



# Supermartingale Certificates for Quantitative Omega-Regular Verification and Control

Thomas A. Henzinger<sup>1</sup>, Kaushik Mallik<sup>2</sup>, Pouya Sadeghi<sup>3</sup>,  
and Đorđe Žikelić<sup>3</sup>

<sup>1</sup> Institute of Science and Technology Austria (ISTA),  
Klosterneuburg, Austria  
tah@ist.ac.at

<sup>2</sup> IMDEA Software Institute, Madrid, Spain  
kaushik.mallik@imdea.org

<sup>3</sup> Singapore Management University, Singapore, Singapore  
{pouyas,dzikelic}@smu.edu.sg



**Abstract.** We present the first supermartingale certificate for quantitative  $\omega$ -regular properties of discrete-time infinite-state stochastic systems. Our certificate is defined on the product of the stochastic system and a limit-deterministic Büchi automaton that specifies the property of interest; hence we call it a limit-deterministic Büchi supermartingale (LDBSM). Previously known supermartingale certificates applied only to quantitative reachability, safety, or reach-avoid properties, and to qualitative (i.e., probability 1)  $\omega$ -regular properties. We also present fully automated algorithms for the template-based synthesis of LDBSMs, for the case when the stochastic system dynamics and the controller can be represented in terms of polynomial inequalities. Our experiments demonstrate the ability of our method to solve verification and control tasks for stochastic systems that were beyond the reach of previous supermartingale-based approaches.

**Keywords:** Supermartingales · Probabilistic verification · Stochastic control

## 1 Introduction

Stochastic (or probabilistic) systems provide a framework for modeling and quantifying uncertainties in computational models. They are ubiquitous in many areas of computer science, including randomized algorithms [55], security and privacy protocols [8], stochastic networks [30], control theory [6], and artificial intelligence [33]. Many of these systems are safety critical in nature; hence formal methods for guaranteeing their correctness have become an important topic of research.

While the correctness of stochastic systems can be accomplished using a variety of complementary approaches, such as abstraction-based [26, 27, 38, 64],

symbolic integration-based [9, 32, 54], or logical calculi-based methods [35, 42, 48], there is a growing interest in martingale-based approaches, which do not only guarantee correctness but also provide *supermartingale certificates* as locally checkable correctness proofs for improving the trustworthiness of stochastic system design. A *supermartingale certificate* is a mathematical witness that the desired specification is satisfied. Their name is due to their usage of supermartingale processes from probability theory [62] towards proving properties of stochastic systems.

Recent years have seen significant theoretical and algorithmic advances in utilizing supermartingale certificates for reasoning about stochastic systems with a wide range of specifications, such as termination and reachability [2, 11, 14, 15, 20, 39, 43], safety [1, 12, 16, 21, 58, 59, 61], Büchi and co-Büchi (aka, stability) [13], and cost analysis [17, 47, 60]. These certificates have also been used for the synthesis and verification of controllers with respect to reachability [29, 37], safety [41, 51, 57, 65], reach-avoid [63, 66], and stability [4] objectives, and extensions to continuous-time stochastic systems were also considered [46]. However, existing supermartingale certificates are limited to atomic specifications or to small fragments of logical specification languages. Designing a new, bespoke supermartingale certificate for each complex automaton-based or composite logical specification is not a feasible approach. Ideally, we would like to have a systematic method for designing supermartingale certificates for an entire, rich *class of specifications*, which captures all different properties that one might care about. The recent work of Abate et al. [3] has considered this problem and proposed a supermartingale certificate for the general class of  $\omega$ -regular specifications. However, their certificate is restricted to properties that hold with probability 1, so-called *qualitative* specifications. To the best of our knowledge, no supermartingale certificates for *quantitative  $\omega$ -regular properties* have been proposed. That is, existing supermartingale certificates are insufficient for reasoning about whether an  $\omega$ -regular specification is satisfied with a probability  $p \in [0, 1]$ , which would allow the fine-grained probabilistic analysis that is often needed in practice.

In this work, we present the first supermartingale certificate for reasoning about the full class of *quantitative  $\omega$ -regular specifications* for discrete-time infinite-state stochastic systems. In order to design our certificate, we use the fact that each  $\omega$ -regular language can be represented by a limit-deterministic Büchi automaton (LDBA) [24]. This allows us to reduce the problem of proving that an  $\omega$ -regular specification is satisfied with probability at least  $p \in [0, 1]$  to the problem of proving that some set of states in the product of the stochastic system and the LDBA is reached infinitely many times with probability at least  $p$ . Hence, we call our certificate a *limit-deterministic Büchi supermartingale (LDBSM)*. The design of LDBSMs builds upon and significantly generalizes several existing supermartingale certificates for both quantitative and qualitative reasoning about stochastic systems.

We supplement our theoretical developments with *fully automated algorithms for the verification and control* of infinite-state stochastic systems using LDB-

SMs. Our algorithms apply to the setting in which the system dynamics and the controller can be represented in terms of a system of polynomial inequalities over real-valued variables. We show how the verification problem (i.e., the synthesis of a supermartingale certificate) and the control problem (i.e., the synthesis of a controller together with a supermartingale certificate) can be reduced to solving a system of polynomial inequalities, for which an off-the-shelf SMT solver may be used. Not only does this give rise to the first *certificate-based* algorithm, but also the *first* algorithm for the verification and control of general *quantitative  $\omega$ -regular specifications* for polynomial dynamical systems with a general form of stochastic uncertainties. We implemented a prototype of our method and experimentally evaluated it on a number of random walk examples with quantitative  $\omega$ -regular specifications. Our results show that our method can reason about a range of quantitative  $\omega$ -regular properties, including those that were beyond the reach of prior works.

Finally, while our focus is on the verification and control of infinite-state stochastic systems, our LDBSM certificates are applicable to general Borel-measurable stochastic dynamic systems, and hence they naturally extend to probabilistic programs with real- and integer-valued variables. Moreover, our automated synthesis algorithm also extends to polynomial probabilistic programs, similarly to existing works on supermartingale-based verification of probabilistic programs [14, 16, 21].

In short, our contributions are as follows:

1. **Theory: Supermartingale certificates.** We present limit-deterministic Büchi supermartingales (LDBSMs), the first supermartingale certificate for arbitrary quantitative  $\omega$ -regular specifications of discrete-time infinite-state stochastic systems.
2. **Automation: Template-based certificate synthesis.** We present the first fully automated algorithms for the verification and control of polynomial stochastic dynamical systems with respect to arbitrary quantitative  $\omega$ -regular specifications.

## 2 Preliminaries

We assume that the reader is familiar with basic notions of probability theory such as probability space, expected value, etc., which can also be found in standard textbooks on probability theory [62]. For a set  $S$ , we use  $\mathcal{D}(S)$  to denote the set of all probability distributions over  $S$ , and use  $2^S$  to denote the set of all subsets of  $S$ .

### 2.1 Stochastic Dynamical Systems and the Problem Statement

**Stochastic Dynamical Systems.** A discrete-time *stochastic dynamical system* (SDS) is a tuple  $\mathcal{S} = (\mathcal{X}, \mathcal{U}, \mathcal{W}, d, f, \text{Init})$ , where  $\mathcal{X} \subseteq \mathbb{R}^n$  is the state space,  $\mathcal{U} \subseteq \mathbb{R}^m$  is the control input space,  $\mathcal{W} \subseteq \mathbb{R}^p$  is the stochastic disturbance space,

$d \in \mathcal{D}(\mathcal{W})$  is the stochastic disturbance distribution,  $f: \mathcal{X} \times \mathcal{U} \times \mathcal{W} \rightarrow \mathcal{X}$  is the *system dynamics* function, and  $\text{Init} \subseteq \mathcal{X}$  is the set of initial system states. We refer to the elements of  $\mathcal{X}$  as *states*, the elements of  $\mathcal{U}$  as *control inputs*, and the elements of  $\mathcal{W}$  as *stochastic disturbances*. A *path* in  $\mathcal{S}$  is an infinite sequence of states  $x_0, x_1, \dots$  in  $\mathcal{X}$  where  $x_0 \in \text{Init}$  and for every  $t \geq 0$ , there exist a control input  $u_t \in \mathcal{U}$  and a stochastic disturbance  $w_t \in \mathcal{W}$  such that  $x_{t+1} = f(x_t, u_t, w_t)$ .

A *controller* in  $\mathcal{S}$  is a function  $\pi: \mathcal{X} \rightarrow \mathcal{U}$ , mapping every state to a control input. To avoid clutter, we define controllers as being memoryless, i.e., only depending on the current state and without using any information about the past. Our setting organically extends to *finite memory* controllers, which are able to remember bounded amount of information from the past. We will use one such finite memory controller in Sect. 2.2, which will use its memory to track the current state of the specification automaton.

Our model of SDS subsumes the cases of SDS without control inputs (using  $|\mathcal{U}| = 1$ , making the control input ineffective) and without disturbances (using  $|\mathcal{W}| = 1$ , making the disturbance ineffective and giving rise to deterministic dynamical systems).

**Semantics of SDS.** We assume that the sets  $\mathcal{X}, \text{Init} \subseteq \mathbb{R}^n$ ,  $\mathcal{U} \subseteq \mathbb{R}^m$  and  $\mathcal{W} \subseteq \mathbb{R}^p$  as well as the functions  $f: \mathcal{X} \times \mathcal{U} \times \mathcal{W} \rightarrow \mathcal{X}$  and  $\pi: \mathcal{X} \rightarrow \mathcal{U}$  are all Borel-measurable. These assumptions are necessary for the semantics of SDS to be mathematically well defined.

Under these assumptions, for every initial state  $x_0 \in \text{Init}$ , the SDS  $\mathcal{S}$  and the controller  $\pi$  together define a continuous-state, discrete-time Markov decision process that takes values in the set of states  $\mathcal{X}$  and whose trajectories correspond to paths in  $\mathcal{S}$  [52]. Initially, the process starts in  $x_0$ . Then, at every time step  $t \in \mathbb{N}_0$ , the next state of the process is defined by the equation  $x_{t+1} = f(x_t, \pi(x_t), w_t)$ , where  $w_t \sim d$  is a stochastic disturbance sampled from the stochastic disturbance distribution  $d$ , independently from the previous samples.

This process gives rise to a probability space over the set of all paths in  $\mathcal{S}$  [52]. We use  $\mathbb{P}_{x_0}^{\mathcal{S}, \pi}$  and  $\mathbb{E}_{x_0}^{\mathcal{S}, \pi}$  to denote the probability measure and the expectation operators in this probability space, respectively.

**Specifications.** A *specification*  $\varphi$  in an SDS  $\mathcal{S}$  is a set of paths in  $\mathcal{S}$ . We write  $\mathbb{P}_{x_0}^{\mathcal{S}, \pi}[\varphi]$  to denote the probability that a path in  $\mathcal{S}$  randomly sampled from the underlying probability space (over all paths) satisfies the specification  $\varphi$ . We will consider  $\omega$ -regular specifications, which constitute a broad class of properties subsuming linear temporal logic (LTL) and computation tree logics [7]. Every  $\omega$ -regular specification can be represented using limit deterministic Büchi automata, which will be defined in Sect. 2.2. We will occasionally use the standard LTL notation [7] for convenience; in particular,  $G a$  will stand for “always  $a$ ,”  $F a$  will stand for “eventually  $a$ ,” and  $aU b$  will stand for “ $a$  until  $b$ .”

**Problem Statement.** We now formally define the formal verification and control problems that we consider in this work. Consider an SDS  $\mathcal{S}$ , an  $\omega$ -regular specification  $\varphi$  in  $\mathcal{S}$ , and a probability threshold  $p \in [0, 1]$ :

1. *Verification problem.* Given a controller  $\pi$ , prove that  $\mathbb{P}_{x_0}^{\mathcal{S}, \pi}[\varphi] \geq p$  for all  $x_0 \in \text{Init}$ .
2. *Control problem.* Synthesize a controller  $\pi$  such that  $\mathbb{P}_{x_0}^{\mathcal{S}, \pi}[\varphi] \geq p$  for all  $x_0 \in \text{Init}$ .

