Positive Almost-Sure Termination of Polynomial Random Walks

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Logic and Computation

eingereicht von

Lorenz Winkler, BSc
Matrikelnummer 12020650

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof.in Dr.in techn. Laura Kovács, MSc

Wien, 19. Juli 2025

Lorenz Winkler

Laura Kovács

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Positive Almost-Sure Termination of Polynomial Random Walks

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Logic and Computation

by

Lorenz Winkler, BSc
Registration Number 12020650

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof.in Dr.in techn. Laura Kovács, MSc

Vienna, July 19, 2025

Lorenz Winkler

Laura Kovács

Declaration of Authorship

Lorenz Winkler, BSc

I hereby declare that I have written this Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.
I further declare that I have used generative AI tools only as an aid, and that my own intellectual and creative efforts predominate in this work. In the appendix "Overview of Generative AI Tools Used" I have listed all generative AI tools that were used in the creation of this work, and indicated where in the work they were used. If whole passages of text were used without substantial changes, I have indicated the input (prompts) I formulated and the IT application used with its product name and version number/date.

Vienna, July 19, 2025

Lorenz Winkler

Acknowledgements

I want to thank my advisor Laura Kovács for the numerous discussions which led to the research presented in this thesis. Her very detailed and thorough feedback was crucial for the quality of this work. Furthermore, I would like to thank Marcel Moosbrugger for introducing me to the topic of probabilistic program analysis and guidance on the Polar analyzer.

This research was partially supported by the European Research Council Consolidator Grant ARTIST 101002685 and the Vienna Science and Technology Fund WWTF 10.47379/ICT19018 grant ProbInG.

Additionally, I want to thank my family for their financial and emotional support throughout my education, especially while writing this thesis.

Kurzfassung

Die Laufzeit eines probabilistischen Programmes ist eine Zufallsvariable. Termination solcher Programme umfasst deshalb den qualitativen Terminationsbegriff “almost-sure termination” (AST) sowie den quantitativen Begriff “positive almost-sure termination” (PAST), welcher eine Aussage über den Erwartungswert der Laufzeit trifft. Ein Programm welches im Sinne von PAST terminiert, terminiert auch im Sinne von AST, jedoch terminiert nicht jedes Programm, welches im Sinne von AST terminiert auch im Sinne von PAST. Der symmetrische Random Walk ist ein Beispiel für ein Programm, welches AST, aber nicht PAST erfüllt.

In dieser Arbeit zeigen wir, dass die Klasse der “Polynomiellen Random Walks” unter bestimmten Umständen PAST erfüllt. In jedem Schleifendurchlauf wird mit einer konstanten Wahrscheinlichkeit p eines von zwei Polynomen in der Anzahl der bisherigen Schleifendurchläufen gewählt, welches die Größe und Richtung des nächsten Schrittes bestimmt. Wir zeigen, dass ein Programm PAST erfüllt, wenn der Grad der Polynome sowohl höher als der Grad des Erwartungswertes der Schritte ist, als auch einen gewissen Schwellenwert $d_{\min}(p)$ übersteigt. Unser Ansatz verwendet keine Beweisregeln und Arithmetische Ausdrücke wie Martingale oder Invarianten. Stattdessen beschränken wir den oberen Rand der Zufallsvariable, welche die Schritte akkumuliert, induktiv, und zeigen mit Hilfe dieser Beschränkung, dass PAST erfüllt sein muss. Weiters implementieren wir die Annäherung dieser Schranke mittels eines genetischen Algorithmus und linearer optimierung.

Abstract

The number of steps until termination of a probabilistic program is a random variable. Probabilistic program termination therefore requires qualitative analysis via almost-sure termination (AST), while also providing quantitative answers via positive almost-sure termination (PAST) on the expected number of steps until termination. While every program which is PAST is AST, the converse is not true. The symmetric random walk with constant step size is a prominent example of a program that is AST but not PAST.

In this thesis we show that a more general class of polynomial random walks is PAST. Our random walks implement a step size that is polynomially increasing in the number of loop iterations and have a constant probability p of choosing either branch. We show that such programs are PAST when the degree of the polynomial is higher than both the degree of the drift and a threshold $d_{\min}(p)$. Our approach does not use proof rules, nor auxiliary arithmetic expressions, such as martingales or invariants. Rather, we establish an inductive bound for the cumulative distribution function of the loop guard, based on which PAST is proven. We implemented the approximation of this threshold, by combining genetic programming, algebraic reasoning and linear programming.

Contents

CHAPTER 1

Introduction

Probabilistic programs extend programs written in classical programming languages by statements that draw samples from stochastic distributions, such as Normal and Bernoulli. The output as well as the number of steps until termination of a probabilistic program are random variables [1], which makes the analysis even harder than in the nonprobabilistic setting [2]. *In this thesis we focus on probabilistic termination and introduce a class of loops for which we provide a new sufficient condition for positive almost-sure termination.*

Probabilistic loop termination [3, 4] requires qualitative arguments via almost-sure termination (AST), and quantitative answers via positive almost-sure termination (PAST) on the expected number of steps until termination. While every program which is PAST is AST, the converse is not true. The symmetric random walk with constant step size is a prominent example of a program that is AST but not PAST. Existing works for proving PAST and/or AST rely on proof rules that need auxiliary arithmetic expressions, such as invariants and martingales, over program variables [5]. In particular, ranking super-martingales (RSMs) or lexicographical RSMs are commonly used ingredients in (P)AST analysis [6]. For probabilistic programs without nondeterminism, RSMs are a sound and complete method for proving termination [7, 8].

However, finding an RSM is challenging, as shown in Figure 1.1. Here, ← denotes variable assignments and $\oplus_p$ captures probabilistic choice: the expression on the left hand side of $\oplus_p$ is chosen with probability p, and the one on the right hand side of $\oplus_p$ is chosen with probability $1-p$. While the program has a finite stopping time, to the best of our knowledge, no RSM of this program has been found until now. As such, existing methods would fail proving PAST for Figure 1.1.

```
$n \leftarrow 0$
$s \leftarrow 0$
while $s \geq 0$ do
    $n \leftarrow n + 1$
    $s \leftarrow s - n + 3 \oplus_{\frac{1}{2}} s + n + 5$
end while
```

Figure 1.1: Program with non-trivial RSM.

Our approach. We overcome challenges of RSM inference in PAST analysis, by identifying a class of loops, called *polynomial random walks* (Chapter 4), for which PAST can be proven without martingale/invariant synthesis. While polynomial random walks are less expressive than arbitrary polynomial loops, their PAST analysis can be automated using genetic programming, algebraic reasoning and linear programming (Chapter 4).

Our approach relies on bounding the tails of random loop variables in order to guarantee that a random variable is close to its mean. Tail bounds [9] are important in providing guarantees on the probability of extreme outcomes. Our work analyzes tail probabilities $\mathbb{P}(X \geq t)$ of the random variable X summing up the steps, to establish that the probability that X exceeds some value t decreases fast with increasing t, and the deviation from 0 is in some sense "controlled". Key to our approach is that variables of polynomial random walks converge to random variables with almost Normal distributions (Lemma 5), whose tail bounds can be approximated (Lemma 6). We therefore transform polynomial random walks into programs with larger expected stopping time, where several steps are accumulated before the loop guard is checked. In other words, we connect polynomial random walk analysis to stochastic processes over random variables with almost Normal distributions whose variance is exponentially growing (Chapter 2). By summing up random variables of such processes, we prove that a sub-Gaussian tail bound is preserved. The cumulative distribution function of this summation can tightly be approximated using an *inductive bound* over random variables.

By proving that an inductive bound always exists, we find that constant-probability polynomial random walks are PAST when the step size grows fast enough (Theorem 2). Our PAST result holds whenever (i) the degree d of the polynomials is larger than the degree of the expected value of the increments, and (ii) d is larger than a threshold $d_{\min}(p)$ parametrized by the probabilistic choice probability p. Our work establishes that Figure 1.1 satisfies this threshold (Table 4.1), implying thus PAST of Figure 1.1 without the need of an RSM.

We implemented our PAST analysis over polynomial random walks in extension of the algebraic program analysis tool `Polar` [10]. To this end, we use linear programming, by relying on `OR-Tools` [11] and the `Gurobi`-solver [12], to derive inductive bounds for fixed parameters of the program transformation over polynomial random walks. We further combined `Polar` with a genetic algorithm, to find the best values for those parameters. Our experimental results in Chapter 4, give practical evidence on the tightness of our inductive bounds on polynomial random walks. Existence of these inductive bounds imply thus PAST of the programs that are analyzed.

Our contributions. We translate the problem of verifying PAST into the problem of tightly approximating tail bounds of random loop variables. We bring the following contributions:

- We introduce the class of polynomial random walks for which we provide sufficient conditions to determine PAST. These conditions do not require user-provided invariants and/or martingales. We determine PAST above a threshold $d_{\min}(p)$, which in turn only depends on the polynomial degree d of the loop updates and the probabilistic choice p in the loop (Chapter 3).

- We show that such a threshold $d_{\min}(p)$ always exists by transforming polynomial random walks into stochastic processes over almost Normal variables (Chapter 2). We prove that such processes admit an inductive bound over their cumulative distribution function, allowing us to tightly approximate our threshold $d_{\min}(p)$.
- We implemented our approach to approximate $d_{\min}(p)$, and hence conclude PAST, in extension of the `Polar` framework (Chapter 4). Our experiments showcase the tightness of our approximation, implying thus PAST.

Impact and dissemination of the thesis. Results of the thesis have been peer-reviewed and accepted for publication and presentation at the International Conference on Quantitative Evaluation of SysTems (QEST) 2025, as follows:

Lorenz Winkler and Laura Kovács. Positive Almost-Sure Termination of Polynomial Random Walks. In Proceedings of the International Conference on Quantitative Evaluation of SysTems (QEST) 2025, LNCS, 2025. To appear.

I have been the main author of the work presented both in this thesis and the above mentioned published paper. Yet, given the collaborative work upon which the thesis is based, the work presented in this thesis is referred to within a plural form (e.g. "we conclude..." instead of "I conclude...").

CHAPTER 2

Almost Normal Variables and Conditioning

This chapter introduces a special class of stochastic processes (Section 2.1) and establishes bound properties for such processes. The stochastic processes we consider are given in Figure 2.1, for which we show that they induce random variables with sub-Gaussian tail bounds (Section 2.2) whose inductive bounds can tightly be approximated (Section 2.3). In Chapter 3 we then prove that polynomial random walks can be transformed into the stochastic process of Figure 2.1, allowing to conclude PAST of polynomial random walks (Theorem 2).

```
$n \leftarrow 0$
$z \sim \mathcal{N}_{c_0}^{\delta_1, C_1}(0, \sigma_0)$
$s \leftarrow s_0 + z$
while $s \geq F_{S_n}^{-1}(\epsilon)$ do
    $n \leftarrow n + 1$
    $z \sim \mathcal{N}_{c_0}^{\delta_1, C_1}(0, \sigma_n)$
    $s \leftarrow s + z$
end while
```

Figure 2.1: Probabilistic programs summing up almost normally distributed variables

In the sequel, we respectively denote by $\mathbb{N}, \mathbb{R}, \mathbb{R}^+$ the set of natural, real, and positive real (including zero) numbers. We reserve $n \in \mathbb{N}$ for the loop iteration (counter). We assume familiarity with probabilistic programs and their semantics, and refer to [13, 9] for details. The probability measure is denoted by $\mathbb{P}$, while we use $\mathbb{E}$ to denote the expected values of random variables. Further, we denote with $(X|E)$ the conditional probability distribution of X given event E.

2.1 Almost Normally Distributed Loop Summations

We consider stochastic processes induced by the probabilistic program of Figure 2.1, which uses the random variable series $\{S_n\}_{n\in\mathbb{N}}$ and $\{Z_n\}_{n\in\mathbb{N}}$ corresponding to the values of s and z in the n-th iteration. We respectively denote by F_{S_n} and F_{Z_n} the cumulative distribution function (cdf) of S_n and Z_n. The notation $z \sim \mathcal{N}_{c_0}^{\delta_1, C_1}(0, \sigma_n)$ indicates that z is drawn from a distribution that is almost equivalent to a Normal distribution $\mathcal{N}(0, \sigma_n)$ with variance σ_n^2, in a sense, that c_0 bounds the absolute deviation of cdf of Z_n from the cdf of the Normal variable. Additionally, δ_1 bounds the shift of the sub-Gaussian tail bound and C_1 is the multiplicative deviation of that bound's variance (see Lemma 1). The initial value of s is also drawn from such an almost Normal distribution with variance σ_0^2. In each loop iteration n, a sample is drawn from an almost Normal distribution with variance σ_n^2 and then added to s. The loop is exited, once s is within the smallest ϵ-fraction of S_n.

