Discounted Properties of Probabilistic Pushdown Automata*

Tomáš Brázdil, Václav Brožek, Jan Holeček, and Antonín Kučera

Faculty of Informatics, Masaryk University,
Botanická 68a, 60200 Brno,
Czech Republic.
{brazdil,brozek,holecek,kucera}@fi.muni.cz

Abstract. We show that several basic discounted properties of probabilistic pushdown automata related both to terminating and non-terminating runs can be efficiently approximated up to an arbitrarily small given precision.

1 Introduction

Discounting formally captures the natural intuition that the far-away future is not as important as the near future. In the discrete time setting, the discount assigned to a state visited after k time units is λ^k, where $0 < \lambda < 1$ is a fixed constant. Thus, the "weight" of states visited lately becomes progressively smaller. Discounting (or inflation) is a key paradigm in economics and has been studied in Markov decision processes as well as game theory [18, 15]. More recently, discounting has been found appropriate also in systems theory (see, e.g., [6]), where it allows to put more emphasis on events that occur early. For example, even if a system is guaranteed to handle every request eventually, it still makes a big difference whether the request is handled early or lately, and discounting provides a convenient formalism for specifying and even quantifying this difference.

In this paper, we concentrate on basic discounted properties of probabilistic pushdown automata (pPDA), which provide a natural model for probabilistic systems with unbounded recursion [9, 4, 10, 3, 14, 12, 13]. Thus, we aim at filling a gap in our knowledge on probabilistic PDA, which has so far been limited only to non-discounted properties. As the main result, we show that several fundamental discounted properties related to long-run behaviour of probabilistic PDA (such as the discounted gain or the total discounted accumulated reward) are expressible as the least solutions of efficiently constructible systems of recursive monotone polynomial equations whose form admits the application of the recent results [17, 8] about a fast convergence of Newton's approximation method. This entails the decidability of the corresponding *quantitative problems* (we ask whether the value of a given discounted long-run property is equal to or bounded by a given rational constant). A more important consequence is that the discounted long-run properties are *computational* in the sense that they can be efficiently

* Supported by the research center Institute for Theoretical Computer Science (ITI), project No. 1M0545.

approximated up to an arbitrarily small given precision. This is very different from the non-discounted case, where the respective quantitative problems are also decidable but no efficient approximation schemes are known[1]. This shows that discounting, besides its natural practical appeal, has also mathematical and computational benefits.

We also consider discounted properties related to terminating runs, such as the discounted termination probability and the discounted reward accumulated along a terminating run. Further, we examine the relationship between the discounted and non-discounted variants of a given property. Intuitively, one expects that a discounted property should be close to its non-discounted variant as the discount approaches 1. This intuition is mostly confirmed, but in some cases the actual correspondence is more complicated.

Concerning the level of originality of the presented work, the results about terminating runs are obtained as simple extensions of the corresponding results for the non-discounted case presented in [9, 14, 10]. New insights and ideas are required to solve the problems about discounted long-run properties of probabilistic PDA (the discounted gain and the total discounted accumulated reward), and also to establish the correspondence between these properties and their non-discounted versions. A more detailed discussion and explanation is postponed to Sections 3 and 4.

Since most of the proofs are rather technical, they are not included in this paper. The main body of the paper contains the summary of the main results, and the relationship to the previous work is carefully discussed at appropriate places. Proofs in full detail can be found in [2].

2 Basic Definitions

In this paper, we use $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Q}$, and $\mathbb{R}$ to denote the sets of positive integers, nonnegative integers, rational numbers, and real numbers, respectively. We also use the standard notation for intervals of real numbers, writing, e.g., $(0, 1]$ to denote the set $\{x \in \mathbb{R} \mid 0 < x \leq 1\}$.

The set of all finite words over a given alphabet Σ is denoted Σ^*, and the set of all infinite words over Σ is denoted Σ^ω. We also use Σ^+ to denote the set $\Sigma^* \setminus \{\varepsilon\}$ where ε is the empty word. The length of a given $w \in \Sigma^* \cup \Sigma^\omega$ is denoted $len(w)$, where the length of an infinite word is ω. Given a word (finite or infinite) over Σ, the individual letters of w are denoted $w(0), w(1), \cdots$.

Let $V \neq \emptyset$, and let $\rightarrow\ \subseteq V \times V$ be a *total* relation (i.e., for every $v \in V$ there is some $u \in V$ such that $v \rightarrow u$). The reflexive and transitive closure of $\rightarrow$ is denoted $\rightarrow^*$. A *path* in $(V, \rightarrow)$ is a finite or infinite word $w \in V^+ \cup V^\omega$ such that $w(i{-}1) \rightarrow w(i)$ for every $1 \leq i < len(w)$. A *run* in $(V, \rightarrow)$ is an infinite path in V. The set of all runs that start with a given finite path w is denoted $Run(w)$.

A *probability distribution* over a finite or countably infinite set X is a function $f : X \rightarrow [0, 1]$ such that $\sum_{x \in X} f(x) = 1$. A probability distribution f over X is *positive* if $f(x) > 0$ for every $x \in X$, and *rational* if $f(x) \in \mathbb{Q}$ for every $x \in X$. A *σ-field* over a set Ω is a set $\mathcal{F} \subseteq 2^\Omega$ that includes Ω and is closed under complement and countable union.

[1] For example, the existing approximation methods for the (non-discounted) gain employ the decision procedure for the existential fragment of $(\mathbb{R}, +, *, \leq)$, which is rather inefficient.

