative question of almost-sure reachability, i.e., whether reachability probability is 1. Ranking supermartingales are amenable to template-based synthesis [10, 11, 13], making them appealing from the automatic analysis point of view. Recently, methods for quantitatively under-approximating reachability probabilities are also proposed in References [15, 41].

The dual question of *over-approximating* reachability probabilities, which can then be used to qualitatively *refute* almost-sure reachability, is also considered. In control theory, supermartingales are used as a probabilistic counterpart of *barrier certificates* [33, 38]. A similar idea is recently used for the purpose of synthesizing *stochastic* invariants for randomized programs [15]. Here, an over-approximation of reachability probability serves as quantitative verification for safety: It gives an upper bound on the probability that the system or the program reaches a bad state.

Table 1 lists four martingale-based techniques for over- and under-approximating reachability probabilities. The table is not meant to be exhaustive—still, it shows that multiple methods have been introduced and studied in different communities (formal verification, control theory, etc.) and with different mathematical backgrounds (ranking functions, martingale concentration inequalities, etc.).

The current work aims at a comprehensive and comparative account of those martingale-based techniques in Table 1. Central to our account is the role of *order-theoretic fixed points*, one of the fundamental techniques in the theory of programming language. More specifically, we characterize our objectives—namely, reachability probability and expected reaching time—as suitable least fixed points. It turns out that a large part of the theory of martingale-based methods can be developed based on this order-theoretic characterization without using mathematical gadgets unique to probabilistic settings such as martingale concentration inequalities. In particular, soundness and completeness of two martingale-based techniques in Table 1—namely, additive ranking supermartingale and nonnegative repulsing supermartingale—immediately follow from our fixed point characterization via Knaster-Tarski theorem. This also unveils the mathematical similarity between these techniques, enabling us to develop their theory in a similar way.

Our contributions are summarized as follows:

- We give a comprehensive and comparative account of different martingale-based techniques for approximating reachability probabilities. We identify their key mathematical principles to be order-theoretic fixed points and martingale concentration inequalities, and we emphasize the role of the former. Specifically, the theory of three martingale techniques in Table 1—additive supermartingale (ARnkSupM), γ -scaled submartingale (γ -ScSubM), and

Table 1. Martingale-based Techniques for Approximation of Reachability Probabilities

name of martingale-based technique	certificate for	from
additive supermartingale (ARnkSupM, Section 5)	$\mathbb{E}(\text{steps to } C) \leq ?$	References [10, 28]
γ -scaled submartingale (γ -SclSubM, Section 6)	$\mathbb{P}(\text{reach } C) \geq ?$	this article for randomized programs, following categorical observations in Reference [41] for Markov chains
ε -decreasing supermartingale (ε -RepSupM, Section 3)	$\mathbb{P}(\text{reach } C) \leq ?$ $\mathbb{P}(\text{reach } C) < 1$	Reference [15], derived from Azuma's martingale concentration inequality
nonnegative supermartingale (NNRepSupM, Section 4)	$\mathbb{P}(\text{reach } C) \leq ?$	this article, derived from the Knaster–Tarski theorem (References [33, 38], without nondeterminism, derived from Markov's concentration inequality)

In the second column, C is the set of terminal states.

nonnegative repulsing supermartingale (NNRepSupM)—are developed based on the aforementioned order-theoretic characterization.

- We introduce two martingale-based techniques that seem to be new, namely, γ -SclSubM and NNRepSupM in Table 1. Their purely probabilistic versions have been in the literature: γ -SclSubM is from a category-theoretic account in Reference [41], and NNRepSupM is from control theory [38]. We extend them to randomized programs that additionally have nondeterminism. Moreover, completeness of ARnkSupM for randomized programs with real-valued variables seems to be new.
- We formalize those techniques, taking randomized programs as the target of analyses. We investigate soundness and completeness of the techniques in Table 1. We work on randomized programs with nondeterminism over continuous variables. While the order-theoretic fixed-point foundation gives us clear theoretical guidance, our general setting requires us to carefully establish measure-theoretic arguments.
- We implement template-based automated synthesis algorithms for γ -SclSubM and NNRepSupM, following References [10, 13]. Our experimental results suggest applicability of NNRepSupM and γ -SclSubM in quantitative reasoning. We also observe that NNRepSupM is applicable to several programs that cannot be handled by ε -RepSupM.

The present article is an extended version of Reference [39] with the following additional materials:

- Full proofs of soundness and completeness results (Propositions 4.2, 5.2, and 6.2) are provided, which are mostly omitted in Reference [39].
- We correct an error in Reference [39, Proposition 2.9]: We claimed $\overline{\mathbb{X}}$ and $\underline{\mathbb{X}}$ are closed operations over the set of Borel measurable functions, which turned out to be not necessarily true. Our correction consists of extending domains of $\overline{\mathbb{X}}$ and $\underline{\mathbb{X}}$ to the set of *upper* and *lower-semianalytic functions*, respectively (Proposition 2.17). The results that rely on this proposition (namely, Propositions 4.2, 5.2, and 6.2) now have modified statements, too.
- We give some additional materials for NNRepSupM in the later part of Section 4. One is an extension of the notion that corresponds to *c-martingale* in Reference [38]; we prove its soundness for over-approximating finite horizon reachability probability through our

order-theoretic framework (Proposition 4.14). Another is a technique to use NNRepSupM for *under*-approximation of reachability probability, under certain additional assumption.

- Analysis on completeness of γ -ScSubM is provided in Section 6.1. While it is not complete for under-approximating reachability probability, it possesses certain convergence property.
- We provide experimental results for more various probabilistic programs in Section 8.
- The comparison between the current work and Reference [38] is extended. It is found in Section 9.

The article is organized as follows: preliminaries are in Section 2, where we introduce our system models (pCFGs) for operational semantics of randomized programs, and review the theory of order-theoretic fixed points (the Knaster–Tarski and Cousot–Cousot theorems). In Section 3–6, we discuss the four techniques in Table 1, offering a unifying account based on order-theoretic fixed points and providing some new techniques and results. In Section 7, we discuss template-based synthesis; its implementation and experiment results are in Section 8. After a discussion of related work in Section 9, we conclude in Section 10.

2 PRELIMINARIES

We first fix some notations. We write $\mathbb{N}$ and $\mathbb{R}$ for the set of all natural numbers (i.e., nonnegative integers) and reals, respectively. We use subscripts to denote subsets of $\mathbb{N}$ and $\mathbb{R}$; for example, $\mathbb{R}_{\geq 0}$ denotes the set of all nonnegative reals. We write X^* , X^+ , X^ω for the sets of all finite, nonempty finite, and infinite sequences of elements of X , respectively. For a set X and $Y \subseteq X$, the complement of Y is denoted by $\bar{Y}$.

The functions **0** and **1** are the real-valued constant functions of which coefficient is 0 and 1, respectively. The Dirac measure on $x \in X$ is denoted by δ_x . For a set X and $A \subseteq X$, the characteristic function of A , i.e., a binary function over X that returns 1 for every $x \in A$ and 0 for every $x \notin A$, is denoted by χ_A . For a function $f : X \times Y \rightarrow Z$, we let $f(x, _) : Y \rightarrow Z$ be the function such that $f(x, _)(y) = f(x, y)$. For a set X , a linearly ordered set $(Y, \leq)$ and functions $f, g : X \rightarrow Y$, we say f is below g and write $f \sqsubseteq g$ if $f(x) \leq g(x)$ for every $x \in X$. We identify any collection F of functions of the type $X \rightarrow Y$ with the partially ordered set $(F, \sqsubseteq)$. For sets X and Y , the set of all functions from X to Y is denoted by Y^X . When X is a finite ordered set, we identify $\mathbb{R}^X$ and the $|X|$ -dimensional vector space $\mathbb{R}^{|X|}$ in a canonical way.

To convey a rigorous argument for real-valued randomized programs with nondeterminism, our definitions and proofs involve notions from measure theory. See, e.g., Reference [8, Section 7] for reference. For topological spaces X and Y , the set of all Borel subsets of X is denoted by $\mathcal{B}(X)$; the set of probability distributions on a measurable space $(X, \mathcal{B}(X))$ is denoted by $\mathcal{D}(X)$; and the set of all Borel, upper-semianalytic, lower-semianalytic, and universally measurable functions from X to Y are denoted by $\mathcal{B}(X, Y)$, $\overline{\mathcal{A}}(X, Y)$, $\underline{\mathcal{A}}(X, Y)$, and $\mathcal{U}(X, Y)$, respectively. The *support* $\text{supp}(d)$ of $d \in \mathcal{D}(X)$ is defined by $\text{supp}(d) = \{x \in X \mid \text{for any open set } A \subseteq X, x \in A \text{ implies } d(A) > 0\}$. A function of the type $f : X \rightarrow \mathcal{D}(Y)$ is called a *stochastic kernel* on Y given X .

For some sets, we assume their underlying topologies as follows: $\mathbb{R}$ is equipped with the topology from its canonical metric; each finite set is equipped with the discrete topology; product and coproduct of topological spaces are equipped with the product and coproduct topology, respectively; and $\mathcal{D}(X)$ is equipped with its weak topology.

2.1 Probabilistic Control Flow Graphs (pCFGs)

We take the notion of pCFG from Reference [2] and use it as our model of probabilistic systems. pCFGs can be thought of as a subclass of Markov decision processes, but tailored for operational

```

1  x := 2;
2  y := 2;
3  t := 0;
4  while t <= 100 do
5    t := t + 1;
6    z := Unif (-2,1);
7    if * then
8      x := x + z
9    else
10     y := y + z
11  fi
12 od

```

Fig. 1. An example of randomized programs. Line 6 means that the value of z is randomly sampled from the interval $[-2, 1]$.

semantics of randomized programs (Section 2.3). We first give a formal definition; see also Example 2.2.

Definition 2.1 (pCFG, [2]). A *probabilistic control flow graph* (pCFG) is a tuple $\Gamma = (L, V, l_{\text{init}}, \vec{x}_{\text{init}}, \mapsto, \text{Up}, \text{Pr}, G)$ consisting of the following components:

- A finite set L of *locations*, equipped with a partition $L = L_N + L_P + L_D + L_A$ into *nondeterministic*, *probabilistic*, *deterministic*, and *assignment* locations.
- A finite set $V = \{x_1, \dots, x_{|V|}\}$ of *program variables*.
- An *initial location* $l_{\text{init}} \in L$, and an *initial valuation vector* $\vec{x}_{\text{init}} \in \mathbb{R}^V$.
- A *transition relation* $\mapsto \subseteq L \times L$, which is total (each location has a successor). For $l \in L \setminus L_A$, we write $\text{succ}(l)$ to denote the set of all successors of l , i.e., $\text{succ}(l) = \{l' \in L \mid l \mapsto l'\}$. We require that each assignment location $l \in L_A$ has a unique successor; in this case, $\text{succ}(l)$ denotes this unique location.
- An *update function* $\text{Up} : L_A \rightarrow V \times (\mathcal{B}(\mathbb{R}^V, \mathbb{R}) \cup \mathcal{D}(\mathbb{R}) \cup \mathcal{B}(\mathbb{R}))$. Here, three components of $\mathcal{B}(\mathbb{R}^V, \mathbb{R}) \cup \mathcal{D}(\mathbb{R}) \cup \mathcal{B}(\mathbb{R})$ represent deterministic, probabilistic and nondeterministic assignment, respectively. We write $L_{AD} = \text{Up}^{-1}(V \times \mathcal{B}(\mathbb{R}^V, \mathbb{R}))$, $L_{AP} = \text{Up}^{-1}(V \times \mathcal{D}(\mathbb{R}))$ and $L_{AN} = \text{Up}^{-1}(V \times \mathcal{B}(\mathbb{R}))$.
- A family $\text{Pr} = (\text{Pr}_l \in \mathcal{D}(\text{succ}(l)))_{l \in L_P}$ of probability distributions.
- A *guard function* $G : L_D \times L \rightarrow \mathcal{B}(\mathbb{R}^V)$ such that, for each $l \in L_D$, the following hold: (collective exhaustion) $\bigcup_{l \mapsto l'} G(l, l') = \mathbb{R}^V$; and (mutual exclusion) $l \mapsto l'$, $l \mapsto l''$ and $l' \neq l''$ imply $G(l, l') \cap G(l, l'') = \emptyset$. We write $\vec{x} \models G(l, l')$ if $\vec{x} \in G(l, l')$.

A *configuration* of a pCFG Γ is a pair $(l, \vec{x}) \in L \times \mathbb{R}^V$ of a location and a vector. A *successor* $(l', \vec{x}')$ of a configuration $(l, \vec{x})$ is a one such that $l \mapsto l'$ and

- if $l \in L_N \cup L_P$, then $\vec{x}' = \vec{x}$;
- if $l \in L_D$, then $\vec{x}' = \vec{x}$ and $\vec{x} \models G(l, l')$; and
- if $l \in L_A$ and $\text{Up}(l) = (x_j, u)$, then $\vec{x}' = \vec{x}(x_j \leftarrow a)$. Here, $\vec{x}(x_j \leftarrow a)$ denotes an update of the vector (the x_j -component of $\vec{x}$ is replaced by a), and (i) $a = u(\vec{x})$ if $l \in L_{AD}$, (ii) $a \in \text{supp}(u)$ if $l \in L_{AP}$, and (iii) $a \in u$ if $l \in L_{AN}$.

Example 2.2. In Figure 2, the pCFG that models the randomized program in Figure 1 is given. Below, we explain the correspondence between Figure 1 and 2 step-by-step.

Each location of the pCFG in Figure 2 represents either termination of the program or a program state in which some line of the program is being executed. For example, l_1 corresponds to line 1 in Figure 1 in which the variable x is updated to 2; l_4 corresponds to line 4 in Figure 1 in which the while condition is checked; l_F corresponds to program termination, and so forth. A configuration,

Fig. 2. The pCFG that models the randomized program in Figure 1. Rectangles, diamonds, and pentagons represent deterministic, nondeterministic, and assignment locations, respectively. For each $i \in \mathbb{N}$, the location l_i corresponds to a program state in which the i th line is going to be executed; the location l_F corresponds to program termination.

a pair of location and valuation vector, therefore determines the program state by specifying the currently executed line and assignment of variables.

Deterministic locations represent program fragments such as while loops and deterministic if branches. For example, the location l_4 in Figure 2 corresponds to the while loop in line 4 of Figure 1. We have $\text{succ}(l_4) = \{l_5, l_F\}$, $G(l_4, l_5) = \{(x, y, t, z) \in \mathbb{R}^V \mid t \in (-\infty, 100]\}$ and $G(l_4, l_F) = \{(x, y, t, z) \in \mathbb{R}^V \mid t \in (100, +\infty)\}$. If the current configuration of the pCFG is $(l_4, (x, y, t, z))$ with $t = 0$, then its successor is $(l_5, (x, y, t, z))$.

Nondeterministic locations represent nondeterministic if branches such as line 7 of Figure 1. In Figure 2, l_7 is a nondeterministic location and we have $\text{succ}(l_7) = \{l_8, l_{10}\}$; the next location is determined by a *scheduler* (Definition 2.3) that is given from outside. The configurations $(l_8, (x, y, t, z))$ and $(l_{10}, (x, y, t, z))$ are both successors of $(l_7, (x, y, t, z))$. Similarly, Probabilistic locations represent probabilistic if branches, i.e., fragments of the form *if prob(p), then ... else ... fi* (Definition 2.11).

Assignment locations represent variable assignments such as lines 1, 2, 3, 5, 6, 8, and 10 in Figure 1. Assignments can be done in either in deterministic, probabilistic or nondeterministic manner. For example, l_5 is a deterministic assignment location (i.e., $l_5 \in L_{AD}$) and we have $\text{Up}(l_5) = (t, \lambda t. t + 1)$; l_6 is a probabilistic assignment location (i.e., $l_6 \in L_{AP}$) and we have $\text{Up}(l_6) = (z, \text{Unif}(-2, 1))$. A configuration $(l_6, (x, y, t + 1, z))$ is a unique successor of $(l_5, (x, y, t, z))$; any configuration $(l_7, (x, y, t, z'))$ such that $z' \in [-2, 1]$ is a successor of $(l_6, (x, y, t, z))$. Nondeterministic assignments represent a program fragment of the form $v := \text{ndet}(A)$ in which a real number $a \in A$ is chosen by a scheduler and assigned to the variable v . ■

A *finite path* of Γ is a finite sequence $c_0, c_1, \dots, c_k$ of configurations where c_i is a successor of c_{i-1} for each i . Similarly, a *run* of Γ is an infinite sequence $c_0, c_1, \dots$ of configurations such that each c_i is a successor of c_{i-1} .

Schedulers resolve nondeterminism. Recall that there are two types of locations at which a non-deterministic choice is required, namely, ones in L_N and L_{AN} . At $l \in L_N$, a successive location $l' \in \text{succ}(l)$ should be chosen; at $l \in L_{AN}$, a valid new assignment of the corresponding variable should be chosen. Therefore, we define a scheduler as a pair of functions that resolve these two situations, respectively.

We require schedulers to be *universally measurable*, which is less restrictive than Borel measurability. This assumption is required to use the Measurable Selection Theorem [8, Proposition 7.50], which is crucial for our theoretical results.

Definition 2.3 (Scheduler). A scheduler σ of a pCFG Γ is a pair (σ_t, σ_a) of functions that satisfies the following. Here, Π_N and Π_{AN} are the sets of all finite paths of Γ that are members of $(L \times \mathbb{R}^V)^*(L_N \times \mathbb{R}^V)$ and $(L \times \mathbb{R}^V)^*(L_{AN} \times \mathbb{R}^V)$, respectively.

- $\sigma_t : \Pi_N \rightarrow \mathcal{D}(L)$ is a universally measurable stochastic kernel on L given Π_N such that $\text{supp}(\sigma_t(\pi \cdot (l, \vec{x}))) \subseteq \text{succ}(l)$ holds for each $\pi \cdot (l, \vec{x}) \in \Pi_N$.

- $\sigma_a : \Pi_{AN} \rightarrow \mathcal{D}(\mathbb{R})$ is a universally measurable stochastic kernel on $\mathbb{R}$ given Π_{AN} such that, for each $\pi \cdot (l, \vec{x}) \in \Pi_{AN}$ that satisfies $\text{Up}(l) = (x_j, u)$, it holds that $\text{supp}(\sigma_a(\pi \cdot (l, \vec{x}))) \subseteq u$.

The set of schedulers of Γ is denoted by Sch_Γ .

Example 2.4. In Figure 2, nondeterminism at l_7 should be resolved by σ_t . For example, one could resolve it by choosing l_8 in the first 50 loops, and randomly choosing l_8 or l_{10} in the rest. Such a strategy is represented by a scheduler such that

$$\begin{aligned} \sigma_t(\pi \cdot (l_7, (x, y, t, z)))(l_8) &= 1, \quad \sigma_t(\pi \cdot (l_7, (x, y, t, z)))(l_{10}) = 0 & \text{if } t < 50, \\ \sigma_t(\pi \cdot (l_7, (x, y, t, z)))(l_8) &= \sigma_t(\pi \cdot (l_7, (x, y, t, z)))(l_{10}) = 0.5 & \text{if } t \geq 50. \end{aligned}$$

In the definition of σ_t , the condition $\text{supp}(\sigma_t(\pi \cdot (l, \vec{x}))) \subseteq \text{succ}(l)$ prohibits σ_t to choose invalid successors. For example, it is not allowed in Figure 2 to choose l_F as the next location from l_7 . ■

If a pCFG Γ and a scheduler σ for Γ are given, then the behavior of Γ is determined in the following sense: suppose several transitions are made in Γ under σ , and we have a sequence $c_0 \dots c_i$ of configurations as a history of transitions. Then the next configuration from this point is drawn from a distribution $\mu_{c_0 \dots c_i}^\sigma$ over $L \times \mathbb{R}^V$, which is determined by Γ , σ and $c_0 \dots c_i$. If we know such a distribution for every finite path, then for a given $c_0 \in L \times \mathbb{R}^V$, $N \in \mathbb{N}$ and measurable $C \subseteq L \times \mathbb{R}^V$, we know the probability that Γ is at a configuration in C after N transitions starting from c_0 .

We define such a family of probability distributions as a stochastic kernel μ_-^σ on $L \times \mathbb{R}^V$ given $(L \times \mathbb{R}^V)^+$ (i.e., a function of the type $(L \times \mathbb{R}^V)^+ \rightarrow \mathcal{D}(L \times \mathbb{R}^V)$). Its formal construction is as follows:

Definition 2.5 (Stochastic Kernel μ_-^σ). For a given pCFG Γ , a scheduler $\sigma \in \text{Sch}_\Gamma$ and a nonempty sequence $\pi \in (L \times \mathbb{R}^V)^+$ of configurations, we define a probability measure $\mu_\pi^\sigma \in \mathcal{D}(L \times \mathbb{R}^V)$ as follows:

- For any $\pi \in (L \times \mathbb{R}^V)^*$ and $(l, \vec{x}) \in L \times \mathbb{R}^V$ such that $\pi \cdot (l, \vec{x})$ is a finite path,
 - if $l \in L_N$, then $\mu_{\pi \cdot (l, \vec{x})}^\sigma = \sum_{l' \mapsto l'} \sigma_t(\pi \cdot (l, \vec{x}))(l') \cdot \delta_{(l', \vec{x})}$,
 - if $l \in L_P$, then $\mu_{\pi \cdot (l, \vec{x})}^\sigma = \sum_{l' \mapsto l'} \text{Pr}_l(l') \cdot \delta_{(l', \vec{x})}$,
 - if $l \in L_D$, then $\mu_{\pi \cdot (l, \vec{x})}^\sigma = \delta_{(l', \vec{x})}$, where l' is the unique location that satisfies $\vec{x} \models G(l, l')$,
 - if $l \in L_A$, $\text{succ}(l) = l'$ and $\text{Up}(l) = (x_j, u)$, then
 - * if $l \in L_{AD}$, then $\mu_{\pi \cdot (l, \vec{x})}^\sigma = \delta_{(l', \vec{x}(x_j \leftarrow u(\vec{x})))}$,
 - * if $l \in L_{AP}$, then $\mu_{\pi \cdot (l, \vec{x})}^\sigma$ is the unique measure that satisfies

$$\forall A_1, \dots, A_{|V|} \in \mathcal{B}(\mathbb{R}). \mu_{\pi \cdot (l, \vec{x})}^\sigma(l', \prod_{1 \leq j' \leq |V|} A_{j'}) = \begin{cases} u(A_j) & (\forall j' \neq j. x_{j'} \in A_{j'}) \\ 0 & (\text{otherwise}), \end{cases}$$

- * if $l \in L_{AN}$, then $\mu_{\pi \cdot (l, \vec{x})}^\sigma$ is the unique measure that satisfies

$$\forall A_1, \dots, A_{|V|} \in \mathcal{B}(\mathbb{R}). \mu_{\pi \cdot (l, \vec{x})}^\sigma(l', \prod_{1 \leq j' \leq |V|} A_{j'}) = \begin{cases} \sigma_a(\pi \cdot (l, \vec{x}))(A_j) & (\forall j' \neq j. x_{j'} \in A_{j'}) \\ 0 & (\text{otherwise}). \end{cases}$$

- For other cases, let $\mu_{\pi \cdot (l, \vec{x})}^\sigma$ be any measure in $\mathcal{D}(L \times \mathbb{R}^V)$ so μ_-^σ is universally measurable (for example, let $\mu_{\pi \cdot (l, \vec{x})}^\sigma = \delta_{(l, \vec{x})}$).

We note that μ_-^σ is universally measurable, and integrals with respect to μ_π^σ is well-defined for every $\sigma \in \text{Sch}_\Gamma$ and $\pi \in (L \times \mathbb{R}^V)^+$ ([8, Proposition 7.46]).

Example 2.6. In Figure 2, if the current configuration is $c = (l_6, (x, y, t, z))$, then the variable z is updated with a value sampled from $\text{Unif}(-2, 1)$ and the location moves to l_7 . Hence, for any π such that $\pi \cdot c$ is a finite path, the measure $\mu_{\pi \cdot c}^\sigma$ is the uniform measure over the compact set $\{l_7\} \times (\{x\} \times \{y\} \times \{t\} \times [-2, 1])$.

Similarly, if the current configuration is $c = (l_7, (x, y, t, z))$, then the location moves to either l_8 or l_{10} according to σ_t and the history of transitions. Hence, if σ_t is as in Example 2.4, then we have $\mu_{\pi \cdot c}^\sigma = \delta_{(l_8, (x, y, t, z))}$ if $t < 50$, or $\mu_{\pi \cdot c}^\sigma = \frac{1}{2}(\delta_{(l_8, (x, y, t, z))} + \delta_{(l_{10}, (x, y, t, z))})$ if $t \geq 50$. ■

2.2 Invariants and Characteristics of pCFG

An invariant is a set of configurations that is consistent as a domain of a given pCFG.

Definition 2.7 ((Pure) Invariant for pCFG). Let Γ be a pCFG. A measurable set $I \in \mathcal{B}(L \times \mathbb{R}^V)$ is called a *(pure) invariant* for Γ if $(l_{\text{init}}, \vec{x}_{\text{init}}) \in I$, and for each $(l, \vec{x}) \in I$, if $(l', \vec{x}')$ is a successor of $(l, \vec{x})$, then $(l', \vec{x}') \in I$.

In what follows, let Γ be a pCFG, $I \in \mathcal{B}(L \times \mathbb{R}^V)$ be an invariant and $C \in \mathcal{B}(I)$ be a *terminal region*. The following two characteristics of Γ are of our interest:

- *The reachability probability*, that is, the probability that Γ eventually reaches the terminal region C . Such a value can be defined once an initial configuration $c_0 \in I$ and a scheduler σ are determined; we define it as a function $\mathbb{P}_{C, \sigma}^{\text{reach}}$ over I parametrized by σ . In particular, we are interested in its supremum and infimum over all schedulers, which we call the *upper* and *lower* reachability probability, respectively.
- *The expected reaching time*, that is, the expected number of steps until Γ reaches C . Similar to the reachability probability, we define it as a function $\mathbb{E}_{C, \sigma}^{\text{steps}}$ over I parametrized by σ , and we are interested in its upper and lower variants.

We give some preliminary notions before their definitions. For given $\sigma \in \text{Sch}_\Gamma$ and a finite path $\pi \in I^*$, let $\sigma_\pi = ((\sigma_\pi)_t, (\sigma_\pi)_a)$ be any scheduler that satisfies $(\sigma_\pi)_t(\pi') = \sigma_t(\pi\pi')$ and $(\sigma_\pi)_a(\pi') = \sigma_a(\pi\pi')$ for each $\pi' \in I^*$ such that $\pi\pi'$ is a finite path. Such a scheduler σ_π can be seen as the reminder of σ after Γ does transitions with its history π . Notice that for each $\pi, \pi' \in I^*$ such that $\pi\pi'$ is a finite path, $(\sigma_\pi)_{\pi'} = \sigma_{\pi\pi'}$ and $\mu_{\pi\pi'}^\sigma = \mu_{\pi'}^{\sigma_\pi}$ hold.