Recall that the stochastic processes induced by $\{S_n\}$ and $\{Z_n\}$ represent a Markov chain [13]. Based on the semantics of Figure 2.1, the series S_n sums variables $Z_0, \ldots, Z_n$, as follows:

- $S_0 = Z_0$;
- $S_{n+1} = (S_n \mid S_n \geq F_{S_n}^{-1}(\epsilon)) + Z_{n+1}$, where $F_{S_n}^{-1}$ is the inverse of the cdf of S_n.

Note that for S_n only the paths are considered, for which the program has not yet terminated. To reason about termination of Figure 2.1, we rely on a right tail bound of the random variable S_n, as it gives an upper limit on the probability of a random variable exceeding a certain value [9]. In other words, the right tail bound of S_n quantifies how likely it is to observe values of S_n in the extreme right (or upper) tail of a distribution. While such extreme cases might yield to the non-termination of Figure 2.1, in Section 2.2 we show that a tail bound for S_n can be derived from the tail bound of Z_n, which ensures that this behavior is unlikely. Moreover, by adjusting standard properties of distributions [13], we also bound the behavior of Z_n, as listed below.

Lemma 1 (CDF deviation and bound)**.** *Let $\{Z_n\}_{n \in \mathbb{N}}$ be a sequence of random variables, as defined in Figure 2.1. Recall that Z_n follows an almost Normal distribution $\mathcal{N}_{c_0}^{\delta_1, C_1}(0, \sigma_n)$. The following holds:*

1. *The cdf of Z_n, denoted as F_{Z_n} deviates from a Normal distribution only by at most c_0. That is, $|F_{Z_n}(z) - \Phi(\frac{z}{\sigma_n})| \leq c_0$, where Φ denotes the cumulative distribution function of the Normal distribution.*
2. *The variable Z_n admits a sub-Gaussian tail bound on its right tail, which is offset by $\delta_1 \sigma_n$ and has variance of $\frac{\sigma_n^2}{C_1}$. That is,*

$$\forall a \in \mathbb{R}^+ : \mathbb{P}(Z_n \geq a + \delta_1 \sigma_n) \leq \exp\left(C_1 \frac{-a^2}{2\sigma_n^2}\right).$$

2.2 Tail Bound for S_n

Lemma 1 limits the behavior of variable z in Figure 2.1. We next show that, in addition to z, also variable s does not grow in an uncontrolled way. Similarly to Lemma 1, we reason about the right tail of S_n and prove that it admits a sub-Gaussian tail bound (Lemma 2); as such, the probability of S_n being significantly greater than its expected value is limited.

Lemma 2 (Preservation of sub-Gaussian tail bound)**.** *Consider the random variable series $\{S_n\}_{n\in\mathbb{N}}$ induced by s in Figure 2.1. Assume that the variance σ_n^2 grows such that the inequality $\sigma_{n+1}^2 \geq d(\sigma_1^2 + \cdots + \sigma_n^2)$ holds for some $d \in \mathbb{R}^+ \setminus \{0\}$. Then, S_n admits a sub-Gaussian (but not centered) tail bound:*

$$\forall a \in \mathbb{R}^+ : \mathbb{P}\Bigg(S_n \geq a + (\frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1 + b)\sqrt{\sum_{i=0}^{n}\sigma_i^2}\Bigg) \;\leq\; \exp\left(C_1 \frac{-a^2}{2\sum_{i=0}^{n}\sigma_i}\right),$$

where $b \geq \frac{\sqrt{2\ln\frac{1}{1-\epsilon}}}{\sqrt{C_1}(\sqrt{(1+d)}-1)}$.

Proof. By induction. *Base case.* $S_0 = Z_0$, and the bound of Z_0 is stronger, since δ_1 and b are positive:

$$\mathbb{P}(S_0 \geq a + (\frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1 + b)\sigma_0) \leq \exp\left(C_1 \frac{-(a + (\frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1 + b)\sigma_0)^2}{2\sigma_0^2}\right)$$
$$\leq \exp\left(C_1 \frac{-a^2}{2\sigma_0^2}\right)$$

Induction step. We know that

$$\mathbb{P}(S_n \geq a + (\frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1 + b)\sqrt{\sum_{i=0}^{n}\sigma_i^2)} \leq \exp\left(C_1 \frac{-a^2}{2\sum_{i=0}^{n}\sigma_i^2}\right).$$

When cutting away a fraction ϵ of the left tail, we multiply the tail probability by $\frac{1}{1-\epsilon}$:

$$\mathbb{P}\left(\left(S_n | S_n > F_{S_n}^{-1}(\epsilon)\right) \geq a + \left(\frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1 + b\right)\sqrt{\sum_{i=0}^{n}\sigma_i^2}\right)$$
$$\leq \frac{1}{1-\epsilon}\exp\left(C_1 \frac{-a^2}{2\sum_{i=0}^{n}\sigma_i^2}\right) \leq \exp\left(C_1\left(\frac{-a^2}{2\sum_{i=0}^{n}\sigma_i^2}\right) + \ln\frac{1}{1-\epsilon}\right).$$

Now inserting the desired bound involving $\sum_{i=0}^{n+1}\sigma_i^2$:

$$\mathbb{P}\left((S_n|S_n > F_{S_n}^{-1}(\epsilon)) \geq a + b\sqrt{\sum_{i=0}^{n+1}\sigma_i^2} + \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1\sqrt{\sum_{i=0}^{n}\sigma_i^2}\right)$$
$$\leq \exp\left(C_1\left(\frac{-\left(a+b\left(\sqrt{\sum_{i=0}^{n+1}\sigma_i^2}-\sqrt{\sum_{i=0}^{n}\sigma_i^2}\right)\right)^2}{2\sum_{i=0}^{n}\sigma_i^2}\right)+\ln\frac{1}{1-\epsilon}\right).$$

We want to show, that:

$$\exp\left(C_1\left(\frac{-\left(a+b\left(\sqrt{\sum_{i=0}^{n+1}\sigma_i^2}-\sqrt{\sum_{i=0}^{n}\sigma_i^2}\right)\right)^2}{2\sum_{i=0}^{n}\sigma_i^2}\right)+\ln\frac{1}{1-\epsilon}\right)$$
$$\leq \exp\left(C_1\left(\frac{-a^2}{2\sum_{i=0}^{n}\sigma_i^2}\right)\right)$$

holds, which is the case, when

$$-\left(a+b\left(\sqrt{\sum\nolimits_{i=0}^{n+1}\sigma_i^2}-\sqrt{\sum\nolimits_{i=0}^{n}\sigma_i^2}\right)\right)^2+\frac{2\left(\sum_{i=0}^{n}\sigma_i^2\right)}{C_1}\ln\frac{1}{1-\epsilon}\leq -a^2$$

By inserting the inequality $\sigma_{n+1}^2 \geq d\sum_{i=0}^{n}\sigma_i^2$, hence $\sum_{i=0}^{n+1}\sigma_i^2 \geq (d+1)\sum_{i=0}^{n}\sigma_i^2$:

$$-\left(a+b\left(\sqrt{(1+d)\sum\nolimits_{i=0}^{n}\sigma_i^2}-\sqrt{\sum\nolimits_{i=0}^{n}\sigma_i^2}\right)\right)^2+\frac{2\left(\sum_{i=0}^{n}\sigma_i^2\right)}{C_1}\ln\frac{1}{1-\epsilon}$$
$$=-\left(a+b(\sqrt{(1+d)}-1)\sqrt{\sum\nolimits_{i=0}^{n}\sigma_i^2}\right)^2+\frac{2\left(\sum_{i=0}^{n}\sigma_i^2\right)}{C_1}\ln\frac{1}{1-\epsilon}\leq -a^2$$

which again is true, when

$$-b^2\left(\sqrt{(1+d)}-1\right)^2\sum\nolimits_{i=0}^{n}\sigma_i^2+\frac{2\left(\sum_{i=0}^{n}\sigma_i^2\right)}{C_1}\ln\frac{1}{1-\epsilon}\leq 0$$

and equivalently when the inequality for b holds

$$b^2\left(\sqrt{(1+d)}-1\right)^2\geq\frac{2}{C_1}\ln\frac{1}{1-\epsilon}\Leftrightarrow b\geq\frac{\sqrt{2\ln\frac{1}{1-\epsilon}}}{\sqrt{C_1}(\sqrt{(1+d)}-1)}.$$

As this is true by the precondition of the lemma, we can conclude, that:

$$\mathbb{P}\left((S_n|S_n > F_{S_n}^{-1}(\epsilon)) \geq a + b\sqrt{\sum_{i=0}^{n+1}\sigma_i^2} + \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1\sqrt{\sum_{i=0}^{n}\sigma_i^2}\right)$$
$$\leq \exp\left(C_1\left(\frac{-a^2}{2\sum_{i=0}^{n}\sigma_i^2}\right)\right).$$

When two random variables are σ_1 and σ_2 sub-gaussian, then their sum is $\sqrt{\sigma_1^2+\sigma_2^2}$ sub-gaussian [14, Lemma 5.4c].

For the random variable $\left((S_n|S_n > F_{S_n}^{-1}(\epsilon)) - (b + \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1)\sqrt{\sum_{i=0}^{n+1}\sigma_i^2}\right)$ we have just established a bound, while for $(Z_{n+1} - \delta_1\sigma_{n+1})$ we have a bound from the precondition of the theorem.

Observe, that we can multiply the factor C_1 into the variance-proxy, resulting in $\sqrt{\frac{1}{C_1}\sum_{i=0}^{n}\sigma_i^2}$ and $\sqrt{\frac{1}{C_1}}\sigma_{n+1}$ sub-gaussian random variables.

Through applying the just mentioned Lemma, we get the following bound:

$$\mathbb{P}\left(\left((S_n|S_n > F_{S_n}^{-1}(\epsilon)) - b\sqrt{\sum_{i=0}^{n+1}\sigma_i^2} - \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1\sqrt{\sum_{i=0}^{n}\sigma_i^2}\right) + (Z_{n+1} - \delta_1\sigma_{n+1}) \geq a\right)$$
$$\leq \exp\left(C_1\left(\frac{-a^2}{2\sum_{i=0}^{n+1}\sigma_i^2}\right)\right)$$

By the growth of the variance, it holds that $\sqrt{\sum_{i=0}^{n}\sigma_i^2} \leq \sqrt{\frac{\sum_{i=0}^{n+1}\sigma_i^2}{1+d}}$. Also, trivially, $\sigma_{n+1} \leq \sqrt{\sum_{i=0}^{n+1}\sigma_i^2}$. Using those inequalities, we get that

$$\delta_1\sigma_{n+1} + \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1\sqrt{\sum_{i=0}^{n}\sigma_i^2} \leq \delta_1\sqrt{\sum_{i=0}^{n+1}\sigma_i^2} + \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1\frac{\sqrt{\sum_{i=0}^{n+1}\sigma_i^2}}{\sqrt{1+d}}$$
$$= \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1\sqrt{\sum_{i=0}^{n+1}\sigma_i^2}$$

and hence can conclude that the induction step holds:

$$\mathbb{P}\left(\left((S_n|S_n > F_{S_n}^{-1}(\epsilon)) - b\sqrt{\sum_{i=0}^{n+1}\sigma_i^2} - \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1\sqrt{\sum_{i=0}^{n}\sigma_i^2}\right) + (Z_{n+1} - \delta_1\sigma_{n+1}) \geq a\right)$$
$$\leq \mathbb{P}\left((S_n|S_n > F_{S_n}^{-1}(\epsilon)) + Z_{n+1} - b\sqrt{\sum_{i=0}^{n+1}\sigma_i^2} - \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1\sqrt{\sum_{i=0}^{n+1}\sigma_i^2} \geq a\right)$$
$$= \mathbb{P}\left((S_n|S_n > F_{S_n}^{-1}(\epsilon)) + Z_{n+1} \geq a + (b + \frac{\sqrt{1+d}}{\sqrt{1+d}-1}\delta_1)\sqrt{\sum_{i=0}^{n+1}\sigma_i^2}\right)$$
$$\leq \exp\left(C_1\left(\frac{-a^2}{2\sum_{i=0}^{n+1}\sigma_i^2}\right)\right)$$

□

2.3 Inductive Bound Set for S_n

While Lemma 2 gives an upper bound for the cdf of S_n, the sub-Gaussian tail bound of S_n is not very sharp for values close to the mean of S_n. In this section we extend Lemma 2 with a *union-bound compositional approach* to tighten cdf bounds, as follows. We (i) split the cdf into $m \in \mathbb{N}$ pieces, (ii) provide lower bounds $B(S_n)$ for each piece of the cdf, and (iii) combine the lower bounds inductively into a tighter upper bound for the cdf of S_n.