A *probability space* is a triple $(\Omega, \mathcal{F}, \mathcal{P})$ where Ω is a set called *sample space*, $\mathcal{F}$ is a σ-field over Ω whose elements are called *events*, and $\mathcal{P} : \mathcal{F} \to [0,1]$ is a *probability measure* such that, for each countable collection $\{X_i\}_{i\in I}$ of pairwise disjoint elements of $\mathcal{F}$ we have that $\mathcal{P}(\bigcup_{i\in I} X_i) = \sum_{i\in I} \mathcal{P}(X_i)$, and moreover $\mathcal{P}(\Omega)=1$. A *random variable* over a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ is a function $X : \Omega \to \mathbb{R} \cup \{\perp\}$, where $\perp \notin \mathbb{R}$ is a special "undefined" symbol, such that $\{\omega \in \Omega \mid X(\omega) \leq c\} \in \mathcal{F}$ for every $c \in \mathbb{R}$. If $\mathcal{P}(X{=}\perp) = 0$, then the *expected value* of X is defined by $\mathbb{E}[X] = \int_{\omega\in\Omega} X(\omega)\, \mathrm{d}\mathcal{P}$.

Definition 1 (Markov Chain). *A* Markov chain *is a triple $M = (S, \to, Prob)$ where S is a finite or countably infinite set of* states, *$\to \subseteq S \times S$ is a total* transition relation*, and $Prob$ is a function which to each state $s \in S$ assigns a positive probability distribution over the outgoing transitions of s. As usual, we write $s \xrightarrow{x} t$ when $s \to t$ and x is the probability of $s \to t$.*

To every $s \in S$ we associate the probability space $(Run(s), \mathcal{F}, \mathcal{P})$ where $\mathcal{F}$ is the σ-field generated by all *basic cylinders* $Run(w)$ where w is a finite path starting with s, and $\mathcal{P} : \mathcal{F} \to [0,1]$ is the unique probability measure such that $\mathcal{P}(Run(w)) = \Pi_{i=1}^{len(w)-1} x_i$ where $w(i{-}1) \xrightarrow{x_i} w(i)$ for every $1 \leq i < len(w)$. If $len(w) = 1$, we put $\mathcal{P}(Run(w)) = 1$.

Definition 2 (probabilistic PDA). *A* probabilistic pushdown automaton (pPDA) *is a tuple $\Delta = (Q, \Gamma, \delta, Prob)$ where Q is a finite set of* control states*, Γ is a finite* stack alphabet*, $\delta \subseteq (Q \times \Gamma) \times (Q \times \Gamma^{\leq 2})$ is a* transition relation *(here $\Gamma^{\leq 2} = \{w \in \Gamma^* \mid len(w) \leq 2\}$), and $Prob : \delta \to (0,1]$ is a rational* probability assignment *such that for all $pX \in Q \times \Gamma$ we have that $\sum_{pX \to q\alpha} Prob(pX \to q\alpha) = 1$.*

A configuration *of Δ is an element of $Q \times \Gamma^*$, and the set of all configurations of Δ is denoted $C(\Delta)$.*

To each pPDA $\Delta = (Q, \Gamma, \delta, Prob_\Delta)$ we associate a Markov chain $M_\Delta = (C(\Delta), \to, Prob)$, where $p\varepsilon \xrightarrow{1} p\varepsilon$ for every $p \in Q$, and $pX\beta \xrightarrow{x} q\alpha\beta$ iff $(pX, q\alpha) \in \delta$, $Prob_\Delta(pX \to q\alpha) = x$, and $\beta \in \Gamma^*$. For all $p, q \in Q$ and $X \in \Gamma$, we use $Run(pXq)$ to denote the set of all $w \in Run(pX)$ such that $w(n) = q\varepsilon$ for some $n \in \mathbb{N}$, and $Run(pX\uparrow)$ to denote the set $Run(pX) \setminus \bigcup_{q\in Q} Run(pXq)$. The runs of $Run(pXq)$ and $Run(pX\uparrow)$ are sometimes referred to as *terminating* and *non-terminating*, respectively.

3 Discounted Properties of Probabilistic PDA

In this section we introduce the family of discounted properties of probabilistic PDA studied in this paper. These notions are not PDA-specific and could be defined more abstractly for arbitrary Markov chains. Nevertheless, the scope of our study is limited to probabilistic PDA, and the notation becomes more suggestive when it directly reflects the structure of a given pPDA.

For the rest of this section, we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob_\Delta)$, a nonnegative *reward function* $f : Q \times \Gamma \to \mathbb{Q}$, and a *discount function* $\lambda : Q \times \Gamma \to [0,1)$. The functions f and λ are extended to all elements of $Q \times \Gamma^+$ by stipulating that $f(pX\alpha) = f(pX)$ and $\lambda(pX\alpha) = \lambda(pX)$, respectively. One can easily generalize the presented arguments also to rewards and discounts that depend on the whole stack content, provided that this dependence is "regular", i.e., can be encoded by a finite-state automaton which reads the stack bottom-up. This extension is obtained just by applying standard techniques that have been used in, e.g., [11]. Also note that λ can assign a different discount to each element of $Q \times \Gamma$, and that the discount can also be 0. Hence, we in fact work with a slightly generalized form of discounting which can also reflect relative speed of transitions.

We start by defining several simple random variables. The definitions are parametrized by the functions f and λ, control states $p, q \in Q$, and a stack symbol $X \in \Gamma$. For every run w and $i \in \mathbb{N}_0$, we use $\lambda(w^i)$ to denote $\Pi_{j=0}^{i} \lambda(w(j))$, i.e., the discount accumulated up to $w(i)$. Note that the initial state of w is also discounted, which is somewhat non-standard but technically convenient (the equations constructed in Section 4 become more readable).

