Aiming Low Is Harder

Induction for Lower Bounds in Probabilistic Program Verification

MARCEL HARK, RWTH Aachen University, Germany
BENJAMIN LUCIEN KAMINSKI, RWTH Aachen University, Germany
JÜRGEN GIESL, RWTH Aachen University, Germany
JOOST-PIETER KATOEN, RWTH Aachen University, Germany

We present a new inductive rule for verifying lower bounds on expected values of random variables after execution of probabilistic loops as well as on their expected runtimes. Our rule is *simple* in the sense that loop body semantics need to be applied only finitely often in order to verify that the candidates are indeed lower bounds. In particular, it is not necessary to find the limit of a sequence as in many previous rules.

CCS Concepts: • **Mathematics of computing** → **Probabilistic algorithms**; *Markov processes*; • **Theory of computation** → Denotational semantics.

Additional Key Words and Phrases: probabilistic programs, verification, weakest precondition, weakest preexpectation, lower bounds, optional stopping theorem, uniform integrability

ACM Reference Format:
Marcel Hark, Benjamin Lucien Kaminski, Jürgen Giesl, and Joost-Pieter Katoen. 2020. Aiming Low Is Harder: Induction for Lower Bounds in Probabilistic Program Verification. *Proc. ACM Program. Lang.* 4, POPL, Article 37 (January 2020), 28 pages. https://doi.org/10.1145/3371105

1 INTRODUCTION AND OVERVIEW

We study probabilistic programs featuring discrete probabilistic choices as well as *unbounded loops*. Randomized algorithms are the classical application of such programs. Recently, applications in *biology*, *quantum computing*, *cyber security*, *machine learning*, and *artificial intelligence* led to rapidly growing interest in probabilistic programming [Gordon et al. 2014].

Formal verification of probabilistic programs is strictly harder than for nonprobabilistic programs [Kaminski et al. 2019]. Given a random variable f, a key verification task is to reason about the *expected value of* f after termination of a program C on input s. If f is the indicator function of an event A, then this expected value is the probability that A has occurred on termination of C.

For verifying probabilistic loops, most approaches share a common, *conceptually very simple*, technique: an *induction rule* for verifying *upper bounds* on expected values, which are characterized as least fixed points (lfp) of a suitable function Φ. This rule, called "Park induction", reads

$$\Phi(I) \sqsubseteq I \qquad \text{implies} \qquad \text{lfp } \Phi \sqsubseteq I\,,$$

i.e., for a candidate upper bound I we check $\Phi(I) \sqsubseteq I$ (for a suitable partial order $\sqsubseteq$) to prove that I is indeed an upper bound on the least fixed point, and hence on the sought–after expected value.

Authors' addresses: Marcel Hark, RWTH Aachen University, Germany, marcel.hark@cs.rwth-aachen.de; Benjamin Lucien Kaminski, RWTH Aachen University, Germany, benjamin.kaminski@cs.rwth-aachen.de; Jürgen Giesl, RWTH Aachen University, Germany, giesl@cs.rwth-aachen.de; Joost-Pieter Katoen, RWTH Aachen University, Germany, katoen@cs.rwth-aachen.de.

2475-1421/2020/1-ART37
https://doi.org/10.1145/3371105

For *lower bounds*, a simple proof principle analogous to Park induction, namely

$$I \sqsubseteq \Phi(I) \qquad \text{implies} \qquad I \sqsubseteq \mathsf{lfp}\ \Phi\,, \qquad \lightning$$

is *unsound* in general. *Sound* rules (see Sect. 9), on the other hand, often suffer from the fact that either f needs to be *bounded*, or that one has to find the *limit of some sequence*, as well as the sequence itself, rendering those rules conceptually much more involved than Park induction.

Our main contribution (Sect. 5, Thm. 37) is to provide relatively *simple* side conditions that can be added to the (unsound) implication above, such that the implication becomes true, i.e.,

$$I \sqsubseteq \Phi(I) \ \wedge \ \begin{matrix}\text{some side}\\ \text{conditions}\end{matrix} \qquad \text{implies} \qquad I \sqsubseteq \mathsf{lfp}\ \Phi\,. \qquad \checkmark$$

In particular, our side conditions will be simple in the sense that (a variation of) Φ needs to be applied to a candidate I only a *finite* number of times, which is beneficial for potential *automation*.

The need for verifying lower bounds on expected values is quite natural: First of all, they help to assess the quality and tightness of upper bounds. Moreover, giving *total correctness* guarantees for probabilistic programs amounts to lower–bounding the correctness probability, e.g., in order to establish membership in complexity classes like RP and PP.

In addition to expected values of random variables at program termination, lower bounds on expected runtimes are also of significant interest: Lower bounds on expected runtimes which depend on secret program variables may compromise the secret, thus allowing for timing side–channel attacks; "very large" lower bounds could indicate potential denial–of–service attacks.

In order to enable practicable reasoning about lower bounds on expected runtimes, we will show how our inductive lower bound rule carries over to expected runtimes (Sect. 8, Thm. 46). As an example to show the applicability of our rule, we will verify that the well–known and notoriously difficult coupon collector's problem [Motwani and Raghavan 1995] (Sect. 8, Ex. 47), modeled by the probabilistic program[1]

$$\begin{aligned}
&x := N \,;\\
&\texttt{while}\,(\,0 < x\,)\,\{\\
&\qquad i := N + 1 \,;\\
&\qquad \texttt{while}\,(\,x < i\,)\,\{\, i := \mathrm{Unif}[1..N]\,\}\,;\\
&\qquad x := x - 1\\
&\}\,,
\end{aligned}$$

has an expected runtime of at least $N\mathcal{H}_N$, where $\mathcal{H}_N$ is the N-th harmonic number.

Our new inductive rules will be stated in terms of so–called expectation transformers [McIver and Morgan 2005] (Sect. 2) and rely on the notions of *uniform integrability* (Sect. 3, in particular 3.4, and Sect. 4), *martingales*, *conditional difference boundedness*, and the *Optional Stopping Theorem* (Sect. 5) from the theory of stochastic processes. However, we do not only *make use* of these notions in order to prove soundness of our induction rule, but instead establish tight connections in terms of these notions between expectation transformers and certain canonical stochastic processes (Sect. 4, Thm. 25 and Sect. 5, Thm. 36). In particular, we will build upon the key result of this connection (Thm. 25) to study exactly how inductive proof rules for both upper and lower bounds can be understood in the realm of these stochastic processes and vice versa (Sect. 5, Thm. 37 and Sect. 7). We see those connections between the theories of expectation transformers and stochastic processes

[1]The random assignment $i := \mathrm{Unif}[1..N]$ does not — strictly speaking — adhere to our syntax of binary probabilistic choices, but it can be modeled in our syntax. For the sake of readability, we opted for $i := \mathrm{Unif}[1..N]$.

as a stepping stone for applying further results from stochastic process theory to probabilistic program analysis and possibly also vice versa.

As a final contribution, we revisit one of the few existing rules for lower bounds due to [McIver and Morgan 2005], which gives *sufficient* criteria for a candidate being a lower bound on the expected value of a *bounded* function f. We show that their rule is also a consequence of uniform integrability and we are moreover able to generalize their rule to a *necessary* and sufficient criterion (Sect. 6, Thm. 41). We demonstrate the usability of our generalization by an example (Sect. 6, Ex. 42).

We refer to [Hark et al. 2019] for more case studies illustrating the effectiveness of our lower bound proof rule, a more detailed introduction to probability theory, and more detailed proofs of our results.

2 WEAKEST PREEXPECTATION REASONING

Weakest preexpectations for probabilistic programs are a generalization of Dijkstra's *weakest preconditions* for nonprobabilistic programs. Dijkstra employs *predicate transformers*, which push a *postcondition* F (a predicate) backward through a nonprobabilistic program C and yield the *weakest precondition* G (another predicate) describing the largest set of states such that whenever C is started in a state satisfying G, C terminates in a state satisfying F.[2]

The *weakest preexpecation calculus* on the other hand employs *expectation transformers* which act on real–valued functions called *expectations*, mapping program states to non–negative reals.[3] These transformers push a *postexpectation* f backward through a probabilistic program C and yield a *preexpectation* g, such that g represents the expected value of f after executing C. The term *expectation* coined by [McIver and Morgan 2005] may appear somewhat misleading at first. We *clearly distinguish between expectations and expected values*: An expectation is hence not an expected value, per se. Instead, we can think of an expectation as a *random variable*. In Bayesian network jargon, expectations are also called *factors*.

Definition 1 (Expectations [Kaminski 2019; McIver and Morgan 2005]**).** *Let* Vars *denote the finite set of program variables and let* $\Sigma = \{s \mid s\colon \mathsf{Vars} \to \mathbb{Q}\}$ *denote the set of program states.*[4]

The set of expectations, denoted by $\mathbb{F}$*, is defined as*

$$\mathbb{F} \;=\; \left\{ f \;\middle|\; f\colon \Sigma \to \overline{\mathbb{R}}_{\geq 0} \right\},$$

where $\overline{\mathbb{R}}_{\geq 0} = \{\, r \in \mathbb{R} \mid r \geq 0 \,\} \cup \{\infty\}$*. We say that* $f \in \mathbb{F}$ *is* finite *and write* $f \ll \infty$*, if* $f(s) < \infty$ *for all* $s \in \Sigma$*. A partial order* $\preceq$ *on* $\mathbb{F}$ *is obtained by point–wise lifting the usual order* $\leq$ *on* $\overline{\mathbb{R}}_{\geq 0}$*, i.e.,*

$$f_1 \;\preceq\; f_2 \quad \text{iff} \quad \forall s \in \Sigma\colon f_1(s) \;\leq\; f_2(s)\,.$$

$(\mathbb{F}, \preceq)$ *is a complete lattice where suprema and infima are constructed point–wise.*

We note that our notion of expectations is more general than the one of McIver and Morgan: Their work builds almost exclusively on *bounded* expectations, i.e., non–negative real–valued functions which are bounded from above by some constant, whereas we allow *unbounded* expectations. As a result, we have that $(\mathbb{F}, \preceq)$ forms a complete lattice, whereas McIver and Morgan's space of bounded expectations does not.

[2] We consider total correctness, i.e., from any state satisfying the weakest precondition G, C definitely terminates.
[3] For simplicity of the presentation, we study the standard case of positive expectations. Mixed–sign expectations mapping to the *full* extended reals require much more technical machinery, see [Kaminski and Katoen 2017].
[4] We choose rationals to have some range of values at hand which are conveniently represented in a computer.

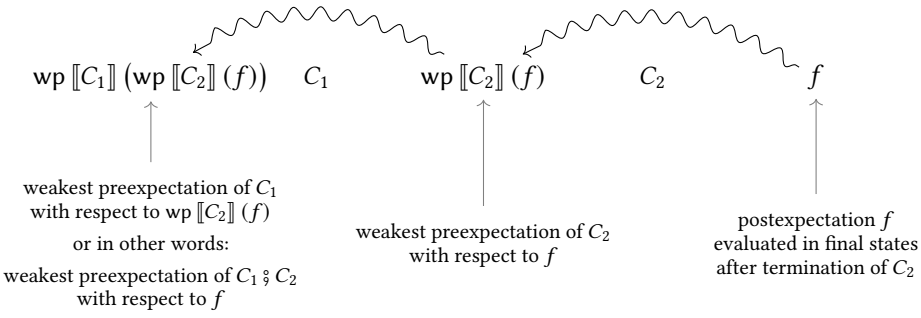


Fig. 1. Backward–moving continuation–passing style weakest preexpectation transformer.

2.1 Weakest Preexpectations

Given program C and postexpectation $f \in \mathbb{F}$, we are interested in the *expected value* of f evaluated in the final states reached after termination of C. More specifically, we are interested in a *function* $g\colon \Sigma \to \overline{\mathbb{R}}_{\geq 0}$ mapping each *initial* state s_0 of C to the respective expected value of f evaluated in the final states reached after termination of C on input s_0. This function g is called the *weakest preexpectation of* C *with respect to* f, denoted $\mathrm{wp}\,[\![C]\!]\,(f)$. Put as an equation, if ${}^{s_0}\mu_C$ is the probability (sub)measure[5] over final states reached after termination of C on initial state s_0, then[6]

$$g(s_0) \;=\; \mathrm{wp}\,[\![C]\!]\,(f)\,(s_0) \;=\; \int_{\Sigma} f \, d\,({}^{s_0}\mu_C)\;.$$

While $\mathrm{wp}\,[\![C]\!]\,(f)$ in fact represents an expected value, f itself does not. In an analogy to Dijkstra's pre– and postconditions, as f is evaluated in the final states after termination of C it is called the *postexpectation*, and as $\mathrm{wp}\,[\![C]\!]\,(f)$ is evaluated in the initial states of C it is called the *preexpectation*.

2.2 The Weakest Preexpectation Calculus

We now show how to determine weakest preexpectations in a systematic and *compositional* manner by recapitulating the *weakest preexpectation calculus* à la McIver and Morgan. This calculus employs expectation transformers which move backward through the program in a *continuation–passing* style, see Fig. 1. If we are interested in the expected value of some postexpectation f after executing the sequential composition $C_1 \fatsemi C_2$, then we can first determine the weakest preexpectation of C_2 with respect to f, i.e., $\mathrm{wp}\,[\![C_2]\!]\,(f)$. Thereafter, we can use the intermediate result $\mathrm{wp}\,[\![C_2]\!]\,(f)$ as *postexpectation* to determine the weakest preexpectation of C_1 with respect to $\mathrm{wp}\,[\![C_2]\!]\,(f)$. Overall, this gives the weakest preexpectation of $C_1 \fatsemi C_2$ with respect to the postexpectation f. The above explanation illustrates the compositional nature of the weakest preexpectation calculus. wp–transformers for all language constructs can be defined by induction on the program structure:

Definition 2 (The wp–Transformer [McIver and Morgan 2005]**).** *Let* pGCL *be the set of programs in the* probabilistic guarded command language. *Then the weakest preexpectation transformer*

$$\mathrm{wp}\colon \mathsf{pGCL} \to \mathbb{F} \to \mathbb{F}$$

[5] ${}^{s_0}\mu_C(s) \in [0,1]$ is the probability that s is the final state reached after termination of C on input s_0. We have $\sum_{s\in\Sigma} {}^{s_0}\mu_C(s) \leq 1$, where the "missing" probability mass is the probability of *nontermination* of C on s_0.
[6] As Σ is countable, the integral can be expressed as $\sum_{s\in\Sigma} {}^{s_0}\mu_C(s) \cdot f(s)$.

Table 1. Rules for the wp–transformer.