Our compositional framework is inductive over the lower bounds of cdf pieces: S_0 satisfies the bound $B(S_0)$ (base case) and, if S_n satisfies the bound $B(S_n)$, then S_{n+1} satisfies $B(S_{n+1})$ (induction step). Our bound $B(S_n)$ is uniquely defined by a set of inequalities using two vectors $\vec{a}_B, \vec{b}_B$ of bounding values, where elements of $\vec{a}_B, \vec{b}_B$ provide the location of and lower bounds on the cdf pieces of S_n. As such, we set:

$$B(S_n) = \left\{ \mathbb{P}\big(S_n \le \vec{a}_{B,1}\sqrt{\sum_{i=0}^{n}\sigma_i^2}\big) \ge \vec{b}_{B,1}, \quad \dots, \quad \mathbb{P}\big(S_n \le \vec{a}_{B,m}\sqrt{\sum_{i=0}^{n}\sigma_i^2}\big) \ge \vec{b}_{B,m} \right\}. \tag{2.1}$$

For simplicity, we assume $\vec{a}_1 \le 0$ and $\vec{b}_1 \ge \epsilon$, in order to ensure that only negative values of s exit the loop therefore enforcing $F_{S_n}^{-1}(\epsilon) \le 0$. If each inequality in $B(S_n)$ is valid, we say that the bound $B(S_n)$ *holds*. By simple arithmetic reasoning, we state the following property over bounds.

Lemma 3 (Partial order of bounds)**.** *The bounds $B(S_n)$ admit an ordering whenever they describe the same intervals and probabilities are ordered. That is:*

$$B'(\cdot) \le B(\cdot) \iff \vec{a}_{B'} = \vec{a}_B \wedge \forall_{1 \le i \le m} : b_{B',i} \ge b_{B,i}$$

Inductive computations of bound set for S_n. With regard to the partial order, we provide our inductive computation for bounds of S_n: from a bound $B(\cdot)$ that holds for S_n, we compute a bound $B'(\cdot)$ that holds for S_{n+1}. If $B'(\cdot) \le B(\cdot)$, then $B(S_m)$ holds for all $m \ge n$, as our computation of new bounds ensures monotonicity of bounds. Doing so and starting with $B(S_n)$, (i) the left tail with a cdf of ϵ is cut away from S_n, yielding S'_n. Then, using Lemma 1, we (ii) add an almost Normal variable with variance $\sigma_{n+1}^2 \ge d\sum_{i=0}^{n}\sigma_i^2$ to the resulting distribution of S'_n, and compute a new bound $B'(S_{n+1})$. Our bound computation uses a union-bound approach for deriving interval boundaries. In addition to the bounds from B, we use tail bounds from Lemma 2, as otherwise obtaining an inductive bound for S_n with $\epsilon > 0$ is not possible.

The next example illustrates our inductive bound set computation for S_n.

Example 1. *Consider an instance of the stochastic process of Figure 2.1, by setting $\epsilon = 0.1$ and $d = 3$ and using a set of almost Normal variables $\{Z_n\}_{n\in\mathbb{N}}$ with parameters $C_1 = 1$, $\delta_1 = 0$, and $c_0 = 10^{-3}$. For this instance of Figure 2.1, we define the bound set:*

$$B(S_n) = \left\{ \mathbb{P}(S_n \le 0) \;\ge 0.1, \quad \mathbb{P}\big(S_n \le \sqrt{\textstyle\sum_{i=0}^{n}\sigma_i^2}\;\big) \;\ge 0.4 \right\}.$$

For the tail bounds of Lemma 2, we (arbitrarily) pick the value $3\sqrt{\sum_{i=0}^{n}\sigma_i^2}$ *and derive:*

$$\mathbb{P}(S_n \geq 3\sqrt{\sum_{i=0}^{n}\sigma_i^2}\ ) \quad \leq \exp(\frac{-(2.54)^2}{2}) \quad \leq 0.04.$$

Therefore, $\mathbb{P}(S_n \leq 3\sqrt{\sum_{i=0}^{n}\sigma_i^2}) \geq 0.96$. *Using these inequalities, we compute new bounds* S_n'. *Here,* $S_n' = (S_n \geq F_{S_n}^{-1}(\epsilon))$ *is the variable obtained from* S_n *by cutting away the left tail with weight* ϵ. *For readability, we denote the summation of variances up to* S_n *through* $\sigma_{S_n}^2 := \sum_{i=0}^{n}\sigma_i^2$, *and similarly* $\sigma_{S_{n+1}}^2 := \sum_{i=0}^{n+1}\sigma_i^2$. *With this notation at hand, we have:*

$$\begin{aligned}
\mathbb{P}(S_{n+1} \leq 0) \quad &\geq \quad \mathbb{P}(S_n' \leq \sigma_{S_n}) \cdot \mathbb{P}(Z_{n+1} \leq -\sigma_{S_n}) + \\
&\quad\quad \mathbb{P}(\sigma_{S_n} \leq S_n' \leq 3\sigma_{S_n}) \cdot \mathbb{P}(Z_{n+1} \leq -3\sigma_{S_n}) \\
\mathbb{P}(S_{n+1} \leq \sigma_{S_{n+1}}) \quad &\geq \quad \mathbb{P}(S_n' \leq \sigma_{S_n}) \cdot \mathbb{P}(Z_{n+1} \leq \sigma_{S_{n+1}} - \sigma_{S_n}) + \\
&\quad\quad \mathbb{P}(\sigma_{S_n} \leq S_n' \leq 3\sigma_{S_n}) \cdot \mathbb{P}(Z_{n+1} \leq \sigma_{S_{n+1}} - 3\sigma_{S_n})
\end{aligned}$$

Note that $\sqrt{\sum_{i=0}^{n+1}\sigma_i^2} \geq \sqrt{1+d}\sqrt{\sum_{i=0}^{n}\sigma_i^2}$ *and* Z_{n+1} *has standard deviation* $\sqrt{3\sum_{i=0}^{n}\sigma_i^2}$. *Using the bound set* $B(S_n)$ *and Lemma 1, we get:*

$$\mathbb{P}\left(S_{n+1} \leq 0\right) \quad \geq \quad \frac{0.4-\epsilon}{1-\epsilon}\left(\Phi(\frac{-1}{\sqrt{3}}) - c_0\right) + \frac{0.56}{1-\epsilon}\left(\Phi(\frac{-3}{\sqrt{3}}) - c_0\right) \approx 0.1188$$

Similarly, for the second bound $\mathbb{P}(S_{n+1} \leq \sqrt{\sum_{i=0}^{n+1}\sigma_i^2}) \gtrapprox 0.4138$. *The bound* B *is inductive as the new bound which we computed for* S_{n+1} *is smaller than the initial bound* B *which is assumed to hold for* S_n. □

Linear Model. To compute an inductive bound B, the parameters $\epsilon, d, c_0, C_1, \delta_1$ must be fixed. Additionally, we require a sorted vector $\vec{a}$ with length m, specifying $\vec{a}_B$, as well as a sorted vector $\vec{c}$ with length k, with values, for which the Chernoff bound should be used. The variables $\vec{b}$, corresponding to $\vec{b}_B$ are the variables of interest. In the following we specify a linear model, where the solution values of $\vec{b}_B$ together with the intervals given as $\vec{a}_B$ specify a valid inductive bound for the given parameters.

We introduce the auxiliary variables $\vec{d}$ of length $m+k$, which instead represent a bound for probability mass of the interval $[\vec{a}_{i-1}; \vec{a}_i]$ and $[\vec{c}_{i-1}; \vec{c}]$ of the conditioned variable $(S_n | S_n \geq F_{S_n}^{-1}(\epsilon))$:

$$\vec{d}_1(1-\epsilon) \leq \vec{b}_1 - \epsilon \wedge \forall_{2 \leq i \leq m} : \vec{d}_i(1-\epsilon) \leq \vec{b}_i - \vec{b}_{i-1}$$

and similarly for the values of the Chernoff bound with $b = \frac{\sqrt{2\ln\frac{1}{1-\epsilon}}}{C_1(\sqrt{(1+d)}-1)}$ (also $\vec{d}_{m+1}$ represents the interval $[\vec{a}_m; \vec{c}_0]$):

$$\vec{d}_{m+1}(1-\epsilon) \leq 1 - \exp\left(C_1 \frac{-(\vec{c}_i - b - \delta_1)^2}{2}\right) - \vec{b}_m \wedge \forall_{2\leq i\leq k}:$$
$$\vec{d}_{m+i}(1-\epsilon) \leq \left(\exp\left(C_1\frac{-(\vec{c}_{i-1} - b - \delta_1)^2}{2}\right) - \exp\left(C_1\frac{-(\vec{c}_i - b - \delta_1)^2}{2}\right)\right)$$

Every inequality must then hold for S_{n+1}:

$$\vec{b}_i \geq \sum_{j=1}^{m} \vec{d}_j \left(\Phi\left(\frac{\vec{a}_i\sqrt{1+d} - \vec{a}_j}{\sqrt{d}}\right) - c_0\right)$$
$$+ \sum_{j=1}^{k} \vec{d}_{m+j} \left(\Phi\left(\frac{\vec{a}_i\sqrt{1+d} - \vec{c}_j}{\sqrt{d}}\right) - c_0\right)$$

for all $i = 1, \ldots, m$.

And ultimately, we can only be less restrictive than the loop exit condition, hence the whole tail that is removed when conditioning must be smaller 0:

$$\vec{a_1} \leq 0 \wedge \vec{b}_1 \geq \epsilon$$

To ensure, that S_0 satisfies the bound, it must also hold, that:

$$(\Phi(\vec{a}_i) - c_0) \geq \vec{b}_i$$

for all $i = 1, \ldots, m$.

On the existence of inductive bounds. Our approach to inductively computing bound sets for S_n relies on an union-bound argument to improve the cdf bound of Lemma 2. Recall that the sub-Gaussian tail bound of S_n only depends on $d \in \mathbb{R}^+ \setminus 0$ and $\epsilon \in \mathbb{R}^+ \setminus 0$. We next show that the existence of an inductive bound set is conditioned only by such a d. Namely, Theorem 1 ensure that, for every d there is an $\epsilon \in \mathbb{R}^+ \setminus 0$, such that an inductive bound exists for the corresponding series $\{S_n\}_{n\in\mathbb{N}}$, with $F_{S_n}^{-1}(\epsilon) \leq 0$. The probability of S_n being smaller than zero is therefore *lower bounded* by some nonzero percentage.

Theorem 1 (Inductive bound set)**.** *For every $d \in \mathbb{R}^+ \setminus 0$ there exists an $\epsilon \in \mathbb{R}^+ \setminus 0$ such that an inductive bound set $B(S_n)$ holds, with $F_{S_n}^{-1}(\epsilon) \leq 0$, given that c_0 converges to 0 and can be chosen arbitrarily small.*

Proof. Lemma 2 implies that $\mathbb{P}(S_n \geq c)$ is bounded and converges to 0 as $c \to \infty$. We choose an arbitrary c' such that the tail bound $\mathbb{P}(S_n \geq c') \leq b' < \frac{1}{2}$ holds. Then, $\mathbb{P}(S_n < c') \geq 1 - b'$ and $\mathbb{P}((S_n|S_n \geq 0) < c') \geq 1 - b' - \epsilon \geq 0$. By union-bound properties, we have $\mathbb{P}\Big(S_{n+1} \leq 0\Big) \geq \mathbb{P}\Big((S_n|S_n \geq 0) < c'\Big) \cdot \mathbb{P}\Big(Z_{n+1} < -c'\Big) \geq (1 - b' - \epsilon)(\Phi(\frac{c'}{\sqrt{d}}) - c_0)$. The bound with $\vec{a} = \Big(0\Big)$ and $\vec{b} = \left(\frac{(1-b')(\Phi(\frac{c'}{\sqrt{d}})-c_0)}{1+(\Phi(\frac{c'}{\sqrt{d}})-c_0)}\right)$ is then inductive and the lower bound is nonzero when c_0 is small enough.

□

CHAPTER 3

Polynomial Random Walks

Chapter 2 showed that stochastic processes defined by Figure 2.1 have bounded behavior, allowing us to lower bound the termination probability via sub-Gaussian tail bounds and inductive bound sets. In this chapter we map the termination analysis of certain polynomial programs, called polynomial random walks, to the framework of Chapter 2. Importantly, *we reduce the problem of verifying PAST of polynomial random walks to the problem of ensuring existence of inductive bounds* (Theorem 2). Our recipe consists of transforming a polynomial random walk program $\mathcal{P}$ to a program that (i) bounds PAST of P and (ii) is equivalent to the stochastic process of Figure 2.1.

```
$n \leftarrow 0$
$y \leftarrow y_0$
while $y > 0$ do
    $n \leftarrow n + 1$
    $x \leftarrow q_1[n] \oplus_p q_2[n]$
    $y \leftarrow y + x$
end while
```

Figure 3.1: Polynomial random walk $\mathcal{P}$

3.1 Programming Model

We define the class of *polynomial random walks* via the programming model of Figure 3.1, where $q_1[n], q_2[n] \in \mathbb{R}[n]$ are arbitrary polynomial expressions in the loop counter n. The *degree of a polynomial random walk program* $\mathcal{P}$, written as $\deg(\mathcal{P})$, is given by the maximum degree of its polynomials, that is $\deg(\mathcal{P}) = \max\{\deg(q_1[n]), \deg(q_2[n])\}$. The series $\{X_n\}_{n\in\mathbb{N}}$, $\{Y_n\}_{n\in\mathbb{N}}$ induced by the random loop variables x, y are next defined.