*Example 1 (Running example: 1D random walk).* For our running example, we will use a simple 1D random walk denoted as  $\mathcal{S}^{\text{rw}}$ , whose state space is the real line  $\mathbb{R}$ , input space is a singleton set  $\{\perp\}$  with a dummy input  $\perp$ , i.e., the system is uncontrolled, stochastic disturbance space is  $\mathcal{W} = [-2, 1]$  with a continuous uniform distribution over  $\mathcal{W}$ , system dynamics is given as

$$f(x, \perp, w) = \begin{cases} x & \text{if } x > 100, \\ x + w & \text{otherwise,} \end{cases} \quad (1)$$

and the set of initial states is  $[2, 3]$ . Although this simple SDS does not contain control inputs in its dynamics, it will be sufficient for us to illustrate our technical contributions; a variation of  $\mathcal{S}^{\text{rw}}$  with control inputs will be considered in Sect. 5. Essentially, at every step, the random walk  $\mathcal{S}^{\text{rw}}$  has a higher probability of moving towards the left than towards the right. If however, it ever crosses 100, it becomes stationary. Consider the specification that is the set of all paths that visit the negative half of the real line infinitely many times, formalized as  $\{x_0 x_1 \dots \mid \forall i \geq 0 . \exists j > i . x_j \leq 0\}$ . Using the standard notation of linear temporal logic, we will write this specification as  $\text{GF}(x \leq 0)$ . For this example, we will refer to the set  $\{x \mid x \leq 0\}$  as the *target*.

## 2.2 Limit-Deterministic Büchi Automata

To specify properties of a given SDS, we label its state space with a finite set of *atomic propositions*  $P$ . A *labeling function*  $L : \mathcal{X} \rightarrow 2^P$  maps each state  $x \in \mathcal{X}$  to the set of atomic propositions that are true in  $x$ , and we assume that each atomic proposition  $p \in P$  has an associated Borel-measurable arithmetic expression  $\text{exp}_p : \mathcal{X} \rightarrow \mathbb{R}$  such that for every  $x \in \mathcal{X}$ ,  $p \in L(x)$  iff  $\text{exp}_p(x) \geq 0$ . Hence, a path  $(x_i)_{i=0}^\infty$  in the SDS gives rise to an *infinite word*  $(L(x_i))_{i=0}^\infty$  over the alphabet  $2^P$ .

We use the classical result that for every  $\omega$ -regular specification  $\varphi$  defined over a finite set of atomic predicates  $P$ , there exists a limit-deterministic Büchi automaton with alphabet  $\Sigma = 2^P$  which accepts the same set of infinite words over  $2^P$  as  $\varphi$  [24].

**Nondeterministic Büchi Automata.** A *nondeterministic Büchi automaton (NBA)* is a tuple  $\mathcal{A} = (Q, q_{\text{init}}, \Sigma, \Delta, \mathcal{F})$  where  $Q$  is a finite set of states,  $q_{\text{init}} \in Q$  is an initial state,  $\Sigma$  is a finite alphabet that includes the empty word symbol  $\epsilon$ ,  $\Delta : Q \times \Sigma \rightarrow 2^Q$  is a nondeterministic transition function, and  $\mathcal{F} \subseteq 2^Q$  is a set of accepting states. An infinite word  $\sigma_0, \sigma_1, \dots$  of letters in the alphabet  $\Sigma$  is *accepted* by  $\mathcal{A}$ , if it gives rise to at least one Büchi accepting run in  $\mathcal{A}$ , i.e., if there exists a run  $q_0, q_1, \dots$  of states in  $Q$  such that  $q_0 = q_{\text{init}}$ , for each  $i \in \mathbb{N}_0$ ,

$q_{i+1} \in \Delta(q_i, \sigma_i)$ , and for every  $i \in \mathbb{N}_0$  there exists a  $j > i$  such that  $q_j \in \mathcal{F}$  (infinitely many visits to  $\mathcal{F}$ ).

**Limit-Deterministic Büchi Automata.** *Limit-deterministic Büchi automata (LDBA)* are particular types of NBAs with a restricted form of nondeterminism. In particular, an NBA  $\mathcal{A} = (Q, q_{\text{init}}, \Sigma, \Delta, \mathcal{F})$  is said to be an LDBA, if there exists a partition  $\{Q_n, Q_d\}$  of the set of states  $Q$  (i.e.,  $Q = Q_n \cup Q_d$  and  $Q_n \cap Q_d = \emptyset$ ) such that for every  $q \in Q_d$  and for every  $\sigma \in \Sigma$ , (i) the transitions from  $q$  are deterministic, i.e.,  $|\Delta(q, \sigma)| = 1$ , and (ii) there is no transition from  $q$  going outside of  $Q_d$ , i.e.,  $\Delta(q, \sigma) \subseteq Q_d$ . We call  $Q_n$  the *nondeterministic part* of the LDBA and  $Q_d$  the *deterministic part* of the LDBA.

As mentioned above, for every  $\omega$ -regular specification  $\varphi$  defined over a finite set of atomic predicates  $P$ , there exists a canonical LDBA  $\mathcal{A}$  with the alphabet  $\Sigma = 2^P$  accepting the same set of infinite words as  $\varphi$ . Hence, a path  $(x_i)_{i=0}^\infty$  in  $\mathcal{S}$  satisfies  $\omega$ -regular specification  $\varphi$  if and only if the infinite word  $(L(x_i))_{i=0}^\infty$  is accepted by the LDBA  $\mathcal{A}$ . Our verification and control problems thus reduce to ensuring that a random run in  $\mathcal{S}$  induces an infinite word over  $2^P$  accepted by the LDBA  $\mathcal{A}$  with probability at least  $p$ .

**Product SDS.** In order to reason about paths and infinite words that they induce in the LDBA, we consider a (synchronous) product of the SDS and the LDBA. A *product SDS* of the SDS  $\mathcal{S} = (\mathcal{X}, \mathcal{U}, \mathcal{W}, d, f, \text{Init})$  and the LDBA  $\mathcal{A} = (Q, q_{\text{init}}, 2^P, \Delta, \mathcal{F})$  is a tuple  $\mathcal{S}^\times = (\mathcal{X}^\times, \mathcal{U}^\times, \mathcal{W}, d, f^\times, \text{Init}^\times)$ , where

- $\mathcal{X}^\times = \mathcal{X} \times Q$  is the state space of the product SDS,
- $\mathcal{U}^\times = \mathcal{U} \times Q$  is the control input space of the product SDS,
- $\mathcal{W} \subseteq \mathbb{R}^p$  is the stochastic disturbance space of the product SDS,
- $d \in \mathcal{D}(\mathcal{W})$  is the stochastic disturbance distribution of the product SDS,
- $f^\times: \mathcal{X}^\times \times \mathcal{U}^\times \times \mathcal{W} \rightarrow \mathcal{X}^\times$  is the system dynamics function, and
- $\text{Init}^\times = \text{Init} \times \{q_{\text{init}}\}$  is the set of initial states of the product SDS.

Similarly as before, a *controller* in  $\mathcal{S}^\times$  is a function  $\pi^\times: \mathcal{X}^\times \rightarrow \mathcal{U}^\times$  mapping every state to a control input. However, we require that for every  $(x, q) \in \mathcal{X}^\times$ , the corresponding control input  $\pi^\times(x, q) = (u, q') \in \mathcal{U} \times Q$  satisfies  $q' \in \Delta(q, L(x))$ . Hence, the product SDS indeed models a synchronous product of the SDS and the LDBA, where the controller  $\pi^\times$  resolves nondeterminism both in the SDS and in the LDBA. To distinguish between the two controller components, we write  $\pi^\times(x, q) = (\pi^{\mathcal{S}}(x, q), \pi^{\mathcal{A}}(x, q))$ . The semantics of the product SDS are defined analogously to Sect. 2.1, with

$$(x_{t+1}, q_{t+1}) = \left( f(x_t, \pi^{\mathcal{S}}(x_t, q_t), w_t), \pi^{\mathcal{A}}(x_t, q_t) \right)$$

for every time step  $t \in \mathbb{N}_0$  and  $w_t$  sampled according to  $d$  and independent from  $w_i$  for every  $i \leq t-1$ . For each initial state  $(x_0, q_{\text{init}}) \in \text{Init}^\times$ , we denote by  $\mathbb{P}_{(x_0, q_{\text{init}})}^{\mathcal{S}^\times, \pi^\times}$  and  $\mathbb{E}_{(x_0, q_{\text{init}})}^{\mathcal{S}^\times, \pi^\times}$  the probability measure and the expectation operator in the probability space of all paths in  $\mathcal{S}^\times$  with controller  $\pi^\times$ , respectively.

Finally, the following proposition will allow us to reduce our verification and control problems to the problems of synthesizing a controller for the product

SDS. Denote by Büchi(B) the set of all runs in  $\mathcal{S}^\times$  that visit states in  $B \subseteq \mathcal{X}^\times$  infinitely many times.

**Proposition 1.** *Consider an SDS  $\mathcal{S}$  and a controller  $\pi$  in  $\mathcal{S}$ . Let  $\varphi$  be an  $\omega$ -regular specification over a finite set of atomic propositions  $P$ ,  $\mathcal{A} = (Q, q_{\text{init}}, 2^P, \Delta, \mathcal{F})$  be an LDBA for the specification  $\varphi$ , and  $\mathcal{S}^\times$  be the product MDP of  $\mathcal{S}$  and  $\mathcal{A}$ . Then, for every controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  in  $\mathcal{S}^\times$  with  $\pi^{\mathcal{S}} = \pi$  and for every initial state  $(x_0, q_{\text{init}}) \in \text{Init}^\times$ , we have*

$$\mathbb{P}_{x_0}^{\mathcal{S}, \pi} [\varphi] \geq \mathbb{P}_{(x_0, q_{\text{init}})}^{\mathcal{S}^\times, \pi^\times} [\text{Büchi}(\mathbb{R}^n \times \mathcal{F})].$$

*Proof.* Consider a controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  in  $\mathcal{S}^\times$  with  $\pi^{\mathcal{S}} = \pi$ . Let  $\Pi$  be the set of all paths in  $\mathcal{S}$ , and  $\Pi^\times$  be the set of all paths in  $\mathcal{S}^\times$ . Define the map  $g : \Pi \rightarrow \Pi^\times$  via

$$g(x_0, x_1, x_2, \dots) = ((x_0, q_0), (x_1, \pi^{\mathcal{A}}(x_0, q_0)), (x_2, \pi^{\mathcal{A}}(x_1, q_1)), \dots).$$

Now, observe that

$$\begin{aligned} \mathbb{P}_{x_0}^{\mathcal{S}, \pi} [\varphi] &= \mathbb{P}_{x_0}^{\mathcal{S}, \pi} \left[ (x_i)_{i=0}^\infty \in \Pi \mid (x_i)_{i=0}^\infty \models \varphi \right] \\ &= \mathbb{P}_{x_0}^{\mathcal{S}, \pi} \left[ (x_i)_{i=0}^\infty \in \Pi \mid (L(x_i))_{i=0}^\infty \text{ infinite word accepted by } \mathcal{A} \right] \\ &\geq \mathbb{P}_{x_0}^{\mathcal{S}, \pi} \left[ (x_i)_{i=0}^\infty \in \Pi \mid (g(x_i))_{i=0}^\infty \text{ visits infinitely many times } \mathbb{R}^n \times \mathcal{F} \right] \\ &= \mathbb{P}_{(x_0, q_{\text{init}})}^{\mathcal{S}^\times, \pi^\times} [\text{Büchi}(\mathbb{R}^n \times \mathcal{F})], \end{aligned}$$

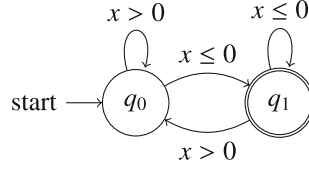
where the inequality holds due to the first set of paths in  $\Pi$  containing the second set of paths in  $\Pi$ . This proves the proposition claim.  $\square$

The inequality in Proposition 1 is strict whenever  $\pi^{\mathcal{A}}$  chooses a sub-optimal resolution of non-determinisms in the LDBA  $\mathcal{A}$ . Consider, e.g., the property  $\text{FG } a$  (eventually always  $a$ ), whose LDBA representation has three states  $q_0, q_1, q_2$ . The initial state is  $q_0$ , and upon seeing an  $a$  the controller  $\pi^{\mathcal{A}}$  needs to non-deterministically decide whether to wait at  $q_0$  or to proceed to the accepting state  $q_1$ . The system can stay in  $q_1$  as long as only  $a$  is seen, and seeing a  $\bar{a}$  forces the automaton (deterministically) to the rejecting sink  $q_2$ . For  $\pi^{\mathcal{S}} = \pi$ , satisfying the specification  $\text{FG } a$  on the SDS with probability  $p > 0$ , if  $\pi^{\mathcal{A}}$  always chooses to stay at  $q_0$  after seeing every  $a$ , then the left side of the inequality will be  $p$  but the right side will be 0.

