Average Case Analysis of the Classical Algorithm for Markov Decision Processes with Büchi Objectives*

Krishnendu Chatterjee[1], Manas Joglekar[2], and Nisarg Shah[3]

1 **IST Austria (Institute of Science and Technology Austria)**
2 **Stanford University**
3 **Carnegie Mellon University**

Abstract

We consider Markov decision processes (MDPs) with specifications given as Büchi (liveness) objectives. We consider the problem of computing the set of *almost-sure* winning vertices from where the objective can be ensured with probability 1. We study for the first time the average case complexity of the classical algorithm for computing the set of almost-sure winning vertices for MDPs with Büchi objectives. Our contributions are as follows: First, we show that for MDPs with constant out-degree the expected number of iterations is at most logarithmic and the average case running time is linear (as compared to the worst case linear number of iterations and quadratic time complexity). Second, for the average case analysis over all MDPs we show that the expected number of iterations is constant and the average case running time is linear (again as compared to the worst case linear number of iterations and quadratic time complexity). Finally we also show that given that all MDPs are equally likely, the probability that the classical algorithm requires more than constant number of iterations is exponentially small.

1998 ACM Subject Classification D.2.4 Formal methods

Keywords and phrases MDPs, Büchi objectives, Average case analysis

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2012.461

1 Introduction

Markov decision processes. *Markov decision processes (MDPs)* are standard models for probabilistic systems that exhibit both probabilistic and nondeterministic behavior [13], and widely used in verification of probabilistic systems [1, 15]. MDPs have been used to model and solve control problems for stochastic systems [12]: there, nondeterminism represents the freedom of the controller to choose a control action, while the probabilistic component of the behavior describes the system response to control actions. MDPs have also been adopted as models for concurrent probabilistic systems [7], probabilistic systems operating in open environments [18], under-specified probabilistic systems [2], and applied in diverse domains [15]. A *specification* describes the set of desired behaviors of the system, which in the verification and control of stochastic systems is typically an ω-regular set of paths. The class of ω-regular languages extends classical regular languages to infinite strings, and provides a robust specification language to express all commonly used specifications, such

* The research was supported by FWF Grant No P 23499-N23, FWF NFN Grant No S11407-N23 (RiSE), ERC Start grant (279307: Graph Games), and Microsoft faculty fellows award.

32nd Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012).
Editors: D. D'Souza, J. Radhakrishnan, and K. Telikepalli; pp. 461–473
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

as safety, liveness, fairness, etc [20]. Parity objectives are a canonical way to define such ω-regular specifications. Thus MDPs with parity objectives provide the theoretical framework to study problems such as the verification and control of stochastic systems.

Qualitative and quantitative analysis. The analysis of MDPs with parity objectives can be classified into qualitative and quantitative analysis. Given an MDP with parity objective, the *qualitative analysis* asks for the computation of the set of vertices from where the parity objective can be ensured with probability 1 (almost-sure winning). The more general *quantitative analysis* asks for the computation of the maximal (or minimal) probability at each state with which the controller can satisfy the parity objective.

Importance of qualitative analysis. The qualitative analysis of MDPs is an important problem in verification that is of interest independent of the quantitative analysis problem. There are many applications where we need to know whether the correct behavior arises with probability 1. For instance, when analyzing a randomized embedded scheduler, we are interested in whether every thread progresses with probability 1 [9]. Even in settings where it suffices to satisfy certain specifications with probability $p < 1$, the correct choice of p is a challenging problem, due to the simplifications introduced during modeling. For example, in the analysis of randomized distributed algorithms it is quite common to require correctness with probability 1 (see, e.g., [16, 14, 19]). Furthermore, in contrast to quantitative analysis, qualitative analysis is robust to numerical perturbations and modeling errors in the transition probabilities, and consequently the algorithms for qualitative analysis are combinatorial. Finally, for MDPs with parity objectives, the best known algorithms and all algorithms used in practice first perform the qualitative analysis, and then perform a quantitative analysis on the result of the qualitative analysis [7, 8, 6]. Thus qualitative analysis for MDPs with parity objectives is one of the most fundamental and core problems in verification of probabilistic systems.