We also define the following notion for convenience. Recall $\bar{C}$ is the complement of C .

Definition 2.8. The *reachability probability* $\mathbb{P}_{C, \sigma}^{\text{reach}=N}(c)$ in N steps from a configuration $c \in I$ to a region $C \in \mathcal{B}(I)$ under a scheduler $\sigma \in \text{Sch}_\Gamma$ is defined by $\mathbb{P}_{C, \sigma}^{\text{reach}=0}(c) = \chi_C(c)$, and for $N \geq 1$ it is inductively defined by

$$\mathbb{P}_{C, \sigma}^{\text{reach}=N+1}(c) = \chi_{\bar{C}}(c) \int_I \mathbb{P}_{C, \sigma_c}^{\text{reach}=N}(c') \mu_c^\sigma(dc'). \quad (1)$$

Also define the *non-reaching probability* $\mathbb{P}_{C, \sigma}^{\text{reach} \leq N}(c)$ within N steps from c to C under σ by $\mathbb{P}_{C, \sigma}^{\text{reach} \leq 0}(c) = \chi_{\bar{C}}(c)$, and for $N \geq 1$ it is inductively defined by

$$\mathbb{P}_{C, \sigma}^{\text{reach} \leq N+1}(c) = \chi_{\bar{C}}(c) \int_I \mathbb{P}_{C, \sigma_c}^{\text{reach} \leq N}(c') \mu_c^\sigma(dc'). \quad (2)$$

As the name suggests, $\mathbb{P}_{C, \sigma}^{\text{reach}=N}(c)$ is the probability that Γ visits C from c for the first time in exactly N steps under σ . The Equation (1) is the probabilistic counterpart of the following obvious observation in non-probabilistic systems: One reaches C (for the first time) in $N + 1$ steps if and only if they are not in C right now, and they reach C in N steps after a transition. Similarly,

$\mathbb{P}_{C,\sigma}^{\neg\text{reach}\leq N}(c)$ is the probability that Γ avoids C for N steps under σ , starting from c . We omit the letter I in descriptions of these functions (i.e., we do not write, e.g., $\mathbb{P}_{I,C,\sigma}^{\text{reach}=N}$), as invariants just determine the domains of functions and do not affect their values at each configuration.

Now, we formally define reachability probability and expected reaching time. We also define a “discounted” variant $\mathbb{P}_{C,\gamma,\sigma}^{\text{reach}}$ of reachability probability, which will be used in the analysis of γ -scaled submartingales (Section 6). It under-approximates reachability probability by definition and has a useful fixed-point property as explained in Section 6.

Definition 2.9 (e.g., [2, 23, 41]). For $\gamma \in (0, 1]$, we define the following functions:

$$\begin{aligned}\mathbb{P}_{C,\sigma}^{\text{reach}\leq N} &= \sum_{i=0}^N \mathbb{P}_{C,\sigma}^{\text{reach}=i}, & \mathbb{E}_{C,\sigma}^{\text{steps}\leq N} &= \left(\sum_{i=0}^N i \cdot \mathbb{P}_{C,\sigma}^{\text{reach}=i} \right) + N \cdot \mathbb{P}_{C,\sigma}^{\neg\text{reach}\leq N}, \\ \mathbb{P}_{C,\sigma}^{\text{reach}} &= \lim_{N \rightarrow \infty} \mathbb{P}_{C,\sigma}^{\text{reach}\leq N}, & \mathbb{E}_{C,\sigma}^{\text{steps}} &= \lim_{N \rightarrow \infty} \mathbb{E}_{C,\sigma}^{\text{steps}\leq N}, \\ \mathbb{P}_{C,\gamma,\sigma}^{\text{reach}\leq N} &= \sum_{i=0}^N \gamma^i \cdot \mathbb{P}_{C,\sigma}^{\text{reach}=i}, & \mathbb{P}_{C,\gamma,\sigma}^{\text{reach}} &= \lim_{N \rightarrow \infty} \mathbb{P}_{C,\gamma,\sigma}^{\text{reach}\leq N}.\end{aligned}$$

Notice that all the limits above always exist. An overline and an underline indicate the upper and lower variants of the function, respectively (i.e., $\overline{\mathbb{P}}_C^{\text{reach}\leq N} = \sup_{\sigma} \mathbb{P}_{C,\sigma}^{\text{reach}\leq N}$ and $\underline{\mathbb{P}}_C^{\text{reach}\leq N} = \inf_{\sigma} \mathbb{P}_{C,\sigma}^{\text{reach}\leq N}$ for example).

Remark 2.10. The reachability probability can be also defined in the following way: For given scheduler σ and an initial configuration c , one can construct a distribution over the set of infinite paths that represents the infinite-horizon behavior of Γ under σ starting from c . For this distribution, one can define $\mathbb{P}_{C,\sigma}^{\text{reach}}(c)$ as the measure of all infinite paths that eventually pass C . This definition is also well used in the literature and is compatible with ours. See Appendix A for the details.

2.3 Randomized Programs: ARP and PRP

The goal of this article is the reachability analysis of imperative programs with probabilistic and nondeterministic branching. We consider two languages taken from References [13, 15], called **affine randomized programs (ARP)** and **polynomial randomized programs (PRP)**. The two languages differ only in the arithmetic expressions allowed in the assignment commands and Boolean expressions. For example, the assignment command $x := xy + x + 1$ is allowed in PRP but not in ARP; $x := 3x + 2y - 1$ is allowed in both, since its right-hand side is an affine expression.

Both ARP and PRP have the standard control structure in imperative languages such as if-branches and while-loops. ARP and PRP additionally have nondeterministic and probabilistic if-branches (if $\star$ then $\dots$ and if **prob**(p) then $\dots$, respectively, where $p \in [0, 1]$). They also have nondeterministic and probabilistic assignment commands: $x := \mathbf{ndet} A$ where a value is chosen from a set $A \subseteq \mathbb{R}$; and $x := d$ where a value is sampled from a probability distribution d over $\mathbb{R}$.

The definition of the semantical model pCFG (Section 2.1) mirrors the structure of these languages. The translation from ARP/PRP to pCFGs is straightforward and omitted. We fix a countably infinite set $\mathcal{V}$ of variables.

Definition 2.11 (Affine Randomized Program). An *affine randomized program* (ARP) is a program defined by the following BNF notation:

$$\begin{aligned}
\langle \text{stmt} \rangle &::= \langle \text{assgn} \rangle \mid \mathbf{skip} \mid \langle \text{stmt} \rangle; \langle \text{stmt} \rangle \\
&\mid \mathbf{if} \langle \text{ndbexpr} \rangle \mathbf{then} \langle \text{stmt} \rangle \mathbf{else} \langle \text{stmt} \rangle \mathbf{fi} \\
&\mid \mathbf{while} \langle \text{bexpr} \rangle \mathbf{do} \langle \text{stmt} \rangle \mathbf{od} \\
\langle \text{assgn} \rangle &::= \langle \text{pvar} \rangle := \langle \text{expr} \rangle \mid \langle \text{pvar} \rangle := \mathbf{sample}(\langle \text{dist} \rangle) \\
&\mid \langle \text{pvar} \rangle := \mathbf{ndet}(\langle \text{dom} \rangle) \\
\langle \text{expr} \rangle &::= \langle \text{constant} \rangle \mid \langle \text{pvar} \rangle \mid \langle \text{constant} \rangle \cdot \langle \text{pvar} \rangle \mid \langle \text{expr} \rangle + \langle \text{expr} \rangle \\
&\mid \langle \text{expr} \rangle - \langle \text{expr} \rangle \\
\langle \text{dom} \rangle &::= \mathbf{Real} \mid \mathbf{Real}[\langle \text{const} \rangle, \langle \text{const} \rangle] \mid \langle \text{dom} \rangle \mathbf{or} \langle \text{dom} \rangle \\
\langle \text{bexpr} \rangle &::= \langle \text{conjexpr} \rangle \mid \langle \text{conjexpr} \rangle \mathbf{or} \langle \text{bexpr} \rangle \\
\langle \text{conjexpr} \rangle &::= \langle \text{literal} \rangle \mid \langle \text{literal} \rangle \mathbf{and} \langle \text{conjexpr} \rangle \\
\langle \text{literal} \rangle &::= \langle \text{expr} \rangle \leq \langle \text{expr} \rangle \mid \langle \text{expr} \rangle \geq \langle \text{expr} \rangle \mid \neg \langle \text{literal} \rangle \\
\langle \text{ndbexpr} \rangle &::= \star \mid \mathbf{prob}(p) \mid \langle \text{bexpr} \rangle \quad (\text{where } p \in [0, 1]) \\
\langle \text{pvar} \rangle &::= v \in \mathcal{V} \quad \langle \text{dist} \rangle ::= d \in \mathcal{D}(\mathbb{R}) \quad \langle \text{const} \rangle ::= c \in \mathbb{R}.
\end{aligned} \tag{3}$$

Definition 2.12 (Polynomial Randomized Program). A *polynomial randomized program* (PRP) is a program defined in almost the same manner as ARP (Definition 2.11) except that an expression $\langle \text{expr} \rangle$ ((3) in Definition 2.11) is defined by the following BNF notation:

$$\langle \text{expr} \rangle ::= \langle \text{constant} \rangle \mid \langle \text{pvar} \rangle \mid \langle \text{expr} \rangle \cdot \langle \text{expr} \rangle \mid \langle \text{expr} \rangle + \langle \text{expr} \rangle \mid \langle \text{expr} \rangle - \langle \text{expr} \rangle.$$

2.4 Order-theoretic Foundation of Fixed Points

Order-theoretic fixed points are central to computer science, for recursive computation, inductive/coinductive datatypes, and reasoning and specification of reactive behaviors, and so on. In general, a fixed-point equation can have multiple solutions; often, we are interested in extremal solutions: least fixed points (lfp's, for liveness, induction, etc.) and greatest ones (gfp's, for safety, coinduction, etc.). The following fundamental results (in a simple setting of complete lattices) give two different characterizations of lfp's and GFP's:

THEOREM 2.13. Let $(\mathcal{L}, \sqsubseteq)$ be a complete lattice, and $f : \mathcal{L} \rightarrow \mathcal{L}$ be a monotone function. Then f has the least fixed point μf and the greatest νf . Moreover,

- (1) (Knaster–Tarski, e.g., [5]) The lfp is the least pre-fixed point: $\mu f = \min\{l \in \mathcal{L} \mid f(l) \sqsubseteq l\}$. Similarly, the GFP is the greatest post-fixed point: $\nu f = \max\{l \in \mathcal{L} \mid l \sqsubseteq f(l)\}$.
- (2) (Cousot–Cousot [16]) The (potentially transfinite) ascending chain $\perp \sqsubseteq \underline{f}(\perp) \sqsubseteq \underline{f}^2(\perp) \sqsubseteq \dots$ stabilizes to μf . Here, $\underline{f}^\alpha(\perp)$ is defined by obvious induction: $\underline{f}^1 = f$; $\underline{f}^{\alpha+1}(\perp) = \underline{f}(\underline{f}^\alpha(\perp))$ for a successor ordinal; and $\underline{f}^\alpha(\perp) = \bigsqcup_{\beta < \alpha} \underline{f}^\beta(\perp)$ for a limit ordinal.

Similarly, the descending chain $\top \supseteq \overline{f}(\top) \supseteq \overline{f}^2(\top) \supseteq \dots$ stabilizes to νf , where $\overline{f}^1 = f$, $\overline{f}^{\alpha+1}(\top) = \overline{f}(\overline{f}^\alpha(\top))$ if α is a successor ordinal, and $\overline{f}^\alpha(\top) = \bigcap_{\beta < \alpha} \overline{f}^\beta(\top)$ if α is a limit ordinal. $\square$

From these characterizations, we can derive the following reasoning principles:

COROLLARY 2.14.

- **(lfp-KT)** $f(l) \sqsubseteq l$ implies $\mu f \sqsubseteq l$.
- **(gfp-KT)** $l \sqsubseteq f(l)$ implies $l \sqsubseteq \nu f$.
- **(lfp-CC)** For each ordinal α , $\underline{f}^\alpha(\perp) \sqsubseteq \mu f$.
- **(gfp-CC)** For each ordinal α , $\nu f \sqsubseteq \overline{f}^\alpha(\top)$. □

The arguments so far are symmetric for lfp's and gfp's. However, if one turns to the common proof methods for lfp specifications (termination, reachability, liveness) and those for gfp specifications (safety), a strong contrast emerges. Here is an example:

LEMMA 2.15. *Let a Kripke frame (i.e., a pair $(S, \mapsto)$ of a set S and a binary relation $\mapsto$ over S) be given, and let $s_0 \in S$, $C \subseteq S$.*

- **(Invariant for safety)** *Let $I \subseteq S$ be an invariant, that is, $I \subseteq \Box I$. Here, $\Box I$ is defined by $\Box I = \{s \in S \mid s \mapsto s' \text{ implies } s' \in I\}$. Assume also that $I \cap C = \emptyset$. Then $s_0 \in I$ implies that there is no path from s_0 to C .*
- **(Ranking function for liveness)** *Let $\eta : S \rightarrow \mathbb{N} \cup \{\infty\}$ be a ranking function for C . That is, (1) for each $s \in S \setminus C$, there is a successor s' such that $\eta(s) \geq \eta(s') + 1$; and (2) for each $s \in S$, $\eta(s) = 0$ implies $s \in C$. Then, $\eta(s_0) \neq \infty$ implies that there is a path from s_0 to C .* □

The difference between the two methods is accounted for by the fact that, in Corollary 2.14, two items give *under*-approximations while the other two give *over*-approximations. It is clear that the invariant method in Lemma 2.15 comes from (gfp-KT) of Corollary 2.14. Its dual, (lfp-KT), gives only an over-approximation l —it can be used for refutation but not for verification. Similarly, ranking functions come from (lfp-CC)—the role of well-foundedness of the value domain $\mathbb{N}$ mirrors the structure of ordinals. Its dual (gfp-CC) only gives an over-approximation of νf . The situation is summarized in the following table.

	Knaster–Tarski	Cousot–Cousot
lfp	over-approx.	under-approx.
gfp	under-approx.	over-approx.

The above foundations underpin our technical developments: This is because reachability probabilities and reaching times are characterized as least fixed points. We note that our semantical domains $\mathcal{L}$ in later sections need not be complete lattices. In those cases, we exploit the ω - and ω^{op} -cpo structures, the corresponding continuity of f , and the *Kleene theorem* (Proposition 4.6). The last is understood as a variation of the Cousot–Cousot theorem.

2.5 The Nexttime Operation

In Section 3–6, we discuss martingale notions that are listed in Table 1. Martingales are, informally, functions over states that satisfy certain monotonicity condition through transitions. For their concrete definitions, we here formalize the notion of “the value of a function at the next state,” which is dependent on the type of transitions.

Definition 2.16 (The “nexttime” Operation $\overline{\mathbb{X}}, \underline{\mathbb{X}}$). Let Γ be a pCFG, I be a pure invariant, and $\mathbb{K} \in \mathcal{B}(\mathbb{R})$ be a Borel set. For a function $\eta : I \rightarrow \mathbb{K}$, we define the function $\overline{\mathbb{X}}\eta$ of the same type as η by $\overline{\mathbb{X}}\eta(c) = \sup_{\sigma} \int_I \eta d\mu_c^\sigma$, provided the right-hand side of the equation is well-defined for each $c \in I$. Equivalently, $\overline{\mathbb{X}}\eta$ is a function that is defined as follows:

- For $l \in L_N$, $(\overline{\mathbb{X}}\eta)(l, \vec{x}) = \max_{l' \mapsto l'} \eta(l', \vec{x})$.
- For $l \in L_P$, $(\overline{\mathbb{X}}\eta)(l, \vec{x}) = \sum_{l' \mapsto l'} \text{Pr}_l(l') \eta(l', \vec{x})$.
- For $l \in L_D$, $(\overline{\mathbb{X}}\eta)(l, \vec{x}) = \eta(l', \vec{x})$ where l' is the unique location such that $\vec{x} \models G(l, l')$.
- For $l \in L_A$, let $\text{Up}(l) = (x_j, u)$.
 - $(\overline{\mathbb{X}}\eta)(l, \vec{x}) = \eta(\text{succ}(l), \vec{x}(x_j \leftarrow u(\vec{x})))$ if u is a measurable function.
 - $(\overline{\mathbb{X}}\eta)(l, \vec{x}) = \int_{x \in \text{supp}(u)} \eta(\text{succ}(l), \vec{x}(x_j \leftarrow x)) du$ if u is a distribution.
 - $(\overline{\mathbb{X}}\eta)(l, \vec{x}) = \sup_{x \in u} \eta(\text{succ}(l), \vec{x}(x_j \leftarrow x))$ if u is a measurable set.

The function $\underline{\mathbb{X}}\eta : I \rightarrow \mathbb{K}$ is defined by $\underline{\mathbb{X}}\eta(c) = \inf_{\sigma} \int_I \eta d\mu_c^{\sigma}$, provided the right-hand side of each equation is well-defined for each $c \in I$.

Notice that $\overline{\mathbb{X}}\eta$ may not be Borel even if η is, as the operation $\overline{\mathbb{X}}$ involves a supremum operation over Sch_{Γ} . As we show in Proposition 2.17 below, the set $\overline{\mathcal{A}}(I, \mathbb{K})$ of *upper semianalytic* functions is suitable for the domain of $\overline{\mathbb{X}}$: see, e.g., Reference [8, Section 7] for its definition and properties. Similarly, the set $\underline{\mathcal{A}}(I, \mathbb{K})$ of *lower semianalytic* functions is suitable for the domain of $\underline{\mathbb{X}}$.

PROPOSITION 2.17. *Let $\mathbb{K}$ be a proper closed convex subset of $\mathbb{R} \cup \{\pm\infty\}$. Then the following hold:*

- (1) *For every $\eta \in \overline{\mathcal{A}}(I, \mathbb{K})$, the function $\overline{\mathbb{X}}\eta$ is well-defined and $\overline{\mathbb{X}}\eta \in \overline{\mathcal{A}}(I, \mathbb{K})$; Also, for every $\eta \in \underline{\mathcal{A}}(I, \mathbb{K})$, the function $\underline{\mathbb{X}}\eta$ is well-defined and $\underline{\mathbb{X}}\eta \in \underline{\mathcal{A}}(I, \mathbb{K})$.*
- (2) *The operators $\overline{\mathbb{X}}$ and $\underline{\mathbb{X}}$ are monotone over $\overline{\mathcal{A}}(I, \mathbb{K})$ and $\underline{\mathcal{A}}(I, \mathbb{K})$, respectively.*
- (3) *The lattice $\overline{\mathcal{A}}(I, \mathbb{K})$ is ω -cpo and ω^{op} -cpo, that is, any ascending (decsending) chain in it has the supremum (infimum). The same holds for $\underline{\mathcal{A}}(I, \mathbb{K})$.*
- (4) *$\overline{\mathbb{X}}$ is ω -continuous, i.e., $\bigsqcup_{n < \omega} \overline{\mathbb{X}}\eta_n = \overline{\mathbb{X}}(\bigsqcup_{n < \omega} \eta_n)$ holds for any $\{\eta_n\}_{n \in \mathbb{N}}$. Also $\underline{\mathbb{X}}$ is ω^{op} -continuous, i.e., $\sqcap_{n < \omega} \underline{\mathbb{X}}\eta_n = \underline{\mathbb{X}}(\sqcap_{n < \omega} \eta_n)$ holds for any $\{\eta_n\}_{n \in \mathbb{N}}$.*

PROOF. The well-definedness part of (1) is straightforward; we have $\underline{\mathbb{X}}\eta \in \underline{\mathcal{A}}(I, \mathbb{K})$ because semianalytic property is preserved through every operation that is done on η in Definition 2.16 [8, Lemma 7.30, Proposition 7.47, Proposition 7.48]; and we have $\overline{\mathbb{X}}\eta \in \overline{\mathcal{A}}(I, \mathbb{K})$ in a similar way to the lower case. The statement (2) is also straightforward; (3) is from Reference [8, Lemma 7.30(2)]; and the upper case of (4) is proved as follows (the lower case is similar):

$$\begin{aligned}
 \bigsqcup_{n < \omega} \overline{\mathbb{X}}\eta_n &= \bigsqcup_{n < \omega} \sup_{\sigma} \int_I \eta_n d\mu_c^{\sigma} \quad (\text{definition of } \overline{\mathbb{X}}) \\
 &= \sup_{\sigma} \bigsqcup_{n < \omega} \int_I \eta_n d\mu_c^{\sigma} \quad (\text{supremums commute}) \\
 &= \overline{\mathbb{X}}(\bigsqcup_{n < \omega} \eta_n). \quad (\text{Monotone Convergence Theorem}) \quad \square
 \end{aligned}$$

Example 2.18. Notice that $\overline{\mathbb{X}}$ is not ω^{op} -continuous as it involves an alternation of supremum and infimum operations. In Figure 3, let $\eta_n = \chi_{\{l_1\} \times [0, \frac{1}{n}]}$. Then $\overline{\mathbb{X}}\eta_n(l_0, 0) = 1$ for each $n < \omega$ and hence $\sqcap_{n < \omega} \overline{\mathbb{X}}\eta_n(l_0, 0) = 1$. However, $\sqcap_{n < \omega} \eta_n = \chi_{\{l_1, 0\}}$ and hence $\overline{\mathbb{X}}(\sqcap_{n < \omega} \eta_n)(l_0, 0) = 0$. Similarly, $\underline{\mathbb{X}}$ is not ω -continuous.

We also witness the lack of continuity when we are interested in the set of all configurations at program termination. In such a case, we would let the nexttime operator act on functions of the type $\eta : I \rightarrow \mathcal{P}(I \cup \{\text{div}\})$, where div represents non-termination. See, e.g., Reference [4] for the details. ■

Fig. 3. An example of incompleteness of ε -RepSupM. Probabilistic locations are depicted by circles. This pCFG satisfies $\overline{\mathbb{P}}_{\{l_5\} \times \mathbb{R}}^{\text{reach}}(l_0, 0) = \frac{1}{2}$ but no ε -RepSupM can refute its a.s. reachability. Indeed, any ε -RepSupM η for $\{l_5\} \times \mathbb{R}$ must satisfy $\lim_{x \rightarrow +0} \eta(l_1, x) = \infty$ due to the ε -decreasing condition, but such an η cannot have κ -bounded differences at $(l_0, 0)$.

3 ε -DECREASING REPULSING SUPERMARTINGALES (ε -REPSUPM)

In Sections 3–6, we will discuss the four martingale-based techniques in Table 1. Through these sections, we assume a pCFG Γ with its invariant I and terminal region $C \in \mathcal{B}(I)$ is given.

First, we briefly review the notion of ε -decreasing repulsing supermartingale (ε -RepSupM) from Reference [15]. It is, to the best of our knowledge, the only existing martingale-based notion for over-approximating reachability probabilities.

Definition 3.1 (ε -RepSupM [15]). Let Γ be a pCFG, I be a pure invariant, and $C \subseteq I$ be a Borel set. An ε -repulsing supermartingale (ε -RepSupM) for C supported by I is a function $\eta \in \overline{\mathcal{A}}(I, \mathbb{R})$ such that (i) $\eta(c) \geq (\overline{\mathbb{X}}\eta)(c) + \varepsilon$ for each $c \in I \setminus C$, (ii) $\eta(c) \geq 0$ for each $c \in C$, and (iii) η has κ -bounded differences for some $\kappa > 0$, i.e., for each $c \in I$ and its successor c' it holds $|\eta(c) - \eta(c')| \leq \kappa$.

THEOREM 3.2 (SOUNDNESS, [15]). Suppose there exists an ε -RepSupM η for C supported by I , which is linear, i.e., η is of the form $\eta(l, \vec{x}) = \sum_{j=1}^{|V|} a_{l,j} x_j$ where $a_{l,j}$ is a fixed constant. Further assume that $\eta(l_{\text{init}}, \vec{x}_{\text{init}}) < 0$ holds. Let $\gamma = e^{-\frac{\varepsilon^2}{2(\kappa+\varepsilon)^2}}$ and $\alpha = e^{\frac{\varepsilon \cdot \eta(l_{\text{init}}, \vec{x}_{\text{init}})}{(\kappa+\varepsilon)^2}}$.

(1) We have the following inequality:

$$\overline{\mathbb{P}}_C^{\text{reach}}(l_{\text{init}}, \vec{x}_{\text{init}}) \leq \alpha \cdot \frac{\gamma^{\lceil |\eta(l_{\text{init}}, \vec{x}_{\text{init}})| / \kappa \rceil}}{1 - \gamma}. \quad (4)$$

(2) If the right-hand side of Equation (4) is greater than 1, then still $\overline{\mathbb{P}}_C^{\text{reach}}(l_{\text{init}}, \vec{x}_{\text{init}}) < 1$ holds. $\square$

We note that for any $\eta \in \overline{\mathcal{A}}(I, \mathbb{R})$ the function $\overline{\mathbb{X}}\eta$ is well-defined because of the κ -bounded differences condition. The bound in Equation (4) is derived from Azuma's concentration inequality, a well-known martingale concentration lemma that exploits κ -bounded differences. ε -RepSupM is not complete: There exist a pCFG Γ and a set C of configurations such that $\overline{\mathbb{P}}_C^{\text{reach}} < 1$ but no ε -RepSupM can prove it. See Figure 3.

4 NONNEGATIVE REPULSING SUPERMARTINGALES (NNREPSUPM)

We move on to another notion for over-approximating reachability probabilities, *nonnegative repulsing supermartingale* (NNRepSupM). We believe this is new. Compared to the notion of ε -RepSupM, NNRepSupM has the following features:

- NNRepSupM is derived from the theory of order-theoretic fixed points (Section 2.4), unlike ε -RepSupM, which relies on Azuma's martingale concentration lemma.
- NNRepSupM is sound and complete, while ε -RepSupM is not complete.

- We experimentally observe that NNRepSupM can give nontrivial bounds for programs that cannot be handled by ε -RepSupM (Section 8).

The definition of NNRepSupM resembles *probabilistic barrier certificates* in control theory [33, 38], or the *while rule* in probabilistic PDL [25]. Our technical contributions are the following: (i) we develop the theory of NNRepSupM in the presence of nondeterminism, while the settings in References [25, 33, 38] are purely probabilistic; and (ii) we characterize NNRepSupM in the general terms of order-theoretic fixed points (Section 2.4), unlike the previous theory in References [33, 38] that relies on Markov's martingale concentration lemma.¹ The latter unveils the mathematical similarity between NNRepSupM and ARnkSupM (Section 5).