$$I_{pXq}(w) = \begin{cases} 1 & \text{if } w \in Run(pXq) \\ 0 & \text{otherwise} \end{cases}$$

$$I^{\lambda}_{pXq}(w) = \begin{cases} \lambda(w^{n-1}) & \text{if } w \in Run(pXq),\ w(n-1) \neq w(n) = q\varepsilon \\ 0 & \text{otherwise} \end{cases}$$

$$R^{f}_{pXq}(w) = \begin{cases} \sum_{i=0}^{n-1} f(w(i)) & \text{if } w \in Run(pXq),\ w(n-1) \neq w(n) = q\varepsilon \\ 0 & \text{otherwise} \end{cases}$$

$$R^{f,\lambda}_{pXq}(w) = \begin{cases} \sum_{i=0}^{n-1} \lambda(w^i) \cdot f(w(i)) & \text{if } w \in Run(pXq),\ w(n-1) \neq w(n) = q\varepsilon \\ 0 & \text{otherwise} \end{cases}$$

$$G^{f}_{pX}(w) = \begin{cases} \lim_{n\to\infty} \frac{\sum_{i=0}^{n} f(w(i))}{n+1} & \text{if } w \in Run(pX\uparrow) \text{ and the limit exists} \\ \perp & \text{otherwise} \end{cases}$$

$$G^{f,\lambda}_{pX}(w) = \begin{cases} \lim_{n\to\infty} \frac{\sum_{i=0}^{n} \lambda(w^i)\cdot f(w(i))}{\sum_{i=0}^{n} \lambda(w^i)} & \text{if } w \in Run(pX\uparrow) \text{ and the limit exists} \\ \perp & \text{otherwise} \end{cases}$$

$$X^{f,\lambda}_{pX}(w) = \begin{cases} \sum_{i=0}^{\infty} \lambda(w^i) f(w(i)) & \text{if } w \in Run(pX\uparrow) \\ 0 & \text{otherwise} \end{cases}$$

The variable I_{pXq} is a simple indicator telling whether a given run belongs to $Run(pXq)$ or not. Hence, $\mathbb{E}[I_{pXq}]$ is the probability of $Run(pXq)$, i.e., the probability of all runs $w \in Run(pX)$ that terminate in $q\varepsilon$. The variable I^{λ}_{pXq} is the discounted version of I_{pXq}, and its expected value can thus be interpreted as the "discounted termination probability", where more weight is put on terminated states visited early. Hence, $\mathbb{E}[I^{\lambda}_{pXq}]$ is a meaningful value which can be used to quantify the difference between two configurations with the same termination probability but different termination time. From now on, we write $[pXq]$ and $[pXq, \lambda]$ instead of $\mathbb{E}[I_{pXq}]$ and $\mathbb{E}[I^{\lambda}_{pXq}]$, respectively, and we also use $[pX\uparrow]$ to denote $1 - \sum_{q \in Q}[pXq]$. The computational aspects of $[pXq]$ have been examined in [9, 14], where it is shown that the family of all $[pXq]$ forms the least solution of an effectively constructible system of monotone polynomial equations. By applying the recent results [17, 8] about a fast convergence of Newton's method, it is possible to approximate the values of $[pXq]$ efficiently (the precise values of $[pXq]$ can be irrational). In Section 4, we generalize these results to $[pXq, \lambda]$.

The variable R^{f}_{pXq} returns to every $w \in Run(pXq)$ the total f-reward accumulated up to $q\varepsilon$. For example, if $f(rY) = 1$ for every $rY \in Q \times \Gamma$, then the variable returns the number of transitions executed before hitting the configuration $q\varepsilon$. In [10], the *conditional* expected value $\mathbb{E}[R^{f}_{pXq} \mid Run(pXq)]$ has been studied in detail. This value can be used to analyze important properties of terminating runs; for example, if f is as above, then

$$\sum_{q \in Q} [pXq] \cdot \mathbb{E}[R^{f}_{pXq} \mid Run(pXq)]$$

is the conditional expected termination time of a run initiated in pX, under the condition that the run terminates (i.e., the stack is eventually emptied). In [10], it has been shown that the family of all $\mathbb{E}[R^{f}_{pXq} \mid Run(pXq)]$ forms the least solution of an effectively constructible system of recursive polynomial equations. One disadvantage of $\mathbb{E}[R^{f}_{pXq} \mid Run(pXq)]$ (when compared to $[pXq, \lambda]$ which also reflects the length of terminating runs) is that this conditional expected value can be infinite even in situations when $[pXq] = 1$.

The discounted version $R^{f,\lambda}_{pXq}$ of R^{f}_{pXq} assigns to each $w \in Run(pXq)$ the total discounted reward accumulated up to $q\varepsilon$. In Section 4, we extend the aforementioned results about $\mathbb{E}[R^{f}_{pXq} \mid Run(pXq)]$ to the family of all $\mathbb{E}[R^{f,\lambda}_{pXq} \mid Run(pXq)]$. The extension is actually based on analyzing the properties of the (unconditional) expected value $\mathbb{E}[R^{f,\lambda}_{pXq}]$. At first glance, $\mathbb{E}[R^{f,\lambda}_{pXq}]$ does not seem to provide any useful information, at least in situations when $[pXq] \neq 1$. However, this expected value can be used to express not only $\mathbb{E}[R^{f,\lambda}_{pXq} \mid Run(pXq)]$, but also other properties such as $\mathbb{E}[G^{f,\lambda}_{pX}]$ or $\mathbb{E}[X^{f,\lambda}_{pX}]$ discussed below, and can be effectively approximated by Newton's method. Hence, the variable $R^{f,\lambda}_{pXq}$ and its expected value provide a crucial technical tool for solving the problems of our interest.

The variable G^{f}_{pX} assigns to each non-terminating run its average reward per transition, provided that the corresponding limit exists. For finite-state Markov chains, the average reward per transition exists for almost all runs, and hence the corresponding expected value (also called *the gain*[2]) always exists. In the case of pPDA, it can happen that $\mathcal{P}(G^{f}_{pX}{=}\bot) > 0$ even if $[pX{\uparrow}] = 1$, and hence the gain $\mathbb{E}[G^{f}_{pX}]$ does not necessarily exist. A concrete example is given in Section 4. In [10], it has been shown that if all $\mathbb{E}[R^{g}_{tYs} \mid Run(tYs)]$ are finite (where $g(rZ) = 1$ for all $rZ \in Q \times \Gamma$), then the gain is guaranteed to exist and can be effectively expressed in first order theory of the reals. This result relies on a construction of an auxiliary finite-state Markov chain with possibly irrational transition probabilities, and this method does not allow for efficient approximation of the gain.