Definition 1 (Random walk variables)**.** *The* random walk variable X_n *corresponding to the loop variable x at iteration n in Figure 3.1 is*

$$X_n = \begin{cases} q_1[n] & \textit{with probability } p \\ q_2[n] & \textit{with probability } 1-p. \end{cases}$$

The random walk variable Y_n *captures the distribution of* y *after iteration* n*, as:*

$$Y_{n+1} = (Y_n | Y_n > 0) + X_{n+1}.$$

The second-order moment of a random variable X_n is written as $Var(X_n)$. For Figure 3.1, we have $\mathbb{E}(X_n) = q_1[n]p + q_2[n](1-p)$ and $Var(X_n) = q_1[n]^2 p + q_2[n]^2(1-p)$, capturing the mean (first moment) and variance (second moment) of X_n; note that both moments of X_n are also polynomials in n.

To prove PAST of Figure 3.1 we need to prove that the expected value of its stopping time is finite [2]. Based on the semantics of Figure 3.1, it is easy to see that the stopping time of Figure 3.1 is given by the first iteration n in which Y_n becomes negative.

Definition 2 (Expected stopping time)**.** *Let* T *be* $\inf\{n \geq 0 : Y_n \leq 0\}$*, where* T *denotes the* stopping time *of the stochastic process induced by the polynomial random walk of Figure 3.1. The* expected stopping time *of Figure 3.1 is defined as* $\mathbb{E}(T) = \sum_{n=0}^{\infty} \mathbb{P}(T \geq n)$*.*

We exploit Definition 2 to show that Figure 3.1 is PAST under additional conditions. Namely, we translate Figure 3.1 into Figure 3.2 and ensure that the stopping time of Figure 3.2 becomes finite above a certain threshold; this threshold depends only on the maximum polynomial degree of Figure 3.1 and the variable k. We then show that finiteness of the stopping time of Figure 3.2 implies PAST of Figure 3.1 (Lemma 4).

```
$n \leftarrow 0$
$y \leftarrow y_0$
while $n \leq n_0$ do
    $n \leftarrow n + 1$
    $x \leftarrow q_1[n] \oplus_p q_2[n]$
    $y \leftarrow y + x$
end while
while $y > g$ do                    ▷ where $g \leq 0$
    $z \leftarrow 0$
    $n' \leftarrow n$
    while $n \leq n' \cdot k$ do
        $n \leftarrow n + 1$
        $x \leftarrow q_1[n] \oplus_p q_2[n]$
        $z \leftarrow z + x$
    end while
    $y \leftarrow y + z$
end while
```

Figure 3.2: Transformed random walk

Program transformation. We translate Figure 3.1 into the stochastic process of Figure 3.2. This program transformation is defined through the parameters n_0, k and g. The loop body of Figure 3.2 is initially executed several times, accumulating n_0 steps. In every iteration of the outer loop k times as many steps as before are summed up, before the loop guard is checked again. Furthermore, the loop guard of Figure 3.2 might be relaxed, as $g \leq 0$. We highlight similarities between Figure 3.2 and the summation of almost Normal variables with conditioning in Figure 2.1: the inner loop of Figure 3.2 computes the value for z by summing up X_i. As argued in Section 3.2, this is similar to drawing z from an almost Normal distribution as in Figure 2.1.

We have that the expected stopping time of Figure 3.2 is larger than of Figure 3.1.

Lemma 4 (Stopping Time Inequality)**.** *Let T' be* $\inf\{n \geq n_0 : Y_n \leq g\}$*, denoting the stopping time of Figure 3.2. Then,* $\mathbb{E}(T) \leq \mathbb{E}(T')$*.*

Proof. For every possible draw of the variables X_n, the termination of Figure 3.2 implies the termination of Figure 3.1

In Figure 3.2, the inner loop is executed at least once per execution of the outer loop's body, hence n is incremented at least once per iteration of the outer loop. When the loop stops after t steps, then either Figure 3.1 also stops after t steps or it has already stopped at some $t' < t$. □

We denote with p_{term} a lower bound for the probability of the outer loop terminating. In what follows, we will ensure the stopping time of the program in Figure 3.2 is finite when the probability p_{term} is high enough and k is small (Lemma 8). The existence of a nonzero lower bound for p_{term} is implied by Theorem 1; we note that p_{term} depends on the probability of choosing a branch p and grows as $k^{2\deg(\mathcal{P})} + 1$ increases. By setting k to its maximum value, we derive a threshold $d_{\min}(p)$ for the degree $\deg(\mathcal{P})$ of the polynomial random walk of Figure 3.2 and prove that the stopping time of Figure 3.1 above this threshold $d_{\min}(p)$ is finite (Theorem 2). We thus use $d_{\min}(p)$ to provide sufficient conditions for deciding PAST of the polynomial random walks in Figure 3.1.

3.2 Loop Summations of Polynomial Random Walk Increments

We now establish the formal connection between the polynomial random walks of Figure 3.2 and the stochastic processes of Figure 2.1. We prove that the loop summation (defined below) of the increments of the random walk in Figure 3.2 is almost normally distributed as given in Lemma 1, when an inequality over the degrees of expected value of the step and its variance is true. This inequality holds, whenever the leading terms of the steps cancel out.

Definition 3 (Random walk loop summation)**.** *The random variables $U_0 = y_0 + X_0 + \cdots + X_{n_0}$ and $U_{n'} = X_{n'} + \cdots + X_{\lceil n' \cdot k \rceil}$ are* (loop) summations *of the random variables X_i of Figure 3.2.*

Lemma 5 then shows that the absolute deviation c_0 from the cdf of the Normal distribution converges to 0. Further, Lemma 6 conjectures that the summation of random walk increments admits a sub-Gaussian tail bound with $C_1 = 4p(1-p)$ and δ_1 converging to 0, thus establishing, that the loop summation follows an almost Normal distribution $\mathcal{N}_{c_0}^{\delta_1, 4p(1-p)}$.

Lemma 5 (Convergence of cdf deviation)**.** *Assume that* $\deg(Var(X_i)) > 2\deg(\mathbb{E}(X_i)) + 1$ *holds for Figure 3.2. Then, the normalizations of the loop summations U_0 and $U_{n'}$ follow* *a Normal distribution up to a constant error c_0, with c_0 converging to 0 with increasing n_0:*

$$\forall n' \geq n_0 : \left| F_{U_{n'}}(z) - \Phi\left(\frac{z}{\sqrt{\sum_{i=n_0}^{\lceil n'\cdot k\rceil} Var(X_i)}}\right)\right| \leq c_0$$

where n_0 is as given in Figure 3.2 and $F_{U'_n}$ denotes the cdf of $U_{n'}$.

Proof. We first consider the centered version of $U_{n'}$:

$$U'_{n'} = X_{n'} - \mathbb{E}(X_{n'}) + \cdots + X_{\lceil n'\cdot k\rceil} - \mathbb{E}(X_{\lceil n'\cdot k\rceil}).$$

Let now $F'_{n'}$ be the cdf of $U'_{n'}$.

By the Berry-Esseen-Theorem [15], it then holds that

$$\sup_{x\in\mathbb{R}} \left| F'_{n'}(x) - \Phi(\frac{x}{\sqrt{\sum_{i=n'}^{\lceil k\cdot n'\rceil} Var(X_i)}})\right| \leq C_0 \frac{\sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}(|X_i - \mathbb{E}(X_i)|^3)}{\left(\sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}((X_i - \mathbb{E}(X_i))^2)\right)^{\frac{3}{2}}}$$

For the sake of readability, we define $b(n') = C_0 \frac{\sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}(|X_i - \mathbb{E}(X_i)|^3)}{\left(\sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}((X_i - \mathbb{E}(X_i))^2)\right)^{\frac{3}{2}}}$.

For the numerator, the following equality holds:

$$\sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}(|X_i - \mathbb{E}(X_i)|^3) = \sum_{i=n'}^{\lceil k\cdot n'\rceil} |q_1[i] - \mathbb{E}(X_i)|^3 \, p + |q_2[i] - \mathbb{E}(X_i)|^3 \, (1-p)$$

$\mathbb{E}(X_i)$ is a polynomial (see Definition 1) and different from $q_1[i]$ and $q_2[i]$, as we have non-zero variance. Therefore the numerator is a polynomial and its degree is:

$$\deg\left(\sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}(|X_i - \mathbb{E}(X_i)|^3)\right) = 3\deg(\mathcal{P}) + 1$$

Note that the $+1$ comes from the summation and the fact, that $k > 1$.

Analogously,

$$\sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}((X_i - \mathbb{E}(X_i))^2) = \sum_{i=n'}^{\lceil k\cdot n'\rceil} (q_1[i] - \mathbb{E}(X_i))^2 \, p + (q_2[i] - \mathbb{E}(X_i))^2 \, (1 - p)$$

and therefore the degree of the denominator is:

$$\deg\left(\left(\sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}((X_i - \mathbb{E}(X_i))^2)\right)^{\frac{3}{2}}\right) = 3\deg(\mathcal{P}) + \frac{3}{2}$$

Since the degree of the denominator is higher $b(n')$ converges to zero.

By definition, $U_{n'} = U'_{n'} + \sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}(X_i)$, and therefore $F_{U'_n}(x) = F'_{n'}(x - \sum_{i=n'}^{\lceil k\cdot n'\rceil} \mathbb{E}(X_i))$.

For the standard normal distribution Φ it holds that $\Phi(x+\delta) \leq \Phi(x) + |\delta|$, as the probability density function of the standard normal distribution is always smaller than 1:

$$\sup_{x\in\mathbb{R}} \left| F_{U_{n'}}(x) - \Phi\left(\frac{x - \sum_{i=n'}^{\lceil k\cdot n'\rceil}\mathbb{E}(X_i)}{\sqrt{\sum_{i=n'}^{\lceil k\cdot n'\rceil} Var(X_i)}}\right)\right|$$
$$\geq \sup_{x\in\mathbb{R}} \left| F_{U_{n'}}(x) - \Phi\left(\frac{x}{\sqrt{\sum_{i=n'}^{\lceil k\cdot n'\rceil} Var(X_i)}}\right)\right| - \frac{\sum_{i=n'}^{\lceil k\cdot n'\rceil}\mathbb{E}(X_i)}{\sqrt{\sum_{i=n'}^{\lceil k\cdot n'\rceil} Var(X_i)}}$$

By the above inequality, it then holds that

$$\sup_{x\in\mathbb{R}} \left| F_{U_{n'}}(x) - \Phi\left(\frac{x}{\sqrt{\sum_{i=n'}^{\lceil k\cdot n'\rceil} Var(X_i)}}\right)\right| \leq b(n') + \frac{\sum_{i=n'}^{\lceil k\cdot n'\rceil}\mathbb{E}(X_i)}{\sqrt{\sum_{i=n'}^{\lceil k\cdot n'\rceil} Var(X_i)}}$$

Since $b(n')$ converges to 0, the whole bound converges when $\deg(Var(X_i)) > 2\deg(\mathbb{E}(X_i)) + 1$.

Analogously, we consider $U'_0 = X_0 - \mathbb{E}(X_0) + \cdots + X_{n_0} - \mathbb{E}(X_{n_0})$, and let F'_0 be its cdf. Following the same argument, it then holds that:

$$\sup_{x\in\mathbb{R}} \left| F'_0(x) - \Phi(\frac{x}{\sqrt{\sum_{i=0}^{n_0} Var(X_i)}})\right| \leq b(0) := C_0 \frac{\sum_{i=0}^{n_0}\mathbb{E}(|X_i - \mathbb{E}(X_i)|^3)}{(\sum_{i=0}^{n_0}\mathbb{E}((X_i - \mathbb{E}(X_i))^2))^{\frac{3}{2}}}$$

and the expression on the right hand side again converges.