Previous results. The qualitative analysis for MDPs with parity objectives is achieved by iteratively applying solutions of the qualitative analysis of MDPs with Büchi objectives [7, 8, 6]. The qualitative analysis of an MDP with a parity objective with d priorities can be achieved by $O(d)$ calls to an algorithm for qualitative analysis of MDPs with Büchi objectives, and hence we focus on MDPs with Büchi objectives. The qualitative analysis problem for MDPs with Büchi objectives has been widely studied. The classical algorithm for the problem was given in [7, 8], and the worst case running time of the classical algorithm is $O(n \cdot m)$ time, where n is the number of vertices, and m is the number of edges of the MDP. Many improved algorithms have also been given in literature, such as [5, 3, 4], and the current best known worst case complexity of the problem is $O(\min\{n^2, m \cdot \sqrt{m}\})$. Moreover, there exists a family of MDPs where the running time of the improved algorithms match the above bound. While the worst case complexity of the problem has been studied, to the best of our knowledge the average case complexity of none of the algorithms has been studied in literature.

Our contribution. In this work we study for the first time the average case complexity of the qualitative analysis of MDPs with Büchi objectives. Specifically we study the average case complexity of the classical algorithm for the following two reasons: (1) the classical algorithm is very simple and appealing as it iteratively uses solutions of the standard graph reachability and alternating graph reachability algorithms, and can be implemented efficiently by symbolic algorithms; and (2) for the more involved improved algorithms it has been established that there are simple variants of the improved algorithms that never require more than an additional linear time as compared to the classical algorithm, and hence the average case complexity of these variants is no more than the average case complexity of the classical algorithm. We study the average case complexity of the classical algorithm and establish that as compared to the quadratic worst case complexity, the average case complexity is linear. Our main contributions are summarized below:

1. *MDPs with constant out-degree.* We first consider MDPs with constant out-degree. In practice, MDPs often have constant out-degree: for example, see [10] for MDPs with large state space but constant number of actions, or [12, 17] for examples from inventory management where MDPs have constant number of actions (the number of actions correspond to the out-degree of MDPs). We consider MDPs where the out-degree of every vertex is fixed and given. The out-degree of a vertex v is d_v and there are constants $d_{\min}$ and $d_{\max}$ such that for every v we have $d_{\min} \leq d_v \leq d_{\max}$. Moreover, every subset of the set of vertices of size d_v is equally likely to be the neighbour set of v, independent of the neighbour sets of other vertices. We show that the expected number of iterations of the classical algorithm is at most logarithmic ($O(\log n)$), and the average case running time is linear ($O(n)$) (as compared to the worst case linear number of iterations and quadratic $O(n^2)$ time complexity of the classical algorithm, and the current best known $O(n \cdot \sqrt{n})$ worst case complexity). The average case complexity of this model implies the same average case complexity for several related models of MDPs with constant out-degree. For further discussion on this, see Remark 3.4 following Theorem 16.
2. *MDPs in the Erdös-Rényi model.* To consider the average case complexity over all MDPs, we consider MDPs where the underlying graph is a random directed graph according to the classical Erdös-Rényi random graph model [11]. We consider random graphs $\mathcal{G}_{n,p}$, over n vertices where each edge exists with probability p (independently of other edges). To analyze the average case complexity over all MDPs with all graphs equally likely we need to consider the $\mathcal{G}_{n,p}$ model with $p = \frac{1}{2}$ (i.e., each edge is present or absent with equal probability, and thus all graphs are considered equally likely). We show a stronger result (than only $p = \frac{1}{2}$) that if $p \geq \frac{c \cdot \log(n)}{n}$, for any constant $c > 2$, then the expected number of iterations of the classical algorithm is constant ($O(1)$), and the average case running time is linear (again as compared to the worst case linear number of iterations and quadratic time complexity). Note that we obtain that the average case (when $p = \frac{1}{2}$) running time for the classical algorithm is linear over all MDPs (with all graphs equally likely) as a special case of our results for $p \geq \frac{c \cdot \log(n)}{n}$, for any constant $c > 2$, since $\frac{1}{2} \geq \frac{3 \cdot \log(n)}{n}$ for $n \geq 17$. Moreover we show that when $p = \frac{1}{2}$ (i.e., all graphs are equally likely), the probability that the classical algorithm will require more than constantly many iterations is exponentially small (less than $\left(\frac{3}{4}\right)^n$).