In Section 4, we examine the properties of the *discounted gain* $\mathbb{E}[G^{f,\lambda}_{pX}]$ which are remarkably different from the aforementioned properties of the gain (these results constitute the first highlight of our paper). First, we *always* have that $\mathcal{P}(G^{f,\lambda}_{pX}{=}\bot \mid Run(pX{\uparrow})) = 0$ whenever $[pX{\uparrow}] > 0$, and hence the discounted gain is

[2] The gain is one of the fundamental concepts in performance analysis.

guaranteed to exist whenever $[pX\uparrow] = 1$. Further, we show that the discounted gain can be efficiently approximated by Newton's method. One intuitively expects that the discounted gain is close to the value of the gain as the discount approaches 1, and we show that this is indeed the case when the gain exists (the proof is not trivial). Thus, we obtain alternative proofs for some of the results about the gain that have been presented in [10], but unfortunately we do not yield an efficient approximation scheme for the (non-discounted) gain, because we were not able to analyze the corresponding convergence rate. More details are given in Section 4.

The variable $X_{pX}^{f,\lambda}$ assigns to each non-terminating run the total discounted reward accumulated along the whole run. Note that the corresponding infinite sum always exists and it is finite. If $[pX\uparrow] = 1$, then the expected value $\mathbb{E}[X_{pX}^{f,\lambda}]$ exactly corresponds to the expected discounted payoff, which is a fundamental and deeply studied concept in stochastic programming (see, e.g., [18, 15]). In Section 4, we show that the family of all $\mathbb{E}[X_{pX}^{f,\lambda}]$ forms the least solution of an effectively constructible system of monotone polynomial equations. Hence, these expected values can also be effectively approximated by Newton's method by applying the results of [17, 8]. We also show how to express $\mathbb{E}[X_{pX}^{f,\lambda} \mid Run(pX\uparrow)]$, which is more relevant in situations when $0 < [pX\uparrow] < 1$.

4 Computing the Discounted Properties of Probabilistic PDA

In this section we show that the (conditional) expected values of the discounted random variables introduced in Section 3 are expressible as the least solutions of efficiently constructible systems of recursive equations. This allows to encode these values in first order theory of the reals, and design efficient approximation schemes for some of them.

Recall that first order theory of the reals $(\mathbb{R}, +, *, \leq)$ is decidable [19], and the existential fragment is even solvable in polynomial space [5]. The following definition explains what we mean by encoding a certain value in $(\mathbb{R}, +, *, \leq)$.

Definition 3. *We say that some $c \in \mathbb{R}$ is* encoded *by a formula $\Phi(x)$ of $(\mathbb{R}, +, *, \leq)$ iff the formula $\forall x.(\Phi(x) \Leftrightarrow x{=}c)$ holds.*

Note that if a given $c \in \mathbb{R}$ is encoded by $\Phi(x)$, then the problems whether $c = \varrho$ and $c \leq \varrho$ for a given rational constant ϱ are decidable (we simply check the validity of the formulae $\Phi(x/\varrho)$ and $\exists x.(\Phi(x) \wedge x \leq \varrho)$, respectively).

For the rest of this section, we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob_\Delta)$, a nonnegative *reward function* $f : Q \times \Gamma \to \mathbb{Q}$, and a *discount function* $\lambda : Q \times \Gamma \to [0, 1)$. As a warm-up, let us first consider the family of expected values $[pXq, \lambda]$. For each of them, we fix the corresponding first order variable $\langle\!\langle \boldsymbol{pXq,\lambda} \rangle\!\rangle$, and construct the following equation (for the sake of readability, each variable occurrence is typeset in boldface):

$$\begin{aligned}\langle\!\langle \boldsymbol{pXq,\lambda} \rangle\!\rangle \;=\; & \sum_{pX \xrightarrow{x} q\varepsilon} x \cdot \lambda(pX) \;+\; \sum_{pX \xrightarrow{x} rY} x \cdot \lambda(pX) \cdot \langle\!\langle \boldsymbol{rYq,\lambda} \rangle\!\rangle \\ & + \sum_{pX \xrightarrow{x} rYZ,\; s \in Q} x \cdot \lambda(pX) \cdot \langle\!\langle \boldsymbol{rYs,\lambda} \rangle\!\rangle \cdot \langle\!\langle \boldsymbol{sZq,\lambda} \rangle\!\rangle\end{aligned} \tag{1}$$

Thus, we produce a finite system of recursive equations (S1). This system is rather similar to the system for termination probabilities $[pXq]$ constructed in [9, 14]. The only modification is the introduction of the discount factor $\lambda(pX)$. A proof of the following theorem is also just a technical extension of the proof given in [9, 14] (see [2]).

Theorem 4. *The tuple of all* $[pXq, \lambda]$ *is the least nonnegative solution of the system (S1).*

Now consider the expected value $\mathbb{E}[R^{f,\lambda}_{pXq}]$. For all $p, q \in Q$ and $X \in \Gamma$ we fix a first order variable $\langle\!\langle \boldsymbol{pXq} \rangle\!\rangle$ and construct the following equation:

$$
\begin{aligned}
\langle\!\langle \boldsymbol{pXq} \rangle\!\rangle = & \sum_{pX \xrightarrow{x} q\varepsilon} x \cdot \lambda(pX) \cdot f(pX) + \sum_{pX \xrightarrow{x} rY} x \cdot \lambda(pX) \cdot ([rYq] \cdot f(pX) + \langle\!\langle \boldsymbol{rYq} \rangle\!\rangle) \\
& + \sum_{pX \xrightarrow{x} rYZ,\ s \in Q} x \cdot \lambda(pX) \cdot ([rYs] \cdot [sZq] \cdot f(pX) + [sZq] \cdot \langle\!\langle \boldsymbol{rYs} \rangle\!\rangle + [rYs, \lambda] \cdot \langle\!\langle \boldsymbol{sZq} \rangle\!\rangle)
\end{aligned}
\tag{2}
$$

Thus, we obtain the system (S2). Note that termination probabilities and discounted termination probabilities are treated as “known constants” in the equations of (S2).