The notion comes with upper and lower variants. They are used to over-approximate $\overline{\mathbb{P}}_C^{\text{reach}}$ and $\underline{\mathbb{P}}_C^{\text{reach}}$, respectively (Definition 2.9). In what follows, we omit the subscript C from notations when it is clear from the context (e.g., we write $\overline{\mathbb{P}}^{\text{reach}}$ instead of $\overline{\mathbb{P}}_C^{\text{reach}}$).

Definition 4.1 (NNRepSupM for pCFG). An *upper nonnegative repulsing supermartingale* (U -NNRepSupM) over a pCFG Γ for a terminal region C supported by an invariant I is an upper semianalytic function $\eta \in \overline{\mathcal{A}}(I, [0, \infty])$ such that

$$(i) \eta(c) \geq 1 \text{ for each } c \in C, \text{ and } (ii) \eta(c) \geq \overline{\mathbb{X}}\eta(c) \text{ for each } c \in I \setminus C.$$

A lower semianalytic function $\eta \in \underline{\mathcal{A}}(I, [0, \infty])$ is a *lower nonnegative repulsing supermartingale* (L -NNRepSupM) if it satisfies the above conditions, but with $\overline{\mathbb{X}}$ replaced with $\underline{\mathbb{X}}$.

Equivalently, upper and lower NNRepSupMs are pre-fixed points of the following monotone endofunctions $\overline{\Phi}$ over $\overline{\mathcal{A}}(I, [0, \infty])$ and $\underline{\Phi}$ over $\underline{\mathcal{A}}(I, [0, \infty])$, respectively:

$$\overline{\Phi}\eta = \chi_C + \chi_{\overline{C}} \cdot \overline{\mathbb{X}}\eta, \quad \underline{\Phi}\eta = \chi_C + \chi_{\underline{C}} \cdot \underline{\mathbb{X}}\eta.$$

We shall formalize soundness and completeness of NNRepSupM in terms of fixed points. Recall that, by soundness of (say, upper) NNRepSupM, we mean that every NNRepSupM overapproximates $\overline{\mathbb{P}}^{\text{reach}}$, that is, if $\eta \in \overline{\mathcal{A}}(I, [0, \infty])$, then $\eta \sqsubseteq \overline{\mathbb{P}}^{\text{reach}}$. Also, by completeness, we mean that there exists an NNRepSupM that uniformly gives an optimal bound, that is, $\overline{\mathbb{P}}^{\text{reach}} \sqsubseteq \overline{\Phi}\overline{\mathbb{P}}^{\text{reach}}$. By virtue of Knaster-Tarski theorem (Theorem 2.13.1), they are identified with the following characterization of reachability probability. Recall that the least fixed point of a monotone operator f is denoted by μf .

PROPOSITION 4.2 (FIXED POINT CHARACTERIZATION OF $\overline{\mathbb{P}}^{\text{reach}}$ AND $\underline{\mathbb{P}}^{\text{reach}}$). *The following are true:*

- (1) $\overline{\mathbb{P}}^{\text{reach}} = \mu \overline{\Phi}$.
- (2) $\underline{\mathbb{P}}^{\text{reach}} = \mu \underline{\Phi}$.

Due to an asymmetry of $\overline{\Phi}$ and $\underline{\Phi}$ (that is, $\overline{\Phi}$ is ω -continuous while $\underline{\Phi}$ is not), the proof of the upper and lower cases are rather different. Below, we provide their proofs separately.

4.1 Proof of the Upper Case

The role of order-theoretic argument is particularly prominent in the upper case: We prove Proposition 4.2.1 via Kleene fixed point theorem and then derive soundness and completeness as its

¹We note that the theory of NNRepSupM can also be developed using Markov's lemma.

corollary. The following lemma is important; the proof requires a careful measure-theoretic argument and will be given in the later part of this section.

LEMMA 4.3. *For each $N \in \mathbb{N}$, we have $\overline{\mathbb{P}}^{\text{reach} \leq N} \in \overline{\mathcal{A}}(I, [0, \infty])$ and $\overline{\mathbb{P}}^{\text{reach} \leq N+1} = \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq N}$.*

A proof of Proposition 4.2.1 is straightforward once this lemma is verified: We obtain $\overline{\mathbb{P}}^{\text{reach}} = \overline{\Phi}^\omega \perp$ via an iterative application of the lemma (notice $\overline{\Phi} \perp = \chi_C = \overline{\mathbb{P}}^{\text{reach} \leq 0}$), and then $\overline{\mathbb{P}}^{\text{reach}} = \mu \overline{\Phi}$ by Kleene fixed point theorem.

Proving this lemma is easy if the system is purely probabilistic; Equation (5) essentially proves it. It is also easy if the system only admits discrete nondeterminism; it is done by just taking the supremum over schedulers in Equation (5) thanks to Monotone Convergence Theorem. In our setting, though, it is nontrivial, because suprema and integrals over uncountable sets do not commute in general. We start with giving some lemmas.

LEMMA 4.4. *Let $\eta \in \overline{\mathcal{A}}(I, [0, \infty])$, $\eta' \in \mathcal{A}(I, [0, \infty])$, and $a > 0$. Then, we have $\overline{\Phi}(\eta + a \cdot \mathbf{1}) \sqsubseteq \overline{\Phi}\eta + a\chi_{\overline{C}}$, $\overline{\Phi}(\eta - a \cdot \mathbf{1}) \sqsupseteq \overline{\Phi}\eta - a\chi_{\overline{C}}$, $\underline{\Phi}(\eta' + a \cdot \mathbf{1}) \sqsubseteq \underline{\Phi}\eta' + a\chi_{\overline{C}}$ and $\underline{\Phi}(\eta' - a \cdot \mathbf{1}) \sqsupseteq \underline{\Phi}\eta' - a\chi_{\overline{C}}$.*

PROOF. We prove the first instance only; the rest are proved in a similar way.

For $c \in C$, we have $\overline{\Phi}(\eta + a \cdot \mathbf{1})(c) = \overline{\Phi}\eta(c) = 1$ (recall, by definition of $\overline{\Phi}$, we have $\overline{\Phi}\eta(c) = 1$ for any $\eta \in \overline{\mathcal{A}}(I, [0, \infty])$ and $c \in C$). For $c \in \overline{C}$, we have $\overline{\Phi}(\eta + a \cdot \mathbf{1})(c) = \sup_\sigma \int_I \eta + a \cdot \mathbf{1} d\mu_c^\sigma \leq \overline{\Phi}\eta(c) + a$. Hence, the desired inequality holds. $\square$

A scheduler is *Markov* if it satisfies $\sigma_t(\pi c) = \sigma_t(\pi' c)$ and $\sigma_a(\pi c') = \sigma_a(\pi' c')$ for every $\pi, \pi' \in (L \times \mathbb{R}^V)^*$ such that $|\pi| = |\pi'|$, $c \in L_N \times \mathbb{R}^V$ and $c' \in L_{AN} \times \mathbb{R}^V$. In other words, a scheduler is Markov if the output is determined by the last element and the length of an input.

If a scheduler σ is Markov, then σ_π is identical for any finite path $\pi \in (L \times \mathbb{R}^V)^n$ with a fixed n . Such a scheduler is denoted by $\sigma_{(n)}$ (in particular, $\sigma_{(0)} = \sigma$).

For $\mathbb{K} \in \mathcal{B}(\mathbb{R}^V)$ and $\eta \in \mathcal{U}(I, \mathbb{K})$, we define $\mathbb{X}_\sigma \eta$ by $\mathbb{X}_\sigma \eta(c) = \int_I \eta(c') \mu_c^\sigma(d c')$ (i.e., $\mathbb{X}_\sigma \eta(c)$ is the value of η after a transition from c under σ). Similarly, we define $\Phi_\sigma \eta$ by $\Phi_\sigma \eta = \chi_C + \chi_{\overline{C}} \cdot \mathbb{X}_\sigma \eta$. Notice that, if $\eta \in \overline{\mathcal{A}}(I, \mathbb{K})$, then $\overline{\mathbb{X}}\eta = \sup_\sigma \mathbb{X}_\sigma \eta$ and $\overline{\Phi}\eta = \sup_\sigma \Phi_\sigma \eta$ hold.²

For a given $\varepsilon > 0$ and $c \in I$, there exists a scheduler σ that satisfies $\mathbb{X}_\sigma \eta(c) \geq \overline{\mathbb{X}}\eta(c) - \varepsilon$. By combining such a σ for each $c \in I$, we can construct a scheduler that satisfies the inequality for every c , that is, $\mathbb{X}_\sigma \eta \sqsupseteq \overline{\mathbb{X}}\eta - \varepsilon \cdot \mathbf{1}$. One thing we have to ensure here is universal measurability of such a scheduler, which can be done via Measurable Selection Theorem [8, Proposition 7.50]. We prove a slightly generalized statement as follows:

LEMMA 4.5. *Let $\varepsilon_n > 0$ and $\eta_n \in \overline{\mathcal{A}}(I, \mathbb{K})$ be given for each $n \in \mathbb{N}$. Then there exists a Markov scheduler σ that satisfies $\mathbb{X}_{\sigma_{(n)}} \eta_n \sqsupseteq \overline{\mathbb{X}}\eta_n - \varepsilon_n \cdot \mathbf{1}$ for each $n \in \mathbb{N}$.*

Similarly, for given $\varepsilon_n > 0$ and $\eta_n \in \mathcal{A}(I, \mathbb{K})$ for each $n \in \mathbb{N}$, there exists a Markov scheduler σ that satisfies $\mathbb{X}_{\sigma_{(n)}} \eta_n \sqsubseteq \underline{\mathbb{X}}\eta_n + \varepsilon_n \cdot \mathbf{1}$ for each $n \in \mathbb{N}$.

PROOF. We prove the upper case only; the lower case is proved in a similar way. Let $\pi \cdot (l, \vec{x}) \in \Pi_N \cup \Pi_{AN}$ such that $|\pi| = n - 1$ be given. We construct $\sigma = \{\sigma_t, \sigma_a\}$ as follows:

- If $l \in L_N$, then σ_t chooses $l' \in \text{succ}(l)$ that maximizes $\mathbb{X}_\sigma \eta_n$. Concretely, we let $\sigma_t(\pi \cdot (l, \vec{x})) = \delta_{\tilde{l}}$, where $\tilde{l}$ satisfies $\eta_n(\tilde{l}, \vec{x}) = \max_{l' \in l'} \eta_n(l', \vec{x})$.

²The measurability argument is somewhat complicated here: The operation $\mathbb{X}_\sigma$ is closed over $\mathcal{U}(I, \mathbb{K})$ [8, Proposition 7.46], but may not be closed over $\overline{\mathcal{A}}(I, \mathbb{K})$ when σ is not Borel. The fact $\overline{\mathbb{X}}\eta \in \overline{\mathcal{A}}(I, \mathbb{K})$ does not follow from the equation $\overline{\mathbb{X}}\eta = \sup_\sigma \mathbb{X}_\sigma \eta$, but from the equivalent definition of $\overline{\mathbb{X}}\eta$ that does not involve σ (see Proposition 2.17).

- If $l \in L_{AN}$ and $\text{Up}(l) = (x_j, u)$ (i.e., σ_a is required to choose a new assignment of x_j from $u \in \mathcal{B}(\mathbb{R})$), then σ_a chooses $y \in u$ for which $\mathbb{X}_\sigma \eta_n$ is ε_n -close to the supremum. Concretely, we let $\varphi : \mathbb{R}^V \rightarrow u$ be a function such that $\eta_n(\text{succ}(l), \vec{x}(x_j \leftarrow \varphi(\vec{x}))) \geq \sup_{y \in u} \eta_n(\text{succ}(l), \vec{x}(x_j \leftarrow y)) - \varepsilon_n$, and let $\sigma_a(\pi \cdot (l, \vec{x})) = \delta_{\varphi(\vec{x})}$.

This σ is Markov, and the desired inequality immediately follows from its construction. It is also universally measurable by taking a universally measurable φ by [8, Proposition 7.50]. $\square$

PROOF OF LEMMA 4.3. First observe the following:

$$\begin{aligned}
 \mathbb{P}_\sigma^{\text{reach} \leq N+1}(c) &= \mathbb{P}_\sigma^{\text{reach}=0}(c) + \sum_{i=1}^{N+1} \mathbb{P}_\sigma^{\text{reach}=i}(c) \\
 &= \chi_C(c) + \sum_{i=0}^N \chi_{\bar{C}}(c) \int_I \mathbb{P}_{\sigma_c}^{\text{reach}=i}(c') \mu_c^\sigma(\text{dc}') \\
 &= \chi_C(c) + \chi_{\bar{C}}(c) \int_I \mathbb{P}_{\sigma_c}^{\text{reach} \leq N}(c') \mu_c^\sigma(\text{dc}'). \tag{5}
 \end{aligned}$$

We prove the lemma by induction on $\mathbb{N}$. Notice we have $\overline{\mathbb{P}}^{\text{reach} \leq 0} = \chi_C \in \overline{\mathcal{A}}(I, [0, \infty])$. Let $M \in \mathbb{N}$ and assume $\overline{\mathbb{P}}^{\text{reach} \leq N} \in \overline{\mathcal{A}}(I, [0, \infty])$ and $\overline{\mathbb{P}}^{\text{reach} \leq N+1} = \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq N}$ hold for $N = 0, \dots, M-1$ (we assume nothing if $M = 0$). Then, we have $\overline{\mathbb{P}}^{\text{reach} \leq M} \in \overline{\mathcal{A}}(I, [0, \infty])$ by $\overline{\mathbb{P}}^{\text{reach} \leq 0} = \chi_C$ if $M = 0$, or by $\overline{\mathbb{P}}^{\text{reach} \leq M} = \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M-1}$ and Proposition 2.17.1 if $M \geq 1$. Hence, $\overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M} \in \overline{\mathcal{A}}(I, [0, \infty])$ is well-defined, and we have $\overline{\mathbb{P}}^{\text{reach} \leq M+1} \sqsubseteq \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M}$ as follows: For each $c \in I$,

$$\begin{aligned}
 \mathbb{P}_\sigma^{\text{reach} \leq M+1}(c) &= \chi_C(c) + \chi_{\bar{C}}(c) \int_I \mathbb{P}_{\sigma_c}^{\text{reach} \leq M}(c') \mu_c^\sigma(\text{dc}') && \text{(Equation (5))} \\
 &\leq \Phi_\sigma \overline{\mathbb{P}}^{\text{reach} \leq M}(c) && \left(\overline{\mathbb{P}}^{\text{reach} \leq M} = \sup_{\sigma} \mathbb{P}_\sigma^{\text{reach} \leq M} \right) \\
 &= \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M}(c). && \left(\overline{\Phi} \eta = \sup_{\sigma} \Phi_\sigma \eta \right)
 \end{aligned}$$

We also have $\overline{\mathbb{P}}^{\text{reach} \leq M+1} \sqsupseteq \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M}$ as follows: For any $\varepsilon > 0$, let σ be a Markov scheduler that satisfies $\Phi_{\sigma(n)} \overline{\mathbb{P}}^{\text{reach} \leq M-n} \sqsupseteq \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M-n} - \varepsilon \cdot \mathbf{1}$ for each $n = 0, \dots, M$ (Lemma 4.5). Then, we have

$$\begin{aligned}
 \Phi_{\sigma(0)} \dots \Phi_{\sigma(M)} \chi_C &\sqsupseteq \Phi_{\sigma(0)} \dots \Phi_{\sigma(M-1)} (\overline{\Phi} \chi_C - \varepsilon \cdot \mathbf{1}) && \text{(definition of } \sigma) \\
 &\sqsupseteq \Phi_{\sigma(0)} \dots \Phi_{\sigma(M-1)} \overline{\Phi} \chi_C - \varepsilon \cdot \mathbf{1} && \text{(Lemma 4.4)} \\
 &= \Phi_{\sigma(0)} \dots \Phi_{\sigma(M-1)} \overline{\mathbb{P}}^{\text{reach} \leq 1} - \varepsilon \cdot \mathbf{1} && \text{(induction hypothesis)} \\
 &\sqsupseteq \dots \sqsupseteq \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M} - (M+1)\varepsilon \cdot \mathbf{1}.
 \end{aligned}$$

Notice that the Equation (5) can be written as $\mathbb{P}_\sigma^{\text{reach} \leq N+1} = \Phi_\sigma \mathbb{P}_{\sigma(1)}^{\text{reach} \leq N}$ if σ is Markov. As $(\sigma(n))_{(1)} = \sigma_{(n+1)}$ holds, by iteratively applying the equation, we have $\mathbb{P}_\sigma^{\text{reach} \leq N+1} = \Phi_{\sigma(0)} \dots \Phi_{\sigma(N)} \chi_C$, and hence $\mathbb{P}_\sigma^{\text{reach} \leq M+1} \sqsupseteq \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M} - (M+1)\varepsilon \cdot \mathbf{1}$. As we can find such a σ for an arbitrary $\varepsilon > 0$, we have $\overline{\mathbb{P}}^{\text{reach} \leq M+1} \sqsupseteq \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq M}$. $\square$

PROOF OF PROPOSITION 4.2.1. By lemma 4.3, we can inductively prove $\overline{\mathbb{P}}^{\text{reach} \leq N} = \overline{\Phi}^{N+1} \perp$, and by taking supremum over $N < \omega$, we have $\overline{\mathbb{P}}^{\text{reach}} = \sup_{N < \omega} \overline{\mathbb{P}}^{\text{reach} \leq N} = \overline{\Phi}^\omega \perp$. Because $\overline{\mathbb{X}}$ is ω -continuous (Proposition 2.17.4), so is $\overline{\Phi}$, and we have $\overline{\mathbb{P}}^{\text{reach}} = \mu \overline{\Phi}$ by Kleene fixed point theorem. $\square$

Last, we prove soundness and completeness as a corollary of Proposition 4.2.1. As stated in Section 2.4, our domain $\overline{\mathcal{A}}(I, [0, \infty])$ is not complete. To see this, consider any $A \subseteq I$ such that $\chi_A \notin \overline{\mathcal{A}}(I, [0, \infty])$, which is the supremum of $\{\chi_{\{c\}} \mid c \in A\} \subseteq \overline{\mathcal{A}}(I, [0, \infty])$. To complete the proof, we need the following counterpart of Knaster-Tarski theorem (Theorem 2.13.1):

PROPOSITION 4.6. *Let F be a monotone endofunction on a lattice L . Then the following hold:*

- *Let L be an ω -cpo and $F : L \rightarrow L$ be ω -continuous. Then μF exists, and $F l \leq l$ implies $\mu F \leq l$.*
- *Let L be an ω^{op} -cpo and $F : L \rightarrow L$ be ω^{op} -continuous. Then $F l \leq l$ implies $\mu F \leq l$, assuming μF exists.*

PROOF. For the ω -continuous case, we have $\mu F = \bigsqcup_{n < \omega} F^n(\perp)$ from Kleene fixed point theorem. As we have $F^n(\perp) \leq l$ for any pre-fixed point l of F and $n \in \mathbb{N}$, we have $\mu F = \bigsqcup_{n < \omega} F^n(\perp) \leq l$.

For the ω^{op} -continuous case, from the assumption, we can derive $\sqcap_{n < \omega} F^n l \leq l$, and as F is ω^{op} -continuous, $\sqcap_{n < \omega} F^n l$ is a fixed point of F . Thus, we have $\mu F \leq \sqcap_{n < \omega} F^n l \leq l$. $\square$

COROLLARY 4.7.

- (1) (Soundness) *If η is a U-NNRepSupM for C supported by I , then for each $c \in I$, we have $\overline{\mathbb{P}}^{\text{reach}}(c) \leq \eta(c)$. In other words, if $\overline{\Phi} \sqsubseteq \eta$, then $\overline{\mathbb{P}}^{\text{reach}} \sqsubseteq \eta$.*
- (2) (Completeness) *There exists a U-NNRepSupM η that gives the optimal bound for $\overline{\mathbb{P}}^{\text{reach}}_C$, that is, $\overline{\Phi} \overline{\mathbb{P}}^{\text{reach}} \sqsubseteq \overline{\mathbb{P}}^{\text{reach}}$.*

PROOF. Immediate from Proposition 4.2.1 and Proposition 4.6. $\square$

4.2 Proof of the Lower Case

Now, we turn to the lower case. The proof is rather different from the upper case because $\underline{\Phi}$ is not ω -continuous, and we cannot use Kleene fixed point theorem. The structure of the proof is similar to the one for the deterministic model in Reference [8]: We separately prove soundness and completeness of L-NNRepSupM and give the fixed-point characterization of $\underline{\mathbb{P}}^{\text{reach}}$ as a corollary. We first note that $\underline{\mathbb{P}}^{\text{reach}}$ is in the domain of $\underline{\Phi}$. See Appendix B for the proof.

PROPOSITION 4.8. *The function $\underline{\mathbb{P}}^{\text{reach}}$ is lower semianalytic.* $\square$

For soundness, we show the following proposition, from which soundness immediately follows:

PROPOSITION 4.9. *If $\eta \sqsupseteq \underline{\Phi} \eta$, then for any $\varepsilon > 0$ there exists a Markov scheduler σ such that $\underline{\mathbb{P}}^\sigma_{\text{reach}} \sqsubseteq \eta + \varepsilon \cdot 1$.*

PROOF. For a given pre-fixed point η of $\underline{\Phi}$ and $\varepsilon > 0$, let σ be a Markov policy such that $\Phi_{\sigma(n)} \eta \sqsubseteq \underline{\Phi} \eta + 2^{-n} \varepsilon \cdot 1$ for each $n \in \mathbb{N}$ (Lemma 4.4). We show the following by induction on $\mathbb{N}$: For each $N \in \mathbb{N}$, $\pi \in I^*$ and $c \in I$ such that πc is a finite path, we have

$$\underline{\mathbb{P}}^{\text{reach} \leq N}_{\sigma_\pi}(c) \leq \eta(c) + \sum_{k=|\pi|}^{|\pi|+N-1} 2^{-(k+1)} \varepsilon. \quad (6)$$

The case of $N = 0$ is immediate, as $\mathbb{P}_\sigma^{\text{reach} \leq 0} = \chi_C$ holds for any scheduler σ . For the step case,

$$\begin{aligned}
 \mathbb{P}_{\sigma_\pi}^{\text{reach} \leq N+1}(c) &= \chi_C(c) + \chi_{\bar{C}}(c) \int_I \mathbb{P}_{\sigma_{\pi c}}^{\text{reach} \leq N}(c') \mu_c^{\sigma_\pi}(dc') && \text{(Equation (5))} \\
 &\leq \chi_C(c) + \chi_{\bar{C}}(c) \int_I \eta(c') \mu_c^{\sigma_\pi}(dc') + \sum_{k=|\pi|+1}^{|\pi|+N} 2^{-(k+1)\varepsilon} && \text{(induction hypothesis)} \\
 &= \Phi_{\sigma_\pi} \eta(c) + \sum_{k=|\pi|+1}^{|\pi|+N} 2^{-(k+1)\varepsilon} && \text{(definition of } \Phi_{\sigma_\pi} \text{)} \\
 &\leq \eta(c) + \sum_{k=|\pi|}^{|\pi|+N} 2^{-(k+1)\varepsilon}. && \text{(definition of } \sigma \text{ and } \underline{\Phi} \eta \sqsubseteq \eta \text{)}
 \end{aligned}$$

Now the claim is shown to be true, and in particular, we have $\mathbb{P}_\sigma^{\text{reach} \leq N}(c) \leq \eta(c) + \sum_{k=1}^N 2^{-k}\varepsilon$. Letting $N \rightarrow \infty$ we have $\mathbb{P}_\sigma^{\text{reach}} \sqsubseteq \eta + \varepsilon \cdot \mathbf{1}$. $\square$

COROLLARY 4.10. *If $\eta \sqsupseteq \underline{\Phi} \eta$, then $\eta \sqsupseteq \underline{\mathbb{P}}^{\text{reach}}$.* $\square$

PROPOSITION 4.11. *It holds that $\underline{\Phi} \underline{\mathbb{P}}^{\text{reach}} = \underline{\mathbb{P}}^{\text{reach}}$.*

PROOF. Taking limit of N in Equation (5), we have

$$\mathbb{P}_\sigma^{\text{reach}}(c) = \chi_C(c) + \chi_{\bar{C}}(c) \int_I \mathbb{P}_{\sigma_c}^{\text{reach}}(c') \mu_c^\sigma(dc'),$$

and in a similar way to show $\overline{\mathbb{P}}^{\text{reach} \leq N+1} \sqsubseteq \overline{\Phi} \overline{\mathbb{P}}^{\text{reach} \leq N}$, we have $\underline{\Phi} \underline{\mathbb{P}}^{\text{reach}} \sqsubseteq \underline{\mathbb{P}}^{\text{reach}}$, that is, $\underline{\mathbb{P}}^{\text{reach}}$ is a pre-fixed point of $\underline{\Phi}$. Hence, by Lemma 4.5 and Proposition 4.9, there exists a Markov scheduler σ such that $\Phi_\sigma \underline{\mathbb{P}}^{\text{reach}} \sqsubseteq \underline{\Phi} \underline{\mathbb{P}}^{\text{reach}} + \varepsilon \cdot \mathbf{1}$ and $\mathbb{P}_{\sigma(1)}^{\text{reach}} \sqsubseteq \underline{\mathbb{P}}^{\text{reach}} + \varepsilon \cdot \mathbf{1}$. For such a σ , we have

$$\mathbb{P}_\sigma^{\text{reach}} = \Phi_\sigma \mathbb{P}_{\sigma(1)}^{\text{reach}} \sqsubseteq \Phi_\sigma \underline{\mathbb{P}}^{\text{reach}} + \varepsilon \cdot \mathbf{1} \sqsubseteq \underline{\Phi} \underline{\mathbb{P}}^{\text{reach}} + 2\varepsilon \cdot \mathbf{1}.$$

As ε can be any positive number, we have $\underline{\mathbb{P}}^{\text{reach}} \sqsubseteq \underline{\Phi} \underline{\mathbb{P}}^{\text{reach}}$. $\square$

PROOF OF PROPOSITION 4.2.2. Immediate from Corollary 4.10 and Proposition 4.11. $\square$

Last, we restate Proposition 4.9, Corollary 4.10, and Proposition 4.11 in a form of Corollary 4.7.

COROLLARY 4.12.