Implications of our results. We now discuss several implications of our results. First, since we show that the classical algorithm has average case linear time complexity, it follows that the average case complexity of qualitative analysis of MDPs with Büchi objectives is linear time. Second, since qualitative analysis of MDPs with Büchi objectives is a more general problem than reachability in graphs (graphs are a special case of MDPs and reachability objectives are a special case of Büchi objectives), the best average case complexity that can be achieved is linear. Hence our results for the average case complexity are tight. Finally, since for the improved algorithms there are simple variants that never require more than linear time as compared to the classical algorithm it follows that the improved algorithms also have average case linear time complexity. Thus we complete the average case analysis of the algorithms for the qualitative analysis of MDPs with Büchi objectives. In summary our results show that the classical algorithm (the most simple and appealing algorithm) has excellent and optimal (linear-time) average case complexity as compared to the quadratic worst case complexity.

Technical contributions. The two key technical difficulties to establish our results are as follows: (1) Though there are many results for random undirected graphs, for the average case analysis of the classical algorithm we need to analyze random directed graphs; and (2) in contrast to other results related to random undirected graphs that prove results for almost all vertices, the classical algorithm stops when all vertices satisfy a certain reachability property; and hence we need to prove results for all vertices (as compared to almost all vertices). In this work we set up novel recurrence relations to estimate the expected number of iterations, and the average case running time of the classical algorithm. Our key technical results prove many interesting inequalities related to the recurrence relation for reachability properties of random directed graphs to establish the desired result. Detailed proofs omitted due to space restriction are available at: http://arxiv.org/abs/1202.4175.

2 Definitions

Markov decision processes (MDPs). A *Markov decision process (MDP)* $G = ((V, E), (V_1, V_P), \delta)$ consists of a directed graph (V, E), a partition (V_1, V_P) of the *finite* set V of vertices, and a probabilistic transition function δ: $V_P \to \mathcal{D}(V)$, where $\mathcal{D}(V)$ denotes the set of probability distributions over the vertex set V. The vertices in V_1 are the *player*-1 vertices, where player 1 decides the successor vertex, and the vertices in V_P are the *probabilistic (or random)* vertices, where the successor vertex is chosen according to the probabilistic transition function δ. We assume that for $u \in V_P$ and $v \in V$, we have $(u, v) \in E$ iff $\delta(u)(v) > 0$, and we often write $\delta(u, v)$ for $\delta(u)(v)$. For a vertex $v \in V$, we write $E(v)$ to denote the set $\{ u \in V \mid (v, u) \in E \}$ of possible out-neighbours, and $|E(v)|$ is the out-degree of v. For technical convenience we assume that every vertex in the graph (V, E) has at least one outgoing edge, i.e., $E(v) \neq \emptyset$ for all $v \in V$.

Plays, strategies and probability measure. An infinite path, or a *play*, of the game graph G is an infinite sequence $\omega = \langle v_0, v_1, v_2, \ldots \rangle$ of vertices such that $(v_k, v_{k+1}) \in E$ for all $k \in \mathbb{N}$. We write Ω for the set of all plays, and for a vertex $v \in V$, we write $\Omega_v \subseteq \Omega$ for the set of plays that start from the vertex v. A *strategy* for player 1 is a function σ: $V^* \cdot V_1 \to \mathcal{D}(V)$ that chooses the probability distribution over the successor vertices for all finite sequences $\vec{w} \in V^* \cdot V_1$ of vertices ending in a player-1 vertex (the sequence represents a prefix of a play). A strategy must respect the edge relation: for all $\vec{w} \in V^*$ and $u \in V_1$, if $\sigma(\vec{w} \cdot u)(v) > 0$, then $v \in E(u)$. Once a starting vertex $v \in V$ and a strategy $\sigma \in \Sigma$ is fixed, the outcome of the MDP is a random walk ω_v^σ for which the probabilities of events are uniquely defined, where an *event* $\mathcal{A} \subseteq \Omega$ is a measurable set of plays. For a vertex $v \in V$ and an event $\mathcal{A} \subseteq \Omega$, we write $\mathbb{P}_v^\sigma(\mathcal{A})$ for the probability that a play belongs to $\mathcal{A}$ if the game starts from the vertex v and player 1 follows the strategy σ.