As opposed to (S1), the equations of system (S2) do not have a good intuitive meaning. At first glance, it is not clear why these equations should hold, and a formal proof of this fact requires advanced arguments. The proof of the following theorem is already non-trivial (see [2]).

Theorem 5. *The tuple of all* $\mathbb{E}[R^{f,\lambda}_{pXq}]$ *is the least nonnegative solution of the system (S2).*

The conditional expected values $\mathbb{E}[R^{f,\lambda}_{pXq} \mid \mathit{Run}(pXq)]$ make sense only if $[pXq] > 0$, which can be checked in time polynomial in the size of Δ because $[pXq] > 0$ iff $pX \rightarrow^* q\varepsilon$, and the reachability problem for PDA is in **P** (see, e.g., [7]). The next theorem says how to express $\mathbb{E}[R^{f,\lambda}_{pXq} \mid \mathit{Run}(pXq)]$ using $\mathbb{E}[R^{f,\lambda}_{pXq}]$.

Theorem 6. *For all* $p, q \in Q$ *and* $X \in \Gamma$ *such that* $[pXq] > 0$ *we have that*

$$
\mathbb{E}[R^{f,\lambda}_{pXq} \mid \mathit{Run}(pXq)] = \frac{\mathbb{E}[R^{f,\lambda}_{pXq}]}{[pXq]} \tag{3}
$$

Now we turn our attention to the discounted long-run properties of probabilistic PDA introduced in Section 3. These results represent the core of our paper.

As we already mentioned, the system (S1) can also be used to express the family of termination probabilities $[pXq]$. This is achieved simply by replacing each $\lambda(pX)$ with 1 (thus, we obtain the equational systems originally presented in [9, 14]). Hence, we can also express the probability of *non-termination*:

$$
[pX\uparrow] = 1 - \sum_{q \in Q} [pXq] \tag{4}
$$

Note that this equation *is not monotone* (by increasing $[pXq]$ we decrease $[pX\uparrow]$), which leads to some complications discussed in Section 4.1.

Now we have all the tools that are needed to construct an equational system for the family of all $\mathbb{E}[X_{pX}^{f,\lambda}]$. For all $p \in Q$ and $X \in \Gamma$, we fix a first order variable $\langle\!\langle \boldsymbol{pX} \rangle\!\rangle$ and construct the following equation, which gives us the system (S5):

$$\begin{aligned}
\langle\!\langle \boldsymbol{pX} \rangle\!\rangle = & \sum_{pX \xrightarrow{x} rY} x \cdot \lambda(pX) \cdot ([rY\uparrow] \cdot f(pX) + \langle\!\langle \boldsymbol{rY} \rangle\!\rangle) \\
& + \sum_{pX \xrightarrow{x} rYZ} x \cdot \lambda(pX) \cdot ([rY\uparrow] \cdot f(pX) + \langle\!\langle \boldsymbol{rY} \rangle\!\rangle) \\
& + \sum_{pX \xrightarrow{x} rYZ,\ s \in Q} x \cdot \lambda(pX) \cdot \Big([rYs] \cdot [sZ\uparrow] \cdot f(pX) + [sZ\uparrow] \cdot \mathbb{E}[R_{rYs}^{f,\lambda}] + [rYs, \lambda] \cdot \langle\!\langle \boldsymbol{sZ} \rangle\!\rangle\Big)
\end{aligned} \tag{5}$$

The equations of (S5) are even less readable than the ones of (S2). However, note that the equations are monotone and efficiently constructible. The proof of the following theorem is based on advanced arguments.

Theorem 7. *The tuple of all* $\mathbb{E}[X_{pX}^{f,\lambda}]$ *is the least nonnegative solution of the system (S5).*

In situations when $[pX\uparrow] < 1$, $\mathbb{E}[X_{pX}^{f,\lambda} \mid Run(pX\uparrow)]$ may be more relevant than $\mathbb{E}[X_{pX}^{f,\lambda}]$. The next theorem says how to express this expected value.

Theorem 8. *For all* $p, q \in Q$ *and* $X \in \Gamma$ *such that* $[pX\uparrow] > 0$ *we have that*

$$\mathbb{E}[X_{pX}^{f,\lambda} \mid Run(pX\uparrow)] = \frac{\mathbb{E}[X_{pX}^{f,\lambda}]}{[pX\uparrow]} \tag{6}$$

Concerning the equations of (S6), there is one notable difference from all of the previous equational systems. The only known method of solving the problem whether $[pX\uparrow] > 0$ employs the decision procedure for the existential fragment of $(\mathbb{R}, +, *, \leq)$, and hence the best known upper bound for this problem is **PSPACE**. This means that the equations of (S6) cannot be constructed efficiently, because there is no efficient way of determining all p, q and X such that $[pX\uparrow] > 0$.

The last discounted property of probabilistic PDA which is to be investigated is the discounted gain $\mathbb{E}[G_{pX}^{f,\lambda}]$. Here, we only managed to solve the special case when λ is a constant function.

Theorem 9. *Let* λ *be a constant discount function such that* $\lambda(rY) = \kappa$ *for all* $rY \in Q \times \Gamma$*, and let* $p \in Q$*,* $X \in \Gamma$ *such that* $[pX\uparrow] = 1$*. Then*

$$\mathbb{E}[G_{pX}^{f,\lambda}] = (1 - \kappa) \cdot \mathbb{E}[X_{pX}^{f,\lambda}] \tag{7}$$

A proof of Theorem 9 is simple. Let $w \in Run(pX\uparrow)$. Since both $\lim_{n\to\infty} \sum_{i=0}^{n} \lambda(w^i)$ and $\lim_{n\to\infty} \sum_{i=0}^{n} \lambda(w^i) f(w(i))$ exist and the latter is equal to $(1-\kappa)^{-1}$ the claim follows from the linearity of the expected value.