And since $U_0 = U'_o + y_0 + \mathbb{E}(X_0) + \cdots + \mathbb{E}(X_{n_0})$, and $\Phi(x+\delta) \leq \Phi(x) + |\delta|$, it holds that:

$$\sup_{x\in\mathbb{R}} \left| F_{U_0}(x) - \Phi\left(\frac{x - \sum_{i=0}^{n_0}\mathbb{E}(X_i)}{\sqrt{\sum_{i=0}^{n_0} Var(X_i)}}\right)\right|$$
$$\geq \sup_{x\in\mathbb{R}} \left| F_{U_0}(x) - \Phi\left(\frac{x}{\sqrt{\sum_{i=0}^{n_0} Var(X_i)}}\right)\right| - \frac{\sum_{i=0}^{n_0}\mathbb{E}(X_i)}{\sqrt{\sum_{i=0}^{n_0} Var(X_i)}}$$

and therefore

$$\sup_{x\in\mathbb{R}} \left| F_{U_0}(x) - \Phi\left(\frac{x}{\sqrt{\sum_{i=0}^{n_0} Var(X_i)}}\right)\right| \leq b(0) + \frac{\sum_{i=0}^{n_0}\mathbb{E}(X_i)}{\sqrt{\sum_{i=0}^{n_0} Var(X_i)}}$$

The bound for the deviation of the cdf of U_0 therefore also converges. By the definition of convergence, we can always find some n_0 for an arbitrary $c_0 > 0$ (at least numerically), which is bigger than both bounds, as required in the lemma. □

Using Lemma 5, we derive that the loop summations of polynomial random walks follow an almost Normal distribution, similarly to the stochastic process of Figure 2.1 in Chapter 2.

Lemma 6 (Tail bound for $U_{n'}$)**.** *Let $\sigma_{U_{n'}}$ be the standard deviation of $U_{n'}$ and assume* $\deg(Var(X_i)) > 2\deg(\mathbb{E}(X_i)) + 1$*. Then, the right tail probability is bounded with δ_1 converging to* 0*, as follows:*

$$\begin{aligned} \mathbb{P}(U_{n'} \geq \lambda\sigma_{U_{n'}} + \delta_1\sigma_{U_{n'}}) &\leq \exp\left(4(1-p)p\frac{-\lambda^2}{2}\right), \quad \textit{and} \\ \mathbb{P}(U_0 \geq \lambda\sigma_{U_0} + \delta_1\sigma_{U_0}) &\leq \exp\left(4(1-p)p\frac{-\lambda^2}{2}\right). \end{aligned}$$

Proof. We again consider the centered version of $U'_{n'}$ as in the proof of Lemma 5. Let $q'_1[i] = q_1[i] - E[X_i]$ and $q'_2[i] = q_2[i] - E[X_i]$ be the increments of $U'_{n'}$.

The moment generating function of $U'_{n'}$ can be bounded using Hoeffding's Lemma [16, Lemma 2.6], since each variable $X_i - \mathbb{E}(X_i)$ has bounded support:

$$M_{U'_{n'}}(t) = \prod_{i=n'}^{\lceil n' \cdot k \rceil} \exp\left(tq'_1[i]\right)p + \exp\left(tq'_2[i]\right)(1-p) \leq \prod_{i=n'}^{\lceil n' \cdot k \rceil} \exp\left(\frac{t^2(q'_1[i] - q'_2[i])^2}{8}\right)$$

Since the increments are zero-mean, $q'_2[i] = -\frac{q'_1[i]p}{(1-p)}$. Therefore,

$$\begin{aligned} \frac{(q'_1[i] - q'_2[i])^2}{Var(X_i)} = \frac{\left(q'_1[i] + \frac{q'_1[i]p}{(1-p)}\right)^2}{q'_1[i]^2 p + \left(\frac{q'_1[i]p}{(1-p)}\right)^2(1-p)} &= \frac{\left(1 + \frac{p}{(1-p)}\right)^2}{p + \left(\frac{p}{(1-p)}\right)^2(1-p)} \\ &= \frac{1}{(1-p)p} \end{aligned}$$

Hence,

$$M_{U'_{n'}}(t) \leq \prod_{i=n'}^{\lceil n' \cdot k \rceil} \exp\left(\frac{t^2 Var(X_i)}{8((1-p)p)}\right) = \exp\left(\frac{t^2 \sum_{i=n'}^{\lceil n' \cdot k \rceil} Var(X_i)}{8((1-p)p)}\right) = \exp\left(\frac{t^2 Var(U'_{n'})}{8((1-p)p)}\right)$$

The Chernoff bound[14, p.77] states, that for any positive t and variable X, with its moment generating function $M_X(t)$ it holds that:

$$\mathbb{P}(X \geq a) \leq e^{-ta} M_X(t).$$

We now apply the Chernoff-bound for the variable $U'_{n'}$ and choose $a = \lambda\sqrt{Var(U'_{n'})}$ and $t = \frac{\lambda 4((1-p)p)}{\sqrt{Var(U'_{n'})}}$:

$$\mathbb{P}(X \geq \lambda\sqrt{Var(U'_{n'})}) \leq \exp\left(4((1-p)p)\left(-\frac{\lambda^2}{2}\right)\right)$$

By the definition of $U'_{n'}$ (note that also $Var(U'_{n'}) = Var(U_{n'})$), it then follows that

$$\mathbb{P}(U_0 \geq \lambda\sqrt{Var(U_{n'})} + \sum_{i=n'}^{\lceil n'\cdot k\rceil} \mathbb{E}(X_i)) \leq \exp\left(4((1-p)p)\left(-\frac{\lambda^2}{2}\right)\right)$$

Analogously, for U'_0 we obtain can obtain the same bound:

$$M_{U'_0}(t) \leq \exp\left(\frac{t^2 Var(U'_0}{8(1-p)p}\right).$$

Then applying the Chernoff-bound:

$$\mathbb{P}(X \geq \lambda\sqrt{Var(U'_0)}) \leq \exp\left(4((1-p)p)\left(-\frac{\lambda^2}{2}\right)\right)$$

and since $U_0 = U'_0 + y_0 + \sum_{i=0}^{n_0} \mathbb{E}(X_i)$:

$$\mathbb{P}(U_0 \geq \lambda\sqrt{Var(U_0)} + y_0 + \sum_{i=0}^{n_0} \mathbb{E}(X_i)) \leq \exp\left(4((1-p)p)\left(-\frac{\lambda^2}{2}\right)\right)$$

By the same argument as in the proof of Lemma 5

$$\delta_1 = \max\left\{\frac{\sum_{i=n'}^{\lceil n'\cdot k\rceil} \mathbb{E}(X_i)}{\sqrt{\sum_{i=n'}^{\lceil n'\cdot k\rceil} Var(X_i)}}, \frac{y_0 + \sum_{i=0}^{n_0} \mathbb{E}(X_i)}{\sqrt{\sum_{i=0}^{n_0} Var(X_i)}}\right\}$$

converges to zero. □

Example 2. *Consider our motivating example from Figure 1.1. In order to ensure that its loop summations follow an almost Normal distribution, with* c_0 *and* δ_1 *converging to zero, we need to ensure that* $\deg(Var(X_i)) > 2\deg(\mathbb{E}(X_i)) + 1$*. This inequality is true, since* $Var(X_i) = (i+1)^2$ *and* $\mathbb{E}(X_i) = 4$*, hence* $\deg(Var(X_i)) = 2$ *and* $\deg(\mathbb{E}(X_i)) = 0$*. Consequently, Lemma 5 and 6 can be used.* □

In the remaining, we define the random variable series $\{Z_n\}_{n\in\mathbb{N}}$ corresponding to the loop summation of the inner loop of Figure 3.2. That is, Z_n captures the program variable z at the end of every iteration of the outer loop of Figure 3.2, with Z_0 being the variable corresponding to z after its first loop. As such,

- $Z_0 = U_0$, and
- $Z_n = U_{n'_{(n)}}$, where $n'_{(n)}$ is the value of n' in the n-th iteration of the outer loop of Figure 3.2.

Further, Y_n is induced by the program variable y of Figure 3.2, capturing the loop summation of Z_n with repeated conditioning. In order to use inductive bound sets as in Theorem 1, the variance of $\{Z_n\}_{n\in\mathbb{N}}$ must grow consistently and exponentially. This is however clearly ensured by choosing $k > 1$ in Figure 3.2, implying the following result.

Lemma 7 (Growth of variance)**.** *The variance $\{\sigma_n^2\}_{n\in\mathbb{N}}$ of $\{Z_n\}_{n\in\mathbb{N}}$ grows exponentially, with δ' converging to 1:*

$$\sigma_{n+1}^2 \geq \left(\delta' k^{2\deg(\mathcal{P})+1} - 1)\right) \sum_{i=0}^{n} \sigma_i^2$$

Proof. Let $q_{\text{var}}[n]$ be the polynomial describing the variance of $X_1 + \cdots + X_n$. Let $m = deg(q_{\text{var}})$ denote its degree. Then $m = \deg(\mathcal{P}){+}1$ and $Var(Z_n) = q_{\text{var}}[\lceil k{\cdot}n'_{(n)}\rceil] - q_{\text{var}}[n'_{(n)}]$. Since $Z_1 + \cdots + Z_{n-1} = X_1 + \cdots + X_{n'_{(n)}}$ and thus $Var(Z_1 + \cdots + Z_{n-1}) = q_{\text{var}}[n'_{(n)}]$.

But then the factor d, such that $Var(Z_n) \geq d(Var(Z_1) + \cdots + Var(Z_{n-1})) = d(Var(Z_1 + \cdots + Z_{n-1}))$ can be computed:

$$q_{\text{var}}[\lceil k \cdot n'_{(n)}\rceil] - q_{\text{var}}[n'_{(n)}] \geq d \cdot q_{\text{var}}[n'_{(n)}]$$

A polynomial can be bounded by its leading term with a multiplicative factor. Specifically, with $\delta_1\delta_2 \in \mathbb{R}^+$:

$$\forall n \geq n'_0(\delta_1, \delta_2) : (1-\delta_1)a_m n^m \leq a_1 x + \cdots + a_m n^m \leq (1+\delta_2)a_m n^m$$

Inserting this in the above equation:

$$(1-\delta_1)a_m(\lceil k \cdot n'_{(n)}\rceil)^m \geq (d+1)(1+\delta_2)a_m n'^m_{(n)}$$
$$\delta' k^m = \frac{(1-\delta_1)}{1+\delta_2} k^m \geq (d+1)$$

□

Lemmas 5 and 6 establish that Z_n follows an almost Normal distribution as in Lemma 1. Together with Lemma 7, this ensures that the right tail of Y_n can be bounded (Lemma 2), and therefore inductive bounds can be used. Based on this bounds, Section 3.3 introduces conditions on the stopping time T of Figure 3.2 being finite, implying thus PAST of Figure 3.1.

3.3 Bounding the Stopping Time and PAST

Recall that, using Definition 2, the expected stopping time $\mathbb{E}(T)$ of Figure 3.2 is determined by the loop summation variables Y_n and is set to:

$$\mathbb{E}(T) = \sum_{n=0}^{\infty} \mathbb{P}(T \geq n).$$

Using Lemma 7, we obtain the following bound on $\mathbb{P}(T \geq n)$, and hence on $\mathbb{E}(T)$.

Lemma 8 (Bounding the stopping time)**.** *Assume that the outer loop of Figure 3.2 terminates with probability* p_{term} *after some* n_0*. Then,*

$$\mathbb{P}(T \geq n) \leq \min\left\{1, Bn^{\frac{\ln(1-p_{\text{term}})}{\ln(k+\frac{1}{n_0})}}\right\}$$

where $B = \frac{1}{(1-p_{\text{term}})^{\log_{k+\frac{1}{n_0}}(n_0)+2}}$*. Therefore, if* $\ln(1-p_{\text{term}}) < -\ln(k+\frac{1}{n_0})$ *holds, then the expected stopping time* $\mathbb{E}(T)$ *is finite.*

Proof. During the nth iteration of the inner loop body, Figure 3.2 could have terminated $\lfloor \log_{k+\tau}(n) - \lceil \log_{k+\tau}(n_0) \rceil \rfloor \leq \log_{k+\tau}(n) - \log_{k+\epsilon}(n_0) - 2$ times. The error τ here accounts for the number of rounding ups, as we take $\lceil kn'_{(n)} \rceil$. A safe choice to approximate τ is $\tau = \frac{1}{n_0}$. With increasing n_0, note that τ converges to 0. As such, the probability of Figure 3.2 not terminating is bounded:

$$\mathbb{P}(T \geq n) \leq (1-p_{\text{term}})^{\log_{k+\tau}(n)-\log_{k+\tau}(n_0)-2} = \frac{n^{\frac{\ln(1-p_{\text{term}})}{\ln(k+\tau)}}}{(1-p_{\text{term}})^{\log_{k+\tau}(n_0)+2}}$$

□

On the finiteness of stopping times. Lemma 8 formulates conditions under which Figure 3.2 has finite stopping time. These conditions effectively only depend on the probability p_{term} and k, as n_0 can be chosen arbitrarily. As such, *finiteness of* $\mathbb{E}(T)$ *and PAST of Figure 3.2 is reduced to finding an inductive bound*, with $d = \delta' k^{2\deg(\mathcal{P})+1} - 1$, $C = p(1-p)$ and ϵ (which is a lower bound for p_{term}) so large, that the inequality in Lemma 8 is satisfies. The terms δ_1, δ' and c_0 can be computed from a finite, arbitrary n_0.