C	$\mathbf{wp}\,[\![C]\!]\,(f)$
`skip`	f
$b := e$	$f\,[b/e]$
`if` $(\varphi)\,\{C_1\}$ `else` $\{C_2\}$	$[\varphi]\cdot \mathsf{wp}\,[\![C_1]\!]\,(f) + [\neg\varphi]\cdot \mathsf{wp}\,[\![C_2]\!]\,(f)$
$\{C_1\}\,[p]\,\{C_2\}$	$p\cdot \mathsf{wp}\,[\![C_1]\!]\,(f) + (1-p)\cdot \mathsf{wp}\,[\![C_2]\!]\,(f)$
$C_1 \fatsemi C_2$	$\mathsf{wp}\,[\![C_1]\!]\,(\mathsf{wp}\,[\![C_2]\!]\,(f))$
`while` $(\varphi)\,\{C'\}$	$\mathsf{lfp}_{\ {}^{\mathsf{wp}}\langle C',\varphi\rangle}\Phi_f$

$$ {}^{\mathsf{wp}}_{\langle\varphi,C\rangle}\Phi_f(X) \;=\; [\neg\varphi]\cdot f + [\varphi]\cdot \mathsf{wp}\,[\![C]\!]\,(X) \qquad \text{characteristic function} $$

is defined according to the rules given in Table 1, *where* $[\varphi]$ *denotes the Iverson–bracket of* φ, *i.e.,* $[\varphi]\,(s)$ *evaluates to* 1 *if* $s \models \varphi$ *and to* 0 *otherwise. Moreover, for any variable* $b \in \mathsf{Vars}$ *and any expression* e, *let* $f\,[b/e]$ *be the expectation with* $f\,[b/e]\,(s) = f(s\,[b/e])$ *for any* $s \in \Sigma$, *where* $s\,[b/e]\,(b) = s(e)$ *and* $s\,[b/e]\,(x) = s(x)$ *for all* $x \in \mathsf{Vars} \setminus \{b\}$.

We call the function ${}^{\mathsf{wp}}_{\langle\varphi,C\rangle}\Phi_f$ *the* characteristic function *of* `while` $(\varphi)\,\{C\}$ *with respect to* f. *Its least fixed point is understood in terms of* $\preceq$. *To increase readability, we omit* wp, φ, C, *or* f *from* Φ *whenever they are clear from the context.*

Example 3 (Applying the wp Calculus). *Consider the probabilistic program* C *given by*

$$ \{\, b := b+5 \,\}\ [4/5]\ \{\, b := 10 \,\}\ . $$

Suppose we want to know the expected value of b *after execution of* C. *For this, we determine* $\mathsf{wp}\,[\![C]\!]\,(b)$. *Using the annotation style shown in* Fig. 2a, *we can annotate the program* C *as shown in* Fig. 2b, *using the rules from* Table 1. *At the top, we read off the weakest preexpectation of* C *with respect to* b, *namely* $\frac{4b}{5} + 6$. *This tells us that the expected value of* b *after termination of* C *on* s_0 *is equal to* $\frac{4\cdot s_0(b)}{5} + 6$.

The wp–transformer satisfies what is sometimes called *healthiness conditions* [Hino et al. 2016; Keimel 2015; McIver and Morgan 2005] or *homomorphism properties* [Back and von Wright 1998]:

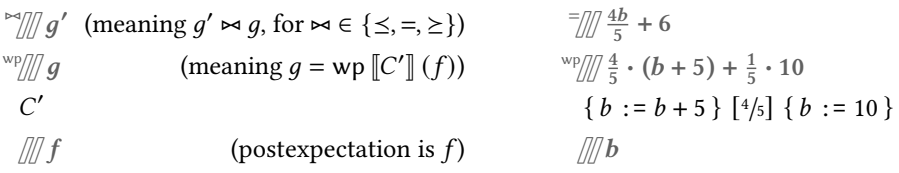
Theorem 4 (Healthiness Conditions [Kaminski 2019; McIver and Morgan 2005]**).** *Let* $C \in \mathsf{pGCL}$, $S \subseteq \mathbb{F}$, $f, g \in \mathbb{F}$, *and* $r \in \mathbb{R}_{\geq 0}$. *Then:*

(1) **Continuity:** $\mathsf{wp}\,[\![C]\!]\,(\sup\ S) \;=\; \sup\ \mathsf{wp}\,[\![C]\!]\,(S)$.
(2) **Strictness:**[7] $\mathsf{wp}\,[\![C]\!]$ *is strict, i.e.,* $\mathsf{wp}\,[\![C]\!]\,(0) = 0$.
(3) **Monotonicity:** $f \preceq g \quad\text{implies}\quad \mathsf{wp}\,[\![C]\!]\,(f) \;\preceq\; \mathsf{wp}\,[\![C]\!]\,(g)$.
(4) **Linearity:** $\mathsf{wp}\,[\![C]\!]\,(r\cdot f + g) \;=\; r\cdot \mathsf{wp}\,[\![C]\!]\,(f) + \mathsf{wp}\,[\![C]\!]\,(g)$.

3 BOUNDS ON WEAKEST PREEXPECTATIONS

For *loop–free* programs, it is generally straightforward to determine weakest preexpectations, simply by applying the rules in Table 1, which guide us along the syntax of C, see Ex. 3. Weakest

[7] Here, we overload notation and denote by 0 the constant expectation that maps every $s \in \Sigma$ to 0.



(a) Style for wp annotations. (b) wp annotations for Ex. 3.

Fig. 2. Annotations for weakest preexpectations. It is more intuitive to read these from the bottom to top.

preexpectations of loops, on the other hand, are generally non–computable least fixed points and we often have to content ourselves with some *approximation* of those fixed points.

For us, a sound approximation is either a *lower* or an *upper* bound on the least fixed point. There are in principle two challenges: (1) finding a candidate bound and (2) verifying that the candidate is indeed an upper or lower bound. In this paper, we study the *latter* problem.

3.1 Upper Bounds

The *Park induction* principle provides us with a very convenient proof rule for verifying upper bounds. In general, this principle reads as follows:

Theorem 5 (Park Induction [Park 1969]**).** *Let $(D, \sqsubseteq)$ be a complete lattice and let $\Phi\colon D \to D$ be continuous.*[8] *Then Φ has a least fixed point in D and for any $I \in D$,*

$$\Phi(I) \sqsubseteq I \qquad \text{implies} \qquad \mathsf{lfp}\ \Phi \sqsubseteq I\,.$$

In the realm of weakest preconditions, Park induction gives rise to the following induction principle:

Corollary 6 (Park Induction for wp [Kaminski 2019; Kozen 1985]**).** *Let Φ_f be the characteristic function of the while loop* `while` $(\varphi)\,\{C\}$ *with respect to postexpectation f and let $I \in \mathbb{F}$. Then*

$$\Phi_f(I) \preceq I \quad \text{implies} \quad \mathsf{wp}\,[\![\texttt{while}\,(\varphi)\,\{C\}]\!]\,(f) \preceq I\,.$$

We call an I that satisfies $\Phi_f(I) \preceq I$ a *superinvariant*. The striking power of Park induction is its simplicity: Once an appropriate candidate I is found (even though this is usually not an easy task), all we have to do is push it through the characteristic function Φ_f *once* and check whether we went down in our underlying partial order. If this is the case, we have verified that I is indeed an upper bound on the least fixed point and thus on the sought–after weakest preexpectation.

Example 7 (Induction for Upper Bounds). *Consider the program C_{geo}, given by*

$$\texttt{while}\,(a \neq 0)\,\{\,\{\,a := 0\,\}\ [1/2]\ \{\,b := b+1\,\}\,\}\,,$$

where we assume $b \in \mathbb{N}$. Suppose we aim at an upper bound on the expected value of b executing C_{geo}. Using the annotation style of Fig. 3a, we can annotate the loop C_{geo} as shown in Fig. 3b, using superinvariant $I = b + [a \neq 0]$, establishing $\Phi_b(I) \preceq I$, and by Cor. 6 establishing $\mathsf{wp}\,[\![C_{geo}]\!]\,(b) \preceq b + [a \neq 0]$. *So the expected value of b after termination of C_{geo} on s_0 is at most $s_0(b) + [s_0(a) \neq 0]$.*

For making a comparison to the lower bound case which we consider later, let us explain why Park induction is sound using the so–called *Tarski–Kantorovich Principle*:

[8] It would even suffice for Φ to be monotonic, but we consider continuous functions throughout this paper.

$\overset{\bowtie}{/\!/\!/}\, I$ (see Fig. 2a)

$\overset{\Phi}{/\!/\!/}\, g$ (meaning $g = [\neg\varphi]\cdot f + [\varphi]\cdot I''$)

`while ( φ ) {`

$\quad \overset{\bowtie}{/\!/\!/}\, I''$ (see Fig. 2a)

$\quad \overset{\text{wp}}{/\!/\!/}\, I'$ (see Fig. 2a)

$\quad$ *Body*

$\quad [\!]\!]\, I$ } (meaning we employ invariant I)

$/\!/\!/\, f$ (postexpectation of loop is f)

(a) Annotation style for loops using invariants.

$\overset{\leq}{/\!/\!/}\, b + [a \neq 0]$

$\overset{\Phi}{/\!/\!/}\, [a = 0]\cdot b \;+\; [a \neq 0]\cdot\Big(b + \tfrac{1}{2}\big(1 + [a \neq 0]\big)\Big)$

`while ( a ≠ 0 ) {`

$\quad \overset{=}{/\!/\!/}\, b \;+\; \tfrac{1}{2}\big(1 + [a \neq 0]\big)$

$\quad \overset{\text{wp}}{/\!/\!/}\, \tfrac{1}{2}\big(b + [0 \neq 0] \;+\; b + 1 + [a \neq 0]\big)$

$\quad \{\, a := 0 \,\}\ [1/2]\ \{\, b := b + 1 \,\}$

$\quad [\!]\!]\, b + [a \neq 0]$ }

$/\!/\!/\, b$

(b) wp loop annotations for Ex. 7.

Fig. 3. Annotation style for loops using invariants and annotations for Ex. 7. Inside the loop, we push an invariant I (provided externally, denoted by $[\!]\!]\, I$) through the loop body, thus obtaining I'' which is (possibly an over- or underapproximation of) $\text{wp}\,[\![\mathit{Body}]\!]\,(I)$. Above the loop head, we then annotate $g = [\neg\varphi]\cdot f + [\varphi]\cdot I''$. In the first line, we establish $g \bowtie I$, for $\bowtie \in \{\leq, \geq\}$. Note that if $\bowtie = \leq$, we have established the precondition of Cor. 6, since we have then overall established

$$\Phi_f(I) \;=\; [\neg\varphi]\cdot f + [\varphi]\cdot \text{wp}\,[\![\mathit{Body}]\!]\,(I) \;\leq\; [\neg\varphi]\cdot f + [\varphi]\cdot I'' \;=\; g \;\leq\; I$$

For reasoning about lower bounds, we will later employ $\bowtie = \geq$.

Theorem 8 (Tarski–Kantorovich Principle, cf. [Jachymski et al. 2000]**).** *Let $(D, \sqsubseteq)$ be a complete lattice, let $\Phi\colon D \to D$ be continuous, and let $I \in D$, such that $I \sqsupseteq \Phi(I)$. Then the sequence $I \sqsupseteq \Phi(I) \sqsupseteq \Phi^2(I) \sqsupseteq \Phi^3(I) \sqsupseteq \cdots$ is a* descending chain *that* converges *to an element*

$$\Phi^\omega(I) \;=\; \lim_{n\to\omega} \Phi^n(I) \;\in\; D\,,$$

which is a fixed point of Φ. In particular, $\Phi^\omega(I)$ is the greatest fixed point of Φ that is $\sqsubseteq I$.

Dually, now let $I \sqsubseteq \Phi(I)$. Then $I \sqsubseteq \Phi(I) \sqsubseteq \Phi^2(I) \sqsubseteq \Phi^3(I) \sqsubseteq \cdots$ is an ascending chain *that* converges *to a fixed point $\Phi^\omega(I) \in D$. Moreover, $\Phi^\omega(I)$ is the* least fixed point of Φ that is $\sqsupseteq I$.

The well–known Kleene Fixed Point Theorem (cf. [Lassez et al. 1982]), which states that $\mathsf{lfp}\,\Phi = \Phi^\omega(\bot)$, where $\bot$ is the least element of D, is a special case of the Tarski–Kantorovich Principle.

In our setting, applying the Tarski–Kantorovich principle to a superinvariant I, the iteration of Φ on I will yield some fixed point $\sqsubseteq I$ and this fixed point is necessarily $\sqsupseteq \mathsf{lfp}\,\Phi$.

3.2 Lower Bounds

For verifying lower bounds, we do not have a rule as simple as Park induction available. In particular, for a given complete lattice $(D, \sqsubseteq)$ and monotonic function $\Phi\colon D \to D$, the rule

$$I \;\sqsubseteq\; \Phi(I) \qquad \text{implies} \qquad I \;\sqsubseteq\; \mathsf{lfp}\;\Phi\,, \qquad \lightning$$

is *unsound* in general. We call an I satisfying $I \sqsubseteq \Phi(I)$ a *subinvariant* and the above rule *simple lower induction*. Generally, we will call an I that is a sub– or a superinvariant an *invariant*. I being an invariant thus expresses mainly its *inductive* nature, namely that I is comparable with $\Phi(I)$ with respect to the partial order $\sqsubseteq$.

An explanation why simple lower induction is unsound is as follows: By Thm. 8, we know from $I \sqsubseteq \Phi(I)$ that $\Phi^\omega(I)$ is the *least fixed point of Φ that is greater than or equal to I*. Since $\Phi^\omega(I)$ is a

fixed point, $\Phi^{\omega}(I) \sqsubseteq \mathsf{gfp}\,\Phi$ holds, but we do not know how I compares to $\mathsf{lfp}\,\Phi$. We only know that if indeed $I \sqsubseteq \mathsf{lfp}\,\Phi$ and $I \sqsubseteq \Phi(I)$, then iterating Φ on I also converges to $\mathsf{lfp}\,\Phi$, i.e.,

$$I \sqsubseteq \mathsf{lfp}\,\Phi \quad \text{and} \quad I \sqsubseteq \Phi(I) \qquad \text{implies} \qquad \Phi^{\omega}(I) = \mathsf{lfp}\,\Phi\,.$$

If, however, $I \sqsubseteq \Phi(I)$ and I is *strictly greater* than $\mathsf{lfp}\,\Phi$, then iterating Φ on I will yield a fixed point strictly greater than $\mathsf{lfp}\,\Phi$, contradicting soundness of simple lower induction.

While we just illustrated by means of the Tarski–Kantorovich principle why the simple lower induction rule is not sound in general, we should note that the rule is not per se absurd: So called *metering functions* [Frohn et al. 2016] basically employ simple lower induction to verify *lower bounds* on runtimes of nonprobabilistic programs [Kaminski 2019, Thm. 7.18]. For weakest preexpectations, however, simple lower induction is unsound:

Counterexample 9 (Simple Induction for Lower Bounds). *Consider the following loop C_{cex}, where $b, k \in \mathbb{N}$*

$$\begin{aligned}&\texttt{while}\,(\,a \neq 0\,)\,\{\\ &\quad \{\,a := 0\,\}\ [1/2]\ \{\,b := b+1\,\}\,;\\ &\quad k := k+1\\ &\}\,.\end{aligned}$$

As in Ex. 7, $\mathsf{wp}\,[\![C_{cex}]\!]\,(b) = b + [a \neq 0]$. *In particular, this weakest preexpectation is independent of k. The corresponding characteristic function is*

$$\Phi_b(X) \;=\; [a=0]\cdot b \;+\; [a\neq 0]\cdot\tfrac{1}{2}\cdot\big(X\,[a/0] \;+\; X\,[b/b+1]\big)\,[k/k+1]\,.$$

Let us consider $I' = b + [a \neq 0]\cdot(1+2^k)$, which does *depend on k. Indeed, one can check that $I' \preceq \Phi_b(I')$, i.e., I' is a subinvariant. If the simple lower induction rule were sound, we would immediately conclude that I' is a* lower bound *on* $\mathsf{wp}\,[\![C_{cex}]\!]\,(b)$*, but this is obviously false since*

$$\mathsf{wp}\,[\![C_{cex}]\!]\,(b) = b + [a \neq 0] \;\prec\; I'\,.$$

3.3 Problem Statement

The purpose of this paper is to present a *sound lower induction rule* of the following form: Let Φ_f be the characteristic function of $\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}$ with respect to f and let $I \in \mathbb{F}$. Then

$$I \preceq \Phi_f(I) \;\wedge\; \begin{array}{c}\text{some side}\\ \text{conditions}\end{array} \qquad \text{implies} \qquad I \preceq \mathsf{lfp}\;\Phi_f\,.$$

We still want our lower induction rule to be simple in the sense that checking the side conditions should be conceptually as simple as checking $I \preceq \Phi_f(I)$. Intuitively, we want to apply the semantics of the loop body only *finitely often*, not ω times, to avoid reasoning about limits of sequences or anything alike. We provide such side conditions in our main contribution, Thm. 37, which transfers the Optional Stopping Theorem of probability theory to weakest preexpectation reasoning.