**Rejecting States.** An LDBA  $\mathcal{A} = (Q, q_{\text{init}}, \Sigma, \Delta, \mathcal{F})$  induces a graph over its states with edges defined by all possible nondeterministic transitions in the automaton, i.e., it induces a graph  $G = (Q, E)$  with  $E = \{(q, q') \in Q \times Q \mid \exists \sigma \in \Sigma. q' \in \Delta(q, \sigma)\}$ .

We denote by  $Q_{\text{reject}} \subseteq Q$  the set of all states in the LDBA from which no state in the set of accepting states  $\mathcal{F}$  can be reached in the graph  $G$ . Thus, if an infinite run  $q_0, q_1, \dots$  in the LDBA contains a state in  $Q_{\text{reject}}$ , then it cannot be an accepting run as it will contain no states in  $\mathcal{F}$  after the first occurrence of a state in  $Q_{\text{reject}}$ . To that end, we call  $Q_{\text{reject}}$  the set of *rejecting states* in the LDBA.

*Example 2.* The specification  $\text{GF}(x \leq 0)$  from the 1D random walk is shown in the LDBA form in Fig. 1. The atomic predicates are given by  $(x \leq 0)$  and  $(x > 0)$ . The product state remains in the automaton state  $q_0$  until  $x > 0$  is true, and each time the target  $x \leq 0$  is true, the automaton visits the state  $q_1$ , which is the accepting state.



**Fig. 1.** The specifications  $\text{GF}(x \leq 0)$  from Example 1 represented using LDBA, which in this case is a deterministic Büchi automaton, where  $q_1$  is the Büchi accepting state.

### 3 Supermartingale Certificate for $\omega$ -Regular Specifications

We now present the theory behind our novel supermartingale certificates for quantitative  $\omega$ -regular specifications, which is the main technical contribution of our work. Definition 1 formally defines the certificates and Theorem 1 establishes their soundness. The computational aspects of the certificate will be discussed in Sect. 4. We also assume that a controller is provided as input, and in Sect. 4 we will show how to compute it when it is not provided.

In what follows, suppose the following are given: an SDS  $\mathcal{S} = (\mathcal{X}, \mathcal{U}, \mathcal{W}, d, f, \text{Init})$ , a controller  $\pi$  in  $\mathcal{S}$ , an  $\omega$ -regular specification  $\varphi$  over a set of atomic propositions  $P$ , and a minimum probability  $p \in [0, 1]$  with which the specification should be satisfied.

#### 3.1 Intuitive Overview

Let  $\mathcal{A} = (Q, q_{\text{init}}, 2^P, \Delta, \mathcal{F})$  be an LDBA that accepts the same set of infinite words over  $2^P$  as the specification  $\varphi$ . Our certificate is defined over the product SDS  $\mathcal{S}^\times$  of the SDS  $\mathcal{S}$  and the LDBA  $\mathcal{A}$ . By Proposition 1, this reduces the problem of certifying

*the specification  $\varphi$  is satisfied with probability  $\geq p$*

to the problem of certifying that

*there exists a controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  in  $\mathcal{S}^\times$  with  $\pi^{\mathcal{S}} = \pi$ , such that the set of states  $\mathbb{R}^n \times \mathcal{F}$  in  $\mathcal{S}^\times$  is reached infinitely many times with probability  $\geq p$ .*

To prove this, it suffices to show that there exists a controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  in  $\mathcal{S}^\times$  with  $\pi^{\mathcal{S}} = \pi$ , such that the following two properties are satisfied:

1. *Safety property.* There exists a set of states  $SI$  in the product SDS  $\mathcal{S}^\times$  that contains all initial states  $\text{Init} \times \{q_0\}$ , no rejecting states  $\mathbb{R}^n \times Q_{\text{reject}}$ , and such that  $SI$  is left with probability at most  $1 - p$ . In other words, the pair  $(SI, p)$  is a *stochastic invariant* [21].
2. *Liveness property.* A random run in the product SDS  $\mathcal{S}^\times$  either visits  $\mathbb{R}^n \times \mathcal{F}$  infinitely many times or eventually leaves the set  $SI$  with probability 1.

The conjunction of the safety and the liveness properties guarantees that the set of states  $\mathbb{R}^n \times \mathcal{F}$  is visited infinitely many times with probability at least  $p$ .

The safety-liveness decomposition of the problem is reflected in the design of our supermartingale certificate. In particular, our certificate  $V$  consists of two components  $V^{\text{safe}}$  and  $V^{\text{live}}$ , called the *safety certificate* and the *liveness certificate*, which are connected in a hierarchical fashion: the certificate  $V^{\text{safe}}$  is independent of  $V^{\text{live}}$  and generates the stochastic invariant  $SI$ , whereas the certificate  $V^{\text{live}}$  depends on  $V^{\text{safe}}$  to use the set  $SI$  as a “safety net” and make sure that the liveness condition is fulfilled. A crucial aspect of this hierarchical composition is that the two certificates agree on their choices of control inputs and resolutions of the nondeterminisms in the LDBA at all time. From this, we obtain a single joint controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  in  $\mathcal{S}^\times$  with  $\pi^{\mathcal{S}} = \pi$ , and formally certify that both the safety and the liveness property specified above are satisfied.

**Invariants.** Supermartingale certificates need to satisfy a set of conditions in the reachable parts of the state space, starting from the initial states. Since computing the exact set of reachable states is in general infeasible, we define our supermartingale certificates with respect to an over-approximation, called a (*supporting*) *invariant*. In this section, we assume that the invariant is given, and in Sect. 4, we will describe how to automatically compute it along with the certificate and the controller.

Formally, a state  $x$  in the SDS  $\mathcal{S}$  is *reachable*, if there exists a path  $x_0, x_1, \dots$  in  $\mathcal{S}$  with  $x_t = x$  for some  $t \in \mathbb{N}_0$ . An *invariant* in  $\mathcal{S}$  is a set of states  $I \subseteq \mathcal{X}$  which contains all reachable states in  $\mathcal{S}$ . The definition of an invariant in a product SDS  $\mathcal{S}^\times$  is analogous.

Invariants are integral components of supermartingale certificates, because they determine within which part of the state space the certificate conditions are required to be fulfilled; outside of the invariant, the certificates may behave arbitrarily. Therefore, our formal definition of LDBSM certificates will include their supporting invariants as their “domains” of operation.

### 3.2 Supermartingale Certificates for LDBA Specifications

To formalize the certificate components  $V^{\text{safe}}$ ,  $V^{\text{live}}$ , and  $I$ , as described above, we build upon and generalize the existing supermartingale certificates for reasoning about quantitative safety and qualitative recurrence (i.e., Büchi) properties.

**Safety Certificates.** Our safety certificate  $V^{\text{safe}}$  draws insight from *repulsing supermartingales* [21], which were introduced for reasoning about quantitative safety. Intuitively, given some set of unsafe states  $U \subseteq \mathcal{X}$  in an SDS  $\mathcal{S}$  under

a controller  $\pi$ , a repulsing supermartingale for  $U$  is a function  $V^{\text{safe}} : \mathcal{X} \rightarrow \mathbb{R}$ , along with an invariant  $I$ , which to every system state assigns a real value that is required to satisfy the following conditions at the states in  $I$ : **(S1)** at initial states, the value of  $V^{\text{safe}}$  is below some negative threshold  $\eta \leq 0$ , **(S2)** at unsafe states in  $U \cap I$ , the value of  $V^{\text{safe}}$  is above the threshold 0, and **(S3)** at states in  $I$  in which  $V^{\text{safe}} \leq 0$ , the value of  $V^{\text{safe}}$  must strictly decrease in expectation by  $\epsilon > 0$  upon every one-step execution of the SDS under the controller  $\pi$ , while also ensuring that the absolute change in its value lies within some interval of length  $M > 0$ , and ensuring that the next state remains within  $I$ . It was shown that, if a repulsing supermartingale for the unsafe set exists, then the SDS stays within the set of states  $SI = \{x \in \mathcal{X} \mid V^{\text{safe}}(x) < 0\}$ —the *stochastic invariant*—and thus does not reach the unsafe set  $U$  with probability at least  $p \geq 1 - \exp(-\frac{8\epsilon\eta}{M^2})$  [59] (the bound of [59] is an improved version of the bound in [21]).

For us, the set of unsafe states of the product SDS  $\mathcal{S}^\times$  is given by  $U = \mathbb{R}^n \times Q_{\text{reject}}$ , where recall that  $Q_{\text{reject}}$  is the set of rejecting states in the LDBA, and our goal is to ensure that there exists a controller  $\pi_{\text{safe}}^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  in  $\mathcal{S}^\times$  with  $\pi^{\mathcal{S}} = \pi$ , under which the product SDS is safe with probability at least  $p$ . To this end, we extend repulsing supermartingales by requiring condition (S3) to be satisfied only for *some* outgoing LDBA transition at a state in  $\mathcal{S}^\times$ —call it condition **(S3+)**. The stochastic invariant also gets generalized to the state space of the product SDS in the obvious way:  $SI^\times = \{(x, q) \in \mathcal{X} \times Q \mid V^{\text{safe}}(x, q) < 0\}$ . The controller  $\pi^{\mathcal{A}}$  then picks that LDBA transition towards satisfying the quantitative safety property.

**Liveness Certificates.** Our liveness certificate  $V^{\text{live}}$  draws insight from the supermartingale certificate of Chakarov et al. [13] for proving that a given set of target states  $T \subseteq \mathcal{X}$  in  $\mathcal{S}$  under a given controller  $\pi$  is visited infinitely many times with probability 1. Intuitively, a supermartingale certificate for  $T$  is a function  $V^{\text{live}} : \mathcal{X} \rightarrow \mathbb{R}$ , along with an invariant  $I$ , which to every system state assigns a real value that is required to satisfy the following conditions at the states in  $I$ : **(L1)** the value of  $V^{\text{live}}$  is non-negative, **(L2)** at non-target states outside  $T$ , the value of  $V^{\text{live}}$  must strictly decrease in expected value by  $\epsilon > 0$  upon every one-step execution of the system under the controller  $\pi$ , and **(L3)** at target states in  $T \cap I$ , the value of  $V^{\text{live}}$  is allowed to increase in expectation by at most  $M > 0$  in every one-step execution of the system under the controller  $\pi$ , while making sure that the next state does not go outside of  $I$ . If such a certificate for the target set  $T$  exists, then  $T$  is reached infinitely many times with probability 1 [13].

For us, the set of target states of the product SDS  $\mathcal{S}^\times$  is given by  $T = \mathbb{R}^n \times \mathcal{F}$ , where recall that  $\mathcal{F}$  is the set of Büchi accepting set in the LDBA. In our LDBSM certificate, we extend the certificate of Chakarov et al. [13] in two ways. First, we present a supermartingale certificate for proving that, with probability 1, *either* the Büchi accepting set  $\mathcal{X} \times \mathcal{F}$  of the product SDS is reached infinitely many times *or* the stochastic invariant set  $SI^\times$  is eventually left—call it condition **(L3+)**. This is in line with the overview in Sect. 3.1 and our requirements for a liveness certificate. Second, in the product MDP  $\mathcal{S}^\times$ , our goal is again to ensure that

there exists a controller  $\pi_{\text{live}}^\times = (\pi^S, \pi^A)$  in  $\mathcal{S}^\times$  with  $\pi^S = \pi$ , under which this property is satisfied. This is accounted for by requiring conditions (L2) and (L3) of the certificate of Chakarov et al. [13] to be satisfied only for *some* outgoing LDBA transition at a state in  $\mathcal{S}^\times$ .