To this end, let $p_{\text{i.b.}}(d, n_0, p)$ be a to-be-determined function that returns the largest ϵ for which an inductive bound exists, which is a lower bound for p_{term}. Then,

$$\mathbb{P}(T \geq n) \leq \inf_{1<k}\left\{\inf_{0\leq n_0}\left\{\min\left\{Bn^{\frac{\ln(1-p_{\text{i.b.}}(\delta' k^{2\deg\{\mathcal{P}+1\}}-1,n_0,p))}{\ln(k+\frac{1}{n_0})}}\right\}\right\}\right\} \tag{3.1}$$

with $B = (1 - p_{\text{i.b.}}(\delta' k^{2\deg\{\mathcal{P}+1\}} - 1, n_0, p))^{-(\log_{k+\frac{1}{n_0}}(n_0)+2)}$. Enforcing (3.1) requires however *solving a non-trivial optimization problem*: we need to approximate the function $p_{\text{i.b.}}(d, n_0, p)$. While in Chapter 4 we show that this approximation can be done using linear programming and a genetic algorithm, the statement of (3.1) has theoretical consequences. The existence of an inductive bound set from Theorem 1 implies that an ϵ for $p_{\text{i.b.}}(d, n_0, p)$ always exist, allowing us to state a PAST condition over polynomial random walks $\mathcal{P}$ from Figure 3.1.

Theorem 2 (PAST of polynomial random walks)**.** *Let* $\mathcal{P}$ *be a polynomial random walk program of Figure 3.1. For every probabilistic choice* p *in* $\mathcal{P}$ *there exists a threshold* $d_{\min}(p)$ *such that* $\mathcal{P}$ *has finite expected stopping time, when* $\deg(\mathcal{P}) > d_{\min}(p)$ *and* $\deg(\mathcal{P}) > \deg(p(q_1[n]) + (1-p)q_2[n])$*.*

Proof. As $\deg(\mathcal{P}) > \deg(p(q_1[n]) + (1-p)q_2[n])$, we get $\deg(Var(X_i)) > 2\deg(\mathbb{E}(X_i)) + 1$. We use Lemmas 5–6 and take an arbitrary d. By Theorem 1, there exists an inductive bound with a nonzero termination probability p_{term}. Lemma 8 implies that the expected stopping time $\mathbb{E}(T)$ is finite when $\ln(1-p_{\text{term}}) < -\ln(k+\frac{1}{n_0})$. Further, Lemma 7 asserts $d = \delta' k^{\deg(\mathcal{P})+1} - 1$. Since $\ln(k+\frac{1}{n_0})$ converges to $\ln(k)$ as n_0 increases, we can conclude that $\mathcal{P}$ has finite expected stopping time. □

Example 3 (PAST of Figure 1.1)**.** *There exists an inductive bound with* $\epsilon \leq 0.1128$ *and* $d \geq 0.4102$ *for* $p = 0.5$*, when the values of* c_0 *and* δ_1 *are small, i.e.* $c_0 = 10^{-8}, \delta_1 = 10^{-5}$*. These constants are chosen so that convergence is guaranteed. The values for* $\vec{a}_B$ *and* $\vec{b}_B$ *are given in Tables 3.1–3.2, and additionally the bound is displayed in Figure 3.3 in red. Additionally, the tail bound is displayed, for which we use the value* $\vec{c} = (6.497321214442595)$*. For showing PAST, we only care about the existence of this bound.*

As $d = \delta' k^{2\deg(\mathcal{P})+1} - 1$ *and* δ' *converges to* 1 *(Lemma 7),* $k \geq 1.1214$ *ensures that* d *is large enough when the degree of a polynomial random walk program is at least* 1 *(that is, at least linear updates).*

Through this bound and Lemma 8, the stopping time of a polynomial random walk program $\mathcal{P}$ *with* $\deg(\mathcal{P}) \geq 1$ *is bounded:* $\mathbb{E}(T) \leq \sum_{n=0}^{\infty} Bn^{\frac{\ln(0.8872)}{\ln 1.1214+\tau}}$*. This stopping time bound has an exponent which is smaller than* -1.04*; therefore, the loop summation of the respective Figure 3.2 is finite and* $d_{min}(0.5) \leq 1$*.*

Using Theorem 2 we conclude that polynomial random walks with linearly (or faster) increasing step size and branching probability 0.5 *have finite expected stopping time and are PAST, given that* $\deg(\mathcal{P}) > \deg(p(q_1[n]) + (1-p)q_2[n])$*. In particular, this is true for Figure 1.1, as shown in Example 2, hence it is PAST.* □

Higher moments of the stopping time We conclude this chapter by noting that solving (3.1) and applying Theorem 2 allows us to derive not only PAST, but also higher moments of the stopping times of polynomial random walks $\mathcal{P}$. That is, the bound we compute for $\mathbb{P}(T \geq n)$ by solving (3.1) is of the form Bn^m and this bound can be used to bound higher moments N of the stopping time. In particular, $\mathbb{E}(T^N) = \sum_{n=0}^{\infty} \mathbb{P}(T^N \geq n) = \sum_{n=0}^{\infty} \mathbb{P}(T \geq \sqrt[N]{n}) \leq \sum_{n=0}^{\infty} Bn^{\frac{m}{N}}$. Therefore, when (3.1) is solved using a bound with $m < -N$, then $\mathbb{E}(T^N)$ is finite.

Table 3.1: Inductive bound for proving PAST of linear random walk (1)

$\vec{a}$	$\vec{b}$
0.00000000	0.11296053
0.02610488	0.12030958
0.05220976	0.12796144
0.07831464	0.13591606
0.10441952	0.14417252
0.13052440	0.15272904
0.15662928	0.16158298
0.18273416	0.17073082
0.20883904	0.18016817
0.23494392	0.18988980
0.26104880	0.19988960
0.28715368	0.21016063
0.31325856	0.22069516
0.33936344	0.23148466
0.36546832	0.24251983
0.39157320	0.25379067
0.41767808	0.26528648
0.44378296	0.27699594
0.46988784	0.28890709
0.49599272	0.30100746
0.52209760	0.31328405
0.54820248	0.32572342
0.57430736	0.33831174
0.60041224	0.35103482
0.62651712	0.36387817
0.65262200	0.37682710
0.67872688	0.38986672
0.70483176	0.40298200
0.73093664	0.41615786
0.75704152	0.42937920
0.78314640	0.44263094
0.80925128	0.45589810
0.83535616	0.46916584
0.86146104	0.48241946
0.88756592	0.49564453
0.91367080	0.50882687
0.93977568	0.52195260
0.96588056	0.53500818
0.99198544	0.54798048
1.01809032	0.56085673
1.04419520	0.57362465
1.07030008	0.58627239
1.09640495	0.59878861
1.12250983	0.61116247
1.14861471	0.62338366
1.17471959	0.63544241
1.20082447	0.64732951
1.22692935	0.65903632
1.25303423	0.67055477
1.27913911	0.68187737
1.30524399	0.69299720
1.33134887	0.70390794
1.35745375	0.71460384
1.38355863	0.72507972
1.40966351	0.73533098
1.43576839	0.74535360
1.46187327	0.75514407
1.48797815	0.76469948
1.51408303	0.77401740
1.54018791	0.78309596
1.56629279	0.79193376
1.59239767	0.80052993
1.61850255	0.80888403
1.64460743	0.81699609
1.67071231	0.82486659
1.69681719	0.83249642
1.72292207	0.83988686
1.74902695	0.84703957
1.77513183	0.85395658
1.80123671	0.86064024
1.82734159	0.86709325
1.85344647	0.87331856
1.87955135	0.87931943
1.90565623	0.88509937
1.93176111	0.89066212
1.95786599	0.89601164
1.98397087	0.90115207
2.01007575	0.90608773
2.03618063	0.91082312
2.06228551	0.91536286
2.08839039	0.91971167
2.11449527	0.92387440
2.14060015	0.92785598
2.16670503	0.93166140
2.19280991	0.93529570
2.21891479	0.93876397
2.24501967	0.94207130
2.27112455	0.94522282
2.29722943	0.94822363
2.32333431	0.95107883
2.34943919	0.95379348
2.37554407	0.95637261
2.40164895	0.95882119
2.42775383	0.96114415
2.45385871	0.96334635
2.47996359	0.96543255
2.50606847	0.96740745
2.53217335	0.96927568
2.55827823	0.97104173
2.58438311	0.97271002
2.61048799	0.97428488
2.63659287	0.97577049
2.66269775	0.97717096
2.68880263	0.97849027
2.71490751	0.97973227
2.74101239	0.98090071
2.76711727	0.98199922
2.79322215	0.98303130
2.81932703	0.98400034
2.84543191	0.98490961
2.87153679	0.98576224
2.89764167	0.98656127
2.92374655	0.98730960
2.94985143	0.98801003
2.97595631	0.98866522
3.00206119	0.98927774
3.02816607	0.98985004
3.05427095	0.99038445
3.08037583	0.99088322
3.10648071	0.99134848

Table 3.2: Inductive bound for proving PAST of linear random walk (2)

$\vec{a}$	$\vec{b}$
3.13258559	0.99178224
3.15869047	0.99218643
3.18479535	0.99256290
3.21090023	0.99291337
3.23700511	0.99323951
3.26310998	0.99354286
3.28921486	0.99382491
3.31531974	0.99408706
3.34142462	0.99433062
3.36752950	0.99455684
3.39363438	0.99476689
3.41973926	0.99496188
3.44584414	0.99514284
3.47194902	0.99531075
3.49805390	0.99546653
3.52415878	0.99561103
3.55026366	0.99574506
3.57636854	0.99586937
3.60247342	0.99598467
3.62857830	0.99609162
3.65468318	0.99619082
3.68078806	0.99628285
3.70689294	0.99636825
3.73299782	0.99644751
3.75910270	0.99652109
3.78520758	0.99658943
3.81131246	0.99665292
3.83741734	0.99671194
3.86352222	0.99676682
3.88962710	0.99681789
3.91573198	0.99686545
3.94183686	0.99690976
3.96794174	0.99695110
3.99404662	0.99698968
4.02015150	0.99702574
4.04625638	0.99705948
4.07236126	0.99709109
4.09846614	0.99712074
4.12457102	0.99714861
4.15067590	0.99717484
4.17678078	0.99719959
4.20288566	0.99722298
4.22899054	0.99724515
4.25509542	0.99726622
4.28120030	0.99728630
4.30730518	0.99730550
4.33341006	0.99732393
4.35951494	0.99734168
4.38561982	0.99735885
4.41172470	0.99737553
4.43782958	0.99739181
4.46393446	0.99740777
4.49003934	0.99742349
4.51614422	0.99743905
4.54224910	0.99745452
4.56835398	0.99746999
4.59445886	0.99748552
4.62056374	0.99750118
4.64666862	0.99751703
4.67277350	0.99753314
4.69887838	0.99754957
4.72498326	0.99756637
4.75108814	0.99758359
4.77719302	0.99760130
4.80329790	0.99761953
4.82940278	0.99763833
4.85550766	0.99765773
4.88161254	0.99767776
4.90771742	0.99769847
4.93382230	0.99771986
4.95992718	0.99774197
4.98603206	0.99776480

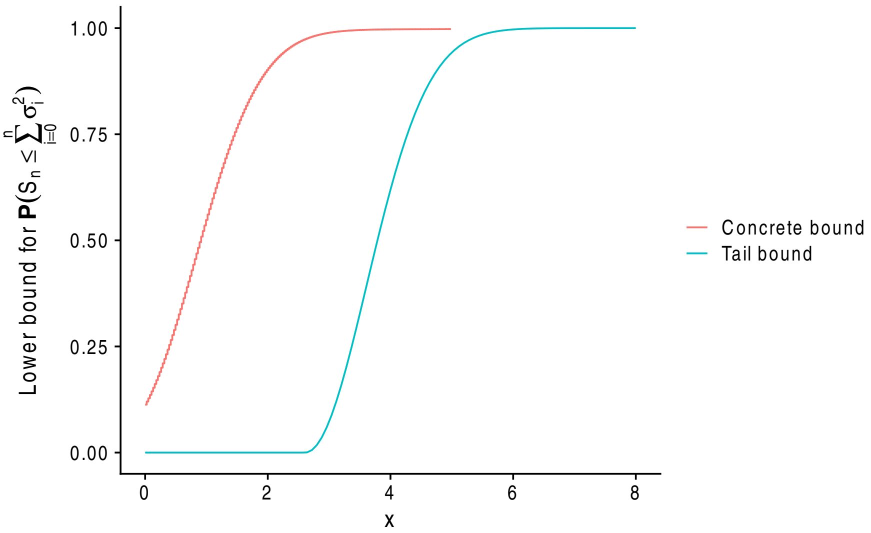


Figure 3.3: Bound for proving PAST of linear random walk

CHAPTER 4

Implementation and Experiments

Theorem 2 states sufficient conditions under which the polynomial random walk programs $\mathcal{P}$ of Figure 3.1 are PAST. These sufficient conditions can be checked by solving inequalities among random walk updates and, importantly, by finding solutions to the optimization problem of (3.1). In this chapter, we detail our implementation to find tight bounds to (3.1), allowing us to conclude PAST of $\mathcal{P}$. Our implementation involves heuristic optimization techniques to find provably correct solutions. Our experiments provide practical evidence on the tightness of computed stopping times bounds and give evidence of the reliability of our approach, despite the absence of convergence guarantees.