Note that the equations of (S7) can be constructed efficiently (in polynomial time), because the question whether $[pX\uparrow] = 1$ is equivalent to checking whether $[pXq] = 0$ for all $q \in Q$, which is equivalent to checking whether $pX \not\to^* q\varepsilon$ for all $q \in Q$. Hence, it suffices to apply a polynomial-time decision procedure for PDA reachability such as [7].

Since all equational systems constructed in this section contain just summation, multiplication, and division, one can easily encode all of the considered discounted properties in $(\mathbb{R}, +, *, \leq)$ in the sense of Definition 3. For a given discounted property c, the corresponding formula $\Phi(x)$ looks as follows:

$$\exists \boldsymbol{v} \left(solution(\boldsymbol{v}) \ \wedge \ (\forall \boldsymbol{u}\ (solution(\boldsymbol{u}) \Rightarrow \boldsymbol{v} \leq \boldsymbol{u}) \ \wedge \ x = \boldsymbol{v}_i \right)$$

Here $\boldsymbol{v}$ and $\boldsymbol{u}$ are tuples of fresh first order variables that correspond (in one-to-one fashion) to the variables employed in the equational systems (S1), (S2), (S3), (S4), (S5), (S6), and (S7). The subformulae $solution(\boldsymbol{v})$ and $solution(\boldsymbol{u})$ say that the variables of $\boldsymbol{v}$ and $\boldsymbol{u}$ form a solution of the equational systems (S1), (S2), (S3), (S4), (S5), (S6), and (S7). Note that the subformulae $solution(\boldsymbol{v})$ and $solution(\boldsymbol{u})$ are indeed expressible in $(\mathbb{R}, +, *, \leq)$, because the right-hand sides of all equational systems contain just summation, multiplication, division, and employ only constants that themselves correspond to some variables in $\boldsymbol{v}$ or $\boldsymbol{u}$. The $\boldsymbol{v}_i$ is the variable of $\boldsymbol{v}$ which corresponds to the considered property c, and the x is the only free variable of the formula $\Phi(x)$. Note that $\Phi(x)$ can be constructed in space which is polynomial in the size of a given pPDA Δ (the main cost is the construction of the system (S6)), but the length of $\Phi(x)$ is only polynomial in the size of Δ, λ, and f. Since the alternation depth of quantifiers in $\Phi(x)$ is fixed, we can apply the result of [16] which says that every fragment of $(\mathbb{R}, +, *, \leq)$ where the alternation depth of quantifiers is bounded by a fixed constant is decidable in exponential time. Thus, we obtain the following theorem:

Theorem 10. *Let c be one of the discounted properties of pPDA considered in this section, i.e., c is either $[pXq, \lambda]$, $\mathbb{E}[R^{f,\lambda}_{pXq}]$, $\mathbb{E}[R^{f,\lambda}_{pXq} \mid Run(pXq)]$, $\mathbb{E}[X^{f,\lambda}_{pX}]$, $\mathbb{E}[X^{f,\lambda}_{pX} \mid Run(pX\uparrow)]$, or $\mathbb{E}[G^{f,\lambda}_{pX}]$ (in the last case we further require that λ is constant). The problems whether $c = \varrho$ and $c \leq \varrho$ for a given rational constant ϱ are in **EXPTIME**.*

The previous theorem extends the results achieved in [9, 10, 14] to discounted properties of pPDA. However, in the case of discounted long-run properties $\mathbb{E}[X^{f,\lambda}_{pX}]$, $\mathbb{E}[X^{f,\lambda}_{pX} \mid Run(pX\uparrow)]$, and $\mathbb{E}[G^{f,\lambda}_{pX}]$, the presented proof is completely different from the non-discounted case. Moreover, the constructed equations take the form which allows to design efficient approximation scheme for these values, and this is what we show in the next subsection.

4.1 The Application of Newton's Method

In this section we show how to apply the recent results [17, 8] about fast convergence of Newton's method for systems of monotone polynomial equations to the discounted properties introduced in Section 3. We start by recalling some notions and results presented in [17, 8].

Monotone systems of polynomial equations (MSPEs) are systems of fixed point equations of the form $x_1 = f_1(x_1, \cdots, x_n), \cdots, x_n = f_n(x_1, \cdots, x_n)$, where each f_i is a polynomial with nonnegative real coefficients. Written in vector form, the system is given as $\boldsymbol{x} = f(\boldsymbol{x})$, and solutions of the system are exactly the fixed points of f. To f we associate the directed graph H_f where the nodes are the variables $x_1, \ldots, x_n$ and (x_i, x_j) is an edge iff x_j appears in f_i. A subset of equations is a *strongly connected component (SCC)* if its associated subgraph is a SCC of H_f.

Observe that each of the systems (S1), (S2), and (S5) forms a MSPE. Also observe that the system (S1) uses only simple coefficients obtained by multiplying transition probabilities of $\varDelta$ with the return values of λ, while the coefficients in (S2) and (S5) are more complicated and also employ constants such as $[rYq]$, $[rYs, \lambda]$, $\mathbb{E}[R^{f,\lambda}_{pXq}]$, or $[rY\!\uparrow]$.