Objectives. We specify *objectives* for the player 1 by providing a set of *winning* plays $\Phi \subseteq \Omega$. We say that a play ω *satisfies* the objective Φ if $\omega \in \Phi$. We consider *ω-regular objectives* [20], specified as parity conditions. We also consider the special case of Büchi objectives.

- *Büchi objectives.* Let B be a set of Büchi vertices. For a play $\omega = \langle v_0, v_1, \ldots \rangle \in \Omega$, we define $\mathrm{Inf}(\omega) = \{ v \in V \mid v_k = v$ for infinitely many $k \}$ to be the set of vertices that occur infinitely often in ω. The Büchi objectives require that some vertex of B be visited infinitely often, and defines the set of winning plays $\text{Büchi}(B) = \{\omega \in \Omega \mid \mathrm{Inf}(\omega) \cap B \neq \emptyset\}$.
- *Parity objectives.* For $c, d \in \mathbb{N}$, we write $[c..d] = \{ c, c+1, \ldots, d \}$. Let p: $V \to [0..d]$ be a function that assigns a *priority* $p(v)$ to every vertex $v \in V$, where $d \in \mathbb{N}$. The *parity objective* is defined as $\mathrm{Parity}(p) = \{ \omega \in \Omega \mid \min\big(p(\mathrm{Inf}(\omega))\big)$ is even $\}$. In other words,

the parity objective requires that the minimum priority visited infinitely often is even. In the sequel we will use Φ to denote parity objectives.

Qualitative analysis: almost-sure winning. Given a player-1 objective Φ, a strategy $\sigma \in \Sigma$ is *almost-sure winning* for player 1 from the vertex v if $\mathbb{P}_v^\sigma(\Phi) = 1$. The *almost-sure winning set* $\langle\langle 1 \rangle\rangle_{almost}(\Phi)$ for player 1 is the set of vertices from which player 1 has an almost-sure winning strategy. The qualitative analysis of MDPs correspond to the computation of the almost-sure winning set for a given objective Φ.

Algorithm for qualitative analysis. The almost-sure winning set for MDPs with parity objectives can be computed using $O(d)$ calls to compute the almost-sure winning set of MDPs with Büchi objectives [6, 7, 8]. Hence we focus on the qualitative analysis of MDPs with Büchi objectives. The algorithms for qualitative analysis for MDPs do not depend on the transition function, but only on the graph $G = ((V, E), (V_1, V_P))$. We now describe the classical algorithm for the qualitative analysis of MDPs with Büchi objectives and the algorithm requires the notion of random attractors.

Random attractor. Given an MDP G, let $U \subseteq V$ be a subset of vertices. The *random attractor* $Attr_P(U)$ is defined inductively as follows: $X_0 = U$, and for $i \geq 0$, let $X_{i+1} = X_i \cup \{ v \in V_P \mid E(v) \cap X_i \neq \emptyset \} \cup \{ v \in V_1 \mid E(v) \subseteq X_i \}$. In other words, X_{i+1} consists of (a) vertices in X_i, (b) probabilistic vertices that have at least one edge to X_i, and (c) player-1 vertices whose all successors are in X_i. Then $Attr_P(U) = \bigcup_{i \geq 0} X_i$. Observe that the random attractor is equivalent to the alternating reachability problem (reachability in AND-OR graphs).