4.1 Computing Tight Bounds on Stopping Times

We solve (3.1) in extension of the `Polar` program analyzer [10]. We use `Polar` to compute closed form expressions for the loop-guard changes of probabilistic branches, allowing us to support programs $\mathcal{P}$ that are even more general than Figure 3.1. We combine `Polar` with linear programming through `OR-Tools` [11] and derive inductive bounds for fixed program transformation parameters. To find the best values for these parameters, we rely on genetic algorithms, such that the fitness functions of these genetic algorithms are controlled by our linear solver. Doing so, we use the `Gurobi`-solver [12] to solve linear models. By integrating algebraic reasoning, linear programming and genetic algorithms, our implementation in `Polar` minimizes the exponent in the bound of $\mathbb{P}(T \geq n)$ in (3.1), which is sufficient to prove PAST and finiteness of further higher moments of $\mathcal{P}$ (Theorem 2). By changing the objective function, our implementation can also minimize an *explicit* bound for the expected stopping time $\mathbb{E}(T)$.

Inferring inductive bound sets. To compute an inductive bound set B in (2.1), the parameters $\epsilon, d, c_0, C_1, \delta_1$ must be fixed. Additionally, we require vectors $\vec{a}_B$ and $\vec{c}_B$, specifying respectively which m lower bounds for the inductive bound-set are computed and which k tail bounds are used. We compute values for the bounds by solving the linear inequality:

$$\vec{b}_i \quad \geq \quad \sum_{j=1}^{m} \vec{d}_j \left(\Phi\left(\frac{\vec{a}_i\sqrt{1+d}-\vec{a}_j}{\sqrt{d}} \right) - c_0 \right) + \sum_{j=1}^{k} \vec{d}_{m+j} \left(\Phi\left(\frac{\vec{a}_i\sqrt{1+d}-\vec{c}_j}{\sqrt{d}} \right) - c_0 \right) \tag{4.1}$$

for $i = 1, \ldots, m$. The vector $\vec{d}$ denotes auxiliary variables, which describe the difference of neighbouring bounds[1]. Additionally, we enforce that the initial, almost Normal, distribution of Z_0 satisfies the bound set B.

Our implementation invokes linear programming over the linear model (4.1) in the form of an indicator function. This function returns 1, when an inductive bound set B is found for the given parameters $\epsilon, d, c_0, C_1, \delta_1$; and 0 otherwise.

Genetic algorithm. We use a genetic algorithm to solve the optimization problem (3.1) and find the best parameter values in (4.1), for which an inductive bound set B exists. Our genetic algorithm repeatedly modifies a collection of individual solutions: we select individuals from the current set of solutions and use them to produce next individuals/solutions. An individual has (i) the properties d, ϵ, and n_0 to capture the program transformation of Figure 3.2 and (ii) the parameters g, s, c to specify the vectors $\vec{a}$ and $\vec{c}$ of (4.1) for the inductive bound B. Specifically, we set $\vec{a}_1 = 0, \vec{a}_2 = \frac{s}{g-1}, \ldots, \vec{a}_g = s$, and $\vec{c} = \begin{pmatrix} c \end{pmatrix}$.

The fitness of an individual is calculated by first calculating the exponent of the bound. In case an explicit bound should be computed, the error-terms c_0 and δ_1 are inferred from n_0. Otherwise, we choose very small values, such as $c_0 = \delta_1 = 10^{-8}$, $\delta' = 1 + 10^{-8}$. Next, we solve our linear model (4.1). If no solution is found, we set $m = 0$; otherwise, we take $m = \frac{\ln(1-\epsilon)}{\ln(k+\frac{1}{n_0})}$ with $k = \left(\frac{(d+1)}{\delta'}\right)^{\frac{1}{2\deg(\mathcal{P})+1}}$. If $m < -1$ and an explicit bound is sought, we compute the summation $\mathbb{E}(T) = \sum_{n=1}^{\infty} \mathbb{P}(T \geq n)$ using the Hurwitz ζ-function [2]. The fitness of an individual is further expressed via the tuple $(\mathbb{E}(T), m)$, which is minimized/optimized with respect to the usual lexicographical ordering.

From a given solution set (generation), a new solution set (population) is generated using random mutations of the parameters. The property d is biased to decrease, while ϵ is biased to increase. Additionally, n_0 is biased to increase when the exponent is not smaller than -1, and biased to decrease otherwise. Furthermore, new individuals are generated by randomly selecting the properties of two parent individuals.

4.2 Experimental Results

We evaluated our approach for computing stopping time bounds, and hence, inferring PAST, using various polynomial random walk programs $\mathcal{P}$. To this end, we took

[1] see Section 2.3

[2] (25.11)NIST:DLMF

Table 4.1: Derived bounds on stopping times for polynomial random walk programs $\mathcal{P}$, with increasing maximal degree $\deg P$ and different probabilistic choices p. The program in the 3rd line of the table corresponds to Figure 1.1.

$\deg(\mathcal{P})$	p	measured exponent	tightest bound
0.25	0.5	-0.744	-0.5589
0.5	0.5	-0.997	-0.7436
1	0.5	-1.508	-1.1189
2	0.5	-2.442	-1.8639
5	0.5	-4.588	-4.0843
3	0.5	-3.448	-2.5971
3	0.9	-3.334	-2.4453
3	0.1	-3.57	-2.4453
3	0.99	-3.152	-1.9321
3	0.01	-3.516	-1.9321
3	0.999	-3.144	-1.359
3	0.001	-3.562	-1.359

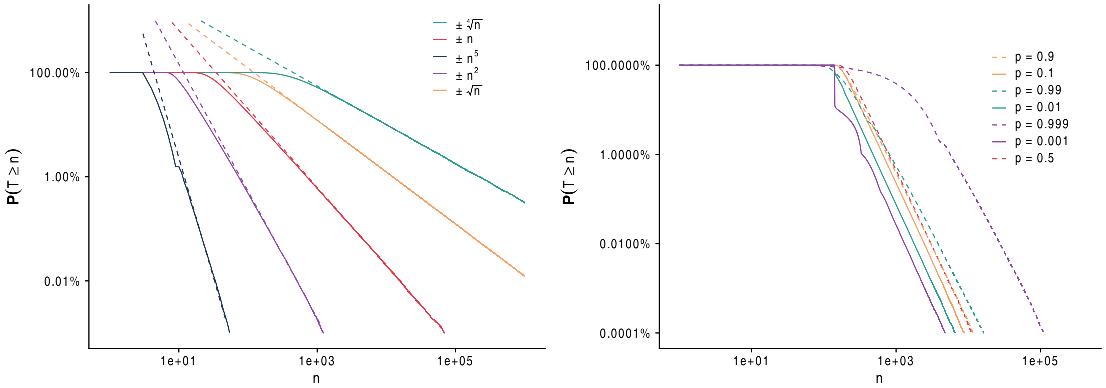
instances of Figure 3.1 with different random walk degrees $\deg P$ and various values of the probabilistic choice p. The PAST analysis of such programs is out of reach of existing tools (see Chapter 5), notably `Amber` [17], `eco-imp` [18], `KoAT` [19], `LexRsm` [6], and `LazyLexRsm` [20].

Table 4.1 summarizes our experiments, with the third line of Table 4.1 being our motivating example from Figure 1.1. Column 3 of Table 4.1 reports the empirical exponents of $\mathbb{P}(T \leq n)$, further detailed in Figure 4.1a. Column 4 of Table 4.1 states the smallest (tightest) exponent obtained through our approach using inductive bounds. Our experiments were run on a machine with 2x AMD EPYC 7502 32-Core processor with one task per core and hyperthreading disabled.

4.2.1 Experimental Analysis

Figure 4.1a displays empirically measured rates of $\mathbb{P}(T \geq n)$ for symmetric random walks with varying degree. These probabilities appear to converge towards a line in the log-log plot, which suggests, that $\mathbb{P}(T \geq n)$ eventually is of form Bn^m, coinciding with the form of our bound. The observed exponent of this probability is the slope of the robust log-log regression lines [21], displayed as dashed lines and displayed in Column 3 of Table 4.1.

In Figure 4.1b we display the stopping times $\mathbb{P}(T \geq n)$ of zero-mean polynomial random walks with different values of p and degree 3. The approximated values of the exponent, as well as the tightest bound found by our method can be found in Table 4.1. The increasing unsharpness for small (or large) values of p stem from the use of Hoeffding's lemma in the proof of Lemma 6. While for individual bounds of centered X_i this bound is sharp, for the product this no longer is the case and the plot suggests, that a tighter bound might be found.



(a) with different degrees

(b) with different percentages

Figure 4.1: Empirical results on stopping times of polynomial random walks

Table 4.2: Explicit bounds and empirical means of stopping time, with various polynomial updates q_1, q_2 and probabilistic choice p of polynomial random walks from Figure 3.1

id	q_1	p	q_2	y_0	empiric $\mathbb{E}(T)$	explicit bound
1	n	0.5	$-n$	100	89.62	9562887
2	n^2	0.5	$-n^2$	100	36.4	17708
3	$n^2 + 2n + 20$	0.5	$-n^2 + 2n + 20$	1000	59.47	213570
4	$\frac{n^3}{0.99}$	0.99	$-\frac{n^3}{0.01}$	10^8	212.8	2671328

4.2.2 Explicit Bound Analysis

Our genetic algorithms can be used to compute explicit bounds on the running times. Figure 4.2 shows the tightest explicit bound found for some random walk programs. The explicit bound is off by several orders of magnitude, as listed in Table 4.2. One of the main reasons for the explicit bound being unsharp stems from the fact that we choose a specific n_0 in (3.1), from which the parameters for the inductive bounds B are computed. This could be improved by computing a bound with multiple segments, and therefore inferring multiple exponents that decrease with growing n.

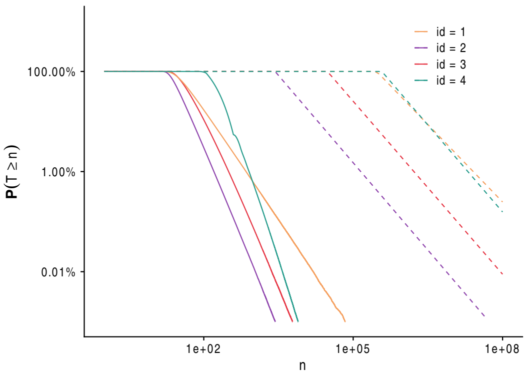
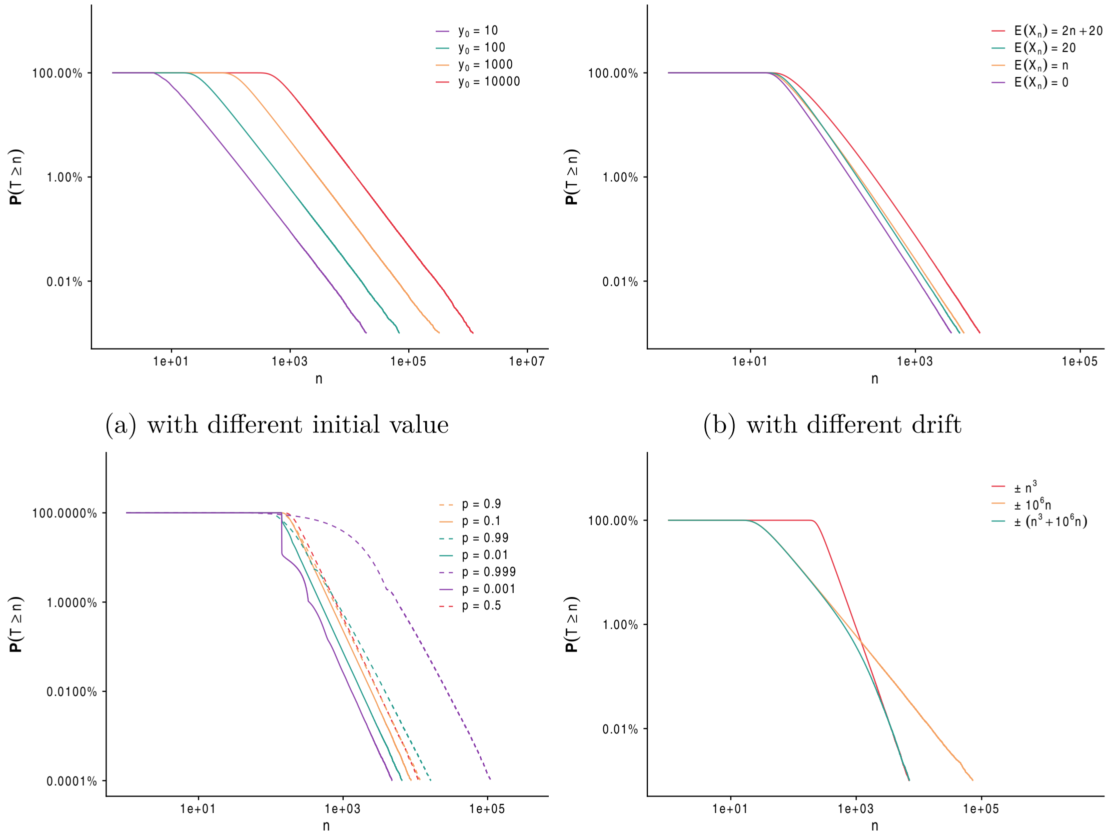