3.4 Uniform Integrability

We now present a sufficient and necessary criterion to under–approximate the least fixed points that we seek for. Let again Φ_f be the characteristic function of $\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}$ with respect to f. Thm. 4 implies that Φ_f is continuous and monotonic.

Let us consider a *subinvariant* I, i.e., $I \preceq \Phi_f(I)$. If we iterate Φ_f on I ad infinitum, then the Tarski–Kantorovich principle (Thm. 8) guarantees that we will converge to some fixed point $\Phi_f^{\omega}(I)$ that is $\succeq I$. From monotonicity of Φ_f and Thm. 8, one can easily show that $\Phi_f^{\omega}(I)$ coincides with $\mathsf{lfp}\;\Phi_f$ *if and only if* I itself was already $\preceq \mathsf{lfp}\;\Phi_f$, i.e.:

Theorem 10 (Subinvariance and Lower Bounds). *For any* subinvariant I*, we have*

$$\Phi_f^\omega(I) \;=\; \mathsf{lfp}\,\Phi_f \qquad \text{iff} \qquad I \;\preceq\; \mathsf{lfp}\,\Phi_f\,.$$

More generally, for *any* expectation X (not necessarily a sub- or superinvariant), if iterating Φ_f on X converges to the least fixed point of Φ_f, then we call X *uniformly integrable for* f:

Definition 11 (Uniform Integrability of Expectations). *Given a loop* `while ( φ ) { C }`*, an expectation* $X \in \mathbb{F}$ *is called* uniformly integrable (u.i.) for $f \in \mathbb{F}$ *if* $\lim_{n\to\omega} \Phi_f^n(X)$ *exists and*

$$\lim_{n\to\omega} \Phi_f^n(X) \;=\; \mathsf{lfp}\,\Phi_f\,.$$

So far, we have thus established the following diagram which we will gradually extend over the next two sections:

$$\begin{array}{ccc} I \text{ u.i. for } f & \xLeftrightarrow{\text{Def. 11}} & \Phi_f^n(I) \xrightarrow{n\to\omega} \mathsf{lfp}\,\Phi_f \\ \Big\Updownarrow \text{Thm. 10 and Def. 11} & & \\ I \preceq \Phi_f(I) \Rightarrow I \preceq \mathsf{lfp}\,\Phi_f & & \end{array}$$

Uniform integrability [Grimmett and Stirzaker 2001] — a notion originally from probability theory — will be essential for the Optional Stopping Theorem in Sect. 5. While, so far, we have studied the function Φ_f solely from an expectation transformer point of view and defined a purely expectation–theoretical notion of uniform integrability without any reference to probability theory, we will study in Sect. 4 the function Φ_f from a stochastic process point of view. Stochastic processes are not inductive per se, whereas expectation transformers make heavy use of induction. We will, however, rediscover the inductiveness also in the realm of stochastic processes. We will also see how our notion of uniform integrability corresponds to uniform integrability in its original sense.

4 FROM EXPECTATIONS TO STOCHASTIC PROCESSES

In this section, we connect concepts from expectation transformers with notions from probability theory. In Sect. 4.1, we recapitulate standard constructions of probability spaces for probabilistic programs, instantiate them in our setting, and present our new results on connecting expectation transformers with stochastic processes (Sect. 4.2) and uniform integrability (Sect. 4.3). Proofs can be found in [Hark et al. 2019, App. C]. For further background on probability theory, we refer to [Hark et al. 2019, App. B] and [Bauer 1971; Grimmett and Stirzaker 2001].

We fix for this section an arbitrary loop `while ( φ ) { C }`. The loop body C may contain loops but we require C to be *universally almost–surely terminating (AST)*, i.e., C terminates on any input with probability 1. The set of program states can be uniquely partitioned into $\Sigma = \Sigma_\varphi \uplus \Sigma_{\neg\varphi}$, with $s \in \Sigma_\varphi$ iff $s \models \varphi$. The set $\Sigma_{\neg\varphi}$ thus contains the terminal states from which the loop is not executed further.

4.1 Canonical Probability Space

We begin with constructing a canonical probability measure and space corresponding to the execution of our loop. As every pGCL program is, operationally, a countable Markov chain, our construction is similar to the standard construction for Markov chains (cf. [Vardi 1985]).

In general, a *measurable space* is a pair $(\Omega, \mathfrak{F})$ consisting of a *sample space* Ω and a *σ–field* $\mathfrak{F}$ of Ω, which is a collection of subsets of Ω, closed under complement and countable union, such that $\Omega \in \mathfrak{F}$. In our setting, a loop `while ( φ ) { C }` induces the following canonical measurable space:

Definition 12 (Loop Space). *The loop* `while ( φ ) { C }` *induces a unique measurable space* $(\Omega^{\text{loop}}, \mathfrak{F}^{\text{loop}})$ *with sample space* $\Omega^{\text{loop}} := \Sigma^{\omega} = \{\vartheta \colon \mathbb{N} \to \Sigma\}$, *i.e., it is the set of all* infinite *sequences of program states (so–called* runs*). For* $\vartheta \in \Omega^{\text{loop}}$, *we denote by* $\vartheta[n]$ *the* n*–th state in the sequence* ϑ *(starting to count at 0). The* σ*–field* $\mathfrak{F}^{\text{loop}}$ *is the smallest* σ*–field that contains all cylinder sets* $Cyl(\pi) = \{\, \pi\vartheta \mid \vartheta \in \Sigma^{\omega} \,\}$, *for all finite prefixes* $\pi \in \Sigma^{+}$, *denoted as*

$$\mathfrak{F}^{\text{loop}} \;=\; \left\langle \left\{ Cyl(\pi) \;\middle|\; \pi \in \Sigma^{+} \right\} \right\rangle_{\sigma} .$$

Intuitively, a run $\vartheta \in \Omega$ is an infinite sequence of states $\vartheta = s_0\, s_1\, s_2\, s_3 \cdots$, where s_0 represents the initial state on which the loop is started and s_i is a state that could be reached after i iterations of the loop. Obviously, some sequences in Ω^{loop} may not actually be admissible by our loop.

We next develop a canonical probability measure corresponding to the execution of the loop, which will assign the measure 0 to inadmissible runs. We start with considering a single loop iteration. The loop body C induces a *family* of distributions[9]

$$^{\bullet}\mu_C \colon \Sigma \to \Sigma \to [0,1] ,$$

where $^{s}\mu_C(s')$ is the probability that after *one* iteration of C on s, the program is in state s'.

The loop `while ( φ ) { C }` induces a *family* of probability measures on $(\Omega^{\text{loop}}, \mathfrak{F}^{\text{loop}})$. This family is parameterized by the initial state of the loop. Using the distributions $^{\bullet}\mu_C$ above, we first define the probability of a finite non–empty prefix of a run, i.e., for $\pi \in \Sigma^{+}$. Here, $^{s}p(\pi)$ is the probability that π is the sequence of states reached after the first loop iterations, when starting the loop in state s. Hence, the family

$$^{\bullet}p \colon \Sigma \to \Sigma^{+} \to [0,1]$$

of distributions on Σ^{+} is defined by

(1) $^{s}p(s') = [s = s']$

(2) $^{s}p(\pi s' s'') = \begin{cases} ^{s}p(\pi s') \cdot [s'' = s'] , & \text{if } s' \in \Sigma_{\neg\varphi} \\ ^{s}p(\pi s') \cdot {}^{s'}\mu_C(s''), & \text{if } s' \in \Sigma_{\varphi} \end{cases} .$

Using the family $^{\bullet}p$, we now obtain a canonical probability measure on the loop space.

Lemma 13 (Loop Measure [Feller 1971, Kolmogorov's Extension Theorem]**).** *There exists a unique family of probability measures* $^{\bullet}\mathbb{P} \colon \Sigma \to \mathfrak{F} \to [0,1]$ *with*

$$^{s}\mathbb{P}\left(Cyl(\pi)\right) \;=\; {}^{s}p(\pi) .$$

We now turn to random variables and their expected values. A mapping $X \colon \Omega \to \overline{\mathbb{R}}_{\geq 0}$ on a probability space $(\Omega, \mathfrak{F}, \mathbb{P})$ is called $(\mathfrak{F}$*–measurable* or *random variable* if for any open set $U \subseteq \overline{\mathbb{R}}_{\geq 0}$ its preimage lies in $\mathfrak{F}$, i.e., $X^{-1}(U) \in \mathfrak{F}$. If $X(\Omega) \subseteq \overline{\mathbb{N}} = \mathbb{N} \cup \{\omega\}$, then this is equivalent to checking $X^{-1}(\{n\}) \in \mathfrak{F}$ for any $n \in \mathbb{N}$. The *expected value* $\mathbb{E}(X)$ of a random variable X is defined as $\mathbb{E}(X) := \int_{\Omega} X d\mathbb{P}$.[10] If X takes only countably many values we have

$$\mathbb{E}(X) \;=\; \int_{\Omega} X \, d\mathbb{P} \;=\; \sum_{r \in X(\Omega)} \mathbb{P}\left(X^{-1}(\{r\})\right) \cdot r .$$

We saw that `while ( φ ) { C }` gives rise to a unique canonical measurable space $(\Omega^{\text{loop}}, \mathfrak{F}^{\text{loop}})$ and to a family of probability measures $^{s}\mathbb{P}$ parameterized by the initial state s on which our loop is started. We now define a corresponding parameterized expected value operator $^{\bullet}\mathbb{E}$.

[9] Since the loop body C is AST, these are distributions and not subdistributions.

[10] Details on integrals for arbitrary measures can be found in [Hark et al. 2019, App. B].

Definition 14 (Expected Value for Loops ${}^{\bullet}\mathbb{E}$). *Let $s \in \Sigma$ and $X\colon \Omega^{\text{loop}} \to \overline{\mathbb{R}}_{\geq 0}$ be a random variable. The* expected value of X *with respect to the loop measure ${}^{s}\mathbb{P}$, parameterized by state s, is defined by ${}^{s}\mathbb{E}(X) \coloneqq \int_{\Omega} X \, d({}^{s}\mathbb{P})$.*

Next, we define a random variable that corresponds to the number of iterations that our loop makes until it terminates.

Definition 15 (Looping Time). *The mapping*

$$T^{\neg\varphi}\colon \quad \Omega^{\text{loop}} \to \overline{\mathbb{N}},\ \vartheta \mapsto \inf\{n \in \mathbb{N} \mid \vartheta[n] \in \Sigma_{\neg\varphi}\}\,,$$

is a random variable and called the looping time *of* `while ( φ ) { C }`*. Here, $\overline{\mathbb{N}} = \mathbb{N} \cup \{\omega\}$ and $\inf \emptyset = \omega$.*

The canonical σ–field $\mathfrak{F}^{\text{loop}}$ contains infinite runs. But after n iterations of the loop we only know the first $n+1$ states $s_0 \cdots s_n$ of a run. Gaining knowledge in this successive fashion can be captured by a so–called *filtration* of the σ–field $\mathfrak{F}^{\text{loop}}$. In general, a filtration is a sequence $(\mathfrak{F}_n)_{n\in\mathbb{N}}$ of subsets of $\mathfrak{F}$, such that $\mathfrak{F}_n \subseteq \mathfrak{F}_{n+1}$ and $\mathfrak{F}_n$ is a σ–field for any $n \in \mathbb{N}$, i.e., $\mathfrak{F}$ is approximated from below.

Definition 16 (Loop Filtration). *The sequence $(\mathfrak{F}_n^{\text{loop}})_{n\in\mathbb{N}}$ is a filtration of $\mathfrak{F}^{\text{loop}}$, where*

$$\mathfrak{F}_n^{\text{loop}} \;=\; \left\langle \{Cyl(\pi) \mid \pi \in \Sigma^{+},\ |\pi| = n+1\} \right\rangle_{\sigma},$$

i.e., $\mathfrak{F}_n^{\text{loop}}$ is the smallest σ–field containing $\{Cyl(\pi) \mid \pi \in \Sigma^{+},\ |\pi| = n+1\}$.[11]

Next, we recall the notion of stopping times from probability theory.

Definition 17 (Stopping Time). *For a probability space $(\Omega, \mathfrak{F}, \mathbb{P})$ with filtration $(\mathfrak{F}_n)_{n\in\mathbb{N}}$, a random variable $T\colon \Omega \to \overline{\mathbb{N}}$ is called a* stopping time *with respect to $(\mathfrak{F}_n)_{n\in\mathbb{N}}$ if for every $n \in \mathbb{N}$ we have $T^{-1}(\{n\}) = \{\vartheta \in \Omega \mid T(\vartheta) = n\} \in \mathfrak{F}_n$.*

Let us reconsider the looping time $T^{\neg\varphi}$ and the loop filtration $(\mathfrak{F}_n^{\text{loop}})_{n\in\mathbb{N}}$. In order to decide for a run $\vartheta = s_0 s_1 \cdots \in \Omega^{\text{loop}}$ whether its looping time is n, we only need to consider the states $s_0 \cdots s_n$. Hence, $(T^{\neg\varphi})^{-1}(\{n\}) \in \mathfrak{F}_n^{\text{loop}}$ for any $n \in \mathbb{N}$ and thus $T^{\neg\varphi}$ is a *stopping time* with respect to $(\mathfrak{F}_n^{\text{loop}})_{n\in\mathbb{N}}$.

Note that $T^{\neg\varphi}$ does *not* reflect the actual runtime of `while ( φ ) { C }`, as it does not take the runtime of the loop body C into account. Instead, $T^{\neg\varphi}$ only counts the number of *loop iterations* of the "outer loop" `while ( φ ) { C }`. This enriches the class of probabilistic programs our technique will be able to analyze, as we will not need to require that the whole program has finite expected runtime, but only that *the outer loop is expected to be executed finitely often.*

4.2 Canonical Stochastic Process

Now we can present our novel results on the connection of weakest preexpectations and stochastic processes. Henceforth, let $f, I \in \mathbb{F}$. Intuitively, f will play the role of the *postexpectation* and I the role of an *invariant* (i.e., I is a sub– or superinvariant). We now present a canonical stochastic process, i.e., a sequence of random variables that captures approximating $\text{wp}\,[\![\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}]\!]\,(f)$ using the invariant I.

Definition 18 (Induced Stochastic Process). *The* stochastic process induced by I, *denoted $\mathbf{X}^{f,I} = (X_n^{f,I})_{n\in\mathbb{N}}$, is given by*

$$X_n^{f,I}\colon \Omega^{\text{loop}} \to \overline{\mathbb{R}}_{\geq 0},\ \vartheta \mapsto \begin{cases} f\left(\vartheta\left[T^{\neg\varphi}(\vartheta)\right]\right), & \text{if } T^{\neg\varphi}(\vartheta) \leq n \\ I(\vartheta[n+1]), & \text{otherwise} \end{cases}.$$

[11] Note that here $\mathfrak{F}^{\text{loop}} = \bigcup_{n\in\mathbb{N}} \mathfrak{F}_n^{\text{loop}}$ which is *not* the case for general filtrations.