Classical algorithm. The classical algorithm for MDPs with Büchi objectives is a simple iterative algorithm, and every iteration uses graph reachability and alternating graph reachability (random attractors). Let us denote the MDP in iteration i by G^i with vertex set V^i. Then in iteration i the algorithm executes the following steps: (i) computes the set Z^i of vertices that can reach the set of Büchi vertices $B \cap V^i$ in G^i; (ii) let $U^i = V^i \setminus Z^i$ be the set of remaining vertices; if U^i is empty, then the algorithm stops and outputs Z^i as the set of almost-sure winning vertices, and otherwise removes $Attr_P(U^i)$ from the graph, and continues to iteration $i+1$. The classical algorithm requires at most $O(n)$ iterations, where $n = |V|$, and each iteration requires at most $O(m)$ time, where $m = |E|$. Moreover the above analysis is tight, i.e., there exists a family of MDPs where the classical algorithm requires $\Omega(n)$ iterations, and total time $\Omega(n \cdot m)$. Hence $\Theta(n \cdot m)$ is the tight worst case complexity of the classical algorithm for MDPs with Büchi objectives. In this work we consider the average case analysis of the classical algorithm.

3 Average Case Analysis for MDPs with Constant Out-degree

In this section we consider the average case analysis of the number of iterations and the running time of the classical algorithm for computing the almost-sure winning set for MDPs with Büchi objectives on family of graphs with constant out-degree (out-degree of every vertex fixed and bounded by two constants $d_{\min}$ and $d_{\max}$).

Family of graphs and results. We consider families of graphs where the vertex set V ($|V| = n$), the target set of Büchi vertices B ($|B| = t$), and the out-degree d_v of each vertex v is fixed across the whole family. The only varying component is the edges of the graph; for each vertex v, every set of vertices of size d_v is equally likely to be the neighbour set of v, independent of neighbours of other vertices. Finally, there exist constants $d_{\min}$ and $d_{\max}$ such that $d_{\min} \leq d_v \leq d_{\max}$ for all vertices v. We will show the following for this family of graphs: (a) if the target set B has size more than $30 \cdot x \cdot \log(n)$, where x is the number of distinct degrees, (i.e., $t \geq 30 \cdot x \cdot \log(n)$), then the expected number of iterations is $O(1)$ and the average running time is $O(n)$; and (b) if the target vertex set B has size at most $30 \cdot x \cdot \log(n)$, then the expected number of iterations required is at most $O(\log(n))$ and average running time is $O(n)$.

Notations. We use n and t for the total number of vertices and the size of the target set, respectively. We will denote by x the number of distinct out-degree d_v's, and let d_i, for $1 \leq i \leq x$ be the distinct out-degrees. Since for all vertices v we have $d_{\min} \leq d_v \leq d_{\max}$, it follows that we have $x \leq d_{\max} - d_{\min} + 1$. Let a_i be the number of vertices with degree d_i and t_i be the number of target (Büchi) vertices with degree d_i.

The event $R(k_1, k_2, ..., k_x)$. The *reverse reachable set* of the target set B is the set of vertices u such that there is a path in the graph from u to a vertex $v \in B$. Let S be any set comprising of k_i vertices of degree d_i, for $1 \leq i \leq x$. We define $R(k_1, k_2, ..., k_x)$ as the probability of the event that all vertices of S can reach B via a path that lies entirely in S. Due to symmetry between vertices, this probability only depends on k_i, for $1 \leq i \leq x$ and is independent of S itself. For ease of notation, we will sometimes denote the event itself by $R(k_1, k_2, ..., k_x)$. We will investigate the reverse reachable set of B, which contains B itself. Recall that t_i vertices in B have degree d_i, and hence we are interested in the case when $k_i \geq t_i$ for all $1 \leq i \leq x$.

