Weakly Non-negative Supermartingales for Omega-Regular Verification

Toru Takisaka, Hongjie Qing, Libo Zhang

Martingale-based methods are central to probabilistic program verification, but strong global non-negativity requirements can exclude simple certificates from tractable template classes. Relaxing this requirement enlarges the search space for automated synthesis, but naive relaxations are unsound in the probabilistic setting. We introduce *Lazy Streett supermartingales* and their lexicographic extension, showing that weak non-negativity can nevertheless be used soundly to certify almost-sure satisfaction of ω-regular properties with polynomial templates under a broad class of sampling distributions, including all bounded-support distributions. This extends prior weakly non-negative methods from termination to general ω-regular verification. We further give a compositional account of lexicographic certificates in terms of one-dimensional ones. Experiments on 170 polynomial probabilistic-program benchmarks show increases of 20.0–23.5 percentage points in verification success over the strongly non-negative baseline.

This is an extended abstract: the full paper can be found on arXiv.

1 Introduction

Background: Martingale-based methods and their algorithmic applicability

Verification of probabilistic programs has been actively studied in recent years. *Martingale-based methods* are among the central techniques for such verification tasks. The basic idea is to find a real-valued function over program states—typically a supermartingale-like certificate—whose value decreases, or is otherwise controlled, in expectation along program executions. Such a function can serve as a certificate for properties such as almost-sure termination [2, 3, 6, 8, 9, 11, 14], recurrence [4], and, more recently, almost-sure satisfaction of ω-regular specifications [1,7,10]. For example, a *ranking supermartingale* (RSM) [3] is a non-negative real-valued function whose value decreases by a positive constant in expectation across every transition from a non-terminal state. Such a function witnesses that the underlying probabilistic program is almost surely terminating.

A recurring theme in these developments is how to improve the *algorithmic applicability* of martingale-based methods. Synthesis algorithms typically restrict their search space to a tractable subclass of supermartingales, such as linear or polynomial ones. Hence, even if a program admits a supermartingale that witnesses the target property, a synthesis algorithm may fail to find one if all such witnesses lie outside its prescribed search space. By algorithmic applicability, we mean the class of programs for which such a synthesis procedure can automatically construct a witnessing supermartingale.

---

* ω-正則検証のための弱非負優マルチンゲール

This is an unrefereed paper. Copyrights belong to the Author(s).

Toru Takisaka, Hongjie Qing, 電子科技大学 計算機科学・工学部, School of Computer Science and Engineering, University of Electronic Science and Technology of China.

Libo Zhang, オークランド大学 計算機科学部, School of Computer Science, The University of Auckland.

```
    y ← 0;
ℓ:  while x ≥ 0 do
    y ← y + 1;
    x ← x − y
    od
```

図 1 A non-probabilistic program admitting a linear weakly non-negative ranking function.

From this viewpoint, one useful way to improve algorithmic applicability is to relax the non-negativity condition imposed on supermartingales. For example, a linear function $\eta(x, y) = x$ is a "weakly non-negative ranking function" that witnesses termination of the non-probabilistic program in Figure 1. Focusing on the value of η at location ℓ, we see that it satisfies the usual requirements of a ranking function—the non-probabilistic counterpart of an RSM—except for a mild violation of global non-negativity: the value of η decreases by at least one in each loop iteration and is non-negative whenever the loop guard $x \geq 0$ is satisfied, *but* it can take an arbitrarily negative value once the loop guard is violated. In fact, the program also admits a globally non-negative ranking function $\eta'(x, y) = \max\{x + 1, 0\}$, but this function is not linear and hence can be harder than η for synthesis algorithms to find. Weak non-negativity has been actively studied in the context of *lexicographic extensions* of ranking functions and supermartingales, which provide another well-known way to improve algorithmic applicability. Recent work reports experimentally that weak non-negativity in lexicographic RSMs improves the verification success rate by 15.6% [14].