Now, in what sense does the stochastic process $\mathbf{X}^{f,I}$ capture approximating the weakest preexpectation of our loop with respect to f by invariant I? $X_n^{f,I}$ takes as argument a run ϑ of the loop and assigns to ϑ a value as follows: If the loop has reached a terminal state within n iterations, it returns the value of the postexpectation f evaluated in that terminal state. If no such terminal state is reached within n steps, it simply *approximates the remainder of the run*, i.e.,

$$\vartheta[0] \cdots \vartheta[n] \underbrace{\vartheta[n{+}1]\, \vartheta[n{+}2]\, \vartheta[n{+}3] \cdots}\,,$$

by returning the value of the invariant I evaluated in $\vartheta[n{+}1]$. We see that $X_n^{f,I}$ needs at most the first $n+2$ states of a run to determine its value. Thus, $X_n^{f,I}$ is not $\mathfrak{F}_n$–measurable but $\mathfrak{F}_{n+1}$–measurable, as there exist runs that agree on the first $n+1$ states but yield different images under $X_n^{f,I}$. Hence, we shift the loop filtration $(\mathfrak{F}_n^{\text{loop}})_{n\in\mathbb{N}}$ by one.

Definition 19 (Shifted Loop Filtration). *The filtration $(\mathfrak{G}_n^{\text{loop}})_{n\in\mathbb{N}}$ of $\mathfrak{F}^{\text{loop}}$ is defined by*

$$\mathfrak{G}_n^{\text{loop}} \;:=\; \mathfrak{F}_{n+1}^{\text{loop}} \;=\; \left\langle \{Cyl(\pi) \mid \pi \in \Sigma^+,\ |\pi| = n+2\} \right\rangle_\sigma \,.$$

Note that $(T^{\neg\varphi})^{-1}(\{n\}) \in \mathfrak{F}_n^{\text{loop}} \subseteq \mathfrak{F}_{n+1}^{\text{loop}} = \mathfrak{G}_n^{\text{loop}}$, so $T^{\neg\varphi}$ is a stopping time w.r.t. $(\mathfrak{G}_n^{\text{loop}})_{n\in\mathbb{N}}$ as well.

Lemma 20 (Adaptedness of Induced Stochastic Process). *$\mathbf{X}^{f,I}$ is adapted to $(\mathfrak{G}_n^{\text{loop}})_{n\in\mathbb{N}}$, i.e., $X_n^{f,I}$ is $\mathfrak{G}_n^{\text{loop}}$–measurable.*

The loop space, the loop measure, and the induced stochastic process $\mathbf{X}^{f,I}$ are not defined by induction on the number of steps performed in the program. The loop space, for instance, contains *all* infinite sequences of states, whether they are admissible by the loop or not. The loop measure filters out the inadmissible runs and gives them probability 0.

Reasoning by invariants and characteristic functions, on the other hand, *is* inductive. We will thus relate iterating a characteristic function on I to the stochastic process $\mathbf{X}^{f,I}$. For this, let Φ_f again be the characteristic function of `while ( φ ) { C }` with respect to f, i.e.,

$$\Phi_f(X) \;=\; [\neg\varphi]\cdot f \;+\; [\varphi]\cdot \text{wp}\,[\![C]\!]\,(X)\,.$$

We now develop a first connection between the stochastic process $\mathbf{X}^{f,I}$ and Φ_f, which involves the notion of conditional expected values with respect to a σ–field, for which we provide some preliminaries here. In general, for $M \subseteq \Omega$, by slight abuse of notation, the *Iverson bracket* $[M] : \Omega \to \overline{\mathbb{R}}_{\geq 0}$ maps $\vartheta \in \Omega$ to 1 if $\vartheta \in M$ and to 0 otherwise. $[M]$ is $\mathfrak{F}$–measurable iff $M \in \mathfrak{F}$. If X is a random variable on $(\Omega, \mathfrak{F}, \mathbb{P})$ and $\mathfrak{G} \subseteq \mathfrak{F}$ is a σ–field with respect to Ω, then the *conditional expected value* $\mathbb{E}\,(X \mid \mathfrak{G}) : \Omega \to \overline{\mathbb{R}}_{\geq 0}$ is a $\mathfrak{G}$–measurable *mapping* such that for every $G \in \mathfrak{G}$ the equality $\mathbb{E}\,(X \cdot [G]) = \mathbb{E}\,(\mathbb{E}\,(X \mid \mathfrak{G}) \cdot [G])$ holds, i.e., restricted to the set G the conditional expected value $\mathbb{E}\,(X \mid \mathfrak{G})$ and X have the same expected value. Hence, $\mathbb{E}\,(X \mid \mathfrak{G})$ is a random variable that is like X, but for elements that are indistinguishable in the subfield $\mathfrak{G}$, i.e., they either are both contained or none of them is contained in a $\mathfrak{G}$–measurable set, it "distributes the value of X equally".

Theorem 21 (Relating $\mathbf{X}^{f,I}$ and Φ_f). *For any $n \in \mathbb{N}$ and any $s \in \Sigma$, we have*

$${}^s\mathbb{E}\left(X_{n+1}^{f,I} \middle| \mathfrak{G}_n^{\text{loop}}\right) \;=\; X_n^{f,\Phi_f(I)}\,.$$

Note that both sides in Thm. 21 are mappings of type $\Omega^{\text{loop}} \to \overline{\mathbb{R}}_{\geq 0}$. Intuitively, Thm. 21 expresses the following: Consider some cylinder $Cyl(\pi) \in \mathfrak{G}_n^{\text{loop}}$, i.e., $\pi = s_0 \cdots s_{n+1} \in \Sigma^{n+2}$ is a sequence of states of length $n+2$. Then, $X_n^{f,\Phi_f(I)}$ and $X_{n+1}^{f,I}$ have the same expected value under ${}^s\mathbb{P}$ on the cylinder set $Cyl(\pi)$ independent of the initial state s of the loop.

Using Thm. 21, one can now explain in which way iterating Φ_f on I represents an expected value, thus revealing the inductive structure inside the induced stochastic process:

Corollary 22 (Relating Expected Values of $\mathbf{X}^{f,I}$ and Iterations of Φ_f). *For any $n \in \mathbb{N}$ and any $s \in \Sigma$, we have*

$$ {}^{s}\mathbb{E}\left(X_n^{f,I}\right) \;=\; \Phi_f^{n+1}(I)(s)\;. $$

Intuitively, Φ_f^{n+1} represents allowing for at most $n+1$ evaluations of the loop guard. For any state $s \in \Sigma$, the number $\Phi_f^{n+1}(I)(s)$ is composed of

(a) f's average value on the final states of those runs starting in s that terminate within $n+1$ guard evaluations, and
(b) I's average value on the $(n+2)$−nd states of those runs starting in s that do *not* terminate within $n+1$ guard evaluations.

We now want to take n to the limit by considering all possible numbers of iterations of the loop body. We will see that this corresponds to evaluating the stochastic process $\mathbf{X}^{f,I}$ at the time when our loop terminates, i.e., the looping time $T^{\neg\varphi}$:

Definition 23 (Canonical Stopped Process). *The mapping*

$$ X^{f,I}_{T^{\neg\varphi}} \colon \Omega^{\text{loop}} \to \overline{\mathbb{R}}_{\geq 0},\, \vartheta \mapsto \begin{cases} X^{f,I}_{T^{\neg\varphi}(\vartheta)}(\vartheta) = f(\vartheta[T^{\neg\varphi}(\vartheta)]), & \text{if } T^{\neg\varphi}(\vartheta) < \infty \\ 0, & \text{otherwise} \end{cases} $$

is the stopped process, *corresponding to $\mathbf{X}^{f,I}$ stopped at stopping time $T^{\neg\varphi}$. As this mapping is independent of I, we write $X^{f}_{T^{\neg\varphi}}$ instead of $X^{f,I}_{T^{\neg\varphi}}$.*

The stopped process now corresponds exactly to the quantity we want to reason about — the value of f evaluated in the final state after termination of our loop. For nonterminating runs we get 0, as there exists no state in which to evaluate f.

We now show that the limit of the induced stochastic process $\mathbf{X}^{f,I}$ corresponds to the stopped process $X^{f}_{T^{\neg\varphi}}$. For the following lemma, note that a statement over runs α holds *almost–surely* in the probability space $(\Omega^{\text{loop}}, \mathfrak{F}^{\text{loop}}, {}^{s}\mathbb{P})$, if ${}^{s}\mathbb{P}\left(\{\, \vartheta \in \Omega \mid \vartheta \text{ satisfies } \alpha \,\}\right) = 1$, i.e., the set of all elements of the sample space satisfying α has probability 1.

Lemma 24 (Convergence of $\mathbf{X}^{f,I}$ to $X^{f}_{T^{\neg\varphi}}$). *The stochastic process $\mathbf{X}^{f,I} \cdot \left[(T^{\neg\varphi})^{-1}(\mathbb{N})\right]$ converges point–wise to $X^{f}_{T^{\neg\varphi}}$, i.e., for all $\vartheta \in \Omega^{\text{loop}}$,*

$$ \lim_{n\to\omega} X_n^{f,I}(\vartheta) \cdot \left[(T^{\neg\varphi})^{-1}(\mathbb{N})\right](\vartheta) \;=\; X^{f}_{T^{\neg\varphi}}(\vartheta)\;. $$

So if `while` $(\varphi)\,\{C\}$ *is universally almost–surely terminating, then $\mathbf{X}^{f,I}$ converges to $X^{f}_{T^{\neg\varphi}}$ almost–surely with respect to the measure ${}^{s}\mathbb{P}$ for any $s \in \Sigma$.*

Intuitively, the factor $\left[(T^{\neg\varphi})^{-1}(\mathbb{N})\right](\vartheta)$ selects those runs ϑ where the looping time $T^{\neg\varphi}$ is finite. If the loop is AST, then this factor can be neglected, because then $\left[(T^{\neg\varphi})^{-1}(\mathbb{N})\right]$ *is* the constant function 1 for the probability measures ${}^{s}\mathbb{P}$. In any case, (i.e., whether the looping time is almost–surely finite or not) the expected value of the *stopped process* captures precisely the weakest preexpectation of our loop with respect to the postexpectation f, since only the terminating runs are taken into account by $X^{f}_{T^{\neg\varphi}}$ and by lfp Φ_f when computing the expected value of f after termination of the loop. So from Cor. 22 and Lem. 24 we get our first main result:

Theorem 25 (Weakest Preexpectation is Expected Value of Stopped Process).

$$ \mathsf{wp}\,[\![\texttt{while}\,(\varphi)\,\{C\}]\!]\,(f) \;=\; \mathsf{lfp}\;\Phi_f \;=\; \lambda s.\; {}^{s}\mathbb{E}\left(X^{f}_{T^{\neg\varphi}}\right). $$

Thm. 25 captures our sought–after least fixed point as an expected value of a canonical stopped process. This is what will allow us to later apply the Optional Stopping Theorem. Moreover, it is crucial for deriving our generalization of an existing rule for lower bounds (cf. Sect. 6) and the connection of upper bounds to the Lemma of Fatou (cf. Sect. 7).

4.3 Uniform Integrability

As we will see in Sect. 5, *uniform integrability* of a certain stochastic process is the central aspect of the Optional Stopping Theorem (Thm. 31). In probability theory, uniform integrability means that taking the expected value and taking the limit of a stochastic process commutes.

Definition 26 (Uniform Integrability of Stochastic Processes, [Grimmett and Stirzaker 2001, Lemma 7.10.(3)]**).** *Let* $\mathbf{X} = (X_n)_{n\in\mathbb{N}}$ *be a stochastic process on a probability space* $(\Omega, \mathfrak{F}, \mathbb{P})$ *with almost–surely existing limit* $\lim_{n\to\omega} X_n$*. The process* $\mathbf{X}$ *is* uniformly integrable *if*

$$\mathbb{E}\left(\lim_{n\to\omega} X_n\right) \;=\; \lim_{n\to\omega} \mathbb{E}\left(X_n\right) \;.$$

Counterexample 27 ([Grimmett and Stirzaker 2001, Sect. 7.10]**).** *Consider the stochastic process* $\mathbf{X} = (X_n)_{n\in\mathbb{N}}$ *on a probability space* $(\Omega, \mathfrak{F}, \mathbb{P})$ *with* $\mathbb{P}\left(Y_n = n\right) = \frac{1}{n} = 1 - \mathbb{P}\left(Y_n = 0\right)$*. Then* $\mathbb{E}\left(X_n\right) = 1$*. Moreover,* $\mathbf{X}$ *converges almost surely to* $Y \equiv 0$*, i.e., the constant function* 0*. So,* $\mathbb{E}\left(Y\right) = 0$*. But*

$$\lim_{n\to\omega} \mathbb{E}\left(X_n\right) = \lim_{n\to\omega} 1 = 1 \neq 0 = \mathbb{E}\left(0\right),$$

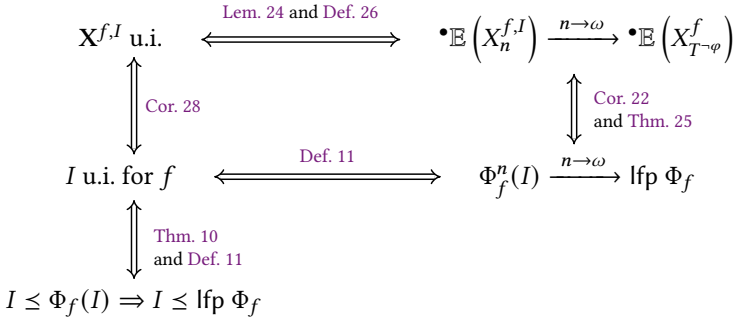
so $\mathbf{X}$ *is not u.i.*

Note that our notion of uniform integrability of expectations from Def. 11 coincides with uniform integrability of the corresponding induced stochastic process.

Corollary 28 (Uniform Integrability of Expectations and Stochastic Processes). *Let the loop* `while` $(\varphi)\,\{C\}$ *be AST.*[12] *Then* I *is uniformly integrable for* f *(in the sense of Def. 11) iff the induced stochastic process* $\mathbf{X}^{f,I}$ *is uniformly integrable (in the sense of Def. 26), i.e.,*

$$\lim_{n\to\omega} \Phi_f^n(I) \;=\; \mathrm{lfp}\,\Phi_f \qquad \text{iff} \qquad \forall s \in \Sigma\colon \quad {}^s\mathbb{E}\left(\lim_{n\to\omega} X_n^{f,I}\right) \;=\; \lim_{n\to\omega} {}^s\mathbb{E}\left(X_n^{f,I}\right) \;.$$

Cor. 28 justifies the naming in Def. 11: an expectation I is uniformly integrable for f iff its induced process $\mathbf{X}^{f,I}$ is uniformly integrable. So, we can now extend the diagram from Sect. 3.4 as follows:



Uniform integrability is very hard to verify in general, both in the realm of stochastic processes as well as in the realm of expectation transformers. Thus, one usually tries to find *sufficient* criteria for uniform integrability that are easier to verify. The very idea of the Optional Stopping Theorem is to provide such sufficient criteria for uniform integrability which then allow deriving a lower bound as we will discuss in the next section.