Consider a set S of vertices that is the reverse reachable set, and let S be composed of k_i vertices of degree d_i and of size k, i.e., $k = |S| = \sum_{i=1}^{x} k_i$. Since S is the reverse reachable set, it follows that for all vertices v in $V \setminus S$, there is no edge from v to a vertex in S (otherwise there would be a path from v to a target vertex and then v would belong to S). Thus there are no incoming edges from $V \setminus S$ to S. Thus for each vertex v of $V \setminus S$, all its neighbours must lie in $V \setminus S$ itself. This happens with probability $\prod_{i \in [1,x], a_i \neq k_i} \left(\frac{\binom{n-k}{d_i}}{\binom{n}{d_i}} \right)^{a_i - k_i}$, since in $V \setminus S$ there are $a_i - k_i$ vertices with degree d_i and the size of $V \setminus S$ is $n - k$ (recall that $[1, x] = \{1, 2, \ldots, x\}$). Note that when $a_i \neq k_i$, there is at least one vertex of degree d_i in $V \setminus S$ that has all its neighbours in $V \setminus S$ and hence $n - k \geq d_i$. For simplicity of notation, we skip mentioning $a_i \neq k_i$ and substitute the term by 1 where $a_i = k_i$. The probability that each vertex in S can reach a target vertex is $R(k_1, k_2, ..., k_x)$. Hence the probability of S being the reverse reachable set is given by: $\prod_{i=1}^{x} \left(\frac{\binom{n-k}{d_i}}{\binom{n}{d_i}} \right)^{a_i - k_i} \cdot R(k_1, k_2, ..., k_x)$. There are $\prod_{i=1}^{x} \binom{a_i - t_i}{k_i - t_i}$ possible ways of choosing $k_i \geq t_i$ vertices (since the target set is contained) out of a_i. Notice that the terms are 1 where $a_i = k_i$. The value k can range from t to n and exactly one of these subsets of V will be the reverse reachable set. So the sum of probabilities of this happening is 1. Hence we have:

$$1 = \sum_{k=t}^{n} \sum_{\sum k_i = k, t_i \leq k_i \leq a_i} \left(\prod_{i=1}^{x} \binom{a_i - t_i}{k_i - t_i} \cdot \left(\frac{\binom{n-k}{d_i}}{\binom{n}{d_i}} \right)^{a_i - k_i} \right) \cdot R(k_1, k_2, ..., k_x) \tag{1}$$

Let

$$\begin{aligned} a_{k_1, k_2, \ldots, k_x} &= \left(\prod_{i=1}^{x} \binom{a_i - t_i}{k_i - t_i} \cdot \left(\frac{\binom{n-k}{d_i}}{\binom{n}{d_i}} \right)^{a_i - k_i} \right) \cdot R(k_1, k_2, ..., k_x); \\ \alpha_k &= \sum_{\sum k_i = k, t_i \leq k_i \leq a_i} a_{k_1, k_2, \ldots, k_x}. \end{aligned}$$

Our goal is to show that for $30 \cdot x \cdot \log(n) \leq k \leq n - 1$, the value of α_k is very small; i.e., we want to get an upper bound on α_k. Note that two important terms in α_k are

$\left(\binom{n-k}{d_i}/\binom{n}{d_i}\right)^{a_i-k_i}$ and $R(k_1, k_2, \ldots, k_x)$. Below we get an upper bound for both of them. Firstly note that when k is small, for any set S comprising of k_i vertices of degree d_i for $1 \le i \le x$ and $|S| = k$, the event $R(k_1, k_2, \ldots, k_x)$ requires each non-target vertex of S to have an edge inside S. Since k is small and all vertices have constant out-degree spread randomly over the entire graph, this is highly improbable. We formalize this intuitive argument in the following lemma.

▶ **Lemma 1** (Upper bound on $R(k_1, k_2, \ldots, k_x)$)**.** *For $k \le n - d_{\max}$*

$$R(k_1, k_2, \ldots, k_x) \le \prod_{i=1}^{x} \left(1 - \left(1 - \frac{k}{n-d_i}\right)^{d_i}\right)^{k_i - t_i} \le \prod_{i=1}^{x} \left(\frac{d_i \cdot k}{n - d_{\max}}\right)^{k_i - t_i}.$$

Now for $\left(\binom{n-k}{d_i}/\binom{n}{d_i}\right)^{a_i-k_i}$, we give an upper bound. First notice that when $a_i \neq k_i$, there is at least one vertex of degree d_i outside the reverse reachable set and it has all its edges outside the reverse reachable set. Hence, the size of the reverse reachable set (i.e. $n-k$) is at least d_i. Thus, $\binom{n-k}{d_i}$ is well defined.