Challenge: Subtlety of weak non-negativity in martingales

This weakening of non-negativity is known to be substantially more delicate in the probabilistic setting: a naive transfer of the non-probabilistic intuition can lead to an unsound martingale notion.

```
    x ← 0; y ← 1;
ℓ:  while x ≥ 0 do
      y ← y + 1;
      if prob(2^−y) then
        x ← −2^y
      fi
    od
```

図 2 A counterexample.

The program in Figure 2 is a known counterexample [6, 15]. The function $\eta(x, y) = x$ satisfies properties analogous to those in the non-probabilistic example above: in each loop iteration, the value of η decreases by one in expectation, namely from 0 to $-2^y \times 2^{-y} = -1$, and it is non-negative whenever the loop guard $x \geq 0$ is satisfied. However, the program does not terminate almost surely: a simple calculation shows that the termination probability is over-approximated by $2^{-2} + 2^{-3} + \cdots = 0.5$. Thus, such a "weakly non-negative RSM" is not sound in general as a certificate of almost-sure termination. The issue arises because a supermartingale condition concerns an *expected value* after a program transition, so negative values can be exploited in a spurious way.

Nevertheless, prior work has shown that this obstruction can sometimes be overcome. A recent paper [14] observed that synthesis algorithms search only within restricted classes of programs and RSMs, and that these restrictions can rule out counterexamples such as the one above. For example, probabilistic branching with a variable probability, such as '**if prob**(x)', plays a crucial role in the counterexample above, whereas existing synthesis algorithms often allow only '**if prob**(q)' with a constant $q \in [0, 1]$. It has been shown in [14] that weakly non-negative RSMs, as well as their lexicographic extension called *lazy lexicographic RSMs*, are sound under suitable assumptions.

The idea developed in [14]—that weakly non-

negative martingales are unsound in general but can become sound under restrictions induced by synthesis algorithms—is still far from fully explored. Their soundness proof is formulated for a reachability property, namely termination; it assumes linear templates; and it relies on specific regularity assumptions on the sampling distributions, called *well-behavedness*. Moreover, the proof is rather intricate: it directly analyzes a multi-dimensional lexicographic object and uses a delicate argument showing how negative values can be "fixed" without destroying the ranking condition. This makes it difficult to adapt their results to different settings, even when such an adaptation is in principle possible.

Contributions

This paper pushes the idea of [14] further and significantly generalizes their results. This paper is an extended abstract that reports a summary of the results: Full formal definitions and proofs, together with the program-MDP semantics, synthesis procedure, experimental evaluation, and related work, are provided in the arXiv preprint corresponding to this paper [13].

We propose a novel martingale notion, called a *Lazy Streett Supermartingale* (Lazy SSM), and its lexicographic extension, called a *Lazy Lexicographic SSM* (Lazy LexSSM). This notion combines two existing martingale notions: Lazy (lexicographic) RSMs [14], RSM notions with the most liberal non-negativity condition to date; and *Streett Supermartingales* (SSMs) [1, 10], recently proposed martingale certificates for almost-sure satisfaction of ω-regular properties.

Table 1 compares the scopes of Lazy RSMs and Lazy SSMs. First, Lazy RSMs certify only almost-sure termination of probabilistic programs, whereas Lazy SSMs certify almost-sure satisfaction of ω-regular properties. Second, the soundness result in [14] assumes that Lazy RSMs are linear; we prove soundness for polynomial Lazy SSMs, a common template class in recent verification algorithms [1,5,7,12]. Third, the existing result requires program arithmetic to be linear: that is, a function f in a program line $x \leftarrow f(\boldsymbol{x})$ must be linear. We allow f to be any measurable function. Fourth, the existing result requires sampling distributions to be *well-behaved*; we replace this assumption with what we call *relative well-behavedness*. This last generalization becomes essential for polynomial extension of soundness, as discussed below.