**The Hierarchical Composition of Safety and Liveness Certificates.** The challenge in combining the safety and liveness components lies in ensuring that the properties implied by the two components are satisfied with respect to the *same controller*  $\pi^\times = (\pi^S, \pi^A)$  in  $\mathcal{S}^\times$  with  $\pi^S = \pi$ ; in other words, we need to ensure that the controllers  $\pi_{\text{safe}}^\times$  and  $\pi_{\text{live}}^\times$  are the same. Otherwise, we cannot conclude that the set of states  $\mathbb{R}^n \times \mathcal{F}$  is reached infinitely many times with probability at least  $p$ . In particular, we need to ensure that condition (S3+) on one hand, and conditions (L2) and (L3+) on the other hand, are always satisfied with respect to the same resolution of non-determinism, i.e., with respect to the same transition  $q' \in \Delta(q, a)$  in the LDBA. We will pinpoint how this is achieved after presenting our novel LDBSM certificate in the following.

**Definition 1 (LDBMS certificates).** *Consider an SDS  $\mathcal{S}$  and a controller  $\pi$  in it. Let  $\mathcal{A} = (Q, q_{\text{init}}, 2^P, \Delta, \mathcal{F})$  be an LDBA,  $\mathcal{S}^\times$  be the product SDS of  $\mathcal{S}$  and  $\mathcal{A}$ , and  $I \subseteq \mathcal{X} \times Q$  be an invariant in  $\mathcal{S}^\times$ .*

*An LDBSM certificate is a triple  $V = (V^{\text{safe}}, V^{\text{live}}, I)$ , where  $V^{\text{safe}} : \mathcal{X} \times Q \rightarrow \mathbb{R}$  and  $V^{\text{live}} : \mathcal{X} \times Q \rightarrow \mathbb{R}$  assign real values to each state in  $\mathcal{S}^\times$ , and  $I \subseteq \mathcal{X} \times Q$  is an invariant in  $\mathcal{S}^\times$ . We require that there exist  $\eta^S \leq 0$ ,  $\epsilon^S > 0$ ,  $M^S > 0$ , and  $\beta^S \in \mathbb{R}$ , as well as  $\epsilon^L > 0$ ,  $M^L > 0$ , such that the following conditions are satisfied:*

(a) Initial condition of the invariant: *the initial state is inside the invariant, i.e.,*

$$\forall (x, q) \in \text{Init}^\times. (x, q) \in I.$$

(b) Initial condition of the safety certificate: *the value of  $V^{\text{safe}}$  at initial states is at most  $\eta^S$ , i.e.,*

$$\forall x \in \text{Init}. V^{\text{safe}}(x, q_{\text{init}}) \leq \eta^S. \quad (\text{Cond. S1})$$

(c) Safety condition of the safety certificate: *the value of  $V^{\text{safe}}$  at reachable rejecting states is non-negative, i.e.,*

$$\forall (x, q) \in (\mathcal{X} \times Q_{\text{reject}}) \cap I. V^{\text{safe}}(x, q) \geq 0. \quad (\text{Cond. S2})$$

(d) Non-negativity condition of the liveness certificate: *the value of  $V^{\text{live}}$  is non-negative at states in the invariant  $I$ , which over-approximates the set of reachable states, i.e.,*

$$\forall (x, q) \in I. V^{\text{live}}(x, q) \geq 0. \quad (\text{Cond. L1})$$

(e) Strict expected decrease of the liveness certificate until  $\mathcal{X} \times \mathcal{F}$  or  $SI = \{(x, q) \in \mathcal{S}^\times \mid V^{\text{safe}}(x, q) < 0\}$  is reached with the safety condition: *we have*

$$\forall (x, q) \in \mathcal{X} \times (Q \setminus (\mathcal{F} \cup Q_{\text{reject}})) \cap I. V^{\text{safe}}(x, q) \leq 0 \implies$$

$$\bigvee_{a \in \Sigma} \bigvee_{q' \in \Delta(q, a)} \left[ \begin{array}{c} \overbrace{\text{SafetyCond}(x, q, a, q')}^{\text{Cond. S3+}} \\ \left( \begin{array}{l} x \models a \wedge \forall w \in \mathcal{W}. (f(x, \pi(x), w), q') \in I \wedge \\ V^{\text{safe}}(x, q) \geq \mathbb{E}_w^\pi [V^{\text{safe}}(f(x, \pi(x), w), q')] + \epsilon^S \wedge \\ \forall w \in \mathcal{W}. \beta^S \leq V^{\text{safe}}(x, q) - V^{\text{safe}}(f(x, \pi(x), w), q') \leq \beta^S + M^S \end{array} \right) \\ \underbrace{V^{\text{live}}(x, q) \geq \mathbb{E}_w^\pi [V^{\text{live}}(f(x, \pi(x), w), q')] + \epsilon^L}_{\text{Cond. L2}} \end{array} \right]$$

(f) Bounded expected increase of the liveness certificate in  $(\mathcal{X} \times \mathcal{F}) \setminus SI$  with the safety condition: *we have*

$$\forall (x, q) \in (\mathcal{X} \times \mathcal{F}) \cap I. V^{\text{safe}}(x, q) \leq 0 \implies$$

$$\bigvee_{a \in \Sigma, \Delta(q, a) = q'} \left[ \begin{array}{c} \overbrace{\text{SafetyCond}(x, q, a, q')}^{\text{Cond. S3+}} \\ \underbrace{\mathbb{E}_w^\pi [V^{\text{live}}(f(x, \pi(x), w), q')] \leq V^{\text{live}}(x, q) + M^L}_{\text{Cond. L3+}} \end{array} \right].$$

Conditions (b), (c), and (d) in Def. 1 formalize the (S1), (S2), and (L1) conditions that were informally introduced beforehand. Conditions (e) and (f) combine (S3+) with (L2) and (L3+), respectively, which are concerned about states that are, respectively, outside of  $\mathcal{F} \cup Q_{\text{reject}}$  and inside of  $\mathcal{F}$ . Additionally, in both (e) and (f), the left hand sides of the implications require that the state  $(x, q)$  is in the stochastic invariant  $SI$ , which is formalized by the inequality  $V^{\text{safe}}(x, q) \leq 0$ , and the right hand sides require the satisfaction of (S3+) which is captured using the formula  $\text{SafetyCond}(x, q, a, q')$ . The formula  $\text{SafetyCond}(x, q, a, q')$  encodes that the strict expected decrease and the bounded differences conditions are satisfied at a state  $(x, q)$  of  $\mathcal{S}^\times$ , upon taking the LDBA transition to  $q'$  by using  $a$ . The assertion  $x \models a$  guarantees the satisfaction of  $a$  at  $x$ , ensuring the availability of the LDBA transition to  $q' \in \Delta(q, a)$ . The clause  $V^{\text{safe}}(x, q) \geq \mathbb{E}_w^\pi [V^{\text{safe}}(f(x, \pi(x), w), q')] + \epsilon^S$  encodes strict expected decrease, i.e., that the value of  $V^{\text{safe}}$  decreases in expected value by at least  $\epsilon^S$  upon one-step execution of the SDS under the controller  $\pi$ . The clause  $\forall w \in \mathcal{W}. \beta^S \leq V^{\text{safe}}(x, q) - V^{\text{safe}}(f(x, \pi(x), w), q') \leq \beta^S + M^S$  encodes the bounded differences condition, i.e., that for every stochastic disturbance  $w \in \mathcal{W}$ , the change in the value of  $V^{\text{safe}}$  is contained in the interval  $[\beta^S, \beta^S + M^S]$ . Finally, the clause  $\forall w \in \mathcal{W}. (f(x, \pi(x), w), q') \in I$  guarantees that after the one-step execution of the SDS under the controller  $\pi$ , the new state is within the invariant  $I$ . Moreover, the right hand sides of the implications of (d) and (e) also respectively implement (L2) and (L3+), which are straightforward.

The following theorem establishes the soundness of our certificate for proving satisfaction of quantitative  $\omega$ -regular specifications in SDSs.

**Theorem 1 (Soundness of LDBSM certificates).** *Consider an SDS  $\mathcal{S}$  and a controller  $\pi$ . Let  $\varphi$  be an  $\omega$ -regular specification in  $\mathcal{S}$ ,  $p \in [0, 1]$  be a probability threshold, and  $\mathcal{A} = (Q, q_{\text{init}}, 2^P, \Delta, \mathcal{F})$  be an LDBA that accepts the same language as  $\varphi$ . Let  $\mathcal{S}^\times$  be the product SDS of  $\mathcal{S}$  and  $\mathcal{A}$ .*

*Suppose that there exists an LDBSM certificate  $V = (V^{\text{safe}}, V^{\text{live}}, I)$  with the associated constants  $\eta^{\mathcal{S}}, \epsilon^{\mathcal{S}}$ , and  $M^{\mathcal{S}}$ , such that  $p \leq 1 - \exp\left(\frac{8 \cdot \eta^{\mathcal{S}} \cdot \epsilon^{\mathcal{S}}}{(M^{\mathcal{S}})^2}\right)$ . Then, there exists a controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  in  $\mathcal{S}^\times$  with  $\pi^{\mathcal{S}} = \pi$ , such that for every initial state  $(x_0, q_{\text{init}}) \in \text{Init}^\times$ , we have*

$$\mathbb{P}_{x_0}^{\mathcal{S}, \pi} \left[ \varphi \right] \geq \mathbb{P}_{(x_0, q_{\text{init}})}^{\mathcal{S}^\times, \pi^\times} \left[ \text{Büchi}(\mathbb{R}^n \times \mathcal{F}) \right] \geq p. \quad (2)$$

*Proof.* We first define  $\pi^{\mathcal{A}} : \mathcal{S}^\times \rightarrow Q$  in order to specify the controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  in  $\mathcal{S}^\times$  with  $\pi^{\mathcal{S}} = \pi$ . In what follows, we fix an ordering of the states in  $Q$ . For each  $(x, q) \in \mathcal{S}^\times$ , we define  $\pi^{\mathcal{A}}(x, q)$  as follows:

- If  $(x, q) \in I$ ,  $V^{\text{safe}}(x, q) \leq 0$  and  $q \in Q \setminus (\mathcal{F} \cup Q_{\text{reject}})$ , we define  $\pi^{\mathcal{A}}(x, q)$  to be  $q' \in Q$  of the smallest order for which the predicate on the right-hand-side of condition ((e)) in Def. 1 is satisfied.
- If  $(x, q) \in I$ ,  $V^{\text{safe}}(x, q) \leq 0$  and  $q \in \mathcal{F}$ , we define  $\pi^{\mathcal{A}}(x, q)$  to be  $q' \in Q$  of the smallest order for which the predicate on the right-hand-side of condition ((f)) in Def. 1 is satisfied.
- If  $(x, q) \in I$ ,  $V^{\text{safe}}(x, q) \leq 0$  and  $q \in Q_{\text{reject}}$ , by condition (c) in Def. 1, if  $(x, q) \in I$  then  $V^{\text{safe}}(x, q) \geq 0$ . Hence,  $V^{\text{safe}}(x, q)$  must be 0. We define  $\pi^{\mathcal{A}}(x, q)$  to be  $q' \in Q$  of the smallest order for which there exists a letter  $a \in \Sigma$  with  $x \models a$  and  $q' \in \Delta(q, a)$ . Such  $q' \in Q$  is guaranteed to exist, as it is always possible to take at least one transition in the LDBA.
- If  $(x, q) \notin I$  or  $V^{\text{safe}}(x, q) > 0$ , we define  $\pi^{\mathcal{A}}(x, q)$  exactly as in the previous case above, i.e.,  $\pi^{\mathcal{A}}(x, q)$  is  $q' \in Q$  of the smallest order for which there exists a letter  $a \in \Sigma$  with  $x \models a$  and  $q' \in \Delta(q, a)$ .