▶ **Lemma 2.** *For any $1 \le i \le x$ such that $a_i \neq k_i$, we have* $\left(\frac{\binom{n-k}{d_i}}{\binom{n}{d_i}}\right)^{a_i-k_i} \le \left(1 - \frac{k}{n}\right)^{d_i \cdot (a_i - k_i)}$.

Next we simplify the expression of α_k by taking care of the summation.

▶ **Lemma 3.** *The probability that the reverse reachable set is of size exactly k is α_k, and $\alpha_k \le n^x \cdot \max_{\sum k_i = k, t_i \le k_i \le a_i} a_{k_1,k_2,\ldots,k_x}$.*

Now we proceed to achieve an upper bound on $a_{k_1,k_2,\ldots,k_x}$. First of all, intuitively if k is small, then $R(k_1, k_2, \ldots, k_x)$ is very small (this can be derived easily from Lemma 1). On the other hand, consider the case when k is very large. In this case there are very few vertices that cannot reach the target set. Hence they must have all their edges within them, which again has very low probability. Note that different factors that bind α_k depend on whether k is small or large. This suggests we should consider these cases separately. Our proof will consist of the following case analysis of the size k of the reverse reachable set: (1) when $30 \cdot x \cdot \log(n) \le k \le c_1 \cdot n$ is *small* (for some constant $c_1 > 0$); (2) when $c_1 \cdot n \le k \le c_2 \cdot n$ is *large* (for all constants $c_2 \ge c_1 > 0$); and (3) when $c_2 \cdot n \le k \le n - d_{\min} - 1$ is *very large*. The analysis of the constants will follow from the proofs. Note that since the target set B (with $|B| = t$) is a subset of its reverse reachable set, we have $k < t$ is infeasible. Hence in all the three cases, we will only consider $k \ge t$. We first consider the case when k is small.

3.1 Small k: $30 \cdot x \cdot \log(n) \le k \le c_1 n$

In this section we will consider the case when $30 \cdot x \cdot \log(n) \le k \le c_1 \cdot n$ for some constant $c_1 > 0$. Note that this case only occurs when $t \le c_1 \cdot n$ (since $k \ge t$). We will assume this throughout this section. We will prove that there exists a constant $c_1 > 0$ such that for all $30 \cdot x \cdot \log(n) \le k \le c_1 \cdot n$ the probability (α_k) that the size of the reverse reachable set is k is bounded by $\frac{1}{n^2}$. Note that we already have a bound on α_k in terms of $a_{k_1,k_2,\ldots,k_x}$ (Lemma 3). We use continuous upper bounds of the discrete functions in $a_{k_1,k_2,\ldots,k_x}$ to convert it into a form that is easy to analyze. Let

$$b_{k_1,k_2,\ldots,k_x} = \prod_{i=1}^{x} \left(\frac{e \cdot (a_i - t_i)}{k_i - t_i}\right)^{k_i - t_i} \cdot e^{-\frac{k}{n} \cdot d_i \cdot (a_i - k_i)} \cdot \left(\frac{d_i \cdot k}{n - d_{\max}}\right)^{k_i - t_i}.$$

▶ **Lemma 4.** *We have $a_{k_1,k_2,\ldots,k_x} \leq b_{k_1,k_2,\ldots,k_x}$.*

Next we show that $b_{k_1,k_2,\ldots,k_x}$ drops exponentially as a function of k. This is the key and non-trivial result of this subsection and requires many involved mathematical inequalities. Note that this is the reason for the logarithmic lower bound on k in this section.

▶ **Lemma 5** (Upper bound on $b_{k_1,k_2,\ldots,k_x}$). *There exists a constant $c_1 > 0$ such that for sufficiently large n and $t \leq k \leq c_1 \cdot n$, we have $b_{k_1,k_2,\ldots,k_x} \leq \left(\frac{9}{10}\right)^k$.*

Taking appropriate bounds on the value of k, we get an upper bound on $a_{k_1,k_2,\ldots,k_x}$. Recall that x is the number of distinct degrees and hence $x \leq d_{\max} - d_{\min} + 1$.