The soundness proof developed in this paper requires reasoning *that is fundamentally different from the existing proof for Lazy RSMs*, mainly because we move from linear to polynomial templates. The extension from reachability to ω-regular properties is conceptually handled by the Streett-supermartingale framework, but the polynomial-template extension raises a separate and more subtle technical challenge: the well-behavedness condition used in the existing Lazy RSM proof becomes too restrictive in the polynomial setting. It even rules out uniform distributions over intervals, thereby severely limiting the practical relevance of the resulting soundness theorem. To overcome this obstacle, we develop a different proof strategy based on *relative well-behavedness*. We show that every bounded-support distribution is relatively well-behaved for polynomial templates, which yields a practically meaningful class of applicable probabilistic programs. We explain this phenomenon in the first part of the technical overview below.

Our technical development also introduces several ideas that make the analysis simpler and better structured. First, we isolate the main technical difficulty in the soundness proof for *single-dimensional* Lazy SSMs, and then derive soundness of their lexicographic extension as a simple corollary. The key observation is that a lexicographic

表 1 **Comparison of lazy lexicographic supermartingale techniques.**

	Lazy LexRSM [14]	Lazy LexSSM (this paper)
Certified property	Reachability	ω-regular properties
Template class	Linear	Polynomial
Program updates	Linear updates	Measurable updates
Assumption on sampling distributions	Well-behavedness	Relative well-behavedness

SSM is simply a sequence of single-dimensional SSMs that jointly certify the target ω-regular property; the second part of the technical overview develops this observation. As a byproduct, we present a "lazy" variant of *lexicographic progress-measure supermartingales* [10], another form of multi-dimensional supermartingale.

Second, we formulate our results over a new semantic model, which we call *program MDPs*, rather than directly over the conventional *probabilistic control flow graph* formalism. The soundness of Lazy RSMs implicitly relies on additional properties of pCFGs that do not hold for general MDPs, and the pCFG formalism makes it difficult to identify which of these properties are essential. Program MDPs abstract the relevant properties of pCFGs as explicit axioms. This gives a clearer account of the scope of our soundness results, while also allowing us to use these axioms to develop sound weakly non-negative martingale notions.

To examine the practical benefit of weak non-negativity, we performed experiments on 170 synthetic benchmarks of polynomial probabilistic programs. Our synthesis algorithm based on Lazy LexSSMs achieved verification-success rates 20.0–23.5 percentage points higher than those of the strongly non-negative counterpart [10].

2 Technical Overview

2.1 Relative Well-behavedness for Polynomial Templates

We use examples to explain well-behavedness, its role in the soundness of linear Lazy RSMs, why the analogous argument fails for polynomial templates, and how relative well-behavedness resolves the problem.

What are well-behaved distributions?

As an example of a well-behaved distribution, we take $\mathrm{Unif}[0,1]$, i.e., the uniform distribution over $[0,1]$. We explain the idea informally in the context of program execution.

Suppose we have a probabilistic program that involves a program line '$\ell:\ x \sim \mathrm{Unif}[0,1]$', which randomly updates a variable x to a value sampled from $\mathrm{Unif}[0,1]$. Write $p = \mathrm{Unif}[0,1]$. Also suppose the program involves two variables (x, y), for simplicity. Then an execution of '$\ell:\ x \sim \mathrm{Unif}[0,1]$' will perform the following update to the program state (ℓ, x, y), where ℓ' is the successor location of ℓ:

$$(\ell, x, y) \xrightarrow{\ell: x \sim p} (\ell', x', y), \qquad x' \sim \mathrm{Unif}[0,1].$$

Now suppose a linear function $\eta_{\ell'}(x, y)$ is associated with ℓ' as part of an RSM. An RSM is a function η over program states, or equivalently, a collection $(\eta_\ell)_{\ell \in L}$ of functions over variables, one for each program location. For example, let $\eta_{\ell'}(x, y) = x - y$. After the above execution, its value is updated to

$$\eta_{\ell',y}(x') = x' - y, \qquad x' \sim \mathrm{Unif}[0,1].$$

Two quantities are relevant: $\Pr(\eta_{\ell',y} < 0)$, the probability of observing $\eta_{\ell',y}(x') < 0$, and $\int_0^1 (\eta_{\ell',y})^-\, dx$, its *mean shortfall*, where $f^- := \max\{-f, 0\}$. These quantities depend on $y \in \mathbb{R}$.