- (1) (Soundness) *If η is an L -NNRepSupM for C supported by I , then for each $\varepsilon > 0$ there is a scheduler $\sigma \in \text{Sch}_\Gamma$ such that, for any $c \in I$, we have $\mathbb{P}_\sigma^{\text{reach}}(c) \leq \eta(c) + \varepsilon$. In particular, for each $c \in I$, we have $\underline{\mathbb{P}}^{\text{reach}}(c) \leq \eta(c)$.*
- (2) (Completeness) *There exists an L -NNRepSupM η that gives the optimal bound for $\underline{\mathbb{P}}_C^{\text{reach}}$, that is, $\underline{\mathbb{P}}^{\text{reach}} \sqsubseteq \underline{\Phi} \underline{\mathbb{P}}^{\text{reach}}$.* $\square$

4.3 Relaxation of the NNRepSupM Condition

In Reference [38], the notion of c -martingale is proposed for approximating finite horizon reachability probabilities. They impose a condition that could be called an “almost-martingale” one, namely, c -martingales may increase their values through transitions but up to the constant c per step. In what follows, we provide a similar relaxation of NNRepSupMs and show its soundness via our order-theoretic framework. For simplicity, we assume the constant is independent of time, while in Reference [38] it is defined as a function over time.

Definition 4.13 (δ -NNRepSupM for pCFG). Let $\delta \geq 0$. An *upper δ -nonnegative repulsing supermartingale* (U - δ -NNRepSupM) over Γ for C supported by I is a function $\eta \in \overline{\mathcal{A}}(I, [0, \infty])$ such that

$$(i) \eta(c) \geq 1 \text{ for each } c \in C, \text{ and } (ii) \eta(c) \geq \overline{\mathbb{X}}\eta(c) - \delta \text{ for each } c \in I \setminus C.$$

The function $\eta \in \underline{\mathcal{A}}(I, [0, \infty])$ is a *lower δ -nonnegative repulsing supermartingale* (L - δ -NNRepSupM) if it satisfies the above conditions, but with $\overline{\mathbb{X}}$ replaced with $\underline{\mathbb{X}}$.

Equivalently, upper and lower δ -NNRepSupMs are pre-fixed points of the following monotone endofunctions $\overline{\Phi}_\delta$ over $\overline{\mathcal{A}}(I, [0, \infty])$ and $\underline{\Phi}_\delta$ over $\underline{\mathcal{A}}(I, [0, \infty])$, respectively:

$$\begin{aligned} \overline{\Phi}_\delta \eta &= \chi_C + \chi_{\overline{C}} \cdot (\overline{\mathbb{X}}\eta - \delta \cdot \mathbf{1}) = \overline{\Phi}\eta - \delta \chi_{\overline{C}}, \\ \underline{\Phi}_\delta \eta &= \chi_C + \chi_{\overline{C}} \cdot (\underline{\mathbb{X}}\eta - \delta \cdot \mathbf{1}) = \underline{\Phi}\eta - \delta \chi_{\overline{C}}. \end{aligned}$$

Notice that a 0-NNRepSupM is just a usual NNRepSupM. each upper and lower δ -NNRepSupM overapproximates $\overline{\mathbb{P}}^{\text{reach} \leq N}$ and $\underline{\mathbb{P}}^{\text{reach} \leq N}$, respectively (Definition 2.9). Their soundness is given as the following proposition: Notice that the obtained bound is looser than the supermartingale itself, which is the price for a relaxed constraint.

PROPOSITION 4.14 (SOUNDNESS). *Let $\delta \geq 0$ and $N \in \mathbb{N}$ be given. If η is a U - δ -NNRepSupM for C supported by I , then for each $c \in I$, we have $\eta(c) + N\delta \geq \overline{\mathbb{P}}^{\text{reach} \leq N}(c)$. Similarly, if η is an L - δ -NNRepSupM for C supported by I , then for each $c \in I$, we have $\eta(c) + N\delta \geq \underline{\mathbb{P}}^{\text{reach} \leq N}(c)$.*

PROOF. We show the upper case only, as the lower case can be shown in an almost identical way. In what follows, we prove the following inequality holds for each $N \in \mathbb{N}$:

$$\eta \sqsupseteq \overline{\Phi}^N \eta - N\delta \chi_{\overline{C}}. \quad (7)$$

Notice that $\eta \sqsupseteq \chi_{\overline{C}} = \overline{\Phi} \perp$ holds, and we know that $\overline{\Phi}^{N+1} \perp = \overline{\mathbb{P}}^{\text{reach} \leq N}$ holds from the proof of Proposition 4.2.1; thus from Equation (7), we conclude $\eta \sqsupseteq \overline{\Phi}^N \eta - N\delta \chi_{\overline{C}} \sqsupseteq \overline{\mathbb{P}}^{\text{reach} \leq N} - N\delta \chi_{\overline{C}}$.

We prove the Equation (7) by induction on $\mathbb{N}$. It is just an identity if $N = 0$, and it is nothing but the U - δ -NNRepSupM condition if $N = 1$. Now assume $\eta \sqsupseteq \overline{\Phi}^N \eta - N\delta \chi_{\overline{C}}$ for $N \geq 1$; then, we have $\overline{\Phi}\eta \sqsupseteq \overline{\Phi}(\overline{\Phi}^N \eta - N\delta \chi_{\overline{C}}) \sqsupseteq \overline{\Phi}^{N+1} \eta - N\delta \chi_{\overline{C}}$, where the second inequality holds due to Lemma 4.4. Thus, we have $\eta \sqsupseteq \overline{\Phi}\eta - \delta \chi_{\overline{C}} \sqsupseteq \overline{\Phi}^{N+1} \eta - (N+1)\delta \chi_{\overline{C}}$. $\square$

4.4 Flipping the Direction of Approximation

So far, we have developed the theory of NNRepSupM as a technique of over-approximation. Here, we show how it can also *under*-approximate reachability probability when we have certain *a priori* knowledge about the program to be analyzed. The idea is similar to an application of ϵ -RepSupM proposed in Reference [15].

As a simple case, let a randomized program without nondeterminism be given, and suppose we know that the program either terminates *or aborts* with probability 1 (such a property can be verified via ARnkSupM by letting the terminal region to be the set of terminating and aborting configurations). In such a case, an NNRepSupM can *under*-approximate reachability probability by over-approximating the probability of abortion. Concretely, if $p \in [0, 1]$ is an upper bound of the probability that the program aborts, then $1 - p$ is a lower bound of the probability that the program terminates. This argument is also valid under the existence of nondeterminism as follows:

PROPOSITION 4.15. *Let $C_1, C_2 \in \mathcal{B}(I)$ be mutually disjoint sets of configurations. For a given $c \in I$, if $\mathbb{P}_{C_1 \cup C_2, \sigma}^{\text{reach}}(c) = 1$ for every $\sigma \in \text{Sch}_\Gamma$, then we have $\overline{\mathbb{P}}_{C_1}^{\text{reach}}(c) + \underline{\mathbb{P}}_{C_2}^{\text{reach}}(c) = 1$.*

PROOF. Notice that $\mathbb{P}_{C_1 \cup C_2, \sigma}^{\text{reach}}(c) = \mathbb{P}_{C_1, \sigma}^{\text{reach}}(c) + \mathbb{P}_{C_2, \sigma}^{\text{reach}}(c)$ holds for every $\sigma \in \text{Sch}_\Gamma$ as C_1 and C_2 are mutually disjoint. By definition, we have $\sup_{\sigma \in \text{Sch}_\Gamma} \mathbb{P}_{C_1, \sigma}^{\text{reach}}(c) = \overline{\mathbb{P}}_{C_1}^{\text{reach}}(c)$. Thus, we have

$$\underline{\mathbb{P}}_{C_2}^{\text{reach}}(c) = \inf_{\sigma \in \text{Sch}_\Gamma} (1 - \mathbb{P}_{C_1, \sigma}^{\text{reach}}(c)) = 1 - \sup_{\sigma \in \text{Sch}_\Gamma} \mathbb{P}_{C_1, \sigma}^{\text{reach}}(c) = 1 - \overline{\mathbb{P}}_{C_1}^{\text{reach}}(c),$$

and hence $\overline{\mathbb{P}}_{C_1}^{\text{reach}}(c) + \underline{\mathbb{P}}_{C_2}^{\text{reach}}(c) = 1$. $\square$

COROLLARY 4.16. *Let C_1 and C_2 as in Proposition 4.15, and suppose $\mathbb{P}_{C_1 \cup C_2, \sigma}^{\text{reach}} = 1$ for every $\sigma \in \text{Sch}_\Gamma$. If $\eta : I \rightarrow [0, 1]$ satisfies $\overline{\mathbb{P}}_{C_1}^{\text{reach}} \sqsubseteq \eta$, then $\underline{\mathbb{P}}_{C_2}^{\text{reach}} \sqsupseteq 1 - \eta$; if it satisfies $\underline{\mathbb{P}}_{C_1}^{\text{reach}} \sqsubseteq \eta$, then $\overline{\mathbb{P}}_{C_2}^{\text{reach}} \sqsupseteq 1 - \eta$. $\square$*

5 ADDITIVE RANKING SUPERMARTINGALES (ARNKSUPM)

We move on to the notion of additive ranking supermartingale (ARnkSupM) in Table 1. It is the best-known martingale-based notion for analysis of randomized programs and is used for over-approximating the expected reaching time. That its value is finite implies almost-sure reachability, too. We review its theory; the reason is to demonstrate that the same order-theoretic structure (see Section 2.4) underlies ARnkSupM and NNRepSupM in the previous section. The completeness result ((2) of Corollary 5.3) for pCFGs with real-valued variables seems new, too; see Section 9 for a detailed comparison to existing works.

We note that completeness of U-ARnkSupM stated below is the one for *strong* almost-sure reachability [6], which asserts existence of $N \in \mathbb{R}$ such that $\forall \sigma. \mathbb{E}_{C, \sigma}^{\text{steps}}(c) < N$ for every $\sigma \in \text{Sch}_\Gamma$. U-ARnkSupM is incomplete for *positive* almost-sure reachability [6, Example 9], that is, it cannot witness the condition $\forall \sigma. \mathbb{E}_{C, \sigma}^{\text{steps}}(c) < \infty$ in general.

Definition 5.1 (ARnkSupM for pCFG, [10]). An *upper additive ranking supermartingale* (U-ARnkSupM) over a pCFG Γ for a terminal region C supported by an invariant I is an upper semianalytic function $\eta \in \overline{\mathcal{A}}(I, [0, \infty])$ that satisfies $\eta(c) \geq 1 + \overline{\mathbb{X}}\eta(c)$ for each $c \in I \setminus C$.

A lower semianalytic function $\eta \in \underline{\mathcal{A}}(I, [0, \infty])$ is a *lower additive ranking supermartingale* (L-ARnkSupM) if it satisfies the above conditions, but with $\overline{\mathbb{X}}$ replaced with $\underline{\mathbb{X}}$.

Equivalently, upper and lower NNRepSupMs are pre-fixed points of the following monotone endofunctions $\overline{\Psi}$ over $\overline{\mathcal{A}}(I, [0, \infty])$ and $\underline{\Psi}$ over $\underline{\mathcal{A}}(I, [0, \infty])$, respectively:

$$\overline{\Psi}(\eta) = \chi_C^-(1 + \overline{\mathbb{X}}\eta), \quad \underline{\Psi}(\eta) = \chi_C^-(1 + \underline{\mathbb{X}}\eta).$$

Similar to the case of NNRepSupM, we formalize soundness and completeness of ARnkSupM in terms of fixed points as follows: Again, we omit the subscript C from notations when it is clear from the context (e.g., we write $\overline{\mathbb{E}}^{\text{steps}}$ instead of $\overline{\mathbb{E}}_C^{\text{steps}}$).

PROPOSITION 5.2 (FIXED POINT CHARACTERIZATION OF $\overline{\mathbb{E}}^{\text{steps}}$ AND $\underline{\mathbb{E}}^{\text{steps}}$). *The following are true:*

- (1) $\overline{\mathbb{E}}^{\text{steps}} = \mu \overline{\Psi}$.
- (2) $\underline{\mathbb{E}}^{\text{steps}} = \mu \underline{\Psi}$.

PROOF. The proof is similar to the one of Proposition 4.2, so we mostly omit it. We just give the following equation that corresponds to Equation (5):

$$\begin{aligned}
\mathbb{E}_{\sigma}^{\text{steps} \leq N+1}(c) &= \left(\sum_{i=0}^{N+1} \mathbb{P}_{\sigma}^{\text{reach}=i}(c) \cdot i \right) + \mathbb{P}_{\sigma}^{\neg \text{reach} \leq N+1}(c) \cdot (N+1) \\
&= \left(\sum_{i=1}^{N+1} \mathbb{P}_{\sigma}^{\text{reach}=i}(c) \right) + \mathbb{P}_{\sigma}^{\neg \text{reach} \leq N+1}(c) \\
&\quad + \left(\sum_{i=0}^N \mathbb{P}_{\sigma}^{\text{reach}=i+1}(c) \cdot i \right) + \mathbb{P}_{\sigma}^{\neg \text{reach} \leq N+1}(c) \cdot N \\
&= \chi_{\bar{C}}(c) \left(1 + \int_I \mathbb{E}_{\sigma_c}^{\text{steps} \leq N}(c_1) \mu_c^{\sigma}(dc_1) \right).
\end{aligned}$$

To derive the last equality, notice that $\mathbb{P}_{\sigma}^{\text{reach} \leq N} + \mathbb{P}_{\sigma}^{\neg \text{reach} \leq N} = 1$ holds for each $N \in \mathbb{N}$ and $\sigma \in \text{Sch}_{\Gamma}$; also recall Equations (1) and (2). $\square$

COROLLARY 5.3.

- (1) (Soundness, e.g., Reference [2]) If η is a U-ARnkSupM for C supported by I , then for each $c \in I$, we have $\overline{\mathbb{E}}^{\text{steps}}(c) \leq \eta(c)$. In other words, if $\overline{\Psi}\eta \sqsubseteq \eta$, then $\overline{\mathbb{E}}^{\text{steps}} \sqsubseteq \eta$. In particular, for each $c \in I$ that satisfies $\eta(c) < \infty$, we have $\underline{\mathbb{P}}^{\text{reach}}(c) = 1$.

Similarly, if η is an L-ARnkSupM for C supported by I , then for each $\varepsilon > 0$ there is a scheduler $\sigma \in \text{Sch}_{\Gamma}$ such that, for any $c \in I$, we have $\mathbb{E}_{\sigma}^{\text{steps}}(c) \leq \eta(c) + \varepsilon$. In particular, for each $c \in I$, we have $\underline{\mathbb{E}}^{\text{steps}}(c) \leq \eta(c)$, and for each $c \in I$ that satisfies $\eta(c) < \infty$, we have $\overline{\mathbb{P}}^{\text{reach}}(c) = 1$.

- (2) (Completeness) There exists a U-ARnkSupM η that gives the optimal bound on $\overline{\mathbb{E}}_C^{\text{steps}}$, that is, $\overline{\Psi}\mathbb{E}^{\text{steps}} \sqsubseteq \overline{\mathbb{E}}^{\text{steps}}$. The same holds for L-ARnkSupM. $\square$

6 γ -SCALED SUBMARTINGALES (γ -SCLSUBM)

Here, we present the theory of γ -scaled submartingales (γ -SclSubM). It is for under-approximating reachability (Table 1). Compared to the well-known method of ARnkSupM, the greatest advantage is in *quantitative* reasoning: The value of a γ -SclSubM is guaranteed to be below the reachability probability (which can be less than 1), while ARnkSupM is useful only if almost-sure reachability holds.

The notion of γ -SclSubM is first introduced in Reference [41] as an instance of a categorical abstraction of ranking functions. The current article's contribution lies in the following: (i) the theoretical developments about γ -SclSubM in concrete (non-categorical) terms; (ii) introduction of nondeterminism (the setting of Reference [41] is purely probabilistic); and (iii) template-based synthesis of γ -SclSubM (see Section 7).

Definition 6.1 (γ -SclSubM for pCFG, [41]). Let $\gamma \in (0, 1)$ be given. An *upper γ -Scaled Submartingale* (U- γ -SclSubM) over a pCFG Γ for a terminal region C supported by an invariant I is an upper semianalytic function $\eta \in \overline{\mathcal{A}}(I, [0, 1])$ that satisfies $\eta(c) \leq \gamma \cdot \overline{\mathbb{X}}\eta(c)$ for each $I \setminus C$.

A *lower γ -Scaled Submartingale* (L- γ -SclSubM) over Γ for C supported by I is a lower semianalytic function $\eta \in \underline{\mathcal{A}}(I, [0, 1])$ that satisfies $\eta(c) \leq \gamma \cdot \underline{\mathbb{X}}\eta(c)$ for each $I \setminus C$.

Equivalently, upper and lower γ -SclSubM are pre-fixed points of the following monotone end-ofunctions $\bar{\Phi}_\gamma$ over $\bar{\mathcal{A}}(I, [0, 1])$ and $\underline{\Phi}_\gamma$ over $\underline{\mathcal{A}}(I, [0, 1])$, respectively:

$$\bar{\Phi}_\gamma(\eta) = \chi_C + \gamma \chi_{\bar{C}} \cdot (\bar{\mathbb{X}}\eta), \quad \underline{\Phi}_\gamma(\eta) = \chi_C + \gamma \chi_{\bar{C}} \cdot (\underline{\mathbb{X}}\eta).$$

Soundness of (say, an upper) γ -SclSubM comes from the fact that it under-approximates the “ γ -discounted” reachability probability $\bar{\mathbb{P}}_\gamma^{\text{reach}}$ (see Definition 2.9). The trick here is as follows: (i) the function $\bar{\mathbb{P}}_\gamma^{\text{reach}}$ is the lfp of $\bar{\Phi}_\gamma$; and (ii) the gfp and lfp of $\bar{\Phi}_\gamma$ coincide. By these facts and (gfp-KT) in Corollary 2.14, we can show that each γ -SclSubM under-approximates $\bar{\mathbb{P}}_\gamma^{\text{reach}}$, which is below the reachability probability $\bar{\mathbb{P}}^{\text{reach}}$ by definition. The overall argument signifies the role of the Knaster–Tarski theorem.

PROPOSITION 6.2 (FIXED POINT CHARACTERIZATION OF $\bar{\mathbb{P}}_\gamma^{\text{reach}}$ AND $\underline{\mathbb{P}}_\gamma^{\text{reach}}$). *The following are true:*

- (1) $\bar{\mathbb{P}}_\gamma^{\text{reach}} = \mu \bar{\Phi}_\gamma = \nu \bar{\Phi}_\gamma$.
- (2) $\underline{\mathbb{P}}_\gamma^{\text{reach}} = \mu \underline{\Phi}_\gamma = \nu \underline{\Phi}_\gamma$.

COROLLARY 6.3 (SOUNDNESS). *If η is a U - γ -SclSubM for C supported by I , then for each $c \in I$, we have $\bar{\mathbb{P}}^{\text{reach}}(c) \geq \eta(c)$.*

Similarly, if η is an L - γ -SclSubM for C supported by I , then for each $c \in I$, we have $\underline{\mathbb{P}}^{\text{reach}}(c) \geq \eta(c)$.

PROOF OF PROPOSITION 6.2. Similar to Lemma 4.3, we have $\bar{\mathbb{P}}_\gamma^{\text{reach} \leq N+1} = \bar{\Phi}_\gamma \bar{\mathbb{P}}_\gamma^{\text{reach} \leq N}$ and $\underline{\mathbb{P}}_\gamma^{\text{reach} \leq N+1} = \underline{\Phi}_\gamma \underline{\mathbb{P}}_\gamma^{\text{reach} \leq N}$, and hence $\bar{\mathbb{P}}_\gamma^{\text{reach}} = \mu \bar{\Phi}_\gamma$.

The proof of $\underline{\mathbb{P}}_\gamma^{\text{reach}} = \mu \underline{\Phi}_\gamma$ can be simpler than that of Proposition 4.2.2 as follows: As for any $\sigma \in \text{Sch}_\Gamma$ and $N \in \mathbb{N}$ it holds that $\bar{\mathbb{P}}_{\sigma, \gamma}^{\text{reach}} \subseteq \bar{\mathbb{P}}_{\sigma, \gamma}^{\text{reach} \leq N} + \gamma^N \cdot \mathbf{1}$, we have $\underline{\mathbb{P}}_\gamma^{\text{reach}} \subseteq \underline{\mathbb{P}}_\gamma^{\text{reach} \leq N} + \gamma^N \cdot \mathbf{1}$, and hence $\underline{\mathbb{P}}_\gamma^{\text{reach}} \subseteq \sup_{N \in \mathbb{N}} \underline{\mathbb{P}}_\gamma^{\text{reach} \leq N} = \underline{\Phi}_\gamma \perp$. The opposite inequality is easily verified, and thus we have $\underline{\mathbb{P}}_\gamma^{\text{reach}} = \underline{\Phi}_\gamma \perp$. By the inequality $\underline{\Phi}_\gamma \perp \subseteq \underline{\mathbb{P}}_\gamma^{\text{reach} \leq N} + \gamma^N \cdot \mathbf{1}$ and (a modification of) Lemma 4.4, we have $\underline{\Phi}_\gamma(\underline{\Phi}_\gamma \perp) \subseteq \underline{\mathbb{P}}_\gamma^{\text{reach} \leq M+1} + \gamma^M \cdot \mathbf{1}$ for any $M \in \mathbb{N}$, and hence $\underline{\Phi}_\gamma(\underline{\Phi}_\gamma \perp) \subseteq \underline{\Phi}_\gamma \perp$. Again the opposite inequality is easily verified, and we have $\underline{\Phi}_\gamma(\underline{\Phi}_\gamma \perp) = \underline{\Phi}_\gamma \perp$, which proves $\underline{\Phi}_\gamma \perp = \underline{\mathbb{P}}_\gamma^{\text{reach}} = \mu \underline{\Phi}_\gamma$.

Now, we prove $\bar{\mathbb{P}}_\gamma^{\text{reach}} = \nu \bar{\Phi}_\gamma$; the lower case is proved in a similar way. Let η be any fixed point of $\bar{\Phi}_\gamma$. We show the following by induction on $\mathbb{N}$: For any $N \in \mathbb{N}$, we have $\eta \subseteq \bar{\mathbb{P}}_\gamma^{\text{reach} \leq N} + \gamma^N \cdot \chi_{\bar{C}}$. For the case of $N = 0$ the inequality is true (it just asserts $\eta \subseteq \mathbf{1}$). For the step case, we have

$$\begin{aligned} \eta(c) &= \bar{\Phi}_\gamma \eta(c) \\ &\leq \bar{\Phi}_\gamma(\bar{\mathbb{P}}_\gamma^{\text{reach} \leq N} + \gamma^N \cdot \chi_{\bar{C}})(c) \\ &= \sup_{\sigma} \left(\chi_C(c) + \chi_{\bar{C}}(c) \cdot \gamma \int_I \bar{\mathbb{P}}_\gamma^{\text{reach} \leq N}(c') + \gamma^N \cdot \chi_{\bar{C}}(c') \mu_c^\sigma(\text{dc}') \right) \\ &\leq \sup_{\sigma} \left(\chi_C(c) + \chi_{\bar{C}}(c) \cdot \gamma \int_I \bar{\mathbb{P}}_\gamma^{\text{reach} \leq N}(c') \mu_c^\sigma(\text{dc}') \right) + \gamma^{N+1} \cdot \chi_{\bar{C}}(c) \\ &= \bar{\Phi}_\gamma \bar{\mathbb{P}}_\gamma^{\text{reach} \leq N}(c) + \gamma^{N+1} \cdot \chi_{\bar{C}}(c) \\ &= \bar{\mathbb{P}}_\gamma^{\text{reach} \leq N+1}(c) + \gamma^{N+1} \cdot \chi_{\bar{C}}(c). \end{aligned}$$

This proves the claim, and thus we have $\eta \sqsubseteq \sup_{N \in \mathbb{N}} \overline{\mathbb{P}}_{\gamma}^{\text{reach} \leq N} = \mu \overline{\Phi}_{\gamma}$ by letting $N \rightarrow \infty$. Hence, $\mu \overline{\Phi}_{\gamma}$ is also the greatest fixed point of $\overline{\Phi}_{\gamma}$. $\square$

Before the proof of soundness, we note the following is proved in a similar way to Proposition 4.6:

COROLLARY 6.4. *Let $F : L \rightarrow L$ be a monotone endofunction on a lattice L . Then the following hold:*

- *If L is an ω -cpo and F is ω -continuous, then $F l \geq l$ implies $\nu F \geq l$, assuming νF exists.*
- *If L is an ω^{op} -cpo and F is ω^{op} -continuous, then νF exists, and $F l \geq l$ implies $\nu F \geq l$.* $\square$

PROOF OF COROLLARY 6.3. If η is a γ -SclSubM, then by Corollary 6.4 and Proposition 6.2, we have $\eta \sqsubseteq \nu \overline{\Phi}_{\gamma} = \overline{\mathbb{P}}_{\gamma}^{\text{reach}} \sqsubseteq \overline{\mathbb{P}}^{\text{reach}}$. Notice that the last inequality is easily derived from the definition of $\overline{\mathbb{P}}_{\gamma}^{\text{reach}}$ and $\overline{\mathbb{P}}^{\text{reach}}$. The lower case is proved in a similar way. $\square$

6.1 On Completeness of γ -SclSubM

In the proof of Proposition 6.2, we have shown that γ -SclSubM under-approximates $\overline{\mathbb{P}}_{\gamma}^{\text{reach}}$ and $\underline{\mathbb{P}}_{\gamma}^{\text{reach}}$. These functions can be strictly below $\overline{\mathbb{P}}^{\text{reach}}$ and $\underline{\mathbb{P}}^{\text{reach}}$, respectively, and hence γ -SclSubM is not complete for under-approximating $\overline{\mathbb{P}}^{\text{reach}}$ and $\underline{\mathbb{P}}^{\text{reach}}$; i.e., it is not always possible to find a γ -SclSubM that gives an optimal bound. Still, in certain cases, one can find a γ -SclSubM that gives an arbitrarily close bound by taking γ sufficiently close to 1. Concretely, we have the following fact:

COROLLARY 6.5. *The following are true:*

- (1) *For any $\varepsilon > 0$, there exists $\gamma \in (0, 1)$ such that $\overline{\mathbb{P}}^{\text{reach} \leq N}(c) - \overline{\mathbb{P}}_{\gamma}^{\text{reach} \leq N}(c) < \varepsilon$ for every $c \in I$.*
- (2) *For any $\varepsilon > 0$, there exists $\gamma \in (0, 1)$ such that $\underline{\mathbb{P}}^{\text{reach} \leq N}(c) - \underline{\mathbb{P}}_{\gamma}^{\text{reach} \leq N}(c) < \varepsilon$ for every $c \in I$.*
- (3) *For any $\varepsilon > 0$ and $c \in I$, there exists $\gamma \in (0, 1)$ such that $\overline{\mathbb{P}}^{\text{reach}}(c) - \overline{\mathbb{P}}_{\gamma}^{\text{reach}}(c) < \varepsilon$.*

PROOF. By definition, we have $\gamma^N \cdot \overline{\mathbb{P}}_{\sigma}^{\text{reach} \leq N} \sqsubseteq \overline{\mathbb{P}}_{\sigma, \gamma}^{\text{reach} \leq N} \sqsubseteq \overline{\mathbb{P}}_{\sigma}^{\text{reach} \leq N}$ for each $\sigma \in \text{Sch}_{\Gamma}$. The claim (1) and (2) are immediate from this inequality.