The problem of finding the least nonnegative solution of a given MSPE $\boldsymbol{x} = f(\boldsymbol{x})$ can be obviously reduced to the problem of finding the least nonnegative solution for $F(\boldsymbol{x}) = \boldsymbol{0}$, where $F(\boldsymbol{x}) = f(\boldsymbol{x}) - \boldsymbol{x}$. The Newton's method for approximating the least solution of $F(\boldsymbol{x}) = \boldsymbol{0}$ is based on computing a sequence $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \cdots$, where $\boldsymbol{x}^{(0)} = \boldsymbol{0}$ and

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^k - (F'(\boldsymbol{x}^k))^{-1} F(\boldsymbol{x}^k)$$

where $F'(\boldsymbol{x})$ is the Jacobian matrix of partial derivatives. If the graph H_f is strongly connected, then the method is guaranteed to converge *linearly* [17, 8]. This means that there is a threshold k_f such that after the initial k_f iterations of the Newton's method, each additional bit of precision requires only 1 iteration. In [8], an upper bound for k_f is presented.

For general MSPE where H_f is not necessarily strongly connected, a more structured method called *Decomposed Newton's method (DNM)* can be used. Here, the component graph of H_f is computed and the SCCs are divided according to their depth. DNM proceeds bottom-up by computing $k \cdot 2^t$ iterations of Newton's method for each of the SCCs of depth t, where t goes from the height of the component graph to 0. After computing the approximations for the SCCs of depth i, the computed values are fixed, the corresponding equations are removed, and the SCCs of depth $i{-}1$ are processed in the same way, using the previously fixed values as constants. This goes on until all SCCs are processed. It was demonstrated in [14] that DNM is guaranteed to converge to the least solution as k increases. In [17], it was shown that DNM is even guaranteed to converge linearly. Note, however, that the number of iterations of the original Newton's method in one iteration of DNM is exponential in the depth of the component graph of H_f.

Now we show how to apply these results to the systems (S1), (S2), and (S5). First, we also add a system (S0) whose least solution is the tuple of all termination probabilities $[pXq]$ (the system (S0) is very similar to the system (S1), the only difference is that each $\lambda(pX)$ is replaced with 1). The systems (S0), (S1), (S2), and (S5) themselves are not necessarily strongly connected, and we use H to denote the height of the component graph of (S0). Note that the height of the component graph of (S1), (S2), and (S5) is at most H. Now, we unify the systems (S0), (S1), (S2), and (S5) into one equation system S. What we obtain is a MSPE with three types of coefficients: transition probabilities of $\varDelta$, the return values of λ, and non-termination probabilities of the form

$[rY{\uparrow}]$ (the system (S4) cannot be added to S because the resulting system would not be monotone). Observe that

- (S0) and (S1) only use the transition probabilities of Δ and the return values of λ as coefficients;
- (S2) also uses the values defined by (S0) and (S1) as coefficients;
- (S5) uses the values defined by (S0), (S1) and (S2) as coefficients, and it also uses coefficients of the form $[rY{\uparrow}]$.

This means that the height of the component graph of S is at most $3H$. Now we can apply the DNM in the way described above, with the following technical modification: after computing the termination probabilities $[rYq]$ (in the system (S0)), we compute an *upper approximation* for each $[rY{\uparrow}]$ according to equation (4), and then subtract an upper bound for the overall error of this upper approximation bound with the same overall error (here we use the technical results of [8]). In this way, we produce a *lower approximation* for each $[rY{\uparrow}]$ which is used as a constant when processing the other SCCs. Now we can apply the aforementioned results about DNM.

Note that once the values of $[pXq]$, $[pXq, \lambda]$, $\mathbb{E}[R^{f,\lambda}_{pXq}]$, and $\mathbb{E}[X^{f,\lambda}_{pX}]$ are computed with a sufficient precision, we can also compute the values of $\mathbb{E}[R^{f,\lambda}_{pXq} \mid Run(pXq)]$ and $\mathbb{E}[G^{f,\lambda}_{pX}]$ by equations given in Theorem 6 and Theorem 9, respectively. Thus, we obtain the following:

Theorem 11. *The values of* $[pXq]$, $[pXq, \lambda]$, $\mathbb{E}[R^{f,\lambda}_{pXq}]$, $\mathbb{E}[X^{f,\lambda}_{pX}]$, $\mathbb{E}[R^{f,\lambda}_{pXq} \mid Run(pXq)]$, *and* $\mathbb{E}[G^{f,\lambda}_{pX}]$ *can be approximated using DNM, which is guaranteed to converge linearly. The number of iterations of the Newton's method which is needed to compute one iteration of DNM is exponential in* H.

In practice, the parameter H stays usually small. A typical application area of PDA are recursive programs, where stack symbols correspond to the individual procedures, procedure calls are modeled by pushing new symbols onto the stack, and terminating a procedure corresponds to popping a symbol from the stack. Typically, there are "groups" of procedures that call each other, and these groups then correspond to strongly connected components in the component graph. Long chains of procedures $P_1, \cdots, P_n$, where each P_i can only call P_j for $j > i$, are relatively rare, and this is the only situation when the parameter H becomes large.

4.2 The Relationship Between Discounted and Non-discounted Properties

In this section we examine the relationship between the discounted properties introduced in Section 3 and their non-discounted variants. Intuitively, one expects that a discounted property should be close to its non-discounted variant as the discount approaches 1. To formulate this precisely, for every $\kappa \in (0, 1)$ we use λ_κ to denote the constant discount function that always returns κ.

The following theorem is immediate. It suffices to observe that the equational systems for the non-discounted properties are obtained from the corresponding equational systems for discounted properties by substituting all $\lambda(pX)$ with 1.