When $y \in (0,1)$, we have

$$\int_0^1 (\eta_{\ell',y})^- \, dx = \int_0^y (y-x)\, dx = \frac{1}{2}\big(\Pr(\eta_{\ell',y} < 0)\big)^2.$$

Thus, for any sequence $y_0, y_1, \ldots$ of reals, the following implication holds:

$$\Pr(\eta_{\ell',y_t} < 0) \longrightarrow 0 \implies \int_0^1 (\eta_{\ell',y_t})^- \, dx \longrightarrow 0. \tag{1}$$

The same convergence relationship holds for every linear $\eta_{\ell'}$. We call this property the *well-behavedness* of Unif$[0,1]$; the full notion additionally constrains the relative convergence rates of the probability and the mean shortfall. Well-behavedness is shown to be a mild requirement: any bounded-support distribution, as well as some unbounded-support ones such as normal distributions, are well-behaved [15, Proposition C.6].

How is it related to the soundness of linear lazy RSMs?

Intuitively, well-behavedness of a distribution p prevents linear lazy RSMs from spuriously exploiting unbounded negative values at a program line '$x \sim p$'. The crucial update pattern in the counterexample from Figure 2 is as follows:

(a) The probability that $\eta(x,y) = x$ becomes negative in the t-th loop iteration is $2^{-(t+1)}$ and hence converges to zero.

(b) Its mean shortfall remains constant: in the t-th iteration it is $2^{t+1} \cdot 2^{-(t+1)} = 1$.

Well-behavedness of p rules out this update pattern along repeated executions of '$x \sim p$' when η is linear. Indeed, (1) says that conditions a and b cannot hold simultaneously.

Why does the analogous polynomial argument fail?

The notion above is tailored to linear lazy RSMs: it asks whether (1) holds for arbitrary linear $\eta_{\ell'}$. A direct extension to polynomial lazy RSMs would impose the same condition on every polynomial $\eta_{\ell'}$.

This requirement is too restrictive: even Unif$[0,1]$ fails it. Consider three variables (x,y,z) and $\eta_{\ell'}(x,y,z) = xy + z$. Setting $y_t = 4^t$ and $z_t = -2^t$ gives

$$\eta_{\ell',y_t,z_t}(x) = 4^t x - 2^t,$$
$$\Pr(\eta_{\ell',y_t,z_t} < 0) \longrightarrow 0,$$
$$\int_0^1 (\eta_{\ell',y_t,z_t})^- \, dx = \frac{1}{2}.$$

Hence (1) fails although the sampling distribution is bounded and elementary.

Our solution: Transition to relative well-behavedness

The example gives a sequence $(y_t, z_t)_{t=0}^{\infty}$—and hence a sequence $(\ell, x_t, y_t, z_t)_{t=0}^{\infty}$ of program states—along which the polynomial $\eta_{\ell'}$ violates (1). This does not by itself threaten the soundness of polynomial Lazy RSMs: the problematic sequence must actually arise along a program run. In fact, there is another fact that prevents such a sequence from appearing freely along a program run:

$$\int_0^1 \eta_{\ell',y_t,z_t} \, dx \longrightarrow +\infty. \tag{2}$$

Because $\eta_{\ell'}$ is supposed to be a part of polynomial lazy RSM, this means we also have $\eta_\ell(x_t, y_t, z_t) \to +\infty$ as $t \to \infty$. This cannot happen "for free" in a program run because the ranking condition of η forces its value to keep decreasing through a program run, in expectation.

Our key findings are twofold. First, this trade-off—any sequence that violates (1) necessarily satisfies (2)—holds for arbitrary polynomial templates, not only under Unif$[0,1]$ but under every bounded-support distribution p. We formalize this property as *relative well-behavedness*; its full definition also includes a convergence-rate requirement. Every bounded-support distribution is relatively well-behaved for polynomial templates.

Second, relative well-behavedness opens a different route to proving soundness of polynomial lazy RSMs and, more generally, polynomial Lazy SSMs. The core technical step is to use the *fractional-power transform* $(\eta^+)^\theta$ of a lazy RSM η as a witness

of almost-sure termination. Combining relative well-behavedness with Jensen's inequality yields a key inequality that transfers the ranking condition of η to $(\eta^+)^\theta$.