For the claim (3), recall from the proof of Proposition 4.2 that $\sup_{N < \omega} \overline{\mathbb{P}}^{\text{reach} \leq N} = \overline{\mathbb{P}}^{\text{reach}}$ holds. For a given $c \in I$ and $\varepsilon > 0$, take N that satisfies $\overline{\mathbb{P}}^{\text{reach}}(c) - \overline{\mathbb{P}}^{\text{reach} \leq N}(c) < \frac{\varepsilon}{2}$, and then take $\gamma \in (0, 1)$ that satisfies $\overline{\mathbb{P}}^{\text{reach} \leq N}(c) - \overline{\mathbb{P}}_{\gamma}^{\text{reach} \leq N}(c) < \frac{\varepsilon}{2}$. $\square$

Example 6.6. This convergence result does not hold for lower infinite horizon case, i.e., there can be a case that $\underline{\mathbb{P}}_{\gamma}^{\text{reach}}(c)$ does not converge to $\underline{\mathbb{P}}^{\text{reach}}(c)$ as $\gamma \rightarrow 1$. Indeed, in Figure 3, we have $\underline{\mathbb{P}}_{\{l_3\} \times \mathbb{R}}^{\text{reach}}(l_0, 0) = \frac{1}{2}$, as the system eventually visits the location l_3 ; but $\underline{\mathbb{P}}_{\{l_3\} \times \mathbb{R}, \gamma}^{\text{reach}}(l_0, 0) = 0$ for any $\gamma \in (0, 1)$, as one can delay the system to reach l_3 for an arbitrarily many steps by choosing small x in the update at l_0 . $\blacksquare$

7 AUTOMATED SYNTHESIS ALGORITHM OF MARTINGALE-BASED CERTIFICATES

In Sections 3–6, we discussed four different martingale techniques. Each kind of martingale under- or over-approximates the corresponding characteristics of the given system (namely, reachability probability or expected reaching time); this property is guaranteed by soundness of martingale techniques. In this section, we introduce algorithms that synthesize martingales for a given randomized program so we can obtain a concrete bound of these characteristics. We provide

algorithms that synthesize our new martingales, U-NNRepSupM (Section 4) and L- γ -SclSubM (Section 6).³ For their construction, we follow a standard technique used to synthesize ARnkSupM and ε -RepSupM [10, 11, 13, 15].

Here, we give an overview of U-NNRepSupM synthesis; L- γ -SclSubM synthesis is also done in a similar way. Suppose we are given (i) a randomized program and a pCFG Γ as its semantics, (ii) a target region C of Γ , and (iii) an invariant I for Γ . Then, we would like to find a U-NNRepSupM that bounds the characteristics $\overline{\mathbb{P}}_C^{\text{reach}}$ from above, as tight as possible. To this end, we can think of the following optimization problem: Here, $f|_I$ is the restriction of f over I .

$$\begin{aligned} & \underset{f: L \times \mathbb{R}^V \rightarrow \mathbb{R}}{\text{minimize}} && f(l_{\text{init}}, \vec{x}_{\text{init}}) \\ & \text{subject to} && (f|_I \text{ is a U-NNRepSupM over } \Gamma \text{ for } C \text{ supported by } I). \end{aligned} \quad (8)$$

We minimize the objective function $f(l_{\text{init}}, \vec{x}_{\text{init}})$, because an NNRepSupM over-approximates $\overline{\mathbb{P}}_C^{\text{reach}}$; to synthesize a martingale for under-approximation such as γ -SclSubM, we maximize $f(l_{\text{init}}, \vec{x}_{\text{init}})$ instead.

While this problem itself is too complex to be efficiently solved, one can come up with its restricted variant over which standard optimization tools (e.g., LP or SDP solver) are feasible. We in particular restrict the search space by fixing a *template* of the function f , i.e., by assuming f is in, e.g., a linear or polynomial form. In what follows, we introduce the details of linear and polynomial template-based synthesis algorithms of U-NNRepSupM and L- γ -SclSubM.

We first introduce necessary notions. Let $V = \{x_1, \dots, x_{|V|}\}$ be a finite set of variables.

Definition 7.1 (Expression). A *linear expression* over V is a formula defined by the following BNF notation: $a ::= r_1 \mid x \mid r_2 a \mid a + a$ where $x \in V$ and $r_1, r_2 \in \mathbb{R}$. We inductively define $\llbracket a \rrbracket : \mathbb{R}^V \rightarrow \mathbb{R}$ by $\llbracket r_1 \rrbracket(c) := r_1$, $\llbracket x \rrbracket(c) := c(x)$, $\llbracket r_2 a \rrbracket(c) := r_2 \cdot \llbracket a \rrbracket(c)$, and $\llbracket a_1 + a_2 \rrbracket(c) := \llbracket a_1 \rrbracket(c) + \llbracket a_2 \rrbracket(c)$. A *polynomial expression* over V is defined by: $b ::= r_1 \mid x \mid b \cdot b \mid b + b$. We define $\llbracket b \rrbracket : \mathbb{R}^V \rightarrow \mathbb{R}$ similarly to $\llbracket a \rrbracket$.

We follow Reference [15] and call linear inequalities *linear constraints*.

Definition 7.2 (Constraint/(Conjunctive) Predicate). A *linear constraint* over V is a formula $a \triangleright 0$ where $\triangleright \in \{\geq, >\}$ and a is a linear expression over V . A *linear conjunctive predicate* over V is a formula $r_1 \wedge \dots \wedge r_m$ where each r_i is a linear constraint. A *linear predicate* over V is a formula $q_1 \vee \dots \vee q_n$ where each q_i is a linear conjunctive predicate. We define $\llbracket a \triangleright 0 \rrbracket$, $\llbracket r_1 \wedge \dots \wedge r_m \rrbracket$, $\llbracket q_1 \vee \dots \vee q_n \rrbracket \subseteq \mathbb{R}^V$ by $\llbracket a \triangleright 0 \rrbracket := \{\vec{x} \mid \llbracket a \rrbracket(\vec{x}) \triangleright 0\}$, $\llbracket r_1 \wedge \dots \wedge r_m \rrbracket := \{\vec{x} \mid \vec{x} \in \llbracket r_1 \rrbracket \cap \dots \cap \llbracket r_m \rrbracket\}$, and $\llbracket q_1 \vee \dots \vee q_n \rrbracket := \{v \mid v \in \llbracket q_1 \rrbracket \cup \dots \cup \llbracket q_n \rrbracket\}$. *Polynomial constraint* and *(conjunctive) predicate* are similarly defined.

Definition 7.3 (Expression/Predicate Map). A *linear expression map* over a pCFG Γ is a family $\{\mathfrak{f}(l)\}_{l \in L}$ of linear expressions over V . Its *semantics* $\llbracket \mathfrak{f} \rrbracket : L \times \mathbb{R}^V \rightarrow \mathbb{R}$ is defined by $\llbracket \mathfrak{f} \rrbracket(l, \vec{x}) := \llbracket \mathfrak{f}(l) \rrbracket(\vec{x})$. A *linear predicate map* over Γ is a tuple $\mathfrak{P} = \{\mathfrak{P}(l)\}_{l \in L}$ of linear predicates over V . We write $\llbracket \mathfrak{P} \rrbracket$ for $\{(l, \vec{x}) \mid \vec{x} \in \llbracket \mathfrak{P}(l) \rrbracket\} \subseteq L \times \mathbb{R}^V$. If each $\mathfrak{P}(l)$ is a linear conjunctive predicate, then it is called a *linear conjunctive predicate map*. Notions of polynomial expression map and polynomial (conjunctive) predicate map are similarly defined.

Last, for expressions $\mathfrak{f}$ and $\mathfrak{g}$, we write $\mathfrak{f} \equiv \mathfrak{g}$ if $\llbracket \mathfrak{f} \rrbracket = \llbracket \mathfrak{g} \rrbracket$.

³Our algorithm works for *upper* NNRepSupM synthesis only; it is not possible to construct a *lower* NNRepSupM synthesis algorithm in a similar way. This is because constraints of a form $\forall c. f(c) \geq \underline{\mathbb{X}}f(c)$ is computationally more complex than $\forall c. f(c) \geq \overline{\mathbb{X}}f(c)$. (They are still decidable because of the decidability of the first-order theory over reals (see, e.g., Reference [40]).) An analogous discussion for ARnkSupM is found in Reference [14]. For a similar reason, we only introduce a synthesis algorithm for L- γ -SclSubM.

7.1 Linear Template-based Synthesis

7.1.1 Linear Template-based Synthesis of U-NNRepSupM. We first consider linear template-based synthesis that can be efficiently executed by an LP solver. It solves the problem (8), which is reduced in the following way:

- (1) We assume the input program is an ARP (Definition 2.11).
- (2) We fix a linear template for f , i.e., we assume the function $f(l, _)$ is linear for each $l \in L$.
- (3) We reduce the axioms of U-NNRepSupM to ones without global quantifiers.

Below, we describe the detail of these reductions step-by-step. First, we restrict an input program to an ARP together with some additional constraints. Concretely, we assume an input is the following triple: a pCFG Γ as a semantics of an ARP, a target region C for Γ , and an invariant I^4 for Γ . Without contradiction, we assume Γ , I , and C have the following properties: Here, we let $x_{|V|+1}$ be a fresh variable and $V' = V \cup \{x_{|V|+1}\}$.

ASSUMPTION 7.4.

- (1) For each $l \in L_A$ such that $\text{Up}(l) = (v, u)$,
 - (a) if $u \in \mathcal{B}(\mathbb{R}^V, \mathbb{R})$, then a linear expression a over V such that $u = \llbracket a \rrbracket$ is given;
 - (b) if $u \in \mathcal{D}(\mathbb{R})$, then its expectation $\mathbb{E}u \in \mathbb{R}$ is known; and
 - (c) if $u \in \mathcal{B}(\mathbb{R})$, then a linear predicate $\mathfrak{P}$ over $\{x_{|V|+1}\}$ such that $u = \llbracket \mathfrak{P} \rrbracket$ is given.
- (2) For each $l \in L_D$ and $l' \in \text{succ}(l)$, a linear predicate $\mathfrak{b}$ over V such that $G(l, l') = \llbracket \mathfrak{b} \rrbracket$ is given.
- (3) The target region C is the semantics of some linear conjunctive predicate map $\mathfrak{C}$, i.e., $C = \llbracket \mathfrak{C} \rrbracket$.
- (4) The invariant I is the semantics of some linear predicate map $\mathfrak{I}$, i.e., $I = \llbracket \mathfrak{I} \rrbracket$.

Next, we fix a linear template for f in (8), i.e., we assume the function $f(l, _) : \mathbb{R}^V \rightarrow \mathbb{R}$ is linear for each $l \in L$. Syntactically, a *linear template* is an expression map $\mathfrak{f}$ of the following form with unknown coefficients $U = \bigcup_{l \in L} \{a_1^l, \dots, a_{|V|}^l, b^l\}$:

$$\mathfrak{f}(l) := a_1^l x_1 + \dots + a_{|V|}^l x_{|V|} + b^l, \quad l \in L. \quad (9)$$

Once we specify a valuation of U (that is, a map $u : U \rightarrow \mathbb{R}$), the template $\mathfrak{f}$ turns into a linear expression map $\mathfrak{f}_u$ defined by $\mathfrak{f}_u(l) := u(a_1^l)x_1 + \dots + u(a_{|V|}^l)x_{|V|} + u(b^l)$ for each $l \in L$. We assume f is given as the semantics of such an expression map, i.e., $f = \llbracket \mathfrak{f}_u \rrbracket$ for some valuation u .

Last, we reduce the axioms of U-NNRepSupM to ones without global quantifiers. Here, the axioms of U-NNRepSupM are formulas $\varphi_1, \dots, \varphi_k$ over U such that

$$\begin{aligned} & \varphi_1, \dots, \varphi_k \text{ are true under a valuation } u \\ \iff & \llbracket \mathfrak{f}_u \rrbracket \upharpoonright_I \text{ is a U-NNRepSupM over } \Gamma \text{ for } C \text{ supported by } I. \end{aligned}$$

Our goal is to reduce each φ_i to a formula of the form $\exists \vec{v}'. \hat{\varphi}_i$, where $\vec{v}' \in \mathbb{R}^{U'}$ is a vector over a fresh variable set U' and $\hat{\varphi}_i$ is a linear conjunctive predicate over $U \cup U'$. To this end, we first claim that each φ_i can be of a form $\forall \vec{x} \in \mathbb{R}^{V'}. \varphi \Rightarrow \psi$ such that

- φ is a linear conjunctive predicate over V' ;
- ψ is a formula of a form $p_1 x_1 + \dots + p_{|V'|} x_{|V'|} + q \geq 0$ where $p_1, \dots, p_{|V'|}, q$ are linear expressions over U .

⁴Such an invariant can be generated in a similar way to an existing method for the discrete variable case, e.g., Reference [24].

Roughly speaking, each φ represents a condition about the current configuration (e.g., $c \in I \cap \bar{C}$) while ψ represents the corresponding U-NNRepSupM condition (e.g., $f(c) \geq \bar{X}f(c)$). See Appendix C for the exact construction.

Now let $\varphi_1, \dots, \varphi_k$ be the axioms of U-NNRepSupM given in a form above. First, by weakening each strict inequality $>$ to nonstrict ones $\geq$,⁵ each φ_i is reduced to a constraint of a form

$$\forall \vec{x} \in \mathbb{R}^{V'}. A\vec{x} \leq \vec{b} \Rightarrow \vec{c}^T \vec{x} \leq d, \quad (10)$$

where $A \in \mathbb{R}^{m \times |V'|}$ is a matrix and $\vec{b} \in \mathbb{R}^m$ is a vector whose elements are real numbers; and $\vec{c}$ is a vector and d is a scalar whose elements are linear expressions over U . It is then not so hard to see that the following formula implies Equation (10). Here, intuitively, $\vec{y}$ and y are witnesses that prove Equation (10).

$$\exists \vec{y} = (y_1, \dots, y_m)^T \in \mathbb{R}^m. \exists y \in \mathbb{R}. (\vec{y} \geq \vec{0} \wedge y \geq 0 \wedge \forall \vec{x} \in \mathbb{R}^{V'}. (\vec{b} - A\vec{x})^T \vec{y} + y = d - \vec{c}^T \vec{x}). \quad (11)$$

Farkas' lemma (see, e.g., Reference [35, Corollary 7.1h]) tells us the converse implication also holds if the set $\{\vec{x} \in \mathbb{R}^{V'} \mid A\vec{x} \leq \vec{b}\}$ is not empty.

THEOREM 7.5 (FARKAS' LEMMA). *For given $A \in \mathbb{R}^{m \times |V'|}$ and $\vec{b} \in \mathbb{R}^m$, suppose there exists $\vec{x} \in \mathbb{R}^{V'}$ such that $A\vec{x} \leq \vec{b}$. Then the implication (10) $\Rightarrow$ (11) holds.* $\square$

By comparing coefficients of the inequality $(\vec{b} - A\vec{x})^T \vec{y} \leq d - \vec{c}^T \vec{x}$, we can see that Equation (11) is satisfied if and only if:

$$\exists \vec{y} = (y_1, \dots, y_m)^T \in \mathbb{R}^m. \vec{y} \geq \vec{0} \wedge A^T \vec{y} = \vec{c} \wedge \vec{b}^T \vec{y} \leq d. \quad (12)$$

Now, we see that each axiom φ_i is reduced to a constraint of a form $\exists \vec{y}. \hat{\varphi}_i$, where $\hat{\varphi}_i$ is a linear conjunctive predicate over $U \cup \{y_1, \dots, y_m\}$. Notice that we need the set $\{y_1, \dots, y_m\}$ of fresh variables for each $i = 1, \dots, k$; we let U' be the disjoint union of these variables used in every $\hat{\varphi}_i$ (hence, each $\hat{\varphi}_i$ is a linear conjunctive predicate over $U \cup U'$).

To sum up, our algorithm is as follows:

- (1) Receive the following as inputs: a pCFG Γ as a semantics of an ARP, a target set C , and an invariant I for Γ that satisfy Assumption 7.4.
- (2) Solve the following optimization problem:

$$\underset{v:U \rightarrow \mathbb{R}, v':U' \rightarrow \mathbb{R}}{\text{minimize}} \quad \llbracket f_v \rrbracket(l_{\text{init}}, \vec{x}_{\text{init}}), \quad \text{subject to} \quad \hat{\varphi}_1 \wedge \dots \wedge \hat{\varphi}_k.$$

- (3) Return $\llbracket f_v \rrbracket(l_{\text{init}}, \vec{x}_{\text{init}})$ as an over-approximation of $\overline{\mathbb{P}}_C^{\text{reach}}(l_{\text{init}}, \vec{x}_{\text{init}})$, where v is the valuation of U found in step (2).

THEOREM 7.6. *If Assumption 7.4 is satisfied, then existence of a linear U-NNRepSupM for Γ for $\mathcal{C}$ at M supported by $\mathfrak{I}$ is decidable in polynomial time.* $\square$

7.1.2 Linear Template-based Synthesis of γ -ScSubM. The linear template-based algorithm for synthesizing an L- γ -ScSubM is very similar to the one for U-NNRepSupM in the previous section. We also note that we first have to fix $\gamma \in (0, 1)$. The bigger γ we fix, the better bound we can obtain.

THEOREM 7.7. *If Assumption 7.4 is satisfied, then existence of a linear lower γ -ScSubM for Γ for $\mathcal{C}$ at M supported by $\mathfrak{I}$ is decidable in polynomial time.* $\square$

⁵This relaxation does not affect soundness of U-NNRepSupM, but it affects completeness. As was done for ARnkSupM in Reference [10], if we use a generalized theorem called Motzkin's *transposition theorem* (see, e.g., Reference [31]) instead of Farkas' lemma, then such a relaxation is not necessary.

7.2 Polynomial Template-based Synthesis

Now, we describe how the problem (8) can be reduced with a polynomial template. It is done in a similar way to the case of the linear template: We restrict the input program to PRPs, use a polynomial template for f , and reduce the axioms of NNRepSupM to ones without global quantifiers. A polynomial template-based algorithm for lower γ -SclSubMs can be constructed similarly.

First, we restrict the form of an input. Similar to the linear case, let (Γ, C, I) be an input, where Γ is a semantics of a PRP. Without contradiction, we assume Γ, C , and I satisfy the following conditions:

ASSUMPTION 7.8.

- Similar conditions to Assumption 7.4.1a, 1c, 2–4 where “linear” is replaced by “polynomial” hold.
- For each $l \in L_A$ such that $\text{Up}(l) = (v, u)$ and u is a distribution on $\mathbb{R}$ and $n \in \mathbb{N}$, its n th moment $\int_{x \in \mathbb{R}} x^n du \in \mathbb{R}$ is known.

Next, we fix a polynomial template for f in (8). Let $d \in \mathbb{N}$ be a maximum degree of the template, and $\mathcal{M}_d$ be the set of monomials whose degrees are no greater than d . A polynomial template is the following expression map $\mathfrak{f}$ with unknown coefficients $U = \{a_h^l \mid l \in L, h \in \mathcal{M}_d\}$:

$$\mathfrak{f}(l) := \sum_{h \in \mathcal{M}_d} a_h^l \cdot h \quad , \quad l \in L.$$

For a given valuation $u : U \rightarrow \mathbb{R}$, a polynomial expression map $\mathfrak{f}_u$ is defined in a similar way to the linear case. We assume f is given as the semantics of such an expression map.

Last, we reduce the axioms of U-NNRepSupM to ones without global quantifiers. In a similar way to the linear case, they can be represented by formulas of a form $\forall \vec{x} \in \mathbb{R}^{V'}. \varphi \Rightarrow \psi$ such that

- φ is a polynomial conjunctive predicate over $V' = V \cup \{x_{|V|+1}\}$ where $x_{|V|+1}$ is a fresh variable for handling nondeterministic assignment;
- ψ is a formula of a form $p_1 h_1 + \dots + p_t h_t \geq 0$ where $p_1, \dots, p_t$ are polynomial expressions U , and $h_1, \dots, h_t$ are monomials over V' .

Now let $\varphi_1, \dots, \varphi_k$ be the axioms of U-NNRepSupM given in a form above. First, by weakening each strict inequality $>$ to nonstrict ones $\geq$, each φ_i is reduced to a constraint of a form

$$\forall \vec{x} \in \mathbb{R}^{V'}. \bigwedge_{i=1}^m g_i \geq 0 \Rightarrow \sum_{h \in \mathcal{M}'_d} p_h \cdot h \geq 0. \quad (13)$$

Here, g_i is a polynomial expression over V' for each i ; $\mathcal{M}'_d$ is the set of monomials over V' whose degrees are no greater than d ; and p_h is a polynomial expression over U for each $h \in \mathcal{M}'_d$.

Next, recall that in the linear case, we reduced the constraint (10) to (11) by specifying witnesses $\vec{y}$ and y that prove (10); we do a similar reduction by specifying a set of polynomial expressions $\{\mathfrak{h}_w\}_{w \subseteq \{1, \dots, m\}}$ over V' . We consider the following reduction of Equation (13):

$$\exists \{\mathfrak{h}_w\}_{w \subseteq \{1, \dots, m\}}. \forall \vec{x} \in \mathbb{R}^{V'}. \sum_{h \in \mathcal{M}'_d} p_h \cdot h = \sum_{w \subseteq \{1, \dots, m\}} \mathfrak{h}_w \cdot \prod_{j \in w} g_j. \quad (14)$$

It is easy to see that Equation (14) implies Equation (13) if we further assume that each $\mathfrak{h}_w$ is non-negative, i.e., $\forall \vec{x} \in \mathbb{R}^{V'}. \mathfrak{h}_w \geq 0$ is true. We in particular look for a witness $\mathfrak{h}_w$ of a form $\mathfrak{h}_w \equiv \vec{y}_k^T A \vec{y}_k$, where $A \in \mathbb{R}^{|\mathcal{M}'_k| \times |\mathcal{M}'_k|}$ is a positive semidefinite matrix and $\vec{y}_k$ is the column vector whose components consist of the monomials in $\mathcal{M}'_k$. It is known (see, e.g., Reference [21]) that an expression $\mathfrak{h}$

can be written in such a form if and only if it is *sum-of-square*, i.e., there are polynomial expressions $i_1, \dots, i_s$ such that $h \equiv \sum_{i=1}^s i_i^2$. Clearly, each sum-of-square expression is nonnegative.

The converse of this reduction also holds under certain conditions. There are several conditions each of which verifies that, see, e.g., Reference [13]; we refer the following *Schmüdgen's Positivstellensatz*.

THEOREM 7.9 (SCHMÜDGEN'S POSITIVSTELLENSATZ, [34]). *Let $g, g_1, \dots, g_m$ be polynomial expressions over V . If $\llbracket \bigwedge_{i=1}^m g_i \geq 0 \rrbracket \subseteq \mathbb{R}^V$ is compact, then $\forall \vec{x} \in \llbracket \bigwedge_{i=1}^m g_i \geq 0 \rrbracket. \llbracket g \rrbracket(\vec{x}) > 0$ if and only if there exists a family $\{h_w\}_{w \in 2^m}$ of sum-of-square polynomial expressions such that $g = \sum_{w \subseteq \{1, \dots, m\}} h_w \cdot \prod_{j \in w} g_j$.* $\square$

Now in an obvious way, identify a set $\{A_w\}_{w \subseteq \{1, \dots, m\}}$ of $|M'_k| \times |M'_k|$ real-valued matrices with a valuation $u_i : U_i \rightarrow \mathbb{R}$ of a set U_i of fresh unknown coefficients. In a similar way that we reduced (11) to (12), we can reduce Equation (14) to a linear conjunctive predicate $\hat{\phi}_i$ over $U \cup U_i$. Also let ψ_i be a formula over U_i that asserts A_w is positive semidefinite for each $w \subseteq \{1, \dots, m\}$. We have argued that Equation (13) can be reduced to a formula $\exists u'. (\hat{\phi}_i \wedge \psi_i)$. Let U' be the disjoint union of $U_1, \dots, U_k$.

To sum up, our algorithm is as follows:

- (1) Receive the following as inputs: a PRP with its semantics Γ , a target region C , and an invariant I for Γ that satisfy Assumption 7.8.
- (2) Solve the following optimization problem:

$$\underset{v: U \rightarrow \mathbb{R}, v': U' \rightarrow \mathbb{R}}{\text{minimize}} \quad \llbracket f_v \rrbracket(l_{\text{init}}, \vec{x}_{\text{init}}) \quad , \quad \text{subject to} \quad (\hat{\phi}_1 \wedge \dots \wedge \hat{\phi}_k) \wedge (\psi_1 \wedge \dots \wedge \psi_k).$$

- (3) Return $\llbracket f_v \rrbracket(l_{\text{init}}, \vec{x}_{\text{init}})$ as an over-approximation of $\overline{\mathbb{P}}_C^{\text{reach}}(l_{\text{init}}, \vec{x}_{\text{init}})$, where v is the valuation found in step (2).

This is a **semidefinite programming (SDP)** problem that is solvable using an SDP solver.

8 IMPLEMENTATION AND EXPERIMENTS

In this section, we implement synthesis algorithms that we introduced in the previous section and perform experiments. An overview of our experiment is as follows:

- We examine the applicability of our new synthesis algorithms, namely, U-NNRepSupM and L- γ -SclSubM ones, to various randomized programs. Our benchmark set consists of a non-deterministic/probabilistic modification of a real-world benchmark set [3] and randomized programs in the literature [1, 11, 14, 29].
- We compare U-NNRepSupM and ε -RepSupM techniques by performing their synthesis algorithms on artificial examples of randomized programs. To highlight the pros and cons of these techniques, we design these examples so either U-NNRepSupM or ε -RepSupM synthesis algorithm cannot give a nontrivial bound.

To describe an input of synthesis algorithms, we consider an augmented syntax of ARPs and PRPs (Section 2.3) with two components “{...}” and “[...]”. They specify an invariant I and a terminal region C , respectively. An example of a program code is shown in Figure 4. If we feed this code to an U-NNRepSupM synthesis algorithm, for example, then the algorithm under-approximates $\overline{\mathbb{P}}_C^{\text{reach}}(c)$ where the line 3 is reached when $x \leq 0$, under an assumption that $x \geq 0$ holds in each program location.