We note that fixing an order of states in  $Q$  and picking a state  $q'$  of the smallest order is done in order to ensure that  $\pi^{\mathcal{A}} : \mathcal{S}^\times \rightarrow Q$  is a measurable function, for the semantics of the controller to be mathematically well defined.

We now show that the controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$  with  $\pi^{\mathcal{S}} = \pi$  and  $\pi^{\mathcal{A}}$  defined as above satisfies the theorem's claim. The first inequality follows from Proposition 1, hence we only need to prove the second inequality. We show that the certificate component  $V^{\text{safe}}$  defines a repulsing supermartingale for the unsafe set  $\mathbb{R}^n \times Q_{\text{reject}}$  [21] in the product SDS  $\mathcal{S}^\times$  controlled by our controller  $\pi^\times = (\pi^{\mathcal{S}}, \pi^{\mathcal{A}})$ . To prove this, we need to show that  $V^{\text{safe}}$  and  $I$  satisfy conditions (S1)-(S3) defined earlier in this section. Condition (S1) is implied by condition ((b)) in Definition 1. Condition (S2) is implied by condition ((c)) in Definition 1. Finally, condition (S3) is implied by conditions ((e)) and ((f)) in Definition 1. Hence,  $V^{\text{safe}}$  indeed defines a repulsing supermartingale for the

unsafe set  $\mathbb{R}^n \times Q_{\text{reject}}$ , and  $I$  is the respective invariant. Thus, by [59, Theorem 5.1], it follows that for every initial state  $(x, q_{\text{init}}) \in \mathcal{X}^\times$ , we have

$$\mathbb{P}_{(x_0, q_{\text{init}})}^{\mathcal{S}^\times, \pi^\times} \left[ \text{Safe} \left( \{ (x, q) \in \mathcal{X}^\times \mid V^{\text{safe}}(x, q) \geq 0 \} \right) \right] \geq 1 - \exp \frac{8 \cdot \eta^S \cdot \epsilon^S}{(M^S)^2} \geq p, \quad (3)$$

where the inequality indeed holds with respect to our controller  $\pi^\times = (\pi^S, \pi^A)$  and  $\text{Safe}(U)$  denotes the set of all paths in  $\mathcal{S}^\times$  that do not contain states from some set  $U$ .

Next, we show that, under our controller  $\pi^\times = (\pi^S, \pi^A)$ , either the set of states  $\mathbb{R}^n \times \mathcal{F}$  is visited infinitely many times or the set of states  $SI = \{ (x, q) \in \mathcal{X}^\times \mid V^{\text{safe}}(x, q) \leq 0 \}$  is eventually left with probability 1. To prove this, we first define a new product SDS  $\mathcal{S}_{\text{new}}^\times$  by modifying its dynamics function  $f^\times: \mathcal{X}^\times \times \mathcal{U}^\times \times \mathcal{W} \rightarrow \mathcal{X}^\times$  to

$$f_{\text{new}}^\times((x, q), (u, q), w) = \begin{cases} f^\times((x, q), (u, q), w), & \text{if } V^{\text{safe}}(x, q) < 0 \\ (x, q), & \text{if } V^{\text{safe}}(x, q) \geq 0. \end{cases}$$

Intuitively, we are redefining the product SDS to make the complement of  $SI$ , i.e., the set of all states at which  $V^{\text{safe}}(x, q) \geq 0$ , into a sink where the system remains stuck forever once entered. Then, in order to prove that either the set of states  $\mathbb{R}^n \times \mathcal{F}$  is visited infinitely many times or the set of states  $SI = \{ (x, q) \in \mathcal{X}^\times \mid V^{\text{safe}}(x, q) \leq 0 \}$  is eventually left with probability 1, it suffices to prove that the set of states  $(\mathbb{R}^n \times \mathcal{F}) \cup \neg SI$  in the *new product SDS*  $\mathcal{S}_{\text{new}}^\times$  is visited infinitely many times with probability 1. To show this, we prove that the certificate component  $V^{\text{live}}$  defines an instance of the supermartingale certificate of [13] for proving that the target set  $(\mathbb{R}^n \times \mathcal{F}) \cup \neg SI$  is reached infinitely many times with probability 1 in the *new product SDS*  $\mathcal{S}_{\text{new}}^\times$  controlled by our controller  $\pi^\times = (\pi^S, \pi^A)$ . To prove this, we need to show that  $V^{\text{live}}$  and  $I$  satisfy conditions **(L1)**-**(L3)** defined earlier in this section. Condition **(L1)** is implied by condition **((d))** in Definition 1. Condition **(L2)** is implied by condition **((e))** in Definition 1. Finally, condition **(L3)** is implied by conditions **((f))** in Definition 1 and by our construction of the new product SDS  $\mathcal{S}_{\text{new}}^\times$ , since it ensures that the value of  $V^{\text{live}}$  remains constant once a state in  $\neg SI$  is reached. Hence,  $V^{\text{live}}$  indeed defines an instance of the supermartingale certificate of [13] for proving that the target set  $(\mathbb{R}^n \times \mathcal{F}) \cup \neg SI$  is reached infinitely many times with probability 1 in  $\mathcal{S}_{\text{new}}^\times$ , and it follows that in the product SDS  $\mathcal{S}^\times$  we have

$$\mathbb{P}_{(x_0, q_{\text{init}})}^{\mathcal{S}^\times, \pi^\times} \left[ \text{Büchi}(\mathbb{R}^n \times \mathcal{F}) \cup \text{Reach}(\neg SI) \right] = 1, \quad (4)$$

where the equality indeed holds with respect to our controller  $\pi^\times = (\pi^S, \pi^A)$ ,  $\text{Büchi}(T)$  denotes the set of all paths in  $\mathcal{S}^\times$  that visit states in  $T$  infinitely many times, and  $\text{Reach}(T)$  denotes the set of all paths in  $\mathcal{S}^\times$  that eventually visit a state in  $T$ .

Combining eq. (3) and eq. (4) above, we conclude that for every initial state  $(x, q_{\text{init}}) \in \mathcal{X}^\times$  we have

$$\begin{aligned}
 \mathbb{P}_{(x_0, q_{\text{init}})}^{S^\times, \pi^\times} [Büchi(\mathbb{R}^n \times \mathcal{F})] &\geq \mathbb{P}_{(x_0, q_{\text{init}})}^{S^\times, \pi^\times} [Büchi(\mathbb{R}^n \times \mathcal{F}) \cup Reach(\neg SI)] - \mathbb{P}_{(x_0, q_{\text{init}})}^{S^\times, \pi^\times} [Reach(\neg SI)] \\
 &\geq \mathbb{P}_{(x_0, q_{\text{init}})}^{S^\times, \pi^\times} [Büchi(\mathbb{R}^n \times \mathcal{F}) \cup Reach(\neg SI)] - (1 - \mathbb{P}_{(x_0, q_{\text{init}})}^{S^\times, \pi^\times} [Safe(\neg SI)]) \\
 &\geq 1 - (1 - p) = p.
 \end{aligned}$$

This concludes the proof of the theorem claim.  $\square$

*Example 3 (LDBSM certificates).* Consider the 1D random walk from Example 1 and 2 with specification GF ( $x \leq 0$ ). The following constitute an LDBSM certificate: The invariant  $I$  in  $q_0$  is  $73 + (1/2)x$  and in  $q_1$  is the constant 0,  $V^{\text{safe}}(\cdot, q_0) = V^{\text{safe}}(\cdot, q_1) = -9 + (5/16)x$ ,  $V^{\text{live}}(\cdot, q_0) = (367/2) + (3/4)x$ ,  $V^{\text{live}}(\cdot, q_1) = (4747/128) - (1/256)x$ , with the variables  $\eta^S = -8$ ,  $\epsilon^S = 5/32$ ,  $M^S = 1$ ,  $\beta^S = -11/32$ ,  $\epsilon^L = 3/8$ , and  $M^L = 260$ . It can be checked that  $\{V^{\text{safe}}, V^{\text{live}}, I\}$  is a valid LDBSM that satisfies all the constraints in Definition 1.

*Remark 1 (On the incompleteness of LDBSM certificates).* Our LDBSM certificates are incomplete: it is possible that a certificate does not exist even though there exists a controller that fulfills the given  $\omega$ -regular specification. The incompleteness of LDBSM certificates follows from the incompleteness of our safety (based on repulsing supermartingales [21]) and liveness certificates (based on almost-sure recurrence certificates of Chakarov et al. [13]). The incompleteness of repulsing supermartingales was discussed in Chatterjee et al. [16], and we omit the details. An example of incompleteness of the liveness certificate is when the target set is closed under system dynamics and cannot be left once entered. In this case, infinitely many visits to the target is equivalent to reaching it once. For almost-sure reachability, the liveness certificate satisfying conditions (L1) and (L2) (see Sect. 3.1) becomes a ranking supermartingale of Chakarov and Sankaranarayanan [11]. Ranking supermartingales are sound and complete for finite expected time reachability [31], but are incomplete for almost-sure reachability.

## 4 Polynomial Template-Based Synthesis Algorithms

We now present our algorithms for automated verification and control of stochastic dynamical systems via our LDBSM certificates. While our theory on LDBSM certificates in Sect. 3 is applicable to general SDS and controllers, the algorithms in this section are restricted to systems that can be specified via polynomial arithmetic.

**Assumption: Polynomial Systems.** We assume that the system dynamics function  $f$ , the controller  $\pi$ , and arithmetic expressions for all atomic propositions appearing in the  $\omega$ -regular specification  $\varphi$  are *polynomial functions* over the state, control input, and stochastic disturbance variables. More generally,

we allow the system dynamics function  $f$  to be piecewise-polynomial. That is, it can be of the form

$$f(x, u, w) = \begin{cases} f_1(x, u, w), & \text{if } \bigwedge_{i=1}^{N_1} g_i^1(x) \leq 0, \\ \dots & \\ f_k(x, u, w), & \text{if } \bigwedge_{i=1}^{N_k} g_i^k(x) \leq 0, \end{cases} \quad (5)$$

where all  $f_i$ 's and  $g_i^j$ 's are polynomial functions over their input variables. Later, we will fix  $d$  as the specified maximum degree of our polynomial templates, and we assume that the first  $d$  moments of  $f(x, u, W)$ —with  $W$  being the random variable representing the distribution over noise—be representable and given as polynomials over  $x$  and  $u$ . For example, the system dynamics function in Example 1 is piecewise linear with  $k = 2$ .

**Input.** Our verification and control algorithms take as input an SDS  $\mathcal{S}$  whose system dynamics function is piecewise polynomial, an LDBA  $\mathcal{A}$  for an  $\omega$ -regular specification  $\varphi$  defined over a set of polynomial atomic prepositions  $P$ , and a minimum probability threshold  $p \in [0, 1]$  with which the  $\omega$ -regular specification  $\varphi$  should be satisfied. The verification algorithm also takes as input the polynomial controller  $\pi$ .

In addition, the algorithms take as input two parameters: (1) the polynomial degree  $d$  of polynomial templates used in the synthesis of the LDBSM certificate and the controller, and (2) the number of polynomial inequalities  $n_I$  used to define the supporting invariant. These two parameters are formally defined in Step 1 below.

The verification algorithm may accept polynomial supporting invariants as inputs, and if an invariant  $I$  is provided, then the algorithm only computes  $V^{\text{safe}}$  and  $V^{\text{live}}$  for the fixed  $I$ . For the control algorithm, the invariant depends on the controller, and is therefore always synthesized as a part of the certificate. For the uniformity of presentation, we will assume that the verification algorithm is not provided an input invariant.

**Output.** Both the verification and the control algorithms return as output the LDBSM certificate  $V = (V^{\text{safe}}, V^{\text{live}}, I)$ , as defined in Definition 1. The control algorithm also returns the SDS controller  $\pi$  computed by the algorithm. If the algorithms fail due to, e.g., the SMT solver timing out or returning “UNSAT,” they return “Unknown.”

**Algorithm Outline.** Our algorithms employ the classical *template-based synthesis* approach to synthesize an LDBSM certificate  $V = (V^{\text{safe}}, V^{\text{live}}, I)$  (and also the controller  $\pi$  for the control problem). This automation approach is similar to those employed in prior works on algorithmic synthesis of supermartingales for analyzing stochastic systems and programs with respect to reachability or safety properties [14, 16, 21, 58].