2.2 Compositional Soundness for Lexicographic Martingales

We next explain how several one-dimensional supermartingales can jointly certify an ω-regular property, and why the resulting object is precisely a lexicographic supermartingale.

Recall that a *Streett condition* is characterized by a pair $(S_{\mathsf{Fin}}, S_{\mathsf{Inf}})$ of sets of program states, called a *Streett pair*; for simplicity, assume that the two sets are disjoint. A run $s_0 s_1 \dots$ satisfies this condition if it visits S_{Fin} only finitely often or visits S_{Inf} infinitely often:

$$\big(\exists N.\, \forall t \geq N.\ s_t \notin S_{\mathsf{Fin}}\big) \quad \text{or} \quad \big(\forall N.\, \exists t \geq N.\ s_t \in S_{\mathsf{Inf}}\big). \tag{3}$$

For any $S' \subseteq S \setminus S_{\mathsf{Inf}}$, the conjunction of the two Streett conditions

$$(S', S_{\mathsf{Inf}}), \qquad (S_{\mathsf{Fin}} \setminus S', S_{\mathsf{Inf}} \cup S') \tag{4}$$

implies $(S_{\mathsf{Fin}}, S_{\mathsf{Inf}})$. Indeed, if S_{Inf} is visited only finitely often, the first pair forces S' to be visited only finitely often; the second then forces $S_{\mathsf{Fin}} \setminus S'$ to be visited only finitely often. Consequently, S_{Fin} is visited only finitely often. When $S' \subseteq S_{\mathsf{Fin}}$, the converse implication also holds, so the decomposition is exact.

More generally, let $S_1, \dots, S_k$ be a partition of S_{Fin}. Define

$$S^{(i)}_{\mathsf{Fin}} = S_i, \tag{5}$$

$$S^{(i)}_{\mathsf{Inf}} = S_{\mathsf{Inf}} \cup \bigcup_{j<i} S_j. \tag{6}$$

Then $(S_{\mathsf{Fin}}, S_{\mathsf{Inf}})$ is equivalent to the conjunction of the Streett pairs $(S^{(i)}_{\mathsf{Fin}}, S^{(i)}_{\mathsf{Inf}})$ for $i = 1, \dots, k$. This decomposition suggests an incremental approach to ω-regular verification using *Streett supermartingales*[†1] (SSMs) [1,10], or their lazy variants. If synthesis fails for $(S_{\mathsf{Fin}}, S_{\mathsf{Inf}})$ directly, it can first seek an SSM for (S', S_{Inf}) with a large subset $S' \subseteq S_{\mathsf{Fin}}$, and then continue with $(S_{\mathsf{Fin}} \setminus S', S_{\mathsf{Inf}} \cup S')$. If successful, this process produces a sequence $\eta_1, \dots, \eta_k$ of SSMs that jointly certifies $(S_{\mathsf{Fin}}, S_{\mathsf{Inf}})$.

Our observation is that such a sequence is exactly the *lexicographic Streett supermartingale* of [10], and that the incremental construction above coincides with its standard synthesis strategy. Other multi-dimensional martingales admit the same explanation: lexicographic ranking supermartingales [2], progress-measure supermartingales, and their lexicographic extension [10] can all be viewed as sequences of one-dimensional SSMs jointly verifying the target property. Accordingly, a Lazy LexSSM can be defined compositionally as a sequence of one-dimensional Lazy SSMs.

This viewpoint substantially simplifies soundness proofs. Conventional arguments directly analyze multi-dimensional processes under a non-standard order, such as lexicographic order with a constant gap. For weakly non-negative certificates, this requires especially delicate reasoning about the possible exploitation of unbounded negative values. The compositional view isolates that difficulty in the one-dimensional case: once soundness of a single Lazy SSM is established, soundness of its lexicographic extension follows as a direct corollary.

†1 Streett supermartingales were introduced in [1] and subsequently generalized by relaxing their martingale constraint [10]. We use "Streett supermartingale" for the generalized notion.