U-NNRepSupM and L- γ -SclSubM synthesis algorithms receive an ARP or PRP (written in the augmented syntax) as an input, and first translate it into a pCFG Γ , an invariant I , and a terminal

```

1  {0 ≤ x} while true do
2  {0 ≤ x}   x := x - 1;
3  {0 ≤ x}   [x ≤ 0]
4             skip
5             od

```

Fig. 4. An augmented syntax of ARPs and PRPs.

region C . The algorithms then solve the optimization problem described in Section 7, and output the result as a bound. Note that the program does not synthesize an invariant nor prove the correctness of the given invariant, and therefore the user has to provide a correct invariant by hand or by using some algorithm, e.g., Reference [24].

The synthesis algorithm of ε -RepSupM in Reference [13] works in a slightly different way. The difference is that it iteratively synthesizes ε -RepSupM's with different κ -parameters to find a better bound. Concretely, (i) it first computes the smallest number κ_0 for which an ε -RepSupM with κ_0 -bounded difference exists; (ii) it then runs the synthesis algorithm of ε -RepSupM with $\kappa = \kappa_0, \kappa_0 + 1, \dots, \kappa_0 + N$, where $N \in \mathbb{N}$ is a fixed parameter. Following Reference [13], we let $\varepsilon = 1$ and $N = 1000$.

All randomized programs used in the experiments are ARPs (which are also PRPs by definition). We use the linear and polynomial templates for U-NNRepSupM synthesis, and the linear template for L- γ -SclSubM and ε -RepSupM syntheses.

All the programs are implemented in OCaml. We use glpk (v4.63) [18] and SDPT3 [36] for the LP and SDP solvers, respectively. For the implementation of the U-NNRepSupM synthesis algorithm with polynomial template, we also make use of a MATLAB toolbox SOSTOOLS (v3.03) [37]. Our implementation is publicly available online.⁶

Experiments were carried out on a Panasonic Let's note CF-SV7 with 8.00 GB RAM and Intel(R) Core(TM) i5-8350U 1.70 GHz processor.

8.1 Experiment I: Performance of New Martingale Techniques on a Benchmark Set

We first perform U-NNRepSupM and L- γ -SclSubM synthesis algorithms for a benchmark set of randomized programs.

Experiment Settings. We perform our algorithms for two sets of randomized programs. The first one is made by modifying a benchmark set in Reference [3] into randomized ones. Each program in the benchmark set is an abstraction of a real-world program. As most of them are non-randomized programs, we did the following modifications to it: (i) initial values of variables are chosen non-deterministically from a handcrafted interval, and (ii) one of the while loops in the program is modified so the failure flag is up with probability 0.0001 in each iteration of the loop. All original programs we use are a.s. terminating over the invariants we set, and hence the modified programs either terminate or abort with probability 1, depending on the state of the failure flag at the end of a run.

The second set consists of the following randomized programs from the literature. The code of ARPs without invariant conditions are given in Appendix E.

- (a) **(Adversarial random walk)** A variation of a random walk, whose analysis is more challenging because of additional adversarial nondeterministic choices [14]. We consider the following three variants:
 - (a-1) **(1D adversarial random walk)** A random walk over $\mathbb{R}$. This models a *discrete queuing system*, and is parametrized by $p_1, p_2 \in [0, 1]$ that determines the distribution of the number of packets that arrive in each round. The system starts with 2

⁶https://github.com/takisaka/Impl_newSupermartingale.

packets in the queue, and in each round 0, 1, or 2 packets are queued with probability p_1, p_2 , and $1 - p_1 - p_2$, respectively. Each packet is processed depending on its type that is nondeterministically chosen: If it is “urgent,” then the packet is immediately processed in one round with probability $1/8$, but with probability $7/8$ it fails to process the packet and produces another packet. If the type is “standard,” then it is processed using two rounds with probability 1. The process terminates if queue gets empty, and aborts (it is not counted as termination) if the queue size exceeds 10.

- (a-2) **(2D adversarial random walk)** A random walk over $\mathbb{R}^2$ with a time limit of 100 rounds. It is parametrized by $M_1, M_2 \in \mathbb{R}$. The process starts from $(x, y) = (2, 2)$ and in each round, either “ x ” or “ y ” is nondeterministically chosen, and it is added by $z \in \mathbb{R}$ that is uniformly chosen from $[M_1, M_2]$. The process terminates if $x \leq 0$ or $y \leq 0$, and aborts if 100 rounds were consumed.
- (a-3) **(variant of 2D adversarial random walk)** A random walk over $\mathbb{R}^2$ with a time limit of 100 rounds. It is again parametrized by $M_1, M_2 \in \mathbb{R}$. The process starts from $(x, y) = (3, 2)$, and in each round either of the following is nondeterministically chosen: (1) $z \in \mathbb{R}$ is uniformly chosen from $[M_1, M_2]$ and added to x (respectively, y) with probability 0.7 (respectively, 0.3); or (2) $z \in \mathbb{R}$ is uniformly chosen from $[-M_2, -M_1]$ and added to y (respectively, x) with probability 0.7 (respectively, 0.3). The process terminates if $x \leq 0$ or $y \leq 0$, and aborts if 100 rounds were consumed.
- (b) **(room temperature control)** A model of an air conditioning system for adjacent two rooms [1, 11]. It is parametrized by real numbers c and p : the former determines the power of the air conditioner, and the latter determines the size of perturbation. We have also added a time limit of 100 as in (a) above. The rooms are named Room 1 and Room 2, and we write x_1 and x_2 for their temperature. The changes of x_1 and x_2 are modeled by the following stochastic difference equation: $x'_i = x_i + a \cdot (x_j - x_i) + b_i \cdot (o - x_i) + c_i \cdot \sigma(x_i) + v_i$ where $i, j \in \{1, 2\}$ and $j \neq i$. The second term models the heat exchange between the rooms and the third term models the heat exchange with the outside whose temperature is a constant o . The fourth term models the influence of the air conditioner controlled by a function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$. The last term v_i is a probabilistic distribution that models the perturbation. Partially referring to Reference [11], we have fixed the parameters as $a = 0.0625$, $b_1 = 0.0375$, $b_2 = 0.025$, $c_1 = c_2 = c$, $o = 6$, $\sigma(x) = 19.5 - x$, and v_1, v_2 independently follow the uniform distribution over $[-p, p]$, where $c, p \in \mathbb{R}$. The process terminates if x_1 or x_2 is out of the comfortable range: $[17, 22]$ for x_1 and $[16, 23]$ for x_2 .

For the first program set, we under-approximate termination probability by U-NNRepSupM and L- γ -SclSubM. We apply Proposition 4.15 to under-approximate $\mathbb{P}_C^{\text{reach}}$ by U-NNRepSupM. Recall that we know each program either terminates or aborts with probability 1 under any scheduler.

For the second program set, we over- and under-approximate termination probability by U-NNRepSupM and L- γ -SclSubM, respectively.

Experimental Results. The results are given in Tables 2–3.⁷ For each model in the first column, we perform three different synthesis algorithms: U-NNRepSupM one, L- γ -SclSubM one with $\gamma = 1 - 10^3$, and L- γ -SclSubM one with $\gamma = 1 - 10^6$. All algorithms use the linear template. For each of these algorithms, a probability bound computed by the algorithm is given in the “bound” column;

⁷The result of polynomial U-NNRepSupM synthesis for (a-1), which appears in Reference [39], is cut here; we found it came with a numerical error.

Table 2. Bound of Termination Probability for Probabilistic/Nondeterministic Modification of Programs in Reference [3]

model	U-NNRepSupM		L- γ -SclSubM			
	linear		linear, $\gamma = 1 - 10^{-3}$		linear, $\gamma = 1 - 10^{-6}$	
	bound	time(s)	bound	time(s)	bound	time(s)
aaron2	≥ 0.879	0.079	≥ 0.000	0.070	≥ 0.876	0.116
ax	≥ 0.990	0.064	≥ 0.735	0.072	≥ 0.989	0.099
complex	≥ 0.996	0.067	≥ 0.786	0.070	≥ 0.996	0.092
counterex1c	≥ 0.989	0.073	≥ 0.608	0.087	≥ 0.988	0.097
cousot9	≥ 0.480	0.051	≥ 0.000	0.068	≥ 0.467	0.079
easy1	≥ 0.980	0.059	≥ 0.503	0.067	≥ 0.979	0.087
easy2	≥ 0.990	0.051	≥ 0.685	0.060	≥ 0.989	0.084
exmini	≥ 0.980	0.066	≥ 0.280	0.070	≥ 0.979	0.093
loops	≥ 0.989	0.066	≥ 0.000	0.077	≥ 0.984	0.101
nd_loop	≥ 0.999	0.062	≥ 0.979	0.071	≥ 0.999	0.067
ndecr	≥ 0.980	0.062	≥ 0.581	0.068	≥ 0.979	0.071
nestedLoop	≥ 0.987	0.065	≥ 0.000	0.088	≥ 0.978	0.103
perfect	≥ 0.980	0.060	≥ 0.000	0.083	≥ 0.959	0.085
random1d	≥ 0.990	0.065	≥ 0.686	0.076	≥ 0.989	0.079
random2d	≥ 0.990	0.065	≥ 0.637	0.079	≥ 0.989	0.100
relation1	≥ 0.999	0.061	≥ 0.980	0.064	≥ 0.999	0.082
rsd	≥ 0.979	0.067	≥ 0.278	0.082	≥ 0.979	0.083
sipma91	≥ 0.977	0.069	≥ 0.000	0.092	≥ 0.976	0.100
speedFails4	≥ 0.979	0.072	≥ 0.472	0.084	≥ 0.979	0.097
speedp1di2	≥ 0.960	0.069	≥ 0.000	0.072	≥ 0.958	0.083
speedp1di3	≥ 0.480	0.065	≥ 0.000	0.079	≥ 0.467	0.093
speedp1di4	≥ 0.979	0.052	≥ 0.475	0.065	≥ 0.979	0.082
terminate	≥ 0.969	0.062	≥ 0.150	0.072	≥ 0.968	0.069
wcet1	≥ 0.979	0.066	≥ 0.173	0.078	≥ 0.979	0.089
wcet2	≥ 0.998	0.062	≥ 0.766	0.073	≥ 0.997	0.075
while2	≥ 0.980	0.065	≥ 0.000	0.065	≥ 0.960	0.068
wise	≥ 0.980	0.067	≥ 0.489	0.077	≥ 0.979	0.089

and the execution time in seconds is given in the “time(s)” column. In Table 2, the tightest bound for each model is highlighted in bold.

In Table 2, we observe that U-NNRepSupM gives lower bounds tighter than 95% to most instances. There are few exceptions such as *aaron2*, *cousot9*, and *speedp1di3*; this is reasonable, because these programs tend to call more while loop iterations than others. While the L- γ -SclSubM synthesis algorithm does not perform very well with $\gamma = 1 - 10^3$, it provides considerably better bounds when we increase γ to $1 - 10^6$.

U-NNRepSupM uniformly gives better bounds than L- γ -SclSubM, but when $\gamma = 1 - 10^6$, the difference is almost negligible. However, L- γ -SclSubM has an advantage that it does not require an *a priori* knowledge needed for under-approximation via U-NNRepSupM (cf. Section 4.4).

In Table 3, we observe that nontrivial bounds are given in several instances, but not as tight as the ones in Table 2. There are at least two situations in which giving a tight bound is difficult. One is the case that the shape of the characteristics (i.e., reachability probability here) is hard to be mimicked

Table 3. Result of Martingale Synthesis for Randomized Programs in the Literature

model	U-NNRepSupM		L- γ -SclSubM			
	linear		linear, $\gamma = 1 - 10^{-3}$		linear, $\gamma = 1 - 10^{-6}$	
	bound	time(s)	bound	time(s)	bound	time(s)
(a-1)						
$p_1 = 0.2, p_2 = 0.4$	≤ 0.825	0.039	≥ 0.000	0.042	≥ 0.000	0.046
$p_1 = 0.8, p_2 = 0.1$	≤ 1.000	0.040	≥ 0.686	0.043	≥ 0.692	0.036
(a-2)						
$M_1 = -1, M_2 = 2$	≤ 1.000	0.040	≥ 0.000	0.043	≥ 0.000	0.036
$M_1 = -2, M_2 = 1$	≤ 1.000	0.051	≥ 0.767	0.044	≥ 0.859	0.047
(a-3)						
$M_1 = -1, M_2 = 2$	≤ 1.000	0.039	≥ 0.000	0.034	≥ 0.000	0.046
$M_1 = -2, M_2 = 1$	≤ 1.000	0.041	≥ 0.787	0.035	≥ 0.871	0.034
(b)						
$c = 0.5, p = 0.1$	≤ 1.000	0.055	≥ 0.000	0.050	≥ 0.000	0.054
$c = 0.1, p = 0.1$	≤ 1.000	0.066	≥ 0.148	0.041	≥ 0.510	0.052

by a template. As a martingale uniformly over- or under-approximates the characteristics, an error is inevitably large if the martingale does not fit the shape of the characteristics very well.

Another is the case that some constraints of martingale are hard to be satisfied by the template we use. For example, suppose a linear U-NNRepSupM η is constrained by the following conditions:

$$\forall x \in [0, 10]. \eta(l, x) \geq 0.4 \cdot \eta(l, x+1) + 0.6 \cdot \eta(l, x-1) \quad , \quad \eta(l, 0) = 1.$$

By letting $\eta(l, x) = ax + b$ and solving the first condition, we obtain $a \geq 0$. This means, together with the second condition, any linear U-NNRepSupM can only give a trivial bound over $\{l\} \times [0, 10]$. Such an issue occurs in, e.g., linear U-NNRepSupM synthesis for the program (a-1) with $p_1 = 0.8, p_2 = 0.1$.

8.2 Experiment II: A Comparison between U-NNRepSupM and ε -RepSupM

We next compare U-NNRepSupM and ε -RepSupM techniques by performing their synthesis algorithms on artificial examples of randomized programs.

Experiment Settings. We examine the following five randomized programs. The code of ARPs without invariant condition are given in Appendix E. Each program is constructed so either U-NNRepSupM or ε -RepSupM synthesis algorithm cannot give a nontrivial bound. We give a brief explanation of the reason in each case; see Appendix D for a formal argument.

- (c-1) **(Asymmetric bounded random walk)** This program is a random walk over an interval $I = [0, x_{\text{fail}}]$, where $x_{\text{fail}} \in \mathbb{N}$. Starting from $x = x_0$, it does an asymmetric random walk (i.e., continue updating x to either $x - 1$ or $x + 1$ with probability p and $1 - p$, respectively) until $x \leq 0$ or $x \geq x_{\text{fail}}$ is witnessed; the former is understood as a termination, and the latter as a failure. In our experiment, we let $p = 0.4$ and consider three different values for x_0 and x_{fail} , namely, $(x_0, x_{\text{fail}}) = (5, 10), (50, 100), (500, 1,000)$.

There is no ε -RepSupM for this randomized program, because it enters a loop when $x \geq 10$ is witnessed, over which ε -decreasing condition is not satisfiable (the value of a supermartingale cannot keep strictly decreasing over a loop).

- (c-2) **(Modification of (c-1))** This program is the same as (c-1) except that, after $x \geq 10$ is witnessed, the value of x keeps increasing instead of staying at the same value.

Table 4. Bounds of Reachability Probability Given by U-NNRepSupM and ε -RepSupM

model	true r. p.	U-NNRepSupM						ε -RepSupM	
		linear		deg. 2 poly.		deg. 4 poly.		$\varepsilon = 1$	
		bound	time(s)	bound	time(s)	bound	time(s)	bound	time(s)
(c-1)									
$x_0 = 5, x_{\text{fail}} = 10$	≈ 0.116	$\leq \mathbf{0.530}$	0.039	N.E.	-	N.E.	-	INFEAS	-
$x_0 = 50, x_{\text{fail}} = 100$	$\approx 4.39 \times 10^{-5}$	$\leq \mathbf{0.504}$	0.040	N.E.	-	N.E.	-	INFEAS	-
$x_0 = 500, x_{\text{fail}} = 1,000$	$\approx 9.00 \times 10^{-89}$	$\leq \mathbf{0.501}$	0.040	N.E.	-	N.E.	-	INFEAS	-
(c-2)									
$x_0 = 5, x_{\text{fail}} = 10$	≈ 0.116	≤ 1.000	0.039	N.E.	-	N.E.	-	$\leq \mathbf{966.300}$	51.884
$x_0 = 50, x_{\text{fail}} = 100$	$\approx 4.39 \times 10^{-5}$	≤ 1.000	0.040	N.E.	-	N.E.	-	$\leq \mathbf{195.806}$	53.612
$x_0 = 500, x_{\text{fail}} = 1,000$	$\approx 9.00 \times 10^{-89}$	≤ 1.000	0.040	N.E.	-	N.E.	-	$\leq \mathbf{2.29 \times 10^{-5}}$	51.082
(c-3)	0.5	$\leq \mathbf{0.500}$	0.030	$\leq \mathbf{0.500}$	13.090	$\leq \mathbf{0.500}$	19.580	INFEAS	-
(c-4)	≈ 0.200	≤ 0.438	0.043	≤ 0.421	19.086	$\leq \mathbf{0.333}$	18.979	INFEAS	-
(c-5)	≈ 0.333	≤ 1.000	0.042	N.E.	-	N.E.	-	$\leq \mathbf{215.515}$	50.332

Unlike (c-1), there exists an ε -RepSupM for this program, because no configuration is visited twice after a failure, and the value of a supermartingale can keep decreasing through transitions.

However, any polynomial U-NNRepSupM cannot give a nontrivial bound for this program. Roughly, any nontrivial U-NNRepSupM for this program is required at some location to be strictly decreasing and nonnegative over an unbounded domain, which is not realizable by a polynomial function.

- (c-3) **(Simple system with infinite branching)** This program first samples a natural number n from the geometric distribution, i.e., for each $m \in \mathbb{N}$, we have $n = m$ with probability $2^{-(m+1)}$. Then it performs an unbiased coin flip after n skips and either terminates or aborts according to the outcome.

There is no ε -RepSupM for this randomized program, because the number of steps from the initial configuration to the terminal region may be arbitrarily large, depending on the outcome of the first probabilistic transition. In such a case, it is impossible to ensure both of ε -decreasing and κ -bounded conditions of ε -RepSupM.

- (c-4) **(Random walk over $[0, 1]$ that exhibits geometric behaviors)** This program first chooses $x \in [0, 1]$ from the uniform distribution, and then it does a random walk *geometrically*, i.e., at each step x is updated to either $2x$ or $0.5x$ rather than $x + 1$ or $x - 1$. The program terminates when $x \geq 1$ is witnessed.

For a similar reason to the previous case, there is no ε -RepSupM for this randomized program.

- (c-5) **(Asymmetric unbounded random walk)** This program is the same as (c-1) except that the random walk is done over $[0, \infty)$ and the program continues random walk even if $x \geq 10$ is witnessed.

For a similar reason to the case of (c-2), any polynomial U-NNRepSupM cannot give a nontrivial bound for this program.

Experimental Results. The results are shown in Table 4. We compared the ε -RepSupM synthesis algorithm with our U-NNRepSupM one with three different templates: linear one, degree 2 polynomial one, and degree 4 polynomial one. The format of the table is the same as Table 2 and 3 except that the true reachability probability is also provided in the second column: see Appendix D for their calculation. In the table, N.E. indicates that the result is not reliable due to numerical errors;

and INFEAS indicates that the ε -RepSupM synthesis algorithm fails to find the smallest κ . The tightest bound for each model is highlighted in bold; recall that ε -RepSupM may give a bound larger than 1, and in such a case it still refutes almost-sure reachability (Theorem 3.2.2).

We observe that both synthesis algorithms are indeed unable to give nontrivial bounds for some models as explained in *Experiment Setting* section, and for each of such models, the other algorithm successfully gives a nontrivial bound. This outcome suggests that one can handle more variety of problems by using both of NNRepSupM and ε -RepSupM techniques.

We also see that polynomial U-NNRepSupM synthesis is successful for some models and gives better bounds than the linear synthesis algorithm are given for (c-4).

9 RELATED WORK

The notion of ranking supermartingale is first proposed by Reference [10] aiming at extending applicability of *quantitative invariants* [27, 28] to randomized programs with real-valued variables, but they do not treat nondeterminism. Soundness of the method under demonic nondeterminism is studied in References [2, 13, 17]; among them, *lexicographic ranking supermartingales* [2] can be seen as an extension of our U-ARnkSupM. Soundness under finite demonic/angelic nondeterminism is shown in Reference [14].

Completeness of U-ARnkSupM for strong almost-sure termination ([6], also see Section 5) is shown for discrete systems with nondeterminism [6, 12]. The closest result to ours is Reference [12], where they study pCFGs with demonic nondeterminism but restrict to integer-valued variables. Our proof also works for real-valued variables.

ARnkSupMs can also be used to evaluate *Tail probability*, i.e., the probability that the runtime of a program exceeds a given threshold. There are a couple of existing techniques [12, 14, 26], where each of them utilizes a different *concentration inequality*. In Reference [26], a fixed point characterization of $\overline{\mathbb{E}}_C^{\text{steps}}$ is extended to the *kth moment of reaching time* to make use of *Markov's inequality*, which is also a tool for proving soundness of NNRepSupM.

Several under- and over-approximation methods for *expected runtimes* of randomized programs, which is defined on its structure rather than on its semantics, is studied in References [23, 32]. *Upper invariants of while-loops* among them is a U-ARnkSupM-like notion in their setting. In Reference [23] soundness and completeness of the upper invariant technique are derived from order-theoretic considerations. They consider randomized programs with demonic nondeterminism, but program variables are discrete, and nondeterministic choices are restricted to finite ones. The idea in Reference [23] is implemented in Reference [32] via **automatic amortized resource analysis (AARA)** techniques.

The definition of NNRepSupM resembles that of *probabilistic barrier certificate* [33, 38], which is used in control theory to give a lower bound of safety probability of Markov processes. There are several things in Reference [38] that are not treated in our article: (i) they also work on continuous-time dynamics, not only discrete-time ones, and (ii) they provide a template-based synthesis algorithm with an exponential template whose construction is nontrivial. However, (i) they restrict their attention to purely probabilistic systems, while our system can also handle nondeterminism, and (ii) we prove completeness of NNRepSupM, which is not discussed in Reference [38]. Finally, the present article covers Reference [38] in the following aspect: (i) our system can mimic any discrete-time Markov process in Reference [38], and (ii) we introduce yet another supermartingale notion that corresponds to *c-martingale* in Reference [38], where we fix c as a constant that is independent of time (see Section 4.3).

ε -RepSupM [15] is also a technique for over-approximating reachability, which is studied to synthesize *stochastic invariants*. It is combined with ranking supermartingales to verify the *persistence* property of programs, too [11, 15]. While there are certain similarities between ε -RepSupMs and NNRepSupMs, they are technically different, because ε -RepSupMs exploit the

κ -bounded differences condition, which is not assumed in our case. Their method is sound for refuting almost-sure reachability, but not complete.

10 CONCLUSIONS AND FUTURE WORK

We gave a comprehensive and comparative account of martingale-based techniques for approximating reachability probabilities. We demonstrated that several different approximation techniques—NNRepSupM, ARnkSupM, and γ -SclSubM—had a common structure of order-theoretic fixed points in their theory, while they originally arose from different communities. The key observation was that the reachability probability and the expected reaching time were the least fixed points of certain monotone endofunctions; soundness and completeness of the first two techniques are derived as its corollaries, and it is the basis for the proof of soundness of γ -SclSubM. We also implemented the techniques above and conducted experiments, and the results suggest applicability of NNRepSupM and γ -SclSubM in quantitative reasoning.

In this article, we have focused on over- and under-approximating (i.e., refuting and verifying) reachability probabilities. For future work, we wish to study more complicated specifications such as recurrence ($\text{GF}\varphi$) and persistence ($\text{FG}\varphi$), too. Some martingale-based techniques have already been used (see, e.g., Reference [11]); we will investigate the use of *lattice-theoretic progress measures*, introduced in Reference [20] as a generalization of progress measures for parity games [22] in the probabilistic settings.

APPENDICES

A AN ALTERNATIVE DEFINITION OF CHARACTERISTICS

When we introduce the reachability probability $\mathbb{P}_{C,\sigma}^{\text{reach}}$ in Section 2.2, we first defined $\mathbb{P}_{C,\sigma}^{\text{reach} \leq N}$ inductively, and then let $\mathbb{P}_{C,\sigma}^{\text{reach}}$ be the sum of $\mathbb{P}_{C,\sigma}^{\text{reach} \leq N}$ over $N < \omega$. Here, we recall another well-known way of defining it via the *cylinder construction*. In what follows, suppose a pCFG Γ with its invariant I and terminal region C is given.

For each scheduler σ , we have a stochastic kernel μ^σ that describes the behavior of Γ under σ (Definition 2.5). By further specifying an initial distribution $p_{\text{init}} \in \mathcal{D}(I)$, we can construct a unique probability measure $\mu_{\infty}^{p_{\text{init}}, \sigma}$ over I^ω that determines the behavior of Γ in the following sense: For a given $A \in \mathcal{B}(I^\omega)$, the value of $\mu_{\infty}^{p_{\text{init}}, \sigma}(A)$ is the probability that Γ generates an infinite path in A under σ and the initial configuration c drawn from p_{init} . The exact construction can be found in Reference [8, Proposition 7.45]. In this article, the following property of $\mu_{\infty}^{p_{\text{init}}, \sigma}$ is relevant: Here, we let $p_{\leq n}$ be the marginal of $p \in \mathcal{D}(X_0 \times X_1 \times \dots)$ on $X_0 \times \dots \times X_n$.

PROPOSITION A.1 (E.G., [8]). *Let $\sigma \in \text{Sch}_\Gamma$ and $p_{\text{init}} \in \mathcal{D}(I)$ be given, and let $p = \mu_{\infty}^{p_{\text{init}}, \sigma}$ be as above. Then for given $n \in \mathbb{N}$ and $C_0, \dots, C_{n+1} \in \mathcal{B}(I)$, the following holds:*

$$p_{\leq n+1}(C_0 \times \dots \times C_{n+1}) = \int_{C_n} \mu_{c_0 \dots c_n}^\sigma(C_{n+1}) p_{\leq n}(d(c_0 \dots c_n)).$$

Intuitively, the above proposition says that the n -steps behavior of Γ under σ and p_{init} can be extended to the $n + 1$ -steps one by μ^σ . A similar equation repeatedly appears when we construct a scheduler from a certain $p \in \mathcal{D}(I^\omega)$ in the next section.

Having $\mu_{\infty}^{p_{\text{init}}, \sigma}$ as the distribution of an infinite path generated by Γ , it is natural to define the reachability probability from c under σ as the $\mu_{\infty}^{c, \sigma}$ -measure of all infinite paths that eventually pass the terminal region C . Concretely, we consider the following function:

Definition A.2. For a given $p \in \mathcal{D}(I^\omega)$, define $\mathbb{Q}_C^{\text{reach}}(p)$ by

$$\mathbb{Q}_C^{\text{reach}}(p) = p(\diamond C),$$

where we let $\Diamond C = \{c_0 c_1 \dots \in I^\omega \mid \exists i. c_i \in C\}$.