In Step 1, both the verification and the control algorithms fix symbolic polynomial templates with *unknown coefficients* for all objects that they need to compute. Namely, the verification algorithm fixes templates for the LDBSM certificate  $V = (V^{\text{safe}}, V^{\text{live}}, I)$ , while the control algorithm also fixes a template for

the controller  $\pi$ . The templates for  $V^{\text{safe}}$ ,  $V^{\text{live}}$ , and  $\pi$  are polynomial *expressions*, whereas the template for  $I$  is a conjunction of  $n_I$  polynomial inequalities of the form “ $P_I^j(x, q) \geq 0$ ,” with each  $P_I^j$  being a polynomial expression. In Step 2, the algorithms encode all certificate conditions in Definition 1 as a system of quantified polynomial entailments over the unknown template coefficients. In Step 3, the resulting system of quantified polynomial entailments is solved by employing existing algorithms and efficient tooling support provided by the POLYQENT tool [19], which reduces the problem to solving a purely existentially quantified system of polynomial constraints for which an off-the-shelf SMT solver is used. In the rest of this section, we provide a more detailed description of these steps.

**Step 1: Setting Up Templates.** The algorithms fix symbolic polynomial templates for all objects that they need to compute:

- *LDBSM certificate.* Suppose  $x_1, \dots, x_n$  represent the state variables for the  $n$  dimensions. For each LDBA state  $q \in Q$ , we set up the polynomial templates for  $V^{\text{safe}}(\cdot, q)$  and  $V^{\text{live}}(\cdot, q)$ , as well as for polynomial expressions  $P_I^1(\cdot, q), \dots, P_I^{n_I}(\cdot, q)$  over the state variables, each of polynomial degree  $d$  and with unknown coefficients whose values are to be found out in the subsequent steps of the algorithm. Here, the polynomial degree  $d$  and the number  $n_I$  of polynomial inequalities that define the invariant  $I$  are algorithm parameters. For example, if  $n = 2$ ,  $d = 2$ , and  $n_I = 1$ , then  $V^{\text{safe}}(x_1, x_2, q) = \theta_{11}^{q,\text{safe}} x_1^2 + \theta_{22}^{q,\text{safe}} x_2^2 + \theta_{12}^{q,\text{safe}} x_1 x_2 + \theta_1^{q,\text{safe}} x_1 + \theta_2^{q,\text{safe}} x_2 + \theta_0^{q,\text{safe}}$ ,  $V^{\text{live}}(x_1, x_2, q) = \theta_{11}^{q,\text{live}} x_1^2 + \theta_{22}^{q,\text{live}} x_2^2 + \theta_{12}^{q,\text{live}} x_1 x_2 + \theta_1^{q,\text{live}} x_1 + \theta_2^{q,\text{live}} x_2 + \theta_0^{q,\text{live}}$ , and  $(x_1, x_2, q) \in I$  is expressed as  $P_I^1(x, q) \geq 0$  where  $P_I^1(x, q) = \theta_{11}^{q,I} x_1^2 + \theta_{22}^{q,I} x_2^2 + \theta_{12}^{q,I} x_1 x_2 + \theta_1^{q,I} x_1 + \theta_2^{q,I} x_2 + \theta_0^{q,I} \geq 0$ .
- *Certificate parameters.* The algorithm introduces variables  $\underline{\eta}^S, \underline{\epsilon}^S, \underline{M}^S, \underline{\beta}^S, \underline{\epsilon}^L, \underline{M}^L$ , whose values will represent the constant parameters  $\eta^S \leq 0, \epsilon^S, M^S, \epsilon^L, M^L > 0$ , and  $\beta^S \in \mathbb{R}$  from Definition 1.
- *Controller (for the control algorithm).* For every LDBA state  $q \in Q$ , we set up a polynomial controller  $\pi(\cdot, q)$  over the state variables with degree  $d$  and with unknown coefficients. For example, if  $n = 2$  and  $d = 2$ , then  $\pi(x_1, x_2, q) = \lambda_{11}^q x_1^2 + \lambda_{22}^q x_2^2 + \lambda_{12}^q x_1 x_2 + \lambda_1^q x_1 + \lambda_2^q x_2 + \lambda_0^q$ , where  $\lambda$ -s are the unknown coefficients.

We will use  $\Omega$  to denote the set of all unknown coefficients (i.e.,  $\theta_{11}^{q,\text{safe}}, \theta_{11}^{q,\text{live}}, \theta_{11}^{q,I}, \dots$ ) in the polynomial templates for  $V^{\text{safe}}$ ,  $V^{\text{live}}$ ,  $I$ , and  $\pi$ , and the variables (i.e.,  $\underline{\epsilon}^S, \underline{\beta}^S, \dots$ ). For a given set of unknowns such as  $\Omega$ , we will write  $\llbracket \Omega \rrbracket$  to denote the set of all possible valuations of the unknowns in  $\Omega$ .

**Step 2: Constraint Collection.** The goal of this step is to symbolically evaluate the conditions in Definition 1, and rearrange them in a way that we finally obtain a system of polynomial inequality predicates in the prenex normal form

$$\exists \omega \in [\Omega] . \bigwedge_i \underbrace{(\forall q \in Q . \forall x \in \mathbb{R}^n . \forall w \in \mathbb{R}^p . \Phi_i^\omega(q, x, w) \implies \Psi_i^\omega(q, x, w))}_{R_i}, \quad (6)$$

where  $\Phi_i^\omega$  and  $\Psi_i^\omega$  are boolean combinations of polynomial inequalities over the variables  $q$ ,  $x$ , and  $w$  and unknown coefficients whose values conform with  $\omega$ . First, observe that the constraints (a)-(d) of Definition 1 generate constraints that are trivially of the form  $R_i$ . We explain the derivation for constraint ((e)), the procedure for ((f)) is analogous:

(A) **Prenex normal form:** First, we need to take out the two quantifications “ $\forall w \in \mathcal{W}$ ” from within “*SafetyCond*( $x, q, a, q'$ ).” We show the process for the clause “ $\forall w \in \mathcal{W}. \beta^S \leq V^{\text{safe}}(x, q) - V^{\text{safe}}(f(x, \pi(x), w), q') \leq \beta^S + M^S$ ,” call it  $\gamma_1$ , and for “ $\forall w \in \mathcal{W}. (f(x, \pi(x), w), q') \in I$ ,” call it  $\gamma_2$ , the procedure is identical. There are two possibilities:

- (i) Suppose the safety component of the LDBSM certificate  $V^{\text{safe}}$  and  $q'$ 's invariant polynomials  $\{P_I^i(\cdot, q')\}_{i \in [1; n_I]}$  are linear, i.e.,  $d = 1$ , the disturbance in the system is additive, i.e.,  $\forall x, u, w . f(x, u, w) = g(x, u) + w$  for some function  $g$ , and the domain  $\mathcal{W}$  of the disturbance is bounded within a given range  $[w_{\min}, w_{\max}]$ . Then, for  $\gamma_1$ , we can altogether suppress “ $\forall w \in \mathcal{W}$ ” and replace *SafetyCond*( $x, q, a, q'$ ) with

$$\left( \begin{array}{l} x \models a \wedge \forall w \in \mathcal{W}. (f(x, \pi(x), w), q') \in I \\ \wedge V^{\text{safe}}(x, q) \geq \mathbb{E}_w^\pi[V^{\text{safe}}(f(x, \pi(x), w), q')] + \epsilon^S \\ \wedge (\beta^S \leq V^{\text{safe}}(x, q) - V^{\text{safe}}(g(x, \pi(x)), q') - V^{\text{safe}}(w_{\min}, q') \leq \beta^S + M^S) \\ \wedge (\beta^S \leq V^{\text{safe}}(x, q) - V^{\text{safe}}(g(x, \pi(x)), q') - V^{\text{safe}}(w_{\max}, q') \leq \beta^S + M^S) \end{array} \right).$$

This transformation is sound since for every  $w \in \mathcal{W}$ , by the additivity of the noise and by the linearity of  $V^{\text{safe}}$ , we have  $V^{\text{safe}}(f(x, u, w)) = V^{\text{safe}}(g(x, u) + w) = V^{\text{safe}}(g(x, u)) + V^{\text{safe}}(w)$ , which is bounded between  $V^{\text{safe}}(g(x, u)) + V^{\text{safe}}(w_{\min})$  and  $V^{\text{safe}}(g(x, u)) + V^{\text{safe}}(w_{\max})$  from the boundedness of the noise and the linearity of  $V^{\text{safe}}$ . We can use the same idea for the clause  $\gamma_2$ .

- (ii) In general, the above restrictions could be insufficient, in which case we resort to the following stricter requirement at the expense of added incompleteness of the solution. The idea is that, instead of allowing  $w$  to depend on the choice of  $a \in \Sigma$  and  $q' \in \Delta(q, a)$ , for  $\gamma_1$ , we require the following the hold true:

$$\forall a \in \Sigma . \forall q' \in \Delta(q, a) . \forall w \in \mathcal{W} . \\ (\beta^S \leq V^{\text{safe}}(x, q) - V^{\text{safe}}(f(x, \pi(x), w), q') \leq \beta^S + M^S).$$

Let the expression above be written as  $D(x, q)$  in short, and it is easy to see that  $D(x, q)$  can be expressed in the desired form (6). We can now write the constraint ((e)) of Definition 1 as:

$$\begin{aligned}
 & D(x, q) \wedge \\
 & \forall x \in \mathbb{R}^n. V^{\text{safe}}(x, q) \leq 0 \implies \bigvee_{a \in \Sigma} \bigvee_{q' \in \Delta(q, a)} \\
 & \left( \begin{array}{l}
 x \models a \wedge \forall w \in \mathcal{W}. (f(x, \pi(x), w), q') \in I \\
 \wedge V^{\text{safe}}(x, q) \geq \mathbb{E}_w^\pi[V^{\text{safe}}(f(x, \pi(x), w), q')] + \epsilon^S \\
 \wedge V^{\text{live}}(x, q) \geq \mathbb{E}_w^\pi[V^{\text{live}}(f(x, \pi(x), w), q')] + \epsilon^L
 \end{array} \right)
 \end{aligned}$$

The same idea will work for  $\gamma_2$ .

- (B) **Simplifications.** After moving the quantification “ $\forall w \in \mathcal{W}$ ” from the inside of “*SafetyCond*( $x, q, a, q'$ )” to the outermost part of 2(b) and 2(c), we substitute all the variables and functions with the templates that we set up in Step 1. For the expected value  $\mathbb{E}_w^\pi[V^{\text{safe}}(f(x, \pi(x), w), q')]$ , we use the linearity of the expectation operator to break it down into a polynomial over  $x$  whose coefficients are functions of the moments of the distribution over  $w$  (assumed to be given as input). For instance, since  $V^{\text{safe}}(x, q)$  is a polynomial over  $x$ , and  $f(x, \pi(x), w) = C_2(x)w^2 + C_1(x)w + C_0(x)$  where  $C_2, C_1, C_0$  are polynomials over  $x$ , it follows that  $V^{\text{safe}}(f(x, \pi(x), w), q) = \hat{C}_2(x)w^2 + \hat{C}_1(x)w + \hat{C}_0(x)$  where  $\hat{C}_2, \hat{C}_1, \hat{C}_0$  are polynomials over  $x$ . This gives us:  $\mathbb{E}_w^\pi[V^{\text{safe}}(f(x, \pi(x), w), q)] = \hat{C}_2(x) \cdot \mathbb{E}_w^\pi[w^2] + \hat{C}_1(x) \cdot \mathbb{E}_w^\pi[w] + \hat{C}_0(x)$ , where  $\mathbb{E}_w^\pi[w^i]$  is the  $i$ -th moment of the distribution (therefore, a constant) of the noise.