[12] It suffices that ${}^s\mathbb{P}\left(T^{\neg\varphi} < \infty\right) = 1$ for any s. But this is equivalent to AST as we required the body of the loop to be AST.

5 THE OPTIONAL STOPPING THEOREM OF WEAKEST PREEXPECTATIONS

In this section, we develop an inductive proof rule for lower bounds on preexpectations by using the results of Sect. 4 and the Optional Stopping Theorem (Thm. 31). The proofs of our results in this section can be found in [Hark et al. 2019, App. D]. Recall that we have fixed a loop $\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}$, a finite postexpectation f, a corresponding characteristic function Φ_f, and another finite expectation I which plays the role of an invariant.

We first introduce the Optional Stopping Theorem from probability theory. It builds upon the concept of submartingales. A *submartingale* is a stochastic process that induces a monotonically increasing sequence of its expected values.

Definition 29 (Submartingale). *Let $(X_n)_{n\in\mathbb{N}}$ be a stochastic process on a probability space $(\Omega, \mathfrak{F}, \mathbb{P})$ adapted to a filtration $(\mathfrak{F}_n)_{n\in\mathbb{N}}$ of $\mathfrak{F}$, i.e., a sequence of random variables $X_n \colon \Omega \to \overline{\mathbb{R}}_{\geq 0}$ such that X_n is $\mathfrak{F}_n$–measurable. Then $(X_n)_{n\in\mathbb{N}}$ is called a* submartingale *with respect to $(\mathfrak{F}_n)_{n\in\mathbb{N}}$ if*

$$\text{(a)} \qquad \mathbb{E}\,(X_n) < \infty \text{ for all } n \in \mathbb{N}, \text{ and}$$
$$\text{(b)} \qquad \mathbb{E}\,(X_{n+1} \mid \mathfrak{F}_n) \geq X_n.$$

It turns out that submartingales are closely related to subinvariants. In fact, I being a *subinvariant* (plus some side conditions) gives us that the stochastic process induced by I is a *submartingale*.

Lemma 30 (Subinvariant Induces Submartingale). *Let I be a* subinvariant*, i.e., $I \preceq \Phi_f(I)$, such that $\Phi_f^n(I) \ll \infty$ for every $n \in \mathbb{N}$, that is, $\Phi_f^n(I)$ only takes finite values. Then the induced stochastic process $\mathbf{X}^{f,I}$ is a submartingale with respect to $(\mathfrak{G}_n^{\text{loop}})_{n\in\mathbb{N}}$.*

Given a submartingale $(X_n)_{n\in\mathbb{N}}$ and a stopping time T, the goal of the Optional Stopping Theorem is to prove a lower bound for the expected value of X_n at the stopping time T. To this end, we define a stochastic process $(X_{n\wedge T})_{n\in\mathbb{N}}$ where for any $\vartheta \in \Omega$, $X_{n\wedge T}(\vartheta) = X_n(\vartheta)$ if n is smaller than the stopping time $T(\vartheta)$ and otherwise, $X_{n\wedge T}(\vartheta) = X_{T(\vartheta)}(\vartheta)$. Hence, $\mathbb{E}\,(\lim_{n\to\omega} X_{n\wedge T})$ is the expected value of X_n at the stopping time T. The Optional Stopping Theorem asserts that the first component X_0 of the stochastic process $(X_n)_{n\in\mathbb{N}}$ is a lower bound for $\mathbb{E}\,(\lim_{n\to\omega} X_{n\wedge T})$ provided that $(X_{n\wedge T})_{n\in\mathbb{N}}$ is uniformly integrable. Moreover, the Optional Stopping Theorem provides a collection of criteria that are sufficient for uniform integrability of $(X_{n\wedge T})_{n\in\mathbb{N}}$.

Theorem 31 (Optional Stopping Theorem [Grimmett and Stirzaker 2001, Theorems 12.3.(1), 12.4.(11), 12.5.(1), 12.5.(2), 12.5.(9)]**).** *Let $(X_n)_{n\in\mathbb{N}}$ be a submartingale and T be a stopping time on a probability space $(\Omega, \mathfrak{F}, \mathbb{P})$ with respect to a filtration $(\mathfrak{F}_n)_{n\in\mathbb{N}}$. Then $\mathbf{X}_{\wedge T} = (X_{n\wedge T})_{n\in\mathbb{N}}$ defined by*

$$X_{n\wedge T} : \Omega \to \overline{\mathbb{R}}_{\geq 0},\ \vartheta \mapsto X_{\min(n,T(\vartheta))}(\vartheta),$$

is also a submartingale w.r.t. $(\mathfrak{F}_n)_{n\in\mathbb{N}}$. If $\mathbf{X}_{\wedge T}$ converges almost–surely and is uniformly integrable,

$$\mathbb{E}\,(X_0) = \mathbb{E}\,(X_{0\wedge T}) \leq \lim_{n\to\omega} \mathbb{E}\,(X_{n\wedge T}) = \mathbb{E}\left(\lim_{n\to\omega} X_{n\wedge T}\right).$$

If one *of the following conditions holds, then $\mathbf{X}_{\wedge T}$ converges almost–surely and is uniformly integrable:*

(a) *T is almost–surely bounded, i.e., there is a constant $N \in \mathbb{N}$ such that $\mathbb{P}\,(T \leq N) = 1$.*

(b) *$\mathbb{E}\,(T) < \infty$ and there is a constant $c \in \mathbb{R}_{\geq 0}$, such that for each $n \in \mathbb{N}$*

$$\mathbb{E}\,(|X_{n+1} - X_n| \mid \mathfrak{F}_n) \leq c \qquad \text{holds almost–surely}.$$

(c) *There exists a constant $c \in \mathbb{R}_{\geq 0}$ such that $X_{n\wedge T} \leq c$ holds almost–surely for every $n \in \mathbb{N}$.*

Our goal now is to transfer the Optional Stopping Theorem from probability theory to the realm of weakest preexpectations in order to obtain inductive proof rules for lower bounds on weakest preexpectations. So far, we have introduced the looping time $T^{\neg\varphi}$ (which is a stopping time w.r.t.

$(\mathfrak{F}_n^{\text{loop}})_{n\in\mathbb{N}}$), presented the connection of subinvariants and submartingales, and defined the concept of uniform integrability also for expectations. Hence, the only missing ingredient is a proper connection of expectations to the condition "$\mathbb{E}\left(|X_{n+1} - X_n| \mid \mathfrak{F}_n\right) \leq c$" in Thm. 31 (b). To translate this concept to expectations, we require that the expectation I has a certain shape depending on the postexpectation f.

Definition 32 (Harmonization). *An expectation I* harmonizes with *$f \in \mathbb{F}$ if it is of the form*

$$I \;=\; [\neg\varphi]\cdot f \;+\; [\varphi]\cdot I'\,,$$

for some expectation $I' \in \mathbb{F}$.

Def. 32 reflects that in terminal states t of the loop the invariant I evaluates to $f(t)$. For an invariant I to harmonize with postexpectation f is a minor restriction on the shape of I. It is usually easy to choose an I that takes the value of f for states in which the loop is not executed at all. Moreover, performing one iteration of Φ_f obviously brings *any* expectation "into shape":

Corollary 33 (Harmonizing Expectations). *For any $f, J \in \mathbb{F}$, $\Phi_f(J)$ harmonizes with f.*

The actual criterion that connects "$\mathbb{E}\left(|X_{n+1} - X_n| \mid \mathfrak{F}_n\right) \leq c$" with the invariant I is called *conditional difference boundedness* (see also [Fioriti and Hermanns 2015; Fu and Chatterjee 2019]):

Definition 34 (Conditional Difference Boundedness). *Let $I \in \mathbb{F}$. We define the function $H\colon \mathbb{F} \to \mathbb{F}$ and the expectation $\Delta I \in \mathbb{F}$ as*[13]

$$H(X) \;=\; [\varphi]\cdot \mathrm{wp}\,[\![C]\!]\,(X) \qquad \text{and} \qquad \Delta I \;=\; \lambda s.\; \big(H\big(|I - I(s)|\big)(s)\big)\;.$$

The expectation I is called conditionally difference bounded *(c.d.b.) if for some constant $c \in \mathbb{R}_{\geq 0}$*

$$\Delta I(s) \;\leq\; c \text{ holds for all } s \in \Sigma.$$

The expectation ΔI expresses the *expected change of I* within one loop iteration. So, if I is c.d.b. it is expected to change at most by a constant in one loop iteration.

Example 35. *Reconsider the program C_{cex} from Counterex. 9 and expectation $I = b + [a \neq 0]$. We will check* conditional difference boundedness *of I, using the function H given by*

$$H(X) \;=\; [a \neq 0]\cdot \tfrac{1}{2}\cdot \big(X\,[a/0] + X\,[b/b+1]\big)\,[k/k+1]\;.$$

We then check the following:

$$\begin{aligned}
\Delta I \;&=\; \big(\lambda s.\; [a\neq 0]\cdot\tfrac{1}{2}\cdot\big(|I - I(s)|\,[a/0] \;+\; |I - I(s)|\,[b/b+1]\big)\,[k/k+1]\big)(s)\\
&=\; \lambda s.\; \big([a\neq 0]\cdot\tfrac{1}{2}\cdot\big(|I\,[a/0] - s(b) - [a\neq 0]\,(s)| \;+\; |I\,[b/b+1] - s(b) - [a\neq 0]\,(s)|\big)\big)(s)\\
&=\; [a\neq 0]\,(s)\cdot\tfrac{1}{2}\cdot\big(|s(b) + [0\neq 0] - s(b) - 1| \;+\; |s(b) + 1 + [a\neq 0]\,(s) - s(b) - 1|\big)\\
&=\; [a\neq 0]\,(s)\cdot\tfrac{1}{2}\cdot\big(|-1| + |1|\big) \;\leq\; 1\,.
\end{aligned}$$

Thus, I is c.d.b. by the constant 1. In contrast, the subinvariant $I' = b + [a \neq 0]\cdot(1 + 2^k)$ from Counterex. 9 is not conditionally difference bounded. Indeed, we would get (cf. [Hark et al. 2019, App. D] *for details)*

$$\Delta I' \;=\; \big([a\neq 0]\cdot(1+2^k)\big)(s)\,,$$

which cannot be bounded by a constant.

Finally, we can connect the expected change of I to a property of the stochastic process $\mathbf{X}^{f,I}$. This is our second major result.

[13] Recall that we have fixed a loop `while` (φ) { C }.

Theorem 36 (Expected Change of *I*). *Let $I \ll \infty$ harmonize with f. Then*

$$ {}^{s}\mathbb{E}\left(\left|X_{n+1}^{f,I} - X_n^{f,I}\right| \,\middle|\, \mathfrak{G}_n^{\text{loop}}\right) \;=\; X_n^{0,\Delta I}\,. $$

The stochastic process $\mathbf{X}^{f,I}$ induced by I exhibits an interesting correspondence: If ΔI is bounded by a constant c (i.e., if I is c.d.b.), then so is $X_n^{0,\Delta I}$ and thus Thm. 36 ensures that precondition (b) of the Optional Stopping Theorem (Thm. 31) is fulfilled. Note that Thm. 36 depends crucially on the fact that $I \ll \infty$ as otherwise the well–definedness of the expectation ΔI cannot be ensured.

Now Lem. 30 allows us to use the Optional Stopping Theorem from probability theory (Thm. 31) to prove a *novel Optional Stopping Theorem for weakest preexpectations*, which collects sufficient conditions for uniform integrability. In particular, due to Thm. 36, *our* Optional Stopping Theorem shows that our notion of conditional difference boundedness is an (easy–to–check) sufficient criterion for uniform integrability and hence, for ensuring that a subinvariant is indeed a lower bound for the weakest preexpectation under consideration. After stating the theorem, we will discuss the intuition of its parts in more detail.

Theorem 37 (Optional Stopping Theorem for Weakest Preexpectation Reasoning). *Consider a loop* `while ( φ ) { C }` *where C is AST. Let $I \ll \infty$ be a subinvariant w.r.t. the postexpectation $f \ll \infty$ (i.e., $I \le \Phi_f(I)$). I is uniformly integrable for f iff I is a* lower bound, *i.e.,*

$$ I \;\le\; \mathsf{lfp}\,\Phi_f \;=\; \mathsf{wp}\,[\![\texttt{while}\,(\varphi)\,\{C\}]\!]\,(f)\,. $$

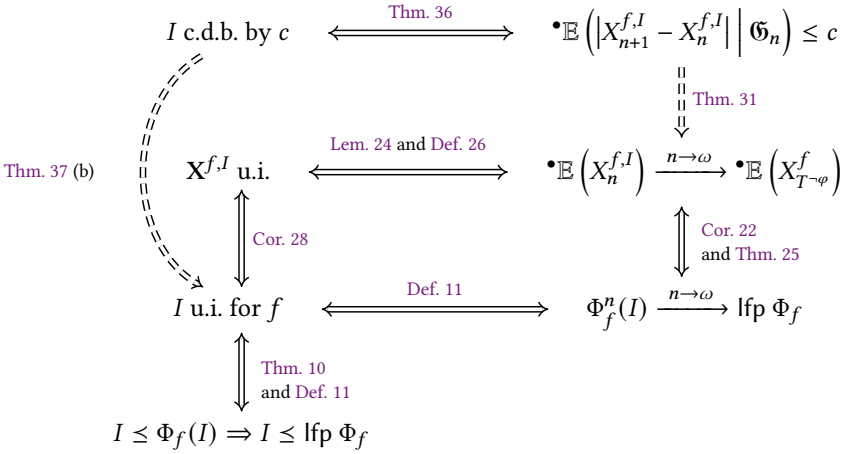
I is uniformly integrable *for f if one of the following three conditions holds:*

(a) *The looping time $T^{\neg\varphi}$ of* `while ( φ ) { C }` *is almost–surely bounded, i.e., for every state $s \in \Sigma$ there exists a constant $N(s) \in \mathbb{N}$ with ${}^{s}\mathbb{P}\,(T^{\neg\varphi} \le N(s)) = 1$ and $\Phi_f^n(I) \ll \infty$ for every $n \in \mathbb{N}$.*

(b) *The expected looping time of* `while ( φ ) { C }` *is finite for every initial state $s \in \Sigma$, I harmonizes with f, $\Phi_f(I) \ll \infty$, and I is conditionally difference bounded.*

(c) *Both f and I are bounded and* `while ( φ ) { C }` *is AST.*

We can now extend the diagram from Sect. 4 connecting the realm of stochastic processes (on the right) and the realm of expectation transformers (on the left) for a universally almost–surely terminating program. The respective Optional Stopping Theorems provide the sufficient criteria for uniform integrability, which is marked by the dashed implications.



Let us elaborate on the different cases of *our* Optional Stopping Theorem (Thm. 37): Case (a) yields an alternative proof for the technique of so–called metering functions by [Frohn et al. 2016]

for *deterministic terminating* loops. As for the severity of the finiteness condition "$\Phi_f^n(I) \ll \infty$ for every $n \in \mathbb{N}$", note that if the body C is loop–free, this condition is vacuously satisfied as I itself is finite and cannot become infinite by finitely iterations of Φ_f. If C contains loops, then we can establish the finiteness condition by finding a finite superinvariant U with $I \preceq U \ll \infty$. In this case, we can also guarantee $\Phi_f^n(I) \ll \infty$.[14]

Case (b) applies whenever the outer loop is expected to be executed finitely often. In particular, this holds if the entire loop terminates positively almost–surely (i.e., within finite expected *runtime*).