We have the following correspondence between $\mathbb{Q}_C^{\text{reach}}$ and $\mathbb{P}_{C,\sigma}^{\text{reach}}$:

PROPOSITION A.3. *For any $c \in I$ and $\sigma \in \text{Sch}_\Gamma$, we have $\mathbb{Q}_C^{\text{reach}}(\mu_{\infty}^{\delta_c, \sigma}) = \mathbb{P}_{C,\sigma}^{\text{reach}}(c)$.*

PROOF. For each $n \in \mathbb{N}$, let $C_n = \{c_0 c_1 \dots \in I^\omega \mid c_n \in C \wedge \forall i < n. c_i \notin C\}$. Then it holds that $\Diamond C = \sum_{n \in \mathbb{N}} C_n$, and by construction of $\mu_{\infty}^{\delta_c, \sigma}$, we have $\mu_{\infty}^{\delta_c, \sigma}(C_n) = \mathbb{P}_{C,\sigma}^{\text{reach}=N}(c)$. $\square$

In a similar way, we can also define $\mathbb{E}_C^{\text{steps}}$ in terms of $\mu_{\infty}^{\delta_c, \sigma}$.

COROLLARY A.4. *Let C_n be as above. For a given $p \in \mathcal{D}(I^\omega)$, define $\mathbb{F}_C^{\text{steps}}(p)$ by*

$$\mathbb{F}_C^{\text{steps}}(p) = \sum_{i=0}^{\infty} i \cdot p(C_n) + (+\infty) \cdot p(\square \bar{C}),$$

where $p(\square \bar{C}) = \{c_0 c_1 \dots \in I^\omega \mid \forall i. c_i \in C\}$ and $(+\infty) \times 0 = 0$. Then for any $c \in I$ and $\sigma \in \text{Sch}_\Gamma$, we have $\mathbb{F}_C^{\text{steps}}(\mu_{\infty}^{\delta_c, \sigma}) = \mathbb{E}_{C,\sigma}^{\text{steps}}(c)$. $\square$

B OMITTED PROOF OF SECTION 4

Here, we give a proof of Proposition 4.8 that states $\underline{\mathbb{P}}^{\text{reach}} = \inf_{\sigma \in \text{Sch}_\Gamma} \underline{\mathbb{P}}_\sigma^{\text{reach}}$ is lower semianalytic. It does not immediately follow from its definition, because each $\underline{\mathbb{P}}_\sigma^{\text{reach}}$ may not be lower semianalytic when σ is not Borel measurable. For the proof, we take a correspondence between Sch_Γ and certain subset of $\mathcal{D}(I^\omega)$ that has a desirable measurability property. In a similar way, we also prove that $\underline{\mathbb{E}}_C^{\text{steps}}$ is lower semianalytic.

Concretely, we saw in the previous section that for given scheduler σ and initial distribution p_{init} , there is a corresponding measure $\mu_{\infty}^{p_{\text{init}}, \sigma} \in \mathcal{D}(I^\omega)$ that describes the behavior of Γ under σ and p_{init} . Conversely, from each $p \in \mathcal{D}(I^\omega)$ that is consistent with the behavior of Γ , we can find σ and p_{init} that constructs p , i.e., $\mu_{\infty}^{p_{\text{init}}, \sigma} = p$. We prove Proposition 4.8 by taking an appropriate infimum over the set of such distributions.

We first formalize the meaning of “consistency” of p . In what follows, let Π_∞ be the set of all infinite paths of Γ . Also notice that, for a finite path $\pi \cdot (l, \vec{x})$ such that $l \in L_D \cup L_P \cup L_{AD} \cup L_{AP}$, the measure $\mu_{\pi \cdot (l, \vec{x})}^\sigma$ is independent of σ (Definition 2.5): We omit the superscript σ in such a case. Recall Π_N and Π_{AN} are the sets of finite paths $\pi \cdot (l, \vec{x})$ of Γ such that $l \in L_N$ and $l \in L_{AN}$, respectively.

Definition B.1. A probability measure $p \in \mathcal{D}(I^\omega)$ is an (infinite horizon) behavior of Γ if it satisfies the following properties:

- $p(\Pi_\infty) = 1$.
- For each $A \in \mathcal{B}(I^{n+1})$ such that $A \cap (\Pi_N \cup \Pi_{AN}) = \emptyset$ (that is, the last location in each $\pi \in A$ is in $L_D \cup L_P \cup L_{AD} \cup L_{AP}$) and $B \in \mathcal{B}(I)$, we have

$$p_{\leq n+1}(A \times B) = \int_A \mu_{c_0 \dots c_n}(B) p_{\leq n}(d(c_0 \dots c_n)). \quad (15)$$

Notice that Equation (15) resembles the equation in proposition A.1. From a behavior p of Γ , we can construct a scheduler σ and an initial distribution p_{init} as follows: The latter is immediately obtained by letting $p_{\text{init}} = p_{\leq 0}$. To obtain σ , we first decompose $p_{\leq n+1}$ into $p_{\leq n}$ and a Borel measurable stochastic kernel $\tilde{\mu}_{\leq n+1}^{(n)} : I^{n+1} \rightarrow \mathcal{D}(I)$ such that, for each $C_0, \dots, C_{n+1} \in \mathcal{B}(I)$,

$$p_{\leq n+1}(C_0 \times \dots \times C_{n+1}) = \int_{C_0 \times \dots \times C_n} \tilde{\mu}_{c_0 \dots c_n}^{(n)}(C_{n+1}) p_{\leq n}(d(c_0 \dots c_n)). \quad (16)$$

It is known that such a stochastic kernel $\tilde{\mu}_-^{(n)}$ always exists [8, Corollary 7.27.2]. Roughly, we would like to find σ such that $\tilde{\mu}_\pi^{(|\pi|-1)} = \mu_\pi^\sigma$ for each $\pi \in \Pi_N \cup \Pi_{AN}$.

Let $\Pi_{N,n} = \Pi_N \cap I^{n+1}$. Also let $\text{succ}(c) \subseteq I$ be the set of all successors of $c \in I$, and $G \subseteq I^{n+2}$ be the set of all possible one-step extensions of paths in $\Pi_{N,n}$, i.e., $G = \{c_0 \dots c_{n+1} \in I^{n+2} \mid c_0 \dots c_n \in \Pi_{N,n} \wedge c_{n+1} \in \text{succ}(c_n)\}$. Notice that, by $p(\Pi_\infty) = 1$, any set $A \subseteq I^{n+1}$ of non-path sequences with the length $n+1$ has the $p_{\leq n}$ -measure zero. In particular, we have $p_{\leq n+1}(G) = p_{\leq n+1}(\Pi_{N,n} \times I) = p_{\leq n}(\Pi_{N,n})$. Hence, by Equation (16),

$$p_{\leq n}(\Pi_{N,n}) = p_{\leq n+1}(G) = \int_{\Pi_{N,n}} \tilde{\mu}_{c_0 \dots c_n}^{(n)}(\text{succ}(c_n)) p_{\leq n}(d(c_0 \dots c_n)).$$

This proves that $\tilde{\mu}_{c_0 \dots c_n}^{(n)}(\text{succ}(c_n)) = 1$ for $(p_{\leq n})$ -a.e. $c_0 \dots c_n \in \Pi_{N,n}$. Thus, by changing the value over $(p_{\leq n})$ -measure zero set if necessary, we can obtain $\tilde{\mu}_{c_0 \dots c_{n-1}}^{(n)}$ that satisfies Equation (16) and $\tilde{\mu}_{c_0 \dots c_n}^{(n)}(\text{succ}(c_n)) = 1$ for every $c_0 \dots c_n \in \Pi_{N,n}$. Now define $\sigma_{t,n} : \Pi_{N,n} \rightarrow \mathcal{D}(I)$ so $\sigma_{t,n}(\pi)$ is the marginal of $\tilde{\mu}_\pi^{(n)}$ on I for each $\pi \in \Pi_{N,n}$. Then, we have $\text{supp}(\sigma_{t,n}(\pi \cdot (l, \vec{x}))) \subseteq \text{succ}(l)$. We do the similar construction for each $\Pi_{N,n}$ and $\Pi_{AN,n} = \Pi_{AN} \cap I^{n+1}$, $n \in \mathbb{N}$, to obtain σ . By definition, we have $\mu_\pi^\sigma = \tilde{\mu}_\pi^{(n)}$ for every $\pi \in \Pi_N \cup \Pi_{AN}$.

Now, we constructed σ and p_{init} from a behavior p . We can recover p from σ and p_{init} as follows:

PROPOSITION B.2. *For a given behavior p of Γ , let σ and p_{init} be as above. Then $\mu_\infty^{p_{\text{init}}, \sigma} = p$.*

PROOF. Let $\tilde{p}_{\leq n} = (p_{\leq n}^{p_0, \sigma})_{\leq n}$. By the uniqueness part of Reference [8, Proposition 7.45], it suffices to show $\tilde{p}_{\leq n} = p_{\leq n}$ for each $n \in \mathbb{N}$. It is immediate from definition that the equation holds for $n = 0$. Assume the equation holds for $n = 0, \dots, k$. Then, we have

$$\begin{aligned} p_{\leq k+1}(C_0 \times \dots \times C_{k+1}) &= \int_{C_0 \times \dots \times C_k} \mu_{c_0 \dots c_k}^\sigma(C_{k+1}) p_n(d(c_0 \dots c_k)) \quad (\text{Equation (15), (16), prop. of } \sigma) \\ &= \int_{C_0 \times \dots \times C_k} \mu_{c_0 \dots c_k}^\sigma(C_{k+1}) \tilde{p}_n(d(c_0 \dots c_k)) \quad (\text{induction hypothesis}) \\ &= \tilde{p}_{\leq k+1}(C_0 \times \dots \times C_{k+1}). \quad (\text{Proposition A.1}) \end{aligned}$$

This proves $\tilde{p}_{\leq k+1} = p_{\leq k+1}$. $\square$

It is not difficult to verify that, for any scheduler σ and p_{init} , the measure $\mu_\infty^{p_{\text{init}}, \sigma}$ is a behavior of Γ . From all the argument above, we have the following proposition. Below, we let Beh_Γ be the set of all behaviors of Γ and $\text{Beh}_\Gamma(q) = \{p \in \text{Beh}_\Gamma \mid p_{\leq 0} = q\}$.

PROPOSITION B.3. *It holds that $\inf_{p \in \text{Beh}_\Gamma(\delta_c)} \mathbb{Q}_C^{\text{reach}}(p) = \underline{\mathbb{P}}_C^{\text{reach}}(c)$ and $\inf_{p \in \text{Beh}_\Gamma(\delta_c)} \mathbb{F}_C^{\text{steps}}(p) = \underline{\mathbb{E}}_C^{\text{steps}}(c)$.*

PROOF. We prove the first equation; the second can be shown similarly. For each $\sigma \in \text{Sch}_\Gamma$, we have $\mu_\infty^{\delta_c, \sigma} \in \text{Beh}_\Gamma(\delta_c)$. Hence, by Proposition A.3, $\inf_{p \in \text{Beh}_\Gamma(\delta_c)} \mathbb{Q}_C^{\text{reach}}(p) \leq \inf_{\sigma \in \text{Sch}_\Gamma} \mathbb{Q}_C^{\text{reach}}(\mu_\infty^{\delta_c, \sigma}) = \underline{\mathbb{P}}_C^{\text{reach}}(c)$. Conversely, for each $p \in \text{Beh}_\Gamma(\delta_c)$, we can construct $\sigma_p \in \text{Sch}_\Gamma$ such that $p = \mu_\infty^{\delta_c, \sigma_p}$, so $\underline{\mathbb{P}}_C^{\text{reach}}(c) \leq \inf_{p \in \text{Beh}_\Gamma(\delta_c)} \mathbb{P}_{C, \sigma_p}^{\text{reach}}(c) = \inf_{p \in \text{Beh}_\Gamma(\delta_c)} \mathbb{Q}_C^{\text{reach}}(p)$. $\square$

Now, we are to prove the desired property of $\underline{\mathbb{P}}_C^{\text{reach}}$ and $\underline{\mathbb{E}}_C^{\text{steps}}$.

COROLLARY B.4. *The functions $\underline{\mathbb{P}}_C^{\text{reach}}$ and $\underline{\mathbb{E}}_C^{\text{steps}}$ are lower semianalytic.*

PROOF. We show $\mathbb{P}_C^{\text{reach}}$ is lower semianalytic, the case of $\mathbb{E}_C^{\text{steps}}$ can be shown similarly. First, the set Beh_Γ is Borel as follows: The set $\mathcal{P}_\infty = \{p \in \mathcal{D}(I^\omega) \mid p(\Pi_\infty) = 1\}$ is closed; and for given A and B as in Definition B.1, the functions $p \mapsto p_{\leq n+1}(A \times B)$ and $p \mapsto \int_A \mu_{c_0 \dots c_n}(B) p_{\leq n}(d(c_0 \dots c_n))$ are Borel measurable, so the set $\mathcal{P}_{A,B} = \{p \in \mathcal{D}(I^\omega) \mid p_{\leq n+1}(A \times B) = \int_A \mu_{c_0 \dots c_n}(B) p_{\leq n}(d(c_0 \dots c_n))\}$ is Borel. As Beh_Γ is the intersection of $\mathcal{P}_\infty$ and $\mathcal{P}_{A,B}$ with an appropriate countable instances of (A, B) , the set Beh_Γ is Borel.

The set $G = \{(q, p) \in I \times \mathcal{D}(I^\omega) \mid p \in \text{Beh}_\Gamma(q)\}$ is also Borel (and hence analytic), because the set $\{(p_{\leq 0}, p) \mid p \in \mathcal{D}(I^\omega)\}$ is the inverse image of the closed set $\{(q, q) \mid q \in \mathcal{D}(I)\}$ under the continuous function $(q, p) \mapsto (q, p_{\leq 0})$. Also the function $\mathbb{Q}_C^{\text{reach}} : \mathcal{D}(I^\omega) \rightarrow [0, 1]$ is Borel measurable, so in particular it is lower semianalytic. Hence, $f(q) = \inf\{\mathbb{Q}_C^{\text{reach}}(p) \mid (q, p) \in G\}$ is a lower semianalytic function over $\mathcal{D}(I)$ [8, Proposition 7.47] and so is $\mathbb{P}_C^{\text{reach}}$ by Proposition B.3. $\square$

C OMITTED DETAILS OF SECTION 7

When we reduce the axioms of U-NNRepSupM in Section 7.1.1, we claim that each of them can be of a form $\forall \vec{x} \in \mathbb{R}^{V'} . \varphi \Rightarrow \psi$ such that

- φ is a linear conjunctive predicate over V' ;
- ψ is a formula of a form $p_1 x_1 + \dots + p_{|V'|} x_{|V'|} + q \geq 0$ where $p_1, \dots, p_{|V'|}, q$ are linear expressions over U .

In this section, we provide their exact construction. See also References [14, 15]. We let $\text{Fml}^\Rightarrow$ be the set of all formulas $\varphi \Rightarrow \psi$ such that φ and ψ are of the forms above.

We first fix some expressions to describe the axioms of U-NNRepSupM. Recall that we are given a pCFG Γ with $V = \{x_1, \dots, x_{|V|}\}$, an invariant $I \subseteq L \times \mathbb{R}^V$, and a target region $C \subseteq L \times \mathbb{R}^V$ that obey Assumption 7.4. For simplicity, we fix a sufficiently large N so each conjunction or disjunction in the axioms is taken over exactly N elements. Those that do not involve N elements can also be written in this form by adding redundant elements (e.g., $r_1 \wedge r_2 \equiv r_1 \wedge r_2 \wedge r_2 \wedge \dots \wedge r_2$).

- For each $l \in L_A$ such that $\text{Up}(l) = (v, u)$ (i.e., x_v is going to be updated via u),
 - if $u \in \mathcal{B}(\mathbb{R}^V, \mathbb{R})$, then let a_l be a linear expression over V such that $u = \llbracket a_l \rrbracket$;
 - if $u \in \mathcal{D}(\mathbb{R})$, then let $r_l = \mathbb{E}u$;
 - if $u \in \mathcal{B}(\mathbb{R})$, then let $\mathfrak{P}_l$ be a linear predicate over $\{x_{|V|+1}\}$ such that $u = \llbracket \mathfrak{P}_l \rrbracket$. Here, assume $\mathfrak{P}_l := \bigvee_{n=1}^N \mathfrak{P}_l^n$, where $\mathfrak{P}_l^n$ is a linear conjunctive predicate for each l, n .
- For each $l \in L_D$ and $l' \in \text{succ}(l)$, let $\mathfrak{G}_{l,l'}$ be a linear predicate over V such that $G(l, l') = \llbracket \mathfrak{G}_{l,l'} \rrbracket$. Here, assume $\mathfrak{G}_{l,l'} := \bigvee_{m=1}^N \mathfrak{G}_{l,l'}^m$, where $\mathfrak{G}_{l,l'}^m$ is a linear conjunctive predicate for each l, l', m .
- Let $\mathfrak{C}$ be a linear conjunctive predicate map such that $C = \llbracket \mathfrak{C} \rrbracket$. Here, assume $\mathfrak{C}(l) := \bigwedge_{i=1}^N r_l^i$, where r_l^i is a linear expression for each i, l .
- Let $\mathfrak{I}$ be a linear predicate map such that $I = \llbracket \mathfrak{I} \rrbracket$. Here, assume $\mathfrak{I}(l) := \bigvee_{j=1}^N \mathfrak{I}_j(l)$, where $\mathfrak{I}_j$ is a linear conjunctive predicate map for each j .

We also introduce the following notations for convenience:

- For the template $\mathfrak{f}$ and an expression a , we define $\mathfrak{f}_{x_v \leftarrow a}$ as an expression map obtained by substituting each x_v in $\mathfrak{f}$ with a .
- For expressions a and b and $\triangleright \in \{>, \geq\}$, we write $a \triangleright b$ to express $a - b \triangleright 0$.
- We define the negation of linear constraint by $\neg(a \geq 0) := -a > 0$ and $\neg(a > 0) := -a \geq 0$.

Now, we provide the axioms of U-NNRepSupM as a finite subset of $\text{Fml}^\Rightarrow$. Recall that the U-NNRepSupM condition for a function $\eta : L \times \mathbb{R}^V \rightarrow \mathbb{R}$ consists of the following:

- (nonnegativity condition) $c \in I \Rightarrow \eta(c) \geq 0$;
- (condition at the target region) $c \in C \Rightarrow \eta(c) \geq 1$;
- (monotonicity condition through a transition) $c \in I \cap \bar{C} \Rightarrow \eta(c) \geq \bar{\mathbb{X}}\eta(c)$.

For each $l \in L$, we give the sets $A_1^l, A_2^l, A_3^l \subseteq \text{Fml}^\Rightarrow$ that correspond to these conditions as follows: While most of them are straightforward interpretations, we note that linearity of $\mathfrak{f}$ is crucial in the definition of A_3^l with $l \in L_{AP}$.

- (Nonnegativity condition) $A_1^l = \left\{ \mathfrak{I}_j(l) \Rightarrow \mathfrak{f}(l) \geq 0 \mid 1 \leq j \leq N \right\}$.
- (Condition at the target region) $A_2^l = \left\{ \mathfrak{C}(l) \Rightarrow \mathfrak{f}(l) \geq 1 \right\}$.
- (Monotonicity condition through a transition) A_3^l is defined as follows:
 - If $l \in L_N$, $A_3^l = \left\{ \mathfrak{I}_j(l) \wedge \neg r_i^l \Rightarrow \mathfrak{f}(l) \geq \mathfrak{f}(l') \mid 1 \leq i, j \leq N, l' \in \text{succ}(l) \right\}$.
 - If $l \in L_P$, $A_3^l = \left\{ \mathfrak{I}_j(l) \wedge \neg r_i^l \Rightarrow \mathfrak{f}(l) \geq \sum_{l' \mapsto l'} \text{Pr}_l(l, l') \mathfrak{f}(l') \mid 1 \leq i, j \leq N \right\}$.
 - If $l \in L_D$, $A_3^l = \left\{ \mathfrak{I}_j(l) \wedge \neg r_i^l \wedge \mathfrak{G}_{l, l'}^m \Rightarrow \mathfrak{f}(l) \geq \mathfrak{f}(l') \mid 1 \leq i, j, m \leq N, l' \in \text{succ}(l) \right\}$.
 - Let $l \in L_A$, $\text{Up}(l) = (v, u)$ and $\text{succ}(l) = l'$.
 - * If u is a measurable function $\mathbb{R}^V \rightarrow \mathbb{R}$ such that $u = \llbracket a_l \rrbracket$, then

$$A_3^l = \left\{ \mathfrak{I}_j(l) \wedge \neg r_i^l \Rightarrow \mathfrak{f}(l) \geq \mathfrak{f}_{x_v \leftarrow a_l}(l') \mid 1 \leq i, j \leq N \right\}.$$

- * If u is a distribution on $\mathbb{R}$ such that $\mathbb{E}u = r_l \in \mathbb{R}$, then

$$A_3^l = \left\{ \mathfrak{I}_j(l) \wedge \neg r_i^l \Rightarrow \mathfrak{f}(l) \geq \mathfrak{f}_{x_v \leftarrow r_l}(l') \mid 1 \leq i, j \leq N \right\}.$$

- * If $u \subseteq \mathbb{R}$ is a set such that $u = \llbracket \mathfrak{P}_l \rrbracket$, then

$$A_3^l = \left\{ \mathfrak{I}_j(l) \wedge \neg r_i^l \wedge \mathfrak{P}_l^n \Rightarrow \mathfrak{f}(l) \geq \mathfrak{f}_{x_v \leftarrow x_{|V|+1}}(l') \mid 1 \leq i, j, n \leq N \right\}.$$

We claim that $A = \bigcup_{l \in L} A_1^l \cup A_2^l \cup A_3^l$ gives the axioms of U-NNRepSupM. For a formula ψ such that $\varphi \Rightarrow \psi \in \text{Fml}^\Rightarrow$, we define $\llbracket \psi \rrbracket \subseteq \mathbb{R}^U \times \mathbb{R}^{V'}$ by

$$\llbracket \psi \rrbracket := \{(u, v) \mid \llbracket p_1 \rrbracket(u)v(x_1) + \dots + \llbracket p_{|V'|} \rrbracket(u)v(x_{|V'|}) + \llbracket q \rrbracket \geq 0\}.$$

PROPOSITION C.1. *Let a pCFG Γ with its terminal region C and I is given, and suppose it obeys Assumption 7.4. Also let $\mathfrak{f}$ be a linear template given as Equation (9). Then for a valuation $u : U \rightarrow \mathbb{R}$ and its vector representation $\vec{u} \in \mathbb{R}^U$, we have:*

$$\begin{aligned} & \left(\forall \vec{x} \in \mathbb{R}^{V'}. \forall l \in L. \forall (\varphi \Rightarrow \psi) \in A. \vec{x} \in \llbracket \varphi \rrbracket \Rightarrow (\vec{u}, \vec{x}) \in \llbracket \psi \rrbracket \right) \\ & \Leftrightarrow (\llbracket \mathfrak{f}_u \rrbracket \upharpoonright_I \text{ is an upper NNRepSupM for } C \text{ supported by } I). \end{aligned}$$

□

D PROOF OF NON-EXISTENCE OF NNREPSUPM AND ε -REPSUPM

Here, we formally prove the hardness of martingale synthesis for randomized programs in Section 8.2. Concretely, we prove that there exists no ε -RepSupM for the programs (c-1), (c-3), and (c-4), and any polynomial (hence, in particular, linear) NNRepSupM for the program (c-2) and (c-5) is trivial. We also explain how the true reachability probability of (c-1)-(c-5) are computed.

D.0.1 (c-1) Bounded Random Walk Over $[0, x_{\text{fail}}]$. The semantics pCFG of (c-1) has its set of locations $L = \{l_1, l_2, l_3, l_4, l_5, l_7, l_{10}, l_F\}$, where each l_i corresponds to the i th line of the program, and l_F corresponds to its termination (which is never reachable in this example; see also Example 2.2). The full description of the semantics pCFG and its terminal region are as follows; for simplicity, we assume $l_F \in L_A$ and omit its transition information.

- $L_N = \emptyset$, $L_P = \{l_4\}$, $L_D = \{l_2, l_3\}$ and $L_A = \{l_1, l_5, l_7, l_{10}, l_F\}$.
- $V = \{x\}$.
- $l_{\text{init}} = l_1$.
- $\vec{x}_{\text{init}} = [x = 0]$.
- $\mapsto = \{(l_1, l_2), (l_2, l_3), (l_2, l_F), (l_3, l_4), (l_3, l_{10}), (l_4, l_5), (l_4, l_7), (l_5, l_2), (l_7, l_2), (l_{10}, l_2)\}$.
- $\text{Up}(l_1) = (x, \lambda x. x_0)$, $\text{Up}(l_5) = (x, \lambda x. x - 1)$, $\text{Up}(l_7) = (x, \lambda x. x + 1)$, and $\text{Up}(l_{10}) = (x, \lambda x. x)$.
- $\text{Pr}(l_4, l_5) = p$, $\text{Pr}(l_4, l_7) = 1 - p$.
- $G(l_2, l_3) = (\text{true})$, $G(l_2, l_F) = (\text{false})$, $G(l_3, l_4) = (x \leq x_{\text{fail}} - 1)$ and $G(l_3, l_{10}) = (x > x_{\text{fail}} - 1)$.
- $C = \{l_2\} \times (-\infty, 0]$.

This pCFG can be seen as the one-dimensional random walk with the initial state $x = x_0$ and two absorbing states $x = 0$ and $x = x_{\text{fail}}$. If $0 < p < 0.5$, then the probability P of eventually reaching $x = 0$ is computed by the following [30]:

$$P = \frac{\left(\frac{p}{1-p}\right)^{x_0} - \left(\frac{p}{1-p}\right)^{x_{\text{fail}}}}{1 - \left(\frac{p}{1-p}\right)^{x_{\text{fail}}}}.$$

There is no ε -RepSupM for this pCFG. Indeed, it can make the following transition:

$$(l_2, [x = x_{\text{fail}}]) \mapsto (l_3, [x = x_{\text{fail}}]) \mapsto (l_{10}, [x = x_{\text{fail}}]) \mapsto (l_2, [x = x_{\text{fail}}]).$$

Any invariant for this pCFG should contain $(l_2, [x = x_{\text{fail}}])$, and no function $\eta : L \times \mathbb{R}^V \rightarrow \mathbb{R}$ can satisfy the ε -decreasing condition over this loop, so one cannot construct an ε -RepSupM for this pCFG.