Theorem 12. *We have that*

– $[pXq] = \lim_{\kappa\uparrow 1}[pXq, \lambda_\kappa]$
– $\mathbb{E}[R^{f}_{pXq}] = \lim_{\kappa\uparrow 1} \mathbb{E}[R^{f,\lambda_\kappa}_{pXq}]$
– $\mathbb{E}[R^{f}_{pXq} \mid Run(pXq)] = \lim_{\kappa\uparrow 1} \mathbb{E}[R^{f,\lambda_\kappa}_{pXq} \mid Run(pXq)]$

The situation with discounted gain $\mathbb{E}[G^{f,\lambda}_{pX}]$ is more complicated. First, let us realize that $\mathbb{E}[G^{f}_{pX}]$ does not necessarily exist even if $[pX\uparrow] = 1$. To see this, consider the pPDA with the following rules:

$$pX \xrightarrow{\frac{1}{2}} pYX,\quad pY \xrightarrow{\frac{1}{2}} pYY,\quad pZ \xrightarrow{\frac{1}{2}} pZZ,\quad pX \xrightarrow{\frac{1}{2}} pZX,\quad pY \xrightarrow{\frac{1}{2}} p\varepsilon,\quad pZ \xrightarrow{\frac{1}{2}} p\varepsilon$$

The reward function f is defined by $f(pX) = f(pY) = 0$ and $f(pZ) = 1$. Intuitively, pX models a one-dimensional symmetric random walk with distinguished zero (X), positive numbers (Z) and negative numbers (Y). Observe that $[pX\uparrow] = 1$. However, one can show that $\mathcal{P}(G^{f}_{pX}{=}\bot) = 1$, which means that $\mathbb{E}[G^{f}_{pX}]$ does not exist. The example is elaborated to a greater detail in [2].

The following theorem says that if the gain *does* exist, then it is equal to the limit of discounted gains as κ approaches 1. The opposite direction, i.e., the question whether the existence of the limit of discounted gains implies the existence of the (non-discounted) gain is left open. The proof of the following theorem is not trivial and relies on several subtle observations (see [2]).

Theorem 13. *If* $\mathbb{E}[G^{f}_{pX}]$ *exists, then* $\mathbb{E}[G^{f}_{pX}] = \lim_{\kappa\uparrow 1} \mathbb{E}[G^{f,\lambda_\kappa}_{pX}]$.

Since $\lim_{\lambda\uparrow 1} \mathbb{E}[G^{f,\lambda}_{pX}]$ can be effectively encoded in first order theory of the reals, we obtain an alternative proof of the result established in [10] saying that the gain is effectively expressible in $(\mathbb{R}, +, *, \leq)$. Actually, we obtain a somewhat *stronger* result, because the formula constructed for $\lim_{\lambda\uparrow 1} \mathbb{E}[G^{f,\lambda}_{pX}]$ encodes the gain whenever it exists, while the (very different) formula constructed in [10] encodes the gain only in situation when a certain sufficient condition (mentioned in Section 3) holds. Unfortunately, Theorem 13 does not yet help us to approximate the gain, because the proof does not give any clue how large κ must be chosen in order to approximate the limit upto a given precision.

5 Conclusions

We have shown that a family of discounted properties of probabilistic PDA can be efficiently approximated by decomposed Newton's method. In some cases, it turned out that the discounted properties are "more computational" than their non-discounted counterparts. An interesting open question is whether the scope of our study can be extended to other discounted properties defined, e.g., in the spirit of [3]. More open questions concern decidability of discounted properties of more complex models like Markov decision process and recursive stochastic games in general.

References

1. *Proceedings of STACS 2005*, volume 3404 of *Lecture Notes in Computer Science*. Springer, 2005.
2. T. Brázdil, V. Brožek, J. Holeček, and A. Kučera. Discounted properties of probabilistic pushdown automata. Technical report FIMU-RS-2008-06, Faculty of Informatics, Masaryk University, 2008.
3. T. Brázdil, J. Esparza, and A. Kučera. Analysis and prediction of the long-run behavior of probabilistic sequential programs with recursion. In *Proceedings of FOCS 2005*, pages 521–530. IEEE Computer Society Press, 2005.
4. T. Brázdil, A. Kučera, and O. Stražovský. On the decidability of temporal properties of probabilistic pushdown automata. In *Proceedings of STACS 2005* [1], pages 145–157.
5. J. Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of STOC'88*, pages 460–467. ACM Press, 1988.
6. L. de Alfaro, T. Henzinger, and R. Majumdar. Discounting the future in systems theory. In *Proceedings of ICALP 2003*, volume 2719 of *Lecture Notes in Computer Science*, pages 1022–1037. Springer, 2003.
7. J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon. Efficient algorithms for model checking pushdown systems. In *Proceedings of CAV 2000*, volume 1855 of *Lecture Notes in Computer Science*, pages 232–247. Springer, 2000.
8. J. Esparza, S. Kiefer, and M. Luttenberger. Convergence thresholds of Newton's method for monotone polynomial equations. In *Proceedings of STACS 2008*, 2008.
9. J. Esparza, A. Kučera, and R. Mayr. Model-checking probabilistic pushdown automata. In *Proceedings of LICS 2004*, pages 12–21. IEEE Computer Society Press, 2004.
10. J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: Expectations and variances. In *Proceedings of LICS 2005*, pages 117–126. IEEE Computer Society Press, 2005.
11. J. Esparza, A. Kučera, and S. Schwoon. Model-checking LTL with regular valuations for pushdown systems. *Information and Computation*, 186(2):355–376, 2003.
12. K. Etessami and M. Yannakakis. Algorithmic verification of recursive probabilistic systems. In *Proceedings of TACAS 2005*, volume 3440 of *Lecture Notes in Computer Science*, pages 253–270. Springer, 2005.
13. K. Etessami and M. Yannakakis. Checking LTL properties of recursive Markov chains. In *Proceedings of 2nd Int. Conf. on Quantitative Evaluation of Systems (QEST'05)*, pages 155–165. IEEE Computer Society Press, 2005.
14. K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proceedings of STACS 2005* [1], pages 340–352.
15. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1996.
16. D. Grigoriev. Complexity of deciding Tarski algebra. *Journal of Symbolic Computation*, 5(1–2):65–108, 1988.
17. S. Kiefer, M. Luttenberger, and J. Esparza. On the convergence of Newton's method for monotone systems of polynomial equations. In *Proceedings of STOC 2007*, pages 217–226. ACM Press, 2007.
18. M.L. Puterman. *Markov Decision Processes*. Wiley, 1994.
19. A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. Univ. of California Press, Berkeley, 1951.