Figure 4.2: Examples of explicit bounds



(a) with different initial value

(b) with different drift

(c) with different percentages

(d) with change of slope

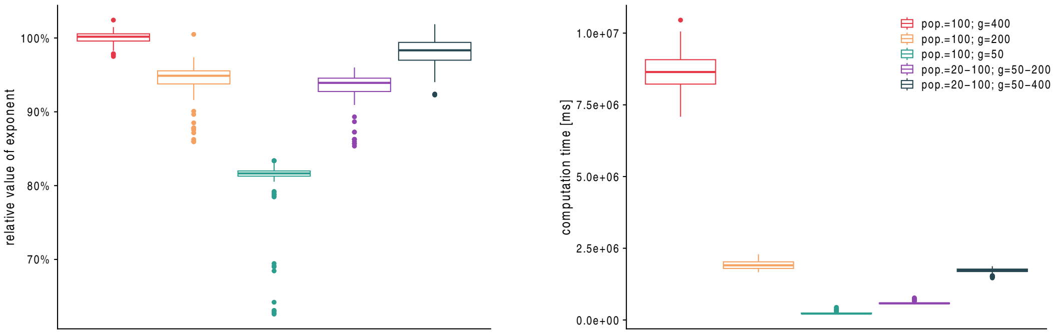
Figure 4.3: Empirical results on stopping times of polynomial random walks

4.2.3 Relation of Converging Terms and Empirical Stopping Time

While the relation of the degree of the polynomials and the slope of the empirically measured $\mathbb{P}(T \geq n)$ has been studied in Section 4.2, the effects of other properties of the polynomials, which cause the terms in the proofs to converge slower, can also be seen in the same kind of plot.

Different initial values are shown in Figure 4.3a, and show that termination happens later, but $\mathbb{P}(T \geq n)$ decays with the same exponent. In our implementation this is accounted for through c_0 (for U_0) and δ_1 in Lemma 6. Figure 4.3b shows random walks with $\pm n^2$ as leading term, and a (strictly positive) polynomial, which is the "drift" of the random walk. Even the random walk with the high linear drift seems to eventually converge to a line with the same slope as the other random walks. This effect is covered through the same constants.

The influence of different values of p is, in addition to C in the tail-bound, covered through c_0 in Lemma 5. Our method computes the bound based on the value of $p(1-p)$



(a) solution quality

(b) computation time

Figure 4.4: Performance of different genetic algorithm configurations

(e.g. the bound for $p = 0.1$ and $p = 0.9$ is the same), since the absolute deviation is captured, and small values of p cause a higher deviation, albeit in the negative,"favorable" direction. This unsharpness can be clearly seen in Figure 4.3c.

A high non-leading term influences the speed of convergence of δ' in Lemma 7. The empirical stopping time then first resembles that of the non-leading term, while later converging towards that of the leading term, as displayed in Figure 4.3d. This also supports the claim, that the explicit stopping time could better be approximated, if multiple segments were used.

4.2.4 Performance of Genetic Algorithm

For the genetic algorithm, higher values of g increase the size of the linear model and hence the computation time. To mitigate this effect, g is chosen relatively small in the beginning, but its value increases with each generation. Similarly, the population size shrinks. Usually, higher values of g allow to decrease d and increase ϵ slightly. Intuitively, the algorithm tries to first find "good values" for the parameters, which in later generations are then improved further. Figure 4.4 shows the relative performance and the running time for different specifications after 50 generations, where the population size and the granularity g are either set to a fixed value, or set to linearly decrease and increase respectively. Figure 4.4a shows that (i) the results of our method are relatively reliably and (ii) changing the parameters over time causes the the absolute value of the exponent of the bound to be slightly lower, but reduces the running time drastically, as displayed in Figure 4.4b. This gives reason to making a population, linearly decreasing from 100 to 20, and a granularity linearly increasing from 50 to 200 the default values in our implementation.

CHAPTER 5

Discussion

Related Work. Reasoning about probabilistic program termination is much harder than for deterministic programs [2], turning the automated analysis of probabilistic loops into a challenging problem. Most approaches rely on proof rules for proving (positive) almost-sure termination [22, 3, 23, 24], which in turn require additional expressions, notably loop invariants and martingale variants, for the applicability in the proof rules. As the generation of invariants and martingales is undecidable in general, automation of these approaches requires user-provided invariants/martingales. Our work is limited to polynomial random walks with the benefit of providing sufficient conditions under which PAST can automatically be inferred. While restrictive, Figure 1.1 shows advantages of our approach: proving PAST of this program using proof rules from [22, 3, 23] requires an auxiliary ranking super-martingale, whose computability is still an open question.

Alternative approaches to automating termination analysis have been proposed by focusing on restrictive classes of probabilistic loops, whose (P)AST analysis becomes (semi-)decidable [25, 17]. Notably, constant probability loops [25] limit probabilistic loop updates to constant increments over random variable and their (P)AST is decidable. A more expressive class of programs is given in [17], with (P)AST analysis shown to be semi-decidable and automated. Key to automation is the ability to inferring (ranking) super-martingales from loop guards and relaxing proof rules to "eventual" reasoning over polynomial loop updates. Our approach complements these works by using arbitrary polynomial updates in polynomial random walks. Such loops cannot be analyzed by [25, 17]; in particular, PAST of Figure 1.1 cannot be inferred.

The analysis of probabilistic programs with arbitrary polynomial updates and control flow is shown to be difficult, especially due to the lack of compositionality [26]. By adjustments of the weakest precondition [27, 28] calculus, runtime bounds are inferred as sufficient conditions for proving PAST in [29, 18]. Control-flow refinement methods are also advocated in [19] to derive runtime bounds on probabilistic loops. Further, lexicographical extensions of synthesizing ranking super-martingales are presented in [6, 20] for the purpose of PAST inference. While powerful, automation of these works depend on the suitable martingales. Unlike our technique, proving PAST of polynomial random walks, in particular of Figure 1.1, cannot yet be achieved by other works.

Conclusion. We study the positive almost-sure termination (PAST) problem of polynomial probabilistic programs implementing random walks with increasing increments. We show that PAST can be proven for polynomial random walks by checking conditions via solving linear inequalities over the polynomial program updates, without requiring additional user input in the form of invariants and/or martingales. Our experiments demonstrate that our approach determines PAST of non-trivial probabilistic programs. Notably, we show PAST for programs beyond the scope of existing methods: for such programs, state-of-the-art works would require ranking super-martingales whose computation is undecidable in general. For such loops, we prove PAST by finding bounds on the probability of termination, depending on the degrees and the branching probability of the polynomial updates. Future work includes the extension of our results to (i) deriving hardness results on PAST decidability for polynomial random walks, and (ii) dealing with probabilistic programs with nondeterminism and more complex updates.

Overview of Generative AI Tools Used

No generative AI tools were used while composing this thesis.

List of Figures

List of Tables

Bibliography

[1] D. Kozen, "A probabilistic PDL," *J. Comput. Syst. Sci.*, 1985.

[2] M. Hark, B. L. Kaminski, J. Giesl, and J. Katoen, "Aiming low is harder: induction for lower bounds in probabilistic program verification," *Proc. ACM Program. Lang.*, vol. 4, no. POPL, pp. 37:1–37:28, 2020.

[3] A. Chakarov and S. Sankaranarayanan, "Probabilistic program analysis with martingales," in *CAV*, 2013, pp. 511–526.

[4] L. M. F. Fioriti and H. Hermanns, "Probabilistic termination: Soundness, completeness, and compositionality," in *POPL*. ACM, 2015, pp. 489–501.

[5] R. Majumdar and V. R. Sathiyanarayana, "Sound and complete proof rules for probabilistic termination," *Proc. ACM Program. Lang.*, vol. 9, no. POPL, pp. 1871–1902, 2025.

[6] S. Agrawal, K. Chatterjee, and P. Novotný, "Lexicographic ranking supermartingales: an efficient approach to termination of probabilistic programs," *Proc. ACM Program. Lang.*, vol. 2, no. POPL, pp. 34:1–34:32, 2018.

[7] O. Bournez and F. Garnier, "Proving positive almost-sure termination," in *RTA*, 2005, pp. 323–337.

[8] K. Chatterjee, H. Fu, and A. K. Goharshady, "Termination analysis of probabilistic programs through positivstellensatz's," in *CAV*, 2016, pp. 3–22.

[9] M. Harchol-Balter, *Introduction to Probability for Computing.* Cambridge University Press, 2023.

[10] M. Moosbrugger, M. Stankovic, E. Bartocci, and L. Kovács, "This is the moment for probabilistic loops," *Proc. ACM Program. Lang.*, vol. 6, no. OOPSLA2, pp. 1497–1525, 2022.

[11] L. Perron and V. Furnon, "Or-tools," Google. [Online]. Available: https://developers.google.com/optimization/

[12] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2025. [Online]. Available: https://www.gurobi.com

[13] R. Durrett, *Probability: Theory and Examples.* Cambridge University Press, 2019.

[14] T. Lattimore and C. Szepesvári, *Bandit algorithms.* Cambridge University Press, 2020.

[15] I. Tyurin, "A refinement of the remainder in the lyapunov theorem," *Theory of Probability & Its Applications*, vol. 56, no. 4, pp. 693–696, 2012.

[16] P. Massart, *Concentration inequalities and model selection: Ecole d'Eté de Probabilités de Saint-Flour XXXIII-2003.* Springer, 2007.

[17] M. Moosbrugger, E. Bartocci, J.-P. Katoen, and L. Kovács, "The probabilistic termination tool Amber," in *FM*, 2021, pp. 667–675.

[18] M. Avanzini, G. Moser, and M. Schaper, "A modular cost analysis for probabilistic programs," *Proc. ACM Program. Lang.*, vol. 4, no. OOPSLA, 2020.

[19] N. Lommen, É. Meyer, and J. Giesl, "Control-flow refinement for complexity analysis of probabilistic programs in KoAT (short paper)," in *IJCAR*, 2024, pp. 233–243.

[20] T. Takisaka, L. Zhang, C. Wang, and J. Liu, "Lexicographic ranking supermartingales with lazy lower bounds," in *CAV*, 2024, pp. 420–442.

[21] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. New York: Springer, 2002. [Online]. Available: https://www.stats.ox.ac.uk/pub/MASS4/

[22] R. Majumdar and V. R. Sathiyanarayana, "Positive almost-sure termination: Complexity and proof rules," *Proc. ACM Program. Lang.*, vol. 8, no. POPL, Jan. 2024.

[23] K. Chatterjee, P. Novotný, and D. Žikelić, "Stochastic invariants for probabilistic termination," in *POPL*, 2017, p. 145–160.

[24] F. Meyer, M. Hark, and J. Giesl, "Inferring expected runtimes of probabilistic integer programs using expected sizes," in *TACAS*, 2021, pp. 250–269.

[25] J. Giesl, P. Giesl, and M. Hark, "Computing expected runtimes for constant probability programs," in *Automated Deduction – CADE 27*, P. Fontaine, Ed. Cham: Springer International Publishing, 2019, pp. 269–286.

[26] B. L. Kaminski, J.-P. Katoen, C. Matheja, and F. Olmedo, "Weakest precondition reasoning for expected runtimes of randomized algorithms," *J. ACM*, vol. 65, no. 5, Aug. 2018.

[27] A. McIver, C. Morgan, B. L. Kaminski, and J. Katoen, "A new proof rule for almost-sure termination," *Proc. ACM Program. Lang.*, vol. 2, no. POPL, pp. 33:1–33:28, 2018.

[28] A. McIver and C. Morgan, *Abstraction, Refinement and Proof for Probabilistic Systems*. Springer, 2005.

[29] V. C. Ngo, Q. Carbonneaux, and J. Hoffmann, “Bounded expectations: resource analysis for probabilistic programs,” in *PLDI*, 2018, p. 496–512.