To the best of our knowledge, Cases (a) and (b) are the first sufficiently simple induction rules for lower bounds that do not require restricting to bounded postexpectations f. While the requirements on the loop's termination behavior gradually weaken along (a) $\rightarrow$ (b) $\rightarrow$ (c), the requirements on the subinvariant I become stricter.

Finally, Case (c) yields an alternative proof of the result of [McIver and Morgan 2005] on inductive lower bounds for bounded expectations in case of AST, which we will generalize in Sect. 6.

When comparing the cases (c) of Thm. 31 and Thm. 37, we notice that Thm. 31 (c) has no restrictions on the stopping time, whereas Thm. 37 (c) requires almost–sure termination. This might spark some hope that AST is not needed in Thm. 37 (c), but the following counterexample shows that this is not the case:

Counterexample 38. *Consider the program*

$$\texttt{while}\,(\,\texttt{true}\,)\,\{\,\texttt{skip}\,\}\ ,$$

together with the bounded postexpectation $f = 1$, i.e., we are interested in the termination probability which is obviously 0*. The corresponding characteristic function is given by*

$$\Phi_1(X) \;=\; [\neg\texttt{true}]\cdot 1 + [\texttt{true}]\cdot \mathsf{wp}\,[\![\texttt{skip}]\!]\,(X) \;=\; X\,,$$

i.e., Φ_1 is the identity map. Trivially, the bounded expectation $I = 1$ is a fixed point of Φ_1, thus in particular I is a subinvariant. Clearly, I is not a lower bound on the actual termination probability, i.e., on lfp Φ_1*. If the condition of almost–sure termination in* Thm. 37 (c) *could be weakened, it has to be ensured that for any program* $\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}$ *with universally almost–surely terminating body C*[15] *and postexpectation $f = 1$, 1 is a lower bound only if the program terminates universally almost–surely. But this means that this property has to be at least as strong as almost–sure termination.*

We reconsider Counterex. 9 illustrating unsoundness of simple lower induction and do *sound* lower induction instead.

Example 39. *Let us continue* Ex. 35*, where we have checked that for the program C_{cex} the expectation $I = b + [a \neq 0]$ is conditionally difference bounded by* 1*. It is easy to check that I is a fixed point of the characteristic function Φ_b with respect to the postexpectation b, which by Park induction gives us a finite upper bound on the least fixed point of Φ_b. But up to now we could not prove that I is indeed equal to the least fixed point. Using* Thm. 37*, we can now do this.*

First of all, we already have $\Phi_b(I) = I \ll \infty$ and since I is a fixed point, it is also a subinvariant. Secondly, the loop is expected to be executed twice.[16] *Finally, $I = b + [a \neq 0] = [\neg(a \neq 0)]\cdot b + [a \neq 0]\cdot(b+1)$ harmonizes with b and is conditionally difference bounded. Hence, the preconditions of* Thm. 37 *(b) are satisfied and I is indeed a lower bound on* lfp Φ_b*. Since I is a fixed point,* it is the least fixed point*, i.e., we have proved* $\mathsf{wp}\,[\![C_{cex}]\!]\,(b) = I$*.*

[14] The reason is that by Thm. 8 we have $U \geq \Phi_f^n(U) \geq \Phi_f^\omega(U)$ and $\Phi_f^\omega(I) \geq \Phi_f^n(I) \geq I$ for all $n \in \mathbb{N}$. By the monotonicity of Φ_f (Thm. 4), $U \geq I$ implies $\Phi^\omega(U) \geq \Phi^\omega(I)$, which gives us $\infty \gg U \geq \Phi^\omega(U) \geq \Phi^\omega(I) \geq \Phi^n(I)$.

[15] Note that in this case 1 is always a subinvariant.

[16] Positive almost–sure termination itself can also be verified by Park induction, see [Kaminski et al. 2016, 2018].

Further case studies demonstrating the effectiveness of our proof rule, as well as an example that cannot be treated by Thm. 37, are provided in [Hark et al. 2019, App. A].

6 LOWER BOUND RULES BY MCIVER AND MORGAN

In Sect. 5, we briefly mentioned the rules for lower bounds for *bounded* expectations by [McIver and Morgan 2005] which are restated in Thm. 40 below. To the best of our knowledge, before our new Thm. 37 these were the only existing *inductive* proof rules for weakest preexpectations.

Theorem 40 ([McIver and Morgan 2005]). *Let $f \in \mathbb{F}$ be a **bounded** postexpectation. Furthermore, let $I' \in \mathbb{F}$ be a **bounded** expectation such that the harmonized expectation $I \in \mathbb{F}$ given by $I = [\neg\varphi]\cdot f + [\varphi]\cdot I'$ is a subinvariant of $\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}$ with respect to f. Finally, let $T = \mathsf{wp}[\![\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}]\!]\,(1)$ be the termination probability of $\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}$. Then:*

(1) *If $I = [G]$ for some predicate G, then $T\cdot I \preceq \mathsf{wp}[\![\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}]\!]\,(f)$.*
(2) *If $[G] \preceq T$ for some predicate G, then $[G]\cdot I \preceq \mathsf{wp}[\![\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}]\!]\,(f)$.*
(3) *If $\varepsilon\cdot I \preceq T$ for some $\varepsilon > 0$, then $I \preceq \mathsf{wp}[\![\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}]\!]\,(f)$.*

Thm. 40 does not make any assumptions on the termination behavior of the loop, so, it is also possible to analyze programs with termination probability < 1. It turns out that Thm. 40 (1) – (3) can be proved easily from our results from Sect. 4 in the case where C is AST where we do *not* need the restriction that I harmonizes with f. In particular, we can show that in Thm. 40 (3) the fact that T is the probability of termination is *insignificant* (see [Hark et al. 2019, App. E]). In fact, it suffices if T is the weakest preexpectation for some arbitrary bounded postexpectation, i.e., a *least fixed point* (see [Hark et al. 2019, App. E] for details and proofs). So, we obtain the following generalized version of Thm. 40 (3) in the case where C is AST which is substantially more powerful: it states a sufficient condition for a subinvariant to be a lower bound but also a *necessary* condition. This is the main new contribution of this section.

Theorem 41 (Generalization of Thm. 40 (3)). *Let $f \in \mathbb{F}$ be a **bounded** postexpectation. Furthermore, let $I \in \mathbb{F}$ be a **bounded** expectation such that I is a subinvariant of $\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}$ with respect to f where C is AST. There exist $\varepsilon > 0$ and $g \in \mathbb{F}$ **bounded** s.t.*

$$\varepsilon\cdot I \preceq \mathsf{wp}[\![\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}]\!]\,(g) \qquad \text{if and only if} \qquad I \preceq \mathsf{wp}[\![\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}]\!]\,(f)\ .$$

Example 42. *Let us consider the program C_{rdw} for an asymmetric random walk*

$$\begin{aligned}&\texttt{while}\,(\,x > 0\,)\,\{\\ &\qquad \{\,x := x-1\,\}\ [1/3]\ \{\,x := x+1\,\}\ ;\\ &\qquad y := \max(y-1, 0)\\ &\}\,,\end{aligned}$$

with $x, y \in \mathbb{N}$ and $y \leq 100$. This program is not *AST but the body of the loop is indeed AST. Furthermore, the postexpectation y is bounded. If $y \leq x$ initially then y is 0 after termination of the program. So, $\mathsf{wp}[\![C_{rdw}]\!]\,(y) \geq [y > x]\cdot\left(\frac{1}{3}\right)^x\cdot(y-x) \coloneqq I$.*

Now consider $f = [y \text{ even}]\cdot 200\cdot y^2 + [y \text{ odd}]\cdot(y+5)^4$. We have $I' \preceq \Phi_f(I')$, where $I' = 400\cdot I$ (see [Hark et al. 2019, App. E]). As we have $\frac{1}{400}\cdot I' \preceq \mathsf{wp}[\![C_{rdw}]\!]\,(y)$ we can conclude from Thm. 41 that $I' \preceq \mathsf{wp}[\![C_{rdw}]\!]\,(f)$. Note that this is easier than relating I' and the termination probability as required by Thm. 40 since the probability of termination of the loop is independent of y.

Of course, Ex. 42 is an artifical example. Nevertheless, it shows a strength of our generalisation: it makes it easier to reason about bounded expectations which are independent of the probability of termination. However, a drawback of Thm. 40 remains: one already needs a lower bound, i.e., one has to be able to read off a lower bound directly from the program.

7 UPPER BOUNDS AND FATOU'S LEMMA

We saw that Park induction for proving upper bounds does not require additional conditions such as conditional difference boundedness or even boundedness of f or I, respectively. The question arises whether this fact is also explainable using our canonical stochastic process. Indeed, the well–known *Lemma of Fatou* provides such an explanation. We will present a specialized variant of it which is sufficient for our purpose.

Lemma 43 (Fatou's Lemma (cf. [Bauer 1971, Lemma 2.7.1])). *Let $(X_n)_{n\in\mathbb{N}}$ be a stochastic process on a probability space $(\Omega, \mathfrak{F}, \mathbb{P})$. Then*

$$\mathbb{E}\left(\lim_{n\to\omega} X_n\right) \;\leq\; \lim_{n\to\omega} \mathbb{E}\left(X_n\right) \;,$$

where the lim *on the left–hand–side is point–wise.*

We can now reprove Park induction for wp using Fatou's Lemma: Let I be a superinvariant, i.e., $\Phi_f(I) \preceq I$. By Thm. 21, the canonical stochastic process $\mathbf{X}^{f,I}$ satisfies

$$^s\mathbb{E}\left(X^{f,I}_{n+1} \mid \mathfrak{G}_n\right) \;=\; X^{f,\Phi_f(I)}_n \;\leq\; X^{f,I}_n \;.$$

By applying $^s\mathbb{E}$ on both sides, we obtain $^s\mathbb{E}\left(X^{f,I}_0\right) \geq {}^s\mathbb{E}\left(X^{f,I}_1\right) \geq \ldots$. This implies

$$^s\mathbb{E}\left(X^{f,I}_0\right) \geq {}^s\mathbb{E}\left(X^{f,I}_n\right) \geq {}^s\mathbb{E}\left(X^{f,I}_n \cdot \left[(T^{\neg\varphi})^{-1}(\mathbb{N})\right]\right) \;, \tag{1}$$

as $X^{f,I}_n \geq X^{f,I}_n \cdot \left[(T^{\neg\varphi})^{-1}(\mathbb{N})\right]$. We conclude

$$\begin{aligned}
&\left(\mathsf{lfp}\ \Phi_f\right)(s) \\
&= {}^s\mathbb{E}\left(X^f_{T^{\neg\varphi}}\right) && \text{(by Thm. 25)} \\
&= {}^s\mathbb{E}\left(\lim_{n\to\omega} X^{f,I}_n \cdot \left[(T^{\neg\varphi})^{-1}(\mathbb{N})\right]\right) && \text{(by Lem. 24)} \\
&\leq \lim_{n\to\omega} {}^s\mathbb{E}\left(X^{f,I}_n \cdot \left[(T^{\neg\varphi})^{-1}(\mathbb{N})\right]\right) && \text{(by Fatou's Lemma)} \\
&\leq \lim_{n\to\omega} {}^s\mathbb{E}\left(X^{f,I}_0\right) && \text{(by (1))} \\
&= {}^s\mathbb{E}\left(X^{f,I}_0\right) \\
&= \Phi_f(I)(s) && \text{(by Cor. 22)} \\
&\leq I(s) \;, && \text{(since } \Phi_f(I) \preceq I\text{)}
\end{aligned}$$

so I is indeed an upper bound on the least fixed point.

Note that here we handle arbitrary loops, i.e., they are *not* necessarily AST. While I being a superinvariant (plus some side conditions) still implies that $\mathbf{X}^{f,I}$ is a supermartingale, the second part of Lem. 24 is not applicable, i.e., in general we have $X^f_{T^{\neg\varphi}} \neq \lim_{n\to\omega} X^{f,I}_n$ if the loop is not AST. So in this case we cannot use classic results from martingale theory. Nevertheless, Fatou's Lemma combined with Thm. 25 and the first part of Lem. 24 provide a connection of Park induction for upper bounds to stochastic processes.

8 LOWER BOUNDS ON THE EXPECTED RUNTIME

So far, we have developed techniques for verifying lower bounds on weakest preexpectations, i.e., expected values of random variables upon program termination. In this section, we transfer

Table 2. Rules for the ert–transformer.

C	$\mathbf{ert}\,[\![C]\!]\,(t)$
`skip`	$1 + t$
$b := e$	$1 + t\,[b/e]$
`if` $(\varphi)\,\{C_1\}$ `else` $\{C_2\}$	$1 + [\varphi]\cdot \mathsf{ert}\,[\![C_1]\!]\,(t) + [\neg\varphi]\cdot \mathsf{ert}\,[\![C_2]\!]\,(t)$
$\{C_1\}\,[p]\,\{C_2\}$	$1 + p\cdot \mathsf{ert}\,[\![C_1]\!]\,(t) + (1-p)\cdot \mathsf{ert}\,[\![C_2]\!]\,(t)$
$C_1 \fatsemi C_2$	$\mathsf{ert}\,[\![C_1]\!]\,(\mathsf{ert}\,[\![C_2]\!]\,(t))$
`while` $(\varphi)\,\{C'\}$	$\mathsf{lfp}_{\langle C',\varphi\rangle}^{\mathsf{ert}}\Phi_t$

$$ {}^{\mathsf{ert}}_{\langle\varphi,C\rangle}\Phi_t(X) \;=\; 1 + [\neg\varphi]\cdot t + [\varphi]\cdot \mathsf{ert}\,[\![C]\!]\,(X) \qquad \text{characteristic function} $$

those techniques to verify lower bounds on *expected runtimes* of probabilistic programs. For this, we employ the ert–transformer [Kaminski et al. 2016, 2018], which is very similar to the wp-transformer: Given program C and post*runtime* $t \in \mathbb{F}$, we are interested in the *expected time* it takes to first execute C and then let time t pass (where t is evaluated in the final states reached after termination of C). Again, the behavior (and the runtime) of C depends on its input, so we are actually interested in a *function* $g \in \mathbb{F}$ mapping *initial* states s_0 to the respective expected time. For more details, see also [Kaminski 2019, Chapter 7]. Similarly to weakest preexpectations, expected runtimes can be determined in a systematic and *compositional* manner by means of the ert calculus:

Definition 44 (The ert–Transformer [Kaminski et al. 2016, 2018]**).** *Let* pGCL *be again the set of programs in the* probabilistic guarded command language. *Then the expected runtime transformer*

$$ \mathsf{ert}\colon \mathsf{pGCL} \to \mathbb{F} \to \mathbb{F} $$

is defined according to the rules given in Table 2. *We call the function* ${}^{\mathsf{ert}}_{\langle\varphi,C\rangle}\Phi_t$ *the* ert–characteristic function *of the loop* `while` $(\varphi)\,\{C\}$ *with respect to* t. *Its least fixed point is understood in terms of the partial order* $\preceq$. *To increase readability, we will again usually omit* ert, φ, C, *or* t *from* Φ *whenever they are clear from the context.*

Example 45 (Applying the ert Calculus). *Consider the probabilistic program* C *given by*

$$ \{b := b+5\}\;[4/5]\;\{b := 10\} \fatsemi $$
$$ \texttt{if}\;(b = 10)\;\{\texttt{skip}\}\;\texttt{else}\;\{\texttt{skip} \fatsemi \texttt{skip}\} $$

Suppose we want to know the expected runtime of C. *Then we need to determine* $\mathsf{ert}\,[\![C]\!]\,(0)$. *Reusing the annotation styles of* Fig. 2a *for* wp, *we make the following* ert *annotations:*

$$ \overset{=}{///}\, 4 + [b \neq 5]\cdot\tfrac{4}{5} $$
$$ \overset{\mathsf{ert}}{///}\, 1 + \tfrac{4}{5}\cdot\big(1+2+[b+5\neq 10]\big) + \tfrac{1}{5}\cdot\big(1+2+[10\neq 10]\big) $$
$$ \{b := b+5\}\;[4/5]\;\{b := 10\} \fatsemi $$
$$ \overset{=}{///}\, 2 + [b \neq 10] $$
$$ \overset{\mathsf{ert}}{///}\, 1 + [b = 10]\cdot(1+0) + [b \neq 10]\cdot(1+1+0) $$
$$ \texttt{if}\;(b = 10)\;\{\texttt{skip}\}\;\texttt{else}\;\{\texttt{skip} \fatsemi \texttt{skip}\} $$
$$ ///\, 0 $$

At the top, we read off the expected runtime of C, namely $4 + [b \neq 5] \cdot \frac{4}{5}$. This tells us that the expected runtime of C is 4 if started in an initial state where b is 5, and $4 + \frac{4}{5} = \frac{24}{5}$ otherwise.