**Step 3: Constraint Solving.** Finally, the system of quantified polynomial entailments collected in Step 2 is solved. This is achieved by first employing Putinar’s theorem [53] (or Farkas’ lemma [28], if all involved polynomials have degree 1) to translate each quantified polynomial entailment into a system of purely existentially quantified polynomial constraints over the unknown template variables, as well as fresh variables introduced by the Putinar’s theorem translation. Since this step is standard in the template-based synthesis literature, including that on the synthesis of supermartingale certificates [14, 16, 21, 58], we omit the details. The resulting system of purely existentially quantified polynomial constraints over real-valued variables is then solved by using an SMT solver. In our implementation, the application of Putinar’s theorem and reduction to SMT solving are achieved via the POLYQENT tool [19]. See [19] for details, including on how boolean combinations of polynomial inequalities are handled prior to applying Putinar’s theorem. It is straightforward to note that if the encoding (6) is satisfiable, then we obtain an LDBSM certificate, and in case of control problem, we also get a controller. Then from the soundness of the POLYQENT tool [19] that we use to solve (6), Theorem 2 follows.

**Theorem 2 (Algorithm soundness).** *Suppose that the verification algorithm returns an LDBSM certificate  $V = (V^{\text{safe}}, V^{\text{live}}, I)$ . Then,  $V$  is an LDBSM certificate with respect to the invariant  $I$  as in Definition 1, and the SDS  $\mathcal{S}$  under controller  $\pi$  satisfies specification  $\varphi$  with probability at least  $p$ .*

Suppose that the control algorithm returns an LDBSM certificate  $V = (V^{\text{safe}}, V^{\text{live}}, I)$  and an SDS controller  $\pi$ . Then,  $V$  is an LDBSM certificate with respect to the invariant  $I$  as in Definition 1, and the SDS  $S$  under controller  $\pi$  satisfies specification  $\varphi$  with probability at least  $p$ . Similar to existing template-based synthesis methods for polynomial program verification and controller synthesis [5, 14], the runtime complexity of our verification and control algorithms is in PSPACE when the polynomial degree  $d$  is treated as a constant parameter. This is because the size of the final SMT query produced by our method is  $\mathcal{O}(n_I * |Q| * n^d)$ . As the final SMT query is a sentence in the existential theory of the reals, the parameterized PSPACE runtime complexity follows.

## 5 Experiments

We implemented our algorithm as a proof-of-concept tool in Python, where our tool automatically translates the given inputs to quantified polynomial entailments by using the procedure outlined in Sect. 4, and then uses the tool POLYQENT [19] as its back end to compute solutions. POLYQENT uses Putinar’s theorem [53] (or Farkas’ lemma [28] for linear systems and certificate and controller templates) and translates the provided entailments into existentially quantified polynomial constraints over unknown template variables, and uses off-the-shelf SMT solvers Z3 [25] and MathSAT [10] to find satisfying assignments of the variables. Our prototype tool is available at <https://github.com/Ipouyall/Omega-Regular-Stoch-Cert>.

We used our tool to automatically compute LDBSM certificates for the verification of the random walk SDS  $\mathcal{S}^{\text{rw}}$  from Example 1, and for the control of a modified version of  $\mathcal{S}^{\text{rw}}$  defined as follows. The system  $\mathcal{S}_c^{\text{rw}}$  has the state space  $\mathbb{R}$ , input space  $\mathcal{U} = [-2, 2]$ , disturbance space  $\mathcal{W} = [0, 1]$  with a continuous uniform distribution over  $\mathcal{W}$ , system dynamics given as

$$f_c(x, u, w) = \begin{cases} x & \text{if } x > 100, \\ x + u + w & \text{otherwise,} \end{cases} \quad (7)$$

and the set of initial states [2, 3]. Both the verification and the control example were solved with a number of different specifications as listed in Table 1. For each specification, the general requirement is to push the state towards the origin (eventually for  $F a$ , always eventually for  $GF a$ , etc.). For the verification problems, the disturbances in  $\mathcal{W} = [-2, 1]$  create an automatic bias towards the origin, which ultimately helps satisfying the specifications. For the control problems, the disturbances in  $\mathcal{W} = [0, 1]$  create a challenge by introducing bias *away* from the origin, so the controller needs to counteract with negative control inputs at each step.

All the experiments were performed on an Apple Macbook Air with M2 chip and 16 GB RAM, and we report the results in Table 1. In both the verification and control problems, the SDS’s have small probabilities of violating the objectives, by reaching the region  $x > 100$  and not being able to come out afterwards.

Therefore, existing approaches supporting only qualitative (almost sure satisfaction)  $\omega$ -regular specifications would not work, and no single supermartingale-based approach for quantitative specifications would support all the specifications at once. In contrast, our tool achieved 99% probability for satisfying the specifications, and computed LDBSM certificates (and controllers for the control problem) within minutes.

**Table 1.** Experimental results for verification and control of the random walk examples. The predicates are:  $a := x \leq 0$  and  $b := x \leq 100$ . Each experiment was run with polynomial degree  $d = 1$ .

	Specification	Probability	Time (s)
Verification (SDS $\mathcal{S}^{\text{rw}}$ )	$F a$	0.9999	0.35
	$GF a$	0.9999	27.28
	$bU a$	0.9999	6.13
	$Gb \wedge F a$	0.9999	27.78
	$Gb$	0.9999	28.60
	$F a \wedge F b$	0.9999	172.28
Control (SDS $\mathcal{S}_c^{\text{rw}}$ )	$F a$	0.9999	0.39
	$GF a$	0.9999	4.90
	$bU a$	0.9999	18.90
	$Gb \wedge F a$	0.9999	46.68
	$Gb$	0.9999	4.04

## 6 Related Work

Earliest known types of certificates in control theory date back to mid-twentieth century, and were developed for certifying stability [34, 36] and invariance [44] of dynamical systems. However, back then, obtaining suitable certificates for general nonlinear systems was largely a manual process. It was in early 2000s, when the seminal work of Parrilo [50] first proposed the use of Positivstellensätze to automate the process of certificate computation, which were later strengthened and refined [49, 56].

In parallel to control theory, from early 2000s, program analysis also experienced an emergence of certificate-based approaches for proving termination of deterministic programs [22, 23]. In the landmark paper by Chakarov and Sankarnarayanan [11], ranking functions for deterministic programs were generalized to ranking supermartingales for probabilistic programs. This has sparked interest in using supermartingale certificates for the analysis of probabilistic programs, with significant advances in supermartingale certificates and automated algorithms for termination [2, 14, 15, 20, 39, 43],

safety [12, 16, 21, 58, 59], Büchi and co-Büchi (stability) [13], and cost analysis [17, 47, 60]. These results were also exported to control theory to solve the verification and control problem for stochastic dynamical systems with logical specifications [51, 63, 66].

A majority of existing supermartingale certificates for stochastic systems only support fragments of  $\omega$ -regular specifications. Only recently, Abate et al. [3] proposed a new class of supermartingale certificates for  $\omega$ -regular specifications represented using Streett automata. However, their certificate is restricted to the probability 1 satisfaction of the specification, contrary to our quantitative certificate that allows arbitrary probability thresholds to be imposed as a requirement for satisfying the specification. It is important to note that our certificate is incomplete, like most other existing supermartingale certificates that are available in the literature, see Remark 1.

It is important to mention that there are also other non-certificate-based approaches to verification and control of stochastic systems. For example, there is a long line of research that falls under the abstraction-based control paradigm, where a given system model is abstracted to a simpler graph, so that graph theoretic techniques can be applied to approximately solve the original verification and control problem [26, 27, 38, 40, 64]. Among these works, only one is known to be able to support quantitative  $\omega$ -regular specifications [26], though their functionalities are complementary to ours: They support general nonlinear systems but only having additive control input and additive stochastic noise with unimodal, symmetric distribution. We support polynomial systems and controllers and stochastic noise that can have arbitrary distributions.

## 7 Discussions

We present LDBSM certificates, the first supermartingale-based certificates for quantitative  $\omega$ -regular verification and control of infinite-state stochastic systems. LDBSM certificates are defined on the product of the system and the equivalent LDBA representation of the given  $\omega$ -regular specification, and they generalize and combine existing safety and liveness supermartingale certificates in a hierarchical manner to achieve soundness. We supplement the theoretical development with a standard template-based synthesis algorithm for solving the verification and control problems in polynomial stochastic systems.

There are several interesting future directions. First, we will explore alternate methods for computing LDBSM certificates, including learning-assisted approaches [18, 66], alongside optimizing our template-based approach that relies on off-the-shelf constraint solvers. Even the state-of-the-art constraint solvers struggled to compute LDBSM certificates, and eliminating this tooling bottleneck will be a priority in future works. Second, in the same vein as the previous one, we will explore compositional approaches for LDBSM certificates for better scalability; such approaches are well-studied for existing classes of certificates [45, 67]. Finally, we will explore completeness questions. Right now, LDBSM certificates provide sound but incomplete solutions: if they exist, then it follows that the specification holds, but not the other way round.

**Acknowledgments.** This work was supported in part by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant (Project ID:22-SIS-SMU-100) and the ERC project ERC-2020-AdG 101020093.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Abate, A., Edwards, A., Giacobbe, M., Punchihewa, H., Roy, D.: Quantitative verification with neural networks. In: CONCUR. LIPIcs, vol. 279, pp. 22:1–22:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2023)
2. Abate, A., Giacobbe, M., Roy, D.: Learning probabilistic termination proofs. In: Silva, A., Leino, K. (eds.) CAV 2021. LNCS, vol. 12760, pp. 3–26. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-81688-9\\_1](https://doi.org/10.1007/978-3-030-81688-9_1)
3. Abate, A., Giacobbe, M., Roy, D.: Stochastic omega-regular verification and control with supermartingales. In: International Conference on Computer Aided Verification, pp. 395–419. Springer, Heidelberg (2024). [https://doi.org/10.1007/978-3-031-65633-0\\_18](https://doi.org/10.1007/978-3-031-65633-0_18)
4. Ansaripour, M., Chatterjee, K., Henzinger, T.A., Lechner, M., Žikelić, Đ.: Learning provably stabilizing neural controllers for discrete-time stochastic systems. In: International Symposium on Automated Technology for Verification and Analysis, pp. 357–379. Springer, Heidelberg (2023). [https://doi.org/10.1007/978-3-031-45329-8\\_17](https://doi.org/10.1007/978-3-031-45329-8_17)
5. Asadi, A., Chatterjee, K., Fu, H., Goharshady, A.K., Mahdavi, M.: Polynomial reachability witnesses via stellensätze. In: PLDI, pp. 772–787. ACM (2021)
6. Åström, K.J.: Introduction to Stochastic Control Theory. Courier Corporation, Chelmsford (2012)
7. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT press, Cambridge (2008)
8. Barthe, G., Gaboardi, M., Grégoire, B., Hsu, J., Strub, P.: Proving differential privacy via probabilistic couplings. In: Grohe, M., Koskinen, E., Shankar, N. (eds.) Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, 5–8 July 2016, pp. 749–758. ACM (2016). <https://doi.org/10.1145/2933575.2934554>
9. Beutner, R., Ong, C.L., Zaiser, F.: Guaranteed bounds for posterior inference in universal probabilistic programming. In: PLDI, pp. 536–551. ACM (2022)
10. Bruttomesso, R., Cimatti, A., Franzén, A., Griggio, A., Sebastiani, R.: The mathsat 4 smt solver: Tool paper. In: Computer Aided Verification: 20th International Conference, CAV 2008 Princeton, NJ, USA, 7–14 July 2008 Proceedings 20, pp. 299–303. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-70545-1\\_28](https://doi.org/10.1007/978-3-540-70545-1_28)
11. Chakarov, A., Sankaranarayanan, S.: Probabilistic program analysis with martingales. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 511–526. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39799-8\\_34](https://doi.org/10.1007/978-3-642-39799-8_34)
12. Chakarov, A., Sankaranarayanan, S.: Expectation Invariants for Probabilistic Program Loops as Fixed Points, pp. 85–100. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-319-10936-7\\_6](https://doi.org/10.1007/978-3-319-10936-7_6)