参 考 文 献

[1] Abate, A., Giacobbe, M., and Roy, D.: Stochastic Omega-Regular Verification and Control with Supermartingales, *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part III*, Gurfinkel, A. and Ganesh, V.(eds.), Lecture Notes in Computer Science, Vol. 14683, Springer, 2024, pp. 395–419.

[2] Agrawal, S., Chatterjee, K., and Novotný, P.: Lexicographic ranking supermartingales: an efficient approach to termination of probabilistic

programs, *Proc. ACM Program. Lang.*, Vol. 2, No. POPL(2018), pp. 34:1–34:32.

[3] Chakarov, A. and Sankaranarayanan, S.: Probabilistic program analysis with martingales, *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25*, Springer, 2013, pp. 511–526.

[4] Chakarov, A., Voronin, Y., and Sankaranarayanan, S.: Deductive Proofs of Almost Sure Persistence and Recurrence Properties, *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, Chechik, M. and Raskin, J.(eds.), Lecture Notes in Computer Science, Vol. 9636, Springer, 2016, pp. 260–279.

[5] Chatterjee, K., Fu, H., and Goharshady, A. K.: Termination analysis of probabilistic programs through Positivstellensatz's, *Computer Aided Verification: 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part I 28*, Springer, 2016, pp. 3–22.

[6] Chatterjee, K., Goharshady, E. K., Novotný, P., Zárevúcky, J., and Zikelic, D.: On Lexicographic Proof Rules for Probabilistic Termination, *Formal Aspects Comput.*, Vol. 35, No. 2(2023), pp. 11:1–11:25.

[7] Henzinger, T. A., Mallik, K., Sadeghi, P., and Zikelic, D.: Supermartingale Certificates for Quantitative Omega-Regular Verification and Control, *Computer Aided Verification - 37th International Conference, CAV 2025, Zagreb, Croatia, July 23-25, 2025, Proceedings, Part II*, Piskac, R. and Rakamaric, Z.(eds.), Lecture Notes in Computer Science, Vol. 15932, Springer, 2025, pp. 29–55.

[8] Huang, M., Fu, H., Chatterjee, K., and Goharshady, A. K.: Modular verification for almost-sure termination of probabilistic programs, *Proc. ACM Program. Lang.*, Vol. 3, No. OOPSLA(2019), pp. 129:1–129:29.

[9] Kenyon-Roberts, A. and Ong, C. L.: Supermartingales, Ranking Functions and Probabilistic Lambda Calculus, *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, IEEE, 2021, pp. 1–13.

[10] Kura, S. and Unno, H.: A Hierarchy of Supermartingales for ω-Regular Verification, *Proc. ACM Program. Lang.*, Vol. 10, No. PLDI(2026), pp. 253–276.

[11] McIver, A., Morgan, C., Kaminski, B. L., and Katoen, J.-P.: A new proof rule for almost-sure termination, *Proceedings of the ACM on Programming Languages*, Vol. 2, No. POPL(2017), pp. 1–28.

[12] Takisaka, T., Oyabu, Y., Urabe, N., and Hasuo, I.: Ranking and Repulsing Supermartingales for Reachability in Randomized Programs, *ACM Trans. Program. Lang. Syst.*, Vol. 43, No. 2(2021), pp. 5:1–5:46.

[13] Takisaka, T., Qing, H., and Zhang, L.: Weakly Non-negative Supermartingales for Omega-Regular Verification, *CoRR*, Vol. abs/2607.17798(2026).

[14] Takisaka, T., Zhang, L., Wang, C., and Liu, J.: Lexicographic Ranking Supermartingales with Lazy Lower Bounds, *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part III*, Gurfinkel, A. and Ganesh, V.(eds.), Lecture Notes in Computer Science, Vol. 14683, Springer, 2024, pp. 420–442.

[15] Takisaka, T., Zhang, L., Wang, C., and Liu, J.: Lexicographic Ranking Supermartingales with Lazy Lower Bounds, *CoRR*, Vol. abs/2304.11363(2024).