The ert– and the wp–transformers are not only similar in definition, but they are closely connected by the following equality [Olmedo et al. 2016]:

$$\mathsf{ert}\,[\![C]\!]\,(t) \;=\; \mathsf{ert}\,[\![C]\!]\,(0) + \mathsf{wp}\,[\![C]\!]\,(t)\;.$$

In addition, reasoning about upper bounds by Park induction works exactly the same way. For reasoning about lower bounds using subinvariants, notice above that $\mathsf{ert}\,[\![C]\!]\,(0)$ is *independent of* t. So, we can combine our derivation of Thm. 37 for lower bounds on wp in Sect. 4 and 5 with the equation above to establish the first inductive rule for verifying lower bounds on expected runtimes:

Theorem 46 (Inductive Lower Bounds on Expected Runtimes). *Let $t, I \in \mathbb{F}$ with $t, I \ll \infty$ and let I harmonize with t. Furthermore, let ${}^{\mathsf{ert}}\Phi_t$ be the* ert*–characteristic function of the loop* $\texttt{while}\,(\varphi)\,\{C\}$ *with respect to t. If I is conditionally difference bounded and ${}^{\mathsf{wp}}\Phi_t(I) \ll \infty$, then*

$$I \;\preceq\; {}^{\mathsf{ert}}\Phi_t(I) \qquad \text{implies} \qquad I \;\preceq\; \mathsf{ert}\,[\![\texttt{while}\,(\varphi)\,\{C\}]\!]\,(t)\;.$$

We call an I that satisfies $I \preceq {}^{\mathsf{ert}}\Phi_t(I)$ a runtime subinvariant.

The proof of Thm. 46 can be found in [Hark et al. 2019, App. F.1]. We now illustrate the applicability of Thm. 46:

Example 47 (Coupon Collector [Pólya 1930]**).** *Consider the well–known coupon collector's problem: There are N different types coupons. A collector wants to collect at least one of each type. Each time she buys a new coupon, its type is drawn uniformly at random. How many coupons does she (expectedly) need to buy in order to have collected at least one coupon of each type?*

We can model this problem by the program C_{cc} for some non–zero natural number $N \in \mathbb{N}$:

$$\begin{aligned}
&x := N\,\fatsemi\\
&\texttt{while}\,(0 < x)\,\{\\
&\quad i := N + 1\,\fatsemi\\
&\quad \texttt{while}\,(x < i)\,\{\\
&\qquad i := \mathrm{Unif}[1..N]\\
&\quad \}\,\fatsemi\\
&\quad x := x - 1\\
&\},
\end{aligned}$$

Variable x represents the number of uncollected coupon types. The inner loop models the buying of new coupons until an uncollected type is drawn.[17]

The expected runtime of C_{cc} is proportional to the expected number of coupons the collector needs to buy. We want to prove that $N\mathcal{H}_N$ is a lower bound on that expected runtime, where $\mathcal{H}_m$ is the m-th harmonic number, i.e., $\mathcal{H}_0 = 0$ and $\mathcal{H}_m = \sum_{k=1}^{m} \frac{1}{k}$. For this, we make the following annotations, reusing the annotation style of Fig. 3a (for more detailed annotations, see [Hark et al. 2019, App. F.2]*):*

[17] The random assignment $i := \mathrm{Unif}[1..N]$ does — strictly speaking — not adhere to our pGCL syntax, but it can be modeled in pGCL. For the sake of readability, we opted for $i := \mathrm{Unif}[1..N]$.

$$
\begin{aligned}
&\overset{=}{/\!/\!/} 1 \;+\; N\mathcal{H}_N \\
&\overset{\text{ert}}{/\!/\!/} 1 \;+\; [0 < N \le N] \cdot N\mathcal{H}_N \;+\; [N < N] \cdot (N\mathcal{H}_N + N - N) \\
&\quad x := N \,; \\
&\overset{\le}{/\!/\!/} [0 < x \le N] \cdot N\mathcal{H}_x \;+\; [N < x] \cdot (N\mathcal{H}_N + N - x) \\
&\overset{=}{/\!/\!/} 1 \;+\; [0 < x \le N] \cdot \left(N\mathcal{H}_x + 3 + \tfrac{N}{x}\right) \;+\; [N < x] \cdot (4 + N\mathcal{H}_N + N - x) \\
&\overset{\Phi}{/\!/\!/} 1 \;+\; [x \le 0] \cdot 0 \;+\; [0 < x] \cdot \left(2 + t + [x = 0] \cdot \infty + [x \le N] \cdot \tfrac{2N}{x}\right) \\
&\quad \texttt{while}\,(\,0 < x\,)\,\{ \\
&\qquad \overset{=}{/\!/\!/} 2 \;+\; t \;+\; [x = 0] \cdot \infty \;+\; [x \le N] \cdot \tfrac{2N}{x} \\
&\qquad \overset{\text{ert}}{/\!/\!/} 1 \;+\; 1 \;+\; t \;+\; [x = 0] \cdot \infty \;+\; [x < N + 1] \cdot 2 \max\left\{\tfrac{N}{x},\, 1\right\} \\
&\qquad\quad i := N + 1 \,; \\
&\qquad \overset{\text{ert}}{/\!/\!/} 1 \;+\; t \;+\; [x = 0] \cdot \infty \;+\; [x < i] \cdot 2 \max\left\{\tfrac{N}{x},\, 1\right\} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by [Hark et al. 2019, App. F.2, Lemma 86])} \\
&\qquad\quad \texttt{while}\,(\,x < i\,)\,\{\, i := \text{Unif}[1..N]\,\} \,; \\
&\qquad \overset{=}{/\!/\!/} \underbrace{1 \;+\; [1 < x \le N + 1] \cdot N\left(\mathcal{H}_x - \tfrac{1}{x}\right) \;+\; [N + 1 < x] \cdot (N\mathcal{H}_N + N - x + 1)}_{=:t} \\
&\qquad \overset{\text{ert}}{/\!/\!/} 1 \;+\; [0 < x - 1 \le N] \cdot N\mathcal{H}_{x-1} \;+\; [N < x - 1] \cdot (N\mathcal{H}_N + N - (x - 1)) \\
&\qquad\quad x := x - 1 \\
&\qquad\quad [\!] \; [0 < x \le N] \cdot N\mathcal{H}_x \;+\; [N < x] \cdot (N\mathcal{H}_N + N - x)\,\} \\
&/\!/\!/ 0
\end{aligned}
$$

By our above annotations, we have shown that

$$I \;=\; [0 < x \le N] \cdot N\mathcal{H}_x \;+\; [N < x] \cdot (N\mathcal{H}_N + N - x)$$

is indeed a runtime subinvariant of the outer loop. Before we finish proving that I is indeed a lower bound on the expected runtime of the outer loop, let us take a closer look at the meaning of I: If I is a lower bound, the outer loop takes at least expected runtime $N\mathcal{H}_x$ if x is between 1 and N, and expected runtime $N\mathcal{H}_N + N - x$ if x is larger than N. In the second case, the too–large x value suggests that we have to collect more coupons than there are different coupons. So we first collect $N - x$ arbitrary "excess coupons" before we enter the "normal coupon collector mode" and collect the remaining N coupons in expected time $N\mathcal{H}_N$. Indeed, $N\mathcal{H}_x$ (without case analysis) is not *a lower bound on the expected runtime and we would in fact fail to prove its subinvariance.*

For the inner loop, we have used the fact that this loop is a so–called independent and identically distributed loop, *for which* exact *expected runtimes can be determined* [Batz et al. 2018, Theorem 4]. *For more details, see* [Hark et al. 2019, App. F.2, Lemma 86]. *We stress that while in this case we had an* exact *expected runtime for the inner loop available by external techniques, a suitable* underapproximation *of the expected runtime of the inner loop using the technique presented in this paper (*Thm. 46*) would have worked as well. Hence, our technique* is *generally applicable to nested loops.*

At the very top of the above annotations, we push I over the initial assignment, thus verifying $1 + N\mathcal{H}_N$ (and hence also $N\mathcal{H}_N$) as lower bound for the entire expected runtime of C_{cc}.

In order to establish that the subinvariant I is in fact a lower bound, we are still left to prove conditional difference boundedness of I. For this, we first make the following annotations:

$/\!/\!/ \left|I\,[x/x-1] - I(s)\right|$ (I does not depend on i)
$i := N+1$
$/\!/\!/ \left|I\,[x/x-1] - I(s)\right|$ (by almost-sure term. of inner loop and [Batz et al. 2018, Lemma 1])
$\texttt{while}\,(\,x < i\,)\,\{\,i := \mathrm{Unif}[1..N]\,\}$
$/\!/\!/ \left|I\,[x/x-1] - I(s)\right|$
$x := x-1$
$/\!/\!/ \left|I - I(s)\right|$

Now that we have determined $\mathsf{wp}\,[\![\textit{outer loop body}]\!]\,(\left|I - I(s)\right|)$*, we finally bound* ΔI*:*

$$\begin{aligned}
\Delta I &= \lambda s.\ \mathsf{wp}\,[\![\textit{outer loop body}]\!]\,(\left|I - I(s)\right|)\\
&= \lambda s.\ \left|I\,[x/x-1] - I(s)\right|\\
&= [x=1]\cdot N \ +\ [1<x<N]\cdot\tfrac{N}{x} \ +\ [x=N+1]\cdot\left(1+\tfrac{1}{N+1}\right) \ +\ [N+1<x] \qquad \text{(by case analysis)}\\
&\preceq \tfrac{1}{2}+N
\end{aligned}$$

Hence, ΔI is bounded by a constant, as N is constant within *the program C_{cc}. Finally, we would still have to show ${}^{\mathsf{wp}}\Phi_t(I) \lessdot \infty$, which is easily checked and thus omitted here. This concludes our lower bound proof for the coupon collector's problem.*

In the example above, we have verified that $N\mathcal{H}_N$ is a lower bound on the expected runtime of the coupon collector program. This lower bound enjoys several nice properties: For one, our lower bound is an *exact* asymptotic lower bound. Another fact is that our lower bound is a *strict* lower bound. The actual runtime is a bit higher, as we have omitted some constants. This is, however, a desirable fact, as often we are only interested in the asymptotic runtime and do not wish to bother with the constants. Notice further, that we never had to find the limit of any sequence. Loop semantics (be it wp or ert) were all applied only finitely many times in order to verify a tight asymptotic lower bound.[18] All in all, the above example demonstrates the effectiveness of our inductive lower bound rule.

9 RELATED WORK

Weakest preexpectation reasoning. The weakest preexpectation calculus goes back to the predicate transformer calculus by [Dijkstra 1975, 1976], which provides an important tool for qualitative formal reasoning about nonprobabilistic programs. The probabilistic and quantitative analog to predicate transformers for nonprobabilistic programs are *expectation transformers* for probabilistic programs. Weakest–preexpectation–style reasoning was first studied in seminal work on probabilistic propositional dynamic logic (PPDL) by [Kozen 1983, 1985]. Its box– and diamond–modalities provide probabilistic versions of Dijkstra's weakest (liberal) preconditions. Amongst others, [Jones 1990], [Morgan et al. 1996], [McIver and Morgan 2005], and [Hehner 2011] have furthered this line of research, e.g., by considering nondeterminism and proof rules for bounding preexpectations in the presence of loops. Work towards automation of weakest preexpectation reasoning was carried out, amongst others, by [Chen et al. 2015], [Cock 2014], [Katoen et al. 2010], and [Feng et al. 2017]. Abstract interpretation of probabilistic programs was studied in this setting by [Monniaux 2005].

[18] This is also true for the technique we used for the inner loop.

Bounds on weakest preexpectations. Rules for bounding weakest preexpectations were considered from very early on. Already [Kozen 1983] provides an induction rule for verifying upper bounds. Pioneering work on lower bounds by means of limits of sequences was carried out by [Jones 1990] and later reconsidered by [Audebaud and Paulin-Mohring 2009]. Proof rules that do not make use of limits were studied by [Morgan 1996] and later more extensively in [McIver and Morgan 2005]. An orthogonal approach to lower bounds by means of bounded model checking was explored by [Jansen et al. 2016].

Advanced weakest preexpectation calculi. Apart from reasoning about expected values of random variables at termination of simple pGCL programs, more advanced expectation–based calculi were invented. For instance, [Morgan and McIver 1999] use expectation transformers to reason about temporal logic. More recently, [Olmedo et al. 2018] studies expectation transformers for probabilistic programs with *conditioning*. [Kaminski et al. 2016, 2018; Olmedo et al. 2016] introduce expectation based calculi to reason about expected runtimes of probabilistic programs. [Batz et al. 2019] present a quantitative separation logic together with a weakest preexpectation calculus for verifying probabilistic programs with pointer–access to dynamic memory.

In all of the above works, the rules for lower bounds rely throughout on finding limits of sequences as well as the sequences themselves. In particular, the proof of the (exact) expected runtime of the coupon collector by [Kaminski et al. 2016] requires a fairly complicated sequence, whereas our invariant in Ex. 47 was conceptually fairly easy and thus more informative for a human.

Martingale–based reasoning. Probabilistic program analysis using martingales was pioneered by [Chakarov and Sankaranarayanan 2013]. Our rules rely on the notions of *uniform integrability* and *conditional difference boundedness* as well as the *Optional Stopping Theorem*. Previous works have also used these notions. [Barthe et al. 2016] focus on synthesizing *exact* martingale expressions. [Fioriti and Hermanns 2015] develop a type system for uniform integrability in order to prove (positive) almost–sure termination[19] of probabilistic programs and give upper bounds on the expected runtime. [Fu and Chatterjee 2019] give lower bounds on expected runtimes. [Kobayashi et al. 2018] provide a semi–decision procedure for lower bounding termination probabilities of probabilistic higher–order recursive programs. [Ngo et al. 2018] perform automated template–driven resource analysis, but infer upper bounds only.

The latter four works analyze the termination behavior of a probabilistic program, whereas we focus on *general* expected values, e.g., of program variables. Furthermore, we do not only *make use* of uniform integrability and/or conditional difference boundedness of some auxiliary stochastic process in order to prove soundness of our proof rules but establish tight connections between expectation–based reasoning via induction and martingale–based reasoning.