D.0.2 (c-2) (Modification of (c-1)). The semantics pCFG of (c-2) is obtained by letting $\text{Up}(l_{10}) = (x, \lambda x. x + 1)$ instead in the previous example. The true reachability of this program is the same as (c-1).

Non-existence of a nontrivial polynomial NNRepSupM for this program is proved in a similar way to (c-5) below (slightly more complicated because of the if branch in line 3).

D.0.3 (c-3) (Simple System with Infinite Branching). The full description of the semantics pCFG and its terminal region are as follows; here, $\text{Geom}(0.5)$ denotes a geometric distribution that takes a value $i \in \mathbb{N} \setminus \{0\}$ in a probability 0.5^i .

- $L_N = \emptyset$, $L_P = \{l_5\}$, $L_D = \{l_2, l_6\}$ and $L_A = \{l_1, l_3, l_7, l_{10}, l_F\}$.
- $V = \{x\}$.
- $l_{\text{init}} = l_1$.
- $\vec{x}_{\text{init}} = [x = 0]$.

- $\mapsto = \{(l_1, l_2), (l_2, l_3), (l_2, l_5), (l_3, l_2), (l_5, l_6), (l_5, l_{10}), (l_6, l_7), (l_7, l_6), (l_{10}, l_F)\}$.
- $\text{Up}(l_1) = (x, \text{Geom}(0.5))$, $\text{Up}(l_3) = \text{Up}(l_7) = (x, \lambda x. x - 1)$ and $\text{Up}(l_{10}) = (x, \lambda x. x)$
- $\text{Pr}(l_5, l_6) = \text{Pr}(l_5, l_{10}) = 0.5$.
- $G(l_2, l_3) = (x \geq 1)$, $G(l_2, l_5) = (x < 1)$ and $G(l_6, l_5) = (\text{true})$.
- $C = \{l_{10}\} \times \mathbb{R}$.

The pCFG always reaches from its initial configuration $(l_1, 0)$ to $(l_5, 0)$ and then reaches either $(l_6, 0)$ or $(l_{10}, 0)$ with the probability 0.5. As $(l_{10}, 0) \in C$ and it is impossible to reach C from $(l_6, 0)$, we have $\overline{\mathbb{P}}_C^{\text{reach}}(l_1, 0) = 0.5$.

The non-existence of an ε -RepSupM for this program is proved as follows: Assuming the contrary, let $\eta : L \times \mathbb{R}^V \rightarrow \mathbb{R}$ be an ε -RepSupM for C that has κ -bounded differences for some $\kappa > 0$. By the ε -repulsing condition (Definition 3.1), we have

$$\eta(l_2, 1) \leq \eta(l_3, 2) - \varepsilon \leq \eta(l_2, 2) - 2\varepsilon \leq \eta(l_3, 3) - 3\varepsilon \leq \eta(l_2, 3) - 4\varepsilon \leq \dots$$

Hence, $\eta(l_2, n) \geq \eta(l_2, 1) + 2(n-1)\varepsilon$, and therefore $\lim_{n \rightarrow \infty} \eta(l_2, n) = \infty$. However, for any $n \in \mathbb{N} \setminus \{0\}$ the configuration (l_2, n) is a successor of $(l_1, 0)$; this contradicts the κ -bounded condition of η .

In contrast, if we define a linear expression map $\mathfrak{f}$ and a linear predicate map $\mathfrak{I}$ by $\mathfrak{f}(l_1) = \mathfrak{f}(l_2) = \mathfrak{f}(l_3) = \mathfrak{f}(l_4) = \frac{1}{2}$, $\mathfrak{f}(l_5) = 1$, $\mathfrak{f}(l_6) = 0$ and $\mathfrak{I}(l) = \text{true}$ for each $l \in L$, then $\mathfrak{f}$ is a linear U-NNRepSupM for C supported by $\mathfrak{I}$.

D.0.4 (c-4) (Random Walk Over $[0, 1]$ that Exhibits Geometric Behaviors).

- $L_N = \emptyset$, $L_P = \{l_3\}$, $L_D = \{l_2\}$ and $L_A = \{l_1, l_4, l_5, l_6\}$.
- $V = \{x\}$.
- $l_{\text{init}} = l_1$
- $\vec{x}_{\text{init}} = [x = 0]$.
- $\mapsto = \{(l_1, l_2), (l_2, l_3), (l_2, l_6), (l_3, l_4), (l_3, l_5), (l_4, l_2), (l_5, l_2), (l_6, l_6)\}$.
- $\text{Up}(l_1) = (x, \text{Uniform}[0, 1])$, $\text{Up}(l_4) = (x, \lambda x. 2 * x)$, $\text{Up}(l_5) = (x, \lambda x. 0.5 * x)$ and $\text{Up}(l_6) = (x, \lambda x. x)$ where $\text{Uniform}[0, 1]$ is the uniform distribution over $[0, 1]$.
- $\text{Pr}(l_3, l_4) = 0.25$, $\text{Pr}(l_3, l_5) = 0.75$.
- $G(l_2, l_3) = (x < 1)$ and $G(l_2, l_6) = (x \geq 1)$.

For a given $x' \in (0, 1)$, if we let (l_2, x') be the initial configuration of the pCFG, then it can be identified with a one-dimensional random walk over $\mathbb{N}$ with one absorbing state $x = 0$ and the initial state $\lceil \log_2(1/x) \rceil$ (for example, if $x' = 0.5$, then configurations $(l_2, 1), (l_2, 0.5), (l_2, 0.25), (l_2, 0.125), \dots$ are identified with the states $x = 0, 1, 2, 3, \dots$ in the random walk). In such an asymmetric random walk, the probability P of eventually reaching $x = 0$ is computed as follows [30]; here, x_0 is the initial value of x and $p \in [0.5, 1]$ is the probability that x is updated to $x + 1$ in each step.

$$P = \left(\frac{1-p}{p} \right)^{x_0}. \quad (17)$$

Our pCFG makes a transition from the initial configuration $(l_1, 0)$ to (l_2, x') with $x' \in [0, 1]$ sampled from $\text{Unif}[0, 1]$. Hence, by taking the mean of $\left(\frac{1-0.75}{0.75} \right)^{\lceil \log_2(1/x) \rceil}$ over $x \in (0, 1]$, we have

$$\overline{\mathbb{P}}_C^{\text{reach}}(l_1, 0) = \int_0^1 \left(\frac{0.25}{0.75} \right)^{\lceil \log_2(1/x) \rceil} dx \approx 0.2.$$

The non-existence of an ε -RepSupM for this program is proved as follows: Let $\eta : L \times \mathbb{R}^V \rightarrow \mathbb{R}$ be an ε -RepSupM for C that has κ -bounded differences for some $\kappa > 0$. Assume $\eta(l_1, 0) = A < 0$.

Then, we have: $A = \eta(l_1, 0) \geq \int_{x \in [0,1]} \eta(l_2, x) + \varepsilon$. Hence, there exists $x > 0$ such that $\eta(l_2, x) \leq A < 0$.

Let $N \in \mathbb{N}$ be the minimum number such that $2^N x \geq 1$. As $(l_6, 2^N x) \in C$, we have $\eta(l_2, 2^N x) \geq \eta(l_6, 2^N x) + \varepsilon \geq 0$, which means that there exists $n' \in \{0, \dots, N-1\}$ such that $\eta(l_2, 2^{n'} x) - \eta(l_2, 2^{n'+1} x) > 0$.

For each $n \in \mathbb{Z}$ such that $n \leq N-1$, we have:

$$\begin{aligned} \eta(l_2, 2^n x) &\geq \eta(l_3, 2^n x) + \varepsilon \\ &\geq 0.25 \cdot \eta(l_4, 2^{n+1} x) + 0.75 \cdot \eta(l_5, 2^{n+1} x) + 2\varepsilon \\ &\geq 0.25 \cdot \eta(l_2, 2^{n+1} x) + 0.75 \cdot \eta(l_2, 2^{n+1} x) + 3\varepsilon. \end{aligned}$$

This implies:

$$\eta(l_2, 2^{n-1} x) - \eta(l_2, 2^n x) \leq \frac{0.25}{0.75} (\eta(l_2, 2^n x) - \eta(l_2, 2^{n+1} x)) - \frac{0.25}{0.75} \cdot 3\varepsilon.$$

By the two discussions above, we can inductively prove $\eta(l_2, 2^n x) - \eta(l_2, 2^{n+1} x) \leq -\frac{1}{3} \cdot 3(n - n')\varepsilon$ for each $n \leq n'$. Hence, $\lim_{n \rightarrow \infty} f(l_2, 2^n x) - f(l_1, 0) = -\infty$, and it contradicts to the κ -bounded condition.

In contrast, if we define a linear expression map $\mathfrak{f}$ and a linear predicate map $\mathfrak{S}$ by $\mathfrak{f}(l_1) = \frac{1}{2}$, $\mathfrak{f}(l_2) = \mathfrak{f}(l_3) = x$, $\mathfrak{f}(l_4) = 2x$, $\mathfrak{f}(l_5) = \frac{1}{2}x$ and $\mathfrak{S}(l) = (x \geq 0)$ for each $l \in L$, then $\mathfrak{f}$ is a linear NNRepSupM for $\mathfrak{C}$ at 1 supported by $\mathfrak{S}$.

D.0.5 (c-5) (Unbounded Random Walk).

- $L_N = \emptyset$, $L_P = \{l_1\}$, $L_D = \emptyset$ and $L_A = \{l_2, l_3\}$.
- $V = \{x\}$.
- $l_{\text{init}} = l_1$
- $\vec{x}_{\text{init}} = [x = 1]$.
- $\mapsto = \{(l_1, l_2), (l_1, l_3), (l_2, l_1), (l_3, l_1)\}$.
- $\text{Up}(l_2) = (x, \lambda x. x - 1)$ and $\text{Up}(l_3) = (x, \lambda x. x + 1)$.
- $\text{Pr}(l_1, l_2) = 0.25$, $\text{Pr}(l_1, l_3) = 0.75$.
- G is the empty function.

We define $C \subseteq L \times \mathbb{R}^V$ by $C := \{(l_1, [x \mapsto r]) \mid r \leq 0\}$ and let $\mathfrak{C}$ be a linear predicate map representing C .

As this is the random walk over $\mathbb{N}$ with one absorbing state $x = 0$ and the initial state $x = 1$, its reachability probability from the initial configuration is $\frac{0.25}{0.75} \approx 0.333$ by Equation (17).

The non-existence of nontrivial polynomial NNRepSupM is proved as follows: By the NNRepSupM condition at outside of C , we have $\eta(l_1, x) \geq 0.25 \cdot \eta(l_1, x-1) + 0.75 \cdot \eta(l_1, x+1)$ for each $x \geq 1$. Now assume $\eta(l_1, 1) < 1$ (which implies $\eta(l_1, 1) < \eta(l_1, 0)$) and $\eta(l_2, _)$ is polynomial. Then by the aforementioned NNRepSupM condition, we can inductively prove $\eta(l_1, x+1) < \eta(l_1, x)$ for each $x \in \mathbb{N}$. Also recall that $\eta(l_1, x) \geq 0$ by the nonnegative condition of NNRepSupM. Clearly, there is no polynomial function that realizes such strict decreasing and nonnegative conditions; hence, we conclude any polynomial NNRepSupM η should satisfy $\eta(l_1, 1) \geq 1$.

E THE CODES OF RANDOMIZED PROGRAMS USED IN EXPERIMENTS

Below, we provide the codes of randomized programs that we used in experiments.

```

1  x := 2;
2  while x <= 10 do
3    if prob( $p_1$ ) then
4      skip
5    else
6      if prob( $\frac{p_2}{1-p_1}$ ) then
7        x := x + 1
8      else
9        x := x + 2
10     fi
11   fi;
12   if * then
13     if prob(0.875) then
14       x := x - 1
15     else
16       skip
17     fi
18   else
19     if prob( $p_1$ ) then
20       skip
21     else
22       if prob( $\frac{p_2}{1-p_1}$ ) then
23         x := x + 1
24       else
25         x := x + 2
26       fi
27     fi;
28     x := x - 1
29   fi;
30   [x <= 0] skip
31 od

```

Fig. 5. (a-1) 1D adversarial random walk [14]. Here, $p_1, p_2 \in [0, 1]$.

```

1  x := 2;
2  y := 2;
3  t := 0;
4  while t <= 100 do
5    if * then
6      z := Unif ( $M_1, M_2$ );
7      x := x + z
8    else
9      z := Unif ( $M_1, M_2$ );
10     y := y + z
11   fi;
12   t := t + 1;
13   [x <= 0] skip;
14   [y <= 0] skip
15 od

```

Fig. 6. (a-2) 2D adversarial random walk [14]. Here, $M_1, M_2 \in \mathbb{R}$.

```

1  x := 3; y := 2; t := 0;
2  while t <= 100 do
3    if * then
4      if prob(0.7) then
5        z := Unif ( $M_1, M_2$ );
6        x := x + z
7      else
8        z := Unif ( $M_1, M_2$ );
9        y := y + z
10     fi
11   else
12     if prob(0.7) then
13       z := Unif ( $-M_2, -M_1$ );
14       y := y + z
15     else
16       z := Unif ( $-M_2, -M_1$ );
17       x := x + z
18     fi
19   fi;
20   t := t + 1;
21   [x <= y] skip
22 od

```

Fig. 7. (a-3) variant of 2D adversarial random walk [14]. Here, $M_1, M_2 \in \mathbb{R}$.

```

1  x0 := 6;
2  x1 := ndet Real[17,22];
3  x2 := ndet Real[16,23];
4  t := 0;
5  while t <= 100 do
6    controller1 := 19.5 - x1;
7    controller2 := 19.5 - x2;
8    noise1 := Unif(-p,p);
9    noise2 := Unif(-p,p);
10   x1 := x1 + 0.0375 * x0 - 0.0375 * x1 + 0.0625 * x2
11   - 0.0625 * x1 + c * controller1 + noise1;
12   x2 := x2 + 0.025 * x0 - 0.025 * x2 + 0.0625 * x1
13   - 0.0625 * x2 + c * controller2 + noise2;
14   t := t + 1;
15   [x1 < 17] skip;
16   [x1 > 22] skip;
17   [x2 < 16] skip;
18   [x2 > 23] skip
19 od

```

Fig. 8. (b) room temperature control [1, 11]. Here, $c, p \in \mathbb{R}$.

```

1  x := x0;
2  [x ≤ 0] while true do
3    if x ≤ xfail - 1 then
4      if prob(p) then
5        x := x - 1
6      else
7        x := x + 1
8      fi
9    else
10     skip
11   fi
12 od

```

Fig. 9. (c-1) bounded random walk over $[0, x_{\text{fail}}]$. Here, $p \in [0, 1]$ and $x_0, x_{\text{fail}} \in \mathbb{N}$.

```

1  x := x0;
2  [x ≤ 0] while true do
3    if x ≤ xfail - 1 then
4      if prob(p) then
5        x := x - 1
6      else
7        x := x + 1
8      fi
9    else
10     x := x + 1
11   fi
12 od

```

Fig. 10. (c-2) A modification of (c-1). Here, $p \in [0, 1]$ and $x_0, x_{\text{fail}} \in \mathbb{N}$.

```

1  x := Geom(0.5);
2  while x ≥ 1 do
3    x := x - 1
4  od;
5  if prob(0.5) then
6    while true do
7      x := x - 1
8    od
9  else
10   skip
11 fi;
12 [true]

```

Fig. 11. (c-3) simple system with infinite branching.

```

1  x := Unif(0, 1);
2  while true do
3    if prob(0.25) then
4      x := 2 * x
5    else
6      x := 0.5 * x
7    fi;
8    [x ≥ 1] skip
9  od

```

Fig. 12. (c-4) random walk over $[0, 1]$ that exhibits geometric behaviors.

```

1  x := 5;
2  while true do
3    if prob(0.25) then
4      x := x - 1
5    else
6      x := x + 1
7    fi;
8    [x ≤ 0] skip
9  od

```

Fig. 13. (c-5) unbounded random walk.

ACKNOWLEDGMENT

We thank Shin-ya Katsumata, Takamasa Okudono, Thomas Haas, and the anonymous referees for helpful comments.

REFERENCES

- [1] Alessandro Abate, Joost-Pieter Katoen, John Lygeros, and Maria Prandini. 2010. Approximate model checking of stochastic hybrid systems. *Eur. J. Cont.* 16, 6 (2010), 624–641.
- [2] Sheshansh Agrawal, Krishnendu Chatterjee, and Petr Novotný. 2018. Lexicographic ranking supermartingales: An efficient approach to termination of probabilistic programs. *Proc. ACM Prog. Lang.* 2, POPL (2018), 34:1–34:32.
- [3] Christophe Alias, Alain Darte, Paul Feautrier, and Laure Gonnord. 2010. Multi-dimensional rankings, program termination, and complexity bounds of flowchart programs. In *Proceedings of the 17th International Static Analysis Symposium, (SAS’10) (Lecture Notes in Computer Science)*, Radhia Cousot and Matthieu Martel (Eds.), Vol. 6337. Springer, 117–133. DOI : https://doi.org/10.1007/978-3-642-15769-1_8
- [4] Krzysztof R. Apt and Gordon D. Plotkin. 1986. Countable nondeterminism and random assignment. *J. ACM* 33, 4 (1986), 724–767. DOI : <https://doi.org/10.1145/6490.6494>
- [5] André Arnold and Damian Niwiński. 2001. *Rudiments of μ -Calculus*. Elsevier.
- [6] Martin Avanzini, Ugo Dal Lago, and Akihisa Yamada. 2018. On probabilistic term rewriting. In *Proceedings of the International Symposium on Functional and Logic Programming (FLOPS’18) (Lecture Notes in Computer Science)*, John P. Gallagher and Martin Sulzmann (Eds.), Vol. 10818. Springer, 132–148.
- [7] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking*. The MIT Press.
- [8] Dimitri P. Bertsekas and Steven E. Shreve. 2007. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific.
- [9] Rastislav Bodík and Rupak Majumdar (Eds.). 2016. In *Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages*. ACM.
- [10] Aleksandar Chakarov and Sriram Sankaranarayanan. 2013. Probabilistic program analysis with martingales. In *Proceedings of the International Conference on Computer Aided Verification (LNCS)*, Natasha Sharygina and Helmut Veith (Eds.), Vol. 8044. Springer, 511–526.
- [11] Aleksandar Chakarov, Yuen-Lam Voronin, and Sriram Sankaranarayanan. 2016. Deductive proofs of almost sure persistence and recurrence properties. In *Proceedings of the 22nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems, (TACAS’16) (LNCS)*, Marsha Chechik and Jean-François Raskin (Eds.), Vol. 9636. Springer, 260–279.
- [12] Krishnendu Chatterjee and Hongfei Fu. 2017. Termination of nondeterministic recursive probabilistic programs. *CoRR* abs/1701.02944 (2017).
- [13] Krishnendu Chatterjee, Hongfei Fu, and Amir Kafshdar Goharshady. 2016. Termination analysis of probabilistic programs through Positivstellensatz’s. In *Proceedings of the International Conference on Computer Aided Verification (LNCS)*, Swarat Chaudhuri and Azadeh Farzan (Eds.), Vol. 9779. Springer, 3–22.
- [14] Krishnendu Chatterjee, Hongfei Fu, Petr Novotný, and Rouzbeh Hasheminezhad. 2016. Algorithmic analysis of qualitative and quantitative termination problems for affine probabilistic programs. In *Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages*. 327–342.
- [15] Krishnendu Chatterjee, Petr Novotný, and Dorde Zikelic. 2017. Stochastic invariants for probabilistic termination. In *Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages (POPL’17)*, Giuseppe Castagna and Andrew D. Gordon (Eds.). ACM, 145–160.
- [16] Radhia Cousot and Patrick Cousot. 1979. Constructive versions of Tarski’s fixed point theorems. *Pacific J. Math.* 82, 1 (1979), 43–57.
- [17] Luis María Ferrer Fioriti and Holger Hermanns. 2015. Probabilistic termination: Soundness, completeness, and compositionality. In *Proceedings of the 42nd SIGPLAN-SIGACT Symposium on Principles of Programming Languages, (POPL’15)*, Sriram K. Rajamani and David Walker (Eds.). ACM, 489–501. DOI : <https://doi.org/10.1145/2676726.2677001>
- [18] A. Makhorin. 2008. GLPK—GNU Linear Programming Kit. <http://www.gnu.org/software/glpk/>.
- [19] Andrew D. Gordon, Thomas A. Henzinger, Aditya V. Nori, and Sriram K. Rajamani. 2014. Probabilistic programming. In *Proceedings of the Conference on the Future of Software Engineering, (FOSE’14)*, James D. Herbsleb and Matthew B. Dwyer (Eds.). ACM, 167–181.
- [20] Ichiro Hasuo, Shunsuke Shimizu, and Corina Cirstea. 2016. Lattice-theoretic progress measures and coalgebraic model checking. In *Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages*. 718–732.
- [21] Roger A. Horn and Charles R. Johnson. 2012. *Matrix Analysis (2nd ed.)*. Cambridge University Press, New York, NY.

- [22] Marcin Jurdzinski. 2000. Small progress measures for solving parity games. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS'00) (LNCS)*, Horst Reichel and Sophie Tison (Eds.), Vol. 1770. Springer, 290–301.
- [23] Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. 2016. Weakest precondition reasoning for expected run-times of probabilistic programs. In *Proceedings of the 25th European Symposium on Programming Languages and Systems, (ESOP'16), Held as Part of the European Joint Conferences on Theory and Practice of Software, (ETAPS'16) (Lecture Notes in Computer Science)*, Peter Thiemann (Ed.), Vol. 9632. Springer, 364–389.
- [24] Joost-Pieter Katoen, Annabelle McIver, Larissa Meinicke, and Carroll C. Morgan. 2010. Linear-invariant generation for probabilistic programs: Automated support for proof-based methods. In *Proceedings of the 17th International Symposium on Static Analysis (SAS'10)*. 390–406.
- [25] Dexter Kozen. 1985. A probabilistic PDL. *J. Comput. Syst. Sci.* 30, 2 (1985), 162–178. DOI: [https://doi.org/10.1016/0022-0000\(85\)90012-1](https://doi.org/10.1016/0022-0000(85)90012-1)
- [26] Satoshi Kura, Natsuki Urabe, and Ichiro Hasuo. 2019. Tail probabilities for randomized program runtimes via martingales for higher moments. In *Proceedings of the 25th International Conference, on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'19), Held as Part of the European Joint Conferences on Theory and Practice of Software, (ETAPS'19) (Lecture Notes in Computer Science)*, Tomás Vojnar and Lijun Zhang (Eds.), Vol. 11428. Springer, 135–153. DOI: https://doi.org/10.1007/978-3-030-17465-1_8
- [27] Annabelle McIver and Carroll Morgan. 2004. *Abstraction, Refinement and Proof for Probabilistic Systems (Monographs in Computer Science)*. SpringerVerlag.
- [28] Annabelle McIver and Carroll Morgan. 2004. Developing and reasoning about probabilistic programs in *pGCL*. In *Proceedings of the 1st Pernambuco Summer School on Software Engineering: Refinement Techniques in Software Engineering (PSSE'04) (LNCS)*, Ana Cavalcanti, Augusto Sampaio, and Jim Woodcock (Eds.), Vol. 3167. Springer, 123–155.
- [29] Annabelle McIver, Carroll Morgan, Benjamin Lucien Kaminski, and Joost-Pieter Katoen. 2018. A new proof rule for almost-sure termination. *Proc. ACM Prog. Lang.* 2, POPL (2018), 33:1–33:28. DOI: <https://doi.org/10.1145/3158121>
- [30] Frederick Mosteller. 2012. *Fifty Challenging Problems in Probability with Solutions*. Dover Publications.
- [31] Arkadi Nemirovski and Cornelis Roos. 2008. Motzkin transposition theorem. *Encyclopedia of Optimization* (2nd edition), C. A. Floudas and P. M. Pardalos (Eds.). Springer, 235–2348. https://doi.org/10.1007/978-0-387-74759-0_405.
- [32] Van Chan Ngo, Quentin Carbonneaux, and Jan Hoffmann. 2018. Bounded expectations: Resource analysis for probabilistic programs. In *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation, (PLDI'18)*, Jeffrey S. Foster and Dan Grossman (Eds.). ACM, 496–512. DOI: <https://doi.org/10.1145/3192366.3192394>
- [33] Stephen Prajna, Ali Jadbabaie, and George J. Pappas. 2004. Stochastic safety verification using barrier certificates. In *Proceedings of the 43rd IEEE Conference on Decision and Control. IEEE, Piscataway, NJ*, 929–934.
- [34] Konrad Schmüdgen. 1991. The K-moment problem for compact semi-algebraic sets. *Math. Ann.* 289, 1 (01 Mar. 1991), 203–206.
- [35] Alexander Schrijver. 1998. *Theory of Linear and Integer Programming*. Wiley.
- [36] K. C. Toh, M. J. Todd, and R. H. Tutuncu, SDPT3—a Matlab software package for semidefinite programming. *Optimization Methods and Software* 11 (1999), 545–581.
- [37] Antonis Papachristodoulou, James Anderson, Giorgio Valmorbida, Stephen Prajna, Pete Seiler, and Pablo A. Parrilo. 2013. SOSTOOLS Version 3.00 Sum of Squares Optimization Toolbox for MATLAB. CoRR abs/1310.4716.
- [38] Jacob Steinhardt and Russ Tedrake. 2012. Finite-time regional verification of stochastic non-linear systems. *Int. J. Robot. Res.* 31, 7 (2012), 901–923.
- [39] Toru Takisaka, Yuichiro Oyabu, Natsuki Urabe, and Ichiro Hasuo. 2018. Ranking and repulsing supermartingales for reachability in probabilistic programs. In *Proceedings of the 16th International Symposium on Automated Technology for Verification and Analysis, (ATVA'18) (Lecture Notes in Computer Science)*, Shuvendu K. Lahiri and Chao Wang (Eds.), Vol. 11138. Springer, 476–493. DOI: https://doi.org/10.1007/978-3-030-01090-4_28
- [40] Alfred Tarski. 1951. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkeley.
- [41] Natsuki Urabe, Masaki Hara, and Ichiro Hasuo. 2017. Categorical liveness checking by corecursive algebras. In *Proceedings of the ACM-IEEE Symposium on Logic in Computer Science (LICS'17)*. IEEE Computer Society, 1–12.
- [42] Moshe Y. Vardi. 1995. An automata-theoretic approach to linear temporal logic. In *Proceedings of the Banff Higher Order Workshop (Lecture Notes in Computer Science)*, Faron Moller and Graham M. Birtwistle (Eds.), Vol. 1043. Springer, 238–266.

Received May 2019; revised January 2021; accepted February 2021