13. Chakarov, A., Voronin, Y.-L., Sankaranarayanan, S.: Deductive proofs of almost sure persistence and recurrence properties. In: Chechik, M., Raskin, J.-F. (eds.) TACAS 2016. LNCS, vol. 9636, pp. 260–279. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-49674-9\\_15](https://doi.org/10.1007/978-3-662-49674-9_15)
14. Chatterjee, K., Fu, H., Goharshady, A.K.: Termination analysis of probabilistic programs through positivstellensatz's. In: Chaudhuri, S., Farzan, A. (eds.) CAV 2016. LNCS, vol. 9779, pp. 3–22. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-41528-4\\_1](https://doi.org/10.1007/978-3-319-41528-4_1)
15. Chatterjee, K., Fu, H., Novotný, P., Hasheminezhad, R.: Algorithmic analysis of qualitative and quantitative termination problems for affine probabilistic programs. *TOPLAS* **40**(2), 7:1–7:45 (2018). <https://doi.org/10.1145/3174800>
16. Chatterjee, K., Goharshady, A.K., Meggendorfer, T., Žikelić, Đ.: Sound and complete certificates for quantitative termination analysis of probabilistic programs. In: International Conference on Computer Aided Verification, pp. 55–78. Springer, Heidelberg (2022). [https://doi.org/10.1007/978-3-031-13185-1\\_4](https://doi.org/10.1007/978-3-031-13185-1_4)
17. Chatterjee, K., Goharshady, A.K., Meggendorfer, T., Žikelić, Đ.: Quantitative bounds on resource usage of probabilistic programs. *Proc. ACM Program. Lang.* **8**(OOPSLA1), 362–391 (2024)
18. Chatterjee, K., Henzinger, T.A., Lechner, M., Žikelić, Đ.: A learner-verifier framework for neural network controllers and certificates of stochastic systems. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pp. 3–25. Springer, Heidelberg (2023). [https://doi.org/10.1007/978-3-031-30823-9\\_1](https://doi.org/10.1007/978-3-031-30823-9_1)
19. Chatterjee, K., et al.: Polyqent: a polynomial quantified entailment solver. arXiv e-prints pp. arXiv–2408 (2024)
20. Chatterjee, K., Kafshdar Goharshady, E., Novotný, P., Zárevúcky, J., Žikelić, Đ.: On lexicographic proof rules for probabilistic termination. *Formal Aspects Comput.* **35**(2), 1–25 (2023)
21. Chatterjee, K., Novotný, P., Zikelic, D.: Stochastic invariants for probabilistic termination. In: Castagna, G., Gordon, A.D. (eds.) Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, 18–20 January 2017, pp. 145–160. ACM (2017). <https://doi.org/10.1145/3009837.3009873>
22. Colón, M.A., Sipma, H.B.: Practical methods for proving program termination. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 442–454. Springer, Heidelberg (2002). [https://doi.org/10.1007/3-540-45657-0\\_36](https://doi.org/10.1007/3-540-45657-0_36)
23. Colón, M.A., Sipma, H.B.: Synthesis of linear ranking functions. In: Margaria, T., Yi, W. (eds.) TACAS 2001. LNCS, vol. 2031, pp. 67–81. Springer, Heidelberg (2001). [https://doi.org/10.1007/3-540-45319-9\\_6](https://doi.org/10.1007/3-540-45319-9_6)
24. Courcoubetis, C., Yannakakis, M.: The complexity of probabilistic verification. *J. ACM* **42**(4), 857–907 (1995)
25. de Moura, L., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 337–340. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-78800-3\\_24](https://doi.org/10.1007/978-3-540-78800-3_24)
26. Dutreix, M., Huh, J., Coogan, S.: Abstraction-based synthesis for stochastic systems with omega-regular objectives. *Nonlinear Anal. Hybrid Syst* **45**, 101204 (2022)
27. Esmaeil Zadeh Soudjani, S., Abate, A.: Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM J. Appl. Dyn. Syst.* **12**(2), 921–956 (2013)

28. Farkas, J.: Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik (Crelles J.)* **1902**(124), 1–27 (1902)
29. Florchinger, P.: Lyapunov-like techniques for stochastic stability. *SIAM J. Control. Optim.* **33**(4), 1151–1169 (1995)
30. Foster, N., Kozen, D., Mamouras, K., Reitblatt, M., Silva, A.: Probabilistic NetKAT. In: Thiemann, P. (ed.) *ESOP 2016*. LNCS, vol. 9632, pp. 282–309. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-49498-1\\_12](https://doi.org/10.1007/978-3-662-49498-1_12)
31. Fu, H., Chatterjee, K.: Termination of nondeterministic probabilistic programs. In: Enea, C., Piskac, R. (eds.) *VMCAI 2019*. LNCS, vol. 11388, pp. 468–490. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-11245-5\\_22](https://doi.org/10.1007/978-3-030-11245-5_22)
32. Gehr, T., Misailovic, S., Vechev, M.T.: PSI: exact symbolic inference for probabilistic programs. In: *CAV* (2016)
33. Ghahramani, Z.: Probabilistic machine learning and artificial intelligence. *Nature* **521**(7553), 452–459 (2015). <https://doi.org/10.1038/nature14541>
34. Kalman, R.E., Bertram, J.E.: Control system analysis and design via the “second method” of lyapunov: I—continuous-time systems (1960)
35. Kaminski, B.L., Katoen, J., Matheja, C., Olmedo, F.: Weakest precondition reasoning for expected runtimes of randomized algorithms. *J. ACM* **65**(5), 30:1–30:68 (2018). <https://doi.org/10.1145/3208102>
36. LaSalle, J., Lefschetz, S.: *Stability by Lyapunov’s Second Method with Applications*. Academic Press, New York (1961)
37. Lechner, M., Žikelić, Đ., Chatterjee, K., Henzinger, T.A.: Stability verification in stochastic control systems via neural network supermartingales. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 7326–7336 (2022)
38. Majumdar, R., Mallik, K., Schmuck, A.K., Soudjani, S.: Symbolic control for stochastic systems via finite parity games. *Nonlinear Anal. Hybrid Syst* **51**, 101430 (2024)
39. Majumdar, R., Sathiyarayanan, V.: Sound and complete proof rules for probabilistic termination. *Proc. ACM Program. Lang.* **9**(POPL), 1871–1902 (2025)
40. Mallik, K., Soudjani, S.E.Z., Schmuck, A.K., Majumdar, R.: Compositional construction of finite state abstractions for stochastic control systems. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 550–557. IEEE (2017)
41. Mathiesen, F.B., Calvert, S.C., Laurenti, L.: Safety certification for stochastic systems via neural barrier functions. *IEEE Control Syst. Lett.* **7**, 973–978 (2022)
42. McIver, A., Morgan, C.: *Abstraction, Refinement and Proof for Probabilistic Systems*. Monographs in Computer Science. Springer, Heidelberg (2005)
43. McIver, A., Morgan, C., Kaminski, B.L., Katoen, J.P.: A new proof rule for almost-sure termination. *Proc. ACM Program. Lang.* **2**(POPL), 1–28 (2017)
44. Nagumo, M.: Über die lage der integralkurven gewöhnlicher differentialgleichungen. *Proc. Physico-Math. Soc. Jpn. 3rd Series* **24**, 551–559 (1942)
45. Nejati, A., Soudjani, S., Zamani, M.: Compositional construction of control barrier functions for continuous-time stochastic hybrid systems. *Automatica* **145**, 110513 (2022)
46. Neustroev, G., Giacobbe, M., Lukina, A.: Neural continuous-time supermartingale certificates. In: *AAAI*, pp. 27538–27546. AAAI Press (2025)
47. Ngo, V.C., Carbonneaux, Q., Hoffmann, J.: Bounded expectations: resource analysis for probabilistic programs. *ACM SIGPLAN Not.* **53**(4), 496–512 (2018)
48. Olmedo, F., Kaminski, B.L., Katoen, J.P., Matheja, C.: Reasoning about recursive probabilistic programs. In: *LICS*, pp. 672–681 (2016). <https://doi.org/10.1145/2933575.2935317>

49. Papachristodoulou, A., Prajna, S.: A tutorial on sum of squares techniques for systems analysis. In: Proceedings of the 2005, American Control Conference, 2005, pp. 2686–2700. IEEE (2005)
50. Parrilo, P.A.: Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. California Institute of Technology (2000)
51. Prajna, S., Jadbabaie, A., Pappas, G.J.: Stochastic safety verification using barrier certificates. In: 2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601), vol. 1, pp. 929–934. IEEE (2004)
52. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley Series in Probability and Statistics. Wiley, Hoboken (1994). <https://doi.org/10.1002/9780470316887>
53. Putinar, M.: Positive polynomials on compact semi-algebraic sets. Indiana Univ. Math. J. **42**(3), 969–984 (1993)
54. Sankaranarayanan, S., Chakarav, A., Gulwani, S.: Static analysis for probabilistic programs: inferring whole program properties from finitely many paths. In: Boehm, H., Flanagan, C. (eds.) ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '13, Seattle, WA, USA, 16–19 June 2013, pp. 447–458. ACM (2013). <https://doi.org/10.1145/2491956.2462179>
55. Segala, R.: Verification of randomized distributed algorithms. In: Brinksma, E., Hermanns, H., Katoen, J.-P. (eds.) EEF School 2000. LNCS, vol. 2090, pp. 232–260. Springer, Heidelberg (2001). [https://doi.org/10.1007/3-540-44667-2\\_6](https://doi.org/10.1007/3-540-44667-2_6)
56. Sloth, C.: On the computation of lyapunov functions for interconnected systems. In: 2016 IEEE Conference on Computer Aided Control System Design (CACSD), pp. 850–855. IEEE (2016)
57. Steinhardt, J., Tedrake, R.: Finite-time regional verification of stochastic non-linear systems. Int. J. Rob. Res. **31**(7), 901–923 (2012)
58. Takisaka, T., Oyabu, Y., Urabe, N., Hasuo, I.: Ranking and repulsing supermartingales for reachability in randomized programs. ACM Trans. Program. Lang. Syst. (TOPLAS) **43**(2), 1–46 (2021)
59. Wang, J., Sun, Y., Fu, H., Chatterjee, K., Goharshady, A.K.: Quantitative analysis of assertion violations in probabilistic programs. In: Freund, S.N., Yahav, E. (eds.) PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual Event, Canada, 20–25 June 2021, pp. 1171–1186. ACM (2021). <https://doi.org/10.1145/3453483.3454102>
60. Wang, P., Fu, H., Goharshady, A.K., Chatterjee, K., Qin, X., Shi, W.: Cost analysis of nondeterministic probabilistic programs. In: McKinley, K.S., Fisher, K. (eds.) Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2019, Phoenix, AZ, USA, 22–26 June 2019, pp. 204–220. ACM (2019). <https://doi.org/10.1145/3314221.3314581>
61. Wang, P., Yang, T., Fu, H., Li, G., Ong, C.L.: Static posterior inference of bayesian probabilistic programming via polynomial solving. Proc. ACM Program. Lang. **8**(PLDI), 1361–1386 (2024)
62. Williams, D.: Probability with Martingales. Cambridge mathematical textbooks, Cambridge University Press, Cambridge (1991)
63. Xue, B., Zhan, N., Fränzle, M.: Reach-avoid analysis for polynomial stochastic differential equations. IEEE Trans. Autom. Control (2023)
64. Zamani, M., Esfahani, P.M., Majumdar, R., Abate, A., Lygeros, J.: Symbolic control of stochastic systems via approximately bisimilar finite abstractions. IEEE Trans. Autom. Control **59**(12), 3135–3150 (2014)

65. Zhi, D., Wang, P., Liu, S., Ong, C.L., Zhang, M.: Unifying qualitative and quantitative safety verification of dnn-controlled systems. In: CAV (2). Lecture Notes in Computer Science, vol. 14682, pp. 401–426. Springer, Heidelberg (2024). [https://doi.org/10.1007/978-3-031-65630-9\\_20](https://doi.org/10.1007/978-3-031-65630-9_20)
66. Žikelić, Đ., Lechner, M., Henzinger, T.A., Chatterjee, K.: Learning control policies for stochastic systems with reach-avoid guarantees. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 11926–11935 (2023)
67. Žikelić, Đ., Lechner, M., Verma, A., Chatterjee, K., Henzinger, T.: Compositional policy learning in stochastic control systems with formal guarantees. Adv. Neural Inf. Process. Syst. **36** (2024)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