Other work on probabilistic program analysis by specialized kinds of martingales includes [Chakarov and Sankaranarayanan 2014], [Chatterjee et al. 2016], [Chatterjee et al. 2017], [Agrawal et al. 2018], [Huang et al. 2018], [Fu and Chatterjee 2019], and [Wang et al. 2019]. For instance, regarding expected runtimes of probabilistic (and possibly nondeterministic) programs, [Fu and Chatterjee 2019] construct *difference bounded* (as opposed to *conditionally* difference bounded, which is a strictly weaker requirement) supermartingales which have to correspond to the *exact* asymptotic expected runtime. In contrast, our rule allows for reasoning about *strict* lower bounds.

10 CONCLUSION

In this paper, we have studied proof rules for lower bounds in probabilistic program verification. Our rules are *simple* in the sense that the invariants need to be "pushed through the loop semantics"

[19] Termination with probability 1 (within finite expected time).

only a *finite* number of times, much like invariants in Hoare logic. In contrast, existing rules for lower bounds of unbounded weakest preexpectations required coming up with an infinite *sequence* of invariants, performing induction to prove relative inductiveness of two subsequent invariants, and then — most unpleasantly — finding the limit of this sequence. The main results of this paper are the following:

(1) We have presented the first *inductive proof rules* (Thm. 37 (a) and (b)) for verifying *lower bounds on (possibly unbounded) weakest preexpectations of probabilistic while loops* using quantitative *invariants*. Our inductive rules are given as an *Optional Stopping Theorem (OST) for weakest preexpectations*. They provide sufficient conditions for the requirement of *uniform integrability* which are much easier to check than uniform integrability in general. Case studies demonstrating the effectiveness but also the limitations of these rules are found in [Hark et al. 2019, App. A].
(2) For proving our OST, we resort to the classical OST from probability theory. However, for most notions that appear in the classical OST, like *uniform integrability* and *conditional difference boundedness*, we were able to find purely expectation–transformer–based counterparts (see Sect. 4 and 5). We thus conjecture that our OST can be proven in purely expectation–theoretic terms, which would most likely simplify the proof of our OST significantly as no probability theory would be required anymore.
(3) We studied the inductive proof rules for lower bounds on *bounded* weakest preexpectations from [McIver and Morgan 2005]. Our results gave rise to a generalization of their proof rule to a *sufficient and necessary* criterion for lower bounds. (Thm. 41).
(4) We have investigated a measure theoretical explanation for why verifying upper bounds using domain theoretical Park induction is conceptually simpler (Sect. 7). The underlying reason is the well–known *Lemma of Fatou*. This leads us to speculate that Fatou's Lemma could be proved in purely domain theoretical terms, perhaps as an instance of Park induction. A successful attempt at a similar idea is due to [Baranga 1991] who proved that the well–known *Banach Contraction Principle* is a particular instance of the Kleene Fixed Point Theorem.
(5) We used the close connection between wp and ert to present the first inductive proof rule for lower bounding expected runtimes (Thm. 46). As an example to demonstrate the power of this rule, we inferred a nontrivial lower bound on the expected runtime of the famous coupon collector's problem (Ex. 47).

Future work includes extending our proof rules for weakest preexpectation reasoning to recursive programs [Olmedo et al. 2016], to probabilistic programs with nondeterminism [McIver and Morgan 2001, 2005], and to *mixed–sign* postexpectations. For the latter, this will likely yield more appealing proof rules for loops than those provided in [Kaminski and Katoen 2017] which currently involve reasoning about sequences. Moreover, we are interested in (partially) automating the synthesis of the quantitative invariants needed in our proof rules.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the German Research Council (DFG) Research Training Group 2236 UnRAVeL and ERC Advanced Grant 787914 FRAPPANT. Furthermore, we would like to thank Florian Frohn and Christoph Matheja for many fruitful discussions on examples and counterexamples.

REFERENCES

Sheshansh Agrawal, Krishnendu Chatterjee, and Petr Novotný. 2018. Lexicographic Ranking Supermartingales: An Efficient Approach to Termination of Probabilistic Programs. *PACMPL* 2, POPL (2018), 34:1–34:32.

Philippe Audebaud and Christine Paulin-Mohring. 2009. Proofs of Randomized Algorithms in Coq. *Science of Computer Programming* 74, 8 (2009), 568–589.

Ralph-Johan Back and Joakim von Wright. 1998. *Refinement Calculus - A Systematic Introduction.* Springer.

Andrei Baranga. 1991. The Contraction Principle as a Particular Case of Kleene's Fixed Point Theorem. *Discrete Mathematics* 98, 1 (1991), 75–79.

Gilles Barthe, Thomas Espitau, Luis María Ferrer Fioriti, and Justin Hsu. 2016. Synthesizing Probabilistic Invariants via Doob's Decomposition. In *Proc. of the International Conference on Computer–Aided Verification (CAV) (Lecture Notes in Computer Science)*, Vol. 9779. Springer, 43–61.

Kevin Batz, Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Thomas Noll. 2019. Quantitative Separation Logic: a Logic for Reasoning about Probabilistic Pointer Programs. *PACMPL* 3, POPL (2019), 34:1–34:29.

Kevin Batz, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. 2018. How Long, O Bayesian Network, will I Sample Thee? - A Program Analysis Perspective on Expected Sampling Times. In *Proc. of the European Symposium on Programming Languages and Systems (ESOP) (Lecture Notes in Computer Science)*, Vol. 10801. Springer, 186–213.

Heinz Bauer. 1971. *Probability Theory and Elements of Measure Theory* (first english ed.). Holt, Rinehart and Winston, Inc., New York.

Aleksandar Chakarov and Sriram Sankaranarayanan. 2013. Probabilistic Program Analysis with Martingales. In *Proc. of the International Conference on Computer–Aided Verification (CAV) (Lecture Notes in Computer Science)*, Vol. 8044. Springer, 511–526.

Aleksandar Chakarov and Sriram Sankaranarayanan. 2014. Expectation Invariants for Probabilistic Program Loops as Fixed Points. In *Proc. of the Static Analysis Symposium (SAS) (Lecture Notes in Computer Science)*, Vol. 8723. Springer, 85–100.

Krishnendu Chatterjee, Hongfei Fu, Petr Novotný, and Rouzbeh Hasheminezhad. 2016. Algorithmic Analysis of Qualitative and Quantitative Termination Problems for Affine Probabilistic Programs. In *Proc. of the Symposium on Principles of Programming Languages (POPL).* ACM, 327–342.

Krishnendu Chatterjee, Petr Novotný, and Dorde Zikelic. 2017. Stochastic Invariants for Probabilistic Termination. In *Proc. of the Symposium on Principles of Programming Languages (POPL).* ACM, 145–160.

Yu-Fang Chen, Chih-Duo Hong, Bow-Yaw Wang, and Lijun Zhang. 2015. Counterexample–Guided Polynomial Loop Invariant Generation by Lagrange Interpolation. In *Proc. of the International Conference on Computer–Aided Verification (CAV) (Lecture Notes in Computer Science)*, Vol. 9206. Springer, 658–674.

David Cock. 2014. pGCL for Isabelle. *Archive of Formal Proofs* (2014).

Edsger Wybe Dijkstra. 1975. Guarded Commands, Nondeterminacy and Formal Derivation of Programs. *Commun. ACM* 18, 8 (1975), 453–457.

Edsger Wybe Dijkstra. 1976. *A Discipline of Programming.* Prentice–Hall.

William Feller. 1971. *An Introduction to Probability Theory and its Applications. Vol. II.* John Wiley & Sons.

Yijun Feng, Lijun Zhang, David Nicolaas Jansen, Naijun Zhan, and Bican Xia. 2017. Finding Polynomial Loop Invariants for Probabilistic Programs. In *Proc. of the International Symposium on Automated Technology for Verification and Analysis (ATVA) (Lecture Notes in Computer Science)*, Vol. 10482. Springer, 400–416.

Luis María Ferrer Fioriti and Holger Hermanns. 2015. Probabilistic Termination: Soundness, Completeness, and Compositionality. In *Proc. of the Symposium on Principles of Programming Languages (POPL).* ACM, 489–501.

Florian Frohn, Matthias Naaf, Jera Hensel, Marc Brockschmidt, and Jürgen Giesl. 2016. Lower Runtime Bounds for Integer Programs. In *Proc. of the International Joint Conference on Automated Reasoning (IJCAR) (Lecture Notes in Computer Science)*, Vol. 9706. Springer, 550–567.

Hongfei Fu and Krishnendu Chatterjee. 2019. Termination of Nondeterministic Probabilistic Programs. In *Proc. of the International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI) (Lecture Notes in Computer Science)*, Vol. 11388. Springer, 468–490.

Andrew D. Gordon, Thomas A. Henzinger, Aditya Vithal Nori, and Sriram K. Rajamani. 2014. Probabilistic Programming. In *Proc. of Future of Software Engineering (FOSE).* ACM, 167–181.

Geoffrey Grimmett and David Stirzaker. 2001. *Probability and Random Processes.* Oxford University Press, Oxford; New York.

Marcel Hark, Benjamin Lucien Kaminski, Jürgen Giesl, and Joost-Pieter Katoen. 2019. Aiming Low Is Harder - Inductive Proof Rules for Lower Bounds on Weakest Preexpectations in Probabilistic Program Verification. *CoRR* abs/1904.01117 (2019). arXiv:1904.01117

Eric Charles Roy Hehner. 2011. A Probability Perspective. *Formal Aspects of Computing* 23, 4 (2011), 391–419.

Wataru Hino, Hiroki Kobayashi, Ichiro Hasuo, and Bart Jacobs. 2016. Healthiness from Duality. In *Proc. of the Annual Symposium on Logic in Computer Science (LICS).* ACM, 682–691.

Mingzhang Huang, Hongfei Fu, and Krishnendu Chatterjee. 2018. New Approaches for Almost–Sure Termination of Probabilistic Programs *(Lecture Notes in Computer Science)*, Vol. 11275. Springer, 181–201.

Jacek Jachymski, Leslaw Gajek, and Piotr Pokarowski. 2000. The Tarski–Kantorovitch Principle and the Theory of Iterated Function Systems. *Bulletin of the Australian Mathematical Society* 61, 2 (2000), 247–261.

Nils Jansen, Christian Dehnert, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Lukas Westhofen. 2016. Bounded Model Checking for Probabilistic Programs. In *Proc. of the International Symposium on Automated Technology for Verification and Analysis (ATVA) (Lecture Notes in Computer Science)*, Vol. 9938. Springer, 68–85.

Claire Jones. 1990. *Probabilistic Non–Determinism.* Ph.D. Dissertation. University of Edinburgh, UK.

Benjamin Lucien Kaminski. 2019. *Advanced Weakest Precondition Calculi for Probabilistic Programs.* Ph.D. Dissertation. RWTH Aachen University, Germany. http://publications.rwth-aachen.de/record/755408/files/755408.pdf

Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. 2019. On the Hardness of Analyzing Probabilistic Programs. *Acta Inf.* 56, 3 (2019), 255–285.

Benjamin Lucien Kaminski and Joost-Pieter Katoen. 2017. A Weakest Pre–expectation Semantics for Mixed–sign Expectations. In *Proc. of the Annual Symposium on Logic in Computer Science (LICS).* IEEE Computer Society, 1–12.

Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. 2016. Weakest Precondition Reasoning for Expected Run–Times of Probabilistic Programs. In *Proc. of the European Symposium on Programming Languages and Systems (ESOP) (Lecture Notes in Computer Science)*, Vol. 9632. Springer, 364–389.

Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. 2018. Weakest Precondition Reasoning for Expected Runtimes of Randomized Algorithms. *Journal of the ACM* 65 (2018).

Joost-Pieter Katoen, Annabelle McIver, Larissa Meinicke, and Carroll Morgan. 2010. Linear-Invariant Generation for Probabilistic Programs: Automated Support for Proof–Based Methods. In *Proc. of the Static Analysis Symposium (SAS) (Lecture Notes in Computer Science)*, Vol. 6337. Springer, 390–406.

Klaus Keimel. 2015. Healthiness Conditions for Predicate Transformers. *Electr. Notes Theor. Comput. Sci.* 319 (2015), 255–270.

Naoki Kobayashi, Ugo Dal Lago, and Charles Grellois. 2018. On the Termination Problem for Probabilistic Higher-Order Recursive Programs. *CoRR* abs/1811.02133 (2018). arXiv:1811.02133

Dexter Kozen. 1983. A Probabilistic PDL. In *Proc. of the Annual Symposium on Theory of Computing (STOC).* 291–297.

Dexter Kozen. 1985. A Probabilistic PDL. *J. Comput. System Sci.* 30, 2 (1985), 162–178.

Jean-Louis Lassez, V. L. Nguyen, and Liz Sonenberg. 1982. Fixed Point Theorems and Semantics: A Folk Tale. *Inform. Process. Lett.* 14, 3 (1982), 112–116.

Annabelle McIver and Carroll Morgan. 2001. Partial Correctness for Probabilistic Demonic Programs. *Theoretical Computer Science* 266, 1-2 (2001), 513–541.

Annabelle McIver and Carroll Morgan. 2005. *Abstraction, Refinement and Proof for Probabilistic Systems.* Springer.

David Monniaux. 2005. Abstract Interpretation of Programs as Markov Decision Processes. *Science of Computer Programming* 58, 1–2 (2005), 179–205.

Carroll Morgan. 1996. Proof Rules for Probabilistic Loops. In *Proc. of BCS–FACS 7th Refinement Workshop.*

Carroll Morgan and Annabelle McIver. 1999. An Expectation–Transformer Model for Probabilistic Temporal Logic. *Logic Journal of the Interest Group in Pure and Applied Logics* 7, 6 (1999), 779–804.

Carroll Morgan, Annabelle McIver, and Karen Seidel. 1996. Probabilistic Predicate Transformers. *ACM Trans. on Programming Languages and Systems* 18, 3 (1996), 325–353.

Rajeev Motwani and Prabhakar Raghavan. 1995. *Randomized Algorithms.* Cambridge University Press.

Van Chan Ngo, Quentin Carbonneaux, and Jan Hoffmann. 2018. Bounded Expectations: Resource Analysis for Probabilistic Programs. In *Proc. of the Conference on Programming Language Design and Implementation (PLDI).* ACM, 496–512.

Federico Olmedo, Friedrich Gretz, Nils Jansen, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Annabelle McIver. 2018. Conditioning in Probabilistic Programming. *ACM Trans. on Programming Languages and Systems* 40, 1 (2018), 4:1–4:50.

Federico Olmedo, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. 2016. Reasoning about Recursive Probabilistic Programs. In *Proc. of the Annual Symposium on Logic in Computer Science (LICS).* ACM, 672–681.

David Park. 1969. Fixpoint Induction and Proofs of Program Properties. *Machine Intelligence* 5 (1969).

George Pólya. 1930. Eine Wahrscheinlichkeitsaufgabe in der Kundenwerbung. *Zeitschrift für Angewandte Mathematik und Mechanik* 10, 1 (1930), 96–97.

Moshe Ya'akov Vardi. 1985. Automatic Verification of Probabilistic Concurrent Finite–State Programs. In *Proc. of the Annual Symposium on Foundations of Computer Science (FOCS).* IEEE Computer Society, 327–338.

Peixin Wang, Hongfei Fu, Amir Kafshdar Goharshady, Krishnendu Chatterjee, Xudong Qin, and Wenjun Shi. 2019. Cost Analysis of Nondeterministic Probabilistic Programs. In *Proc. of the Conference on Programming Language Design and Implementation (PLDI).* ACM, 204–220.