▶ **Lemma 6** (Upper bound on $a_{k_1,k_2,\ldots,k_x}$). *There exists a constant $c_1 > 0$ such that for sufficiently large n with $t \leq c_1 \cdot n$ and for all $30 \cdot x \cdot \log(n) \leq k \leq c_1 \cdot n$, we have $a_{k_1,k_2,\ldots,k_x} < \frac{1}{n^{3\cdot x}}$.*

▶ **Lemma 7** (Main lemma for small k). *There exists a constant $c_1 > 0$ such that for sufficiently large n with $t \leq c_1 \cdot n$ and for all $30 \cdot x \cdot \log(n) \leq k \leq c_1 \cdot n$, the probability that the size of the reverse reachable set S is k is at most $\frac{1}{n^2}$.*

Proof. The probability that the reverse reachable set is of size k is given by α_k. By Lemma 3 and Lemma 6 we have α_k is at most $n^x \cdot n^{-3\cdot x} = n^{-2\cdot x} \leq \frac{1}{n^2}$. ∎

3.2 Large k: $c_1 \cdot n \leq k \leq c_2 \cdot n$

In this section we will show that for all constants c_1 and c_2, with $0 < c_1 \leq c_2$, when $t \leq c_2 \cdot n$ the probability α_k is at most $\frac{1}{n^2}$ for all $c_1 \cdot n \leq k \leq c_2 \cdot n$. We start with a few notations. Let $a_i = p_i \cdot n, t_i = y_i \cdot n, k_i = s_i \cdot n$ for $1 \leq i \leq x$ and $k = s \cdot n$ for $c_1 \leq s < c_2$. We first present a bound on $a_{k_1,k_2,\ldots,k_x}$ in Lemma 8. In the following two lemmas we obtain an upper bound for the bound in Lemma 8. All the lemmas require to prove non-trivial mathematical inequalities to achieve the result.

▶ **Lemma 8.** *For all constants c_1 and c_2 with $0 < c_1 \leq c_2$ and for all $c_1 \cdot n \leq k \leq c_2 \cdot n$, we have $a_{k_1,k_2,\ldots,k_x} \leq (n+1)^x \cdot \mathsf{Term}_1 \cdot \mathsf{Term}_2$, where*

$$\mathsf{Term}_1 = \left(\prod_{i=1}^{x} \left(\frac{p_i - y_i}{s_i - y_i}\right)^{s_i - y_i} \left(\frac{p_i - y_i}{p_i - s_i}\right)^{p_i - s_i} (1-s)^{d_i(p_i - s_i)} (1-(1-s)^{d_i})^{s_i - y_i} \right)^n, \quad \text{and}$$

$$\mathsf{Term}_2 = \prod_{i=1}^{x} \left(\frac{1 - \left(1 - \frac{s}{1-d_i/n}\right)^{d_i}}{1-(1-s)^{d_i}} \right)^{n(s_i - y_i)}.$$

On simplification, the base of the exponent in Term_2 can be shown to be upper bounded by $1 + c^*/n$ for some constant $c^* > 0$. Since $s_i - y_i \leq 1$ and x is a constant, we have the following.

▶ **Lemma 9.** Term_2 *of Lemma 8 is upper bounded by a constant.*

For Term_1, we maximize the base of the exponent with respect to every d_i. When all d_i's take their optimal values, the value of the base becomes 1. But using the fact that $d_i \geq 2$ for all i, we show that not all the d_i's can take their optimal values simultaneously and we prove the following.

▶ **Lemma 10.** *There exists a constant $0 < \eta < 1$ such that* Term_1 *of Lemma 8 is at most* η^n.

▶ **Lemma 11** (Main lemma for large k)**.** *For all constants c_1 and c_2 with $0 < c_1 \le c_2$, when n is sufficiently large and $t \le c_2 \cdot n$, for all $c_1 \cdot n \le k \le c_2 \cdot n$, the probability that the size of the reverse reachable set S is k is at most $\frac{1}{n^2}$.*

Proof. By Lemma 8 we have $a_{k_1,k_2,\ldots,k_x} \le (n+1)^x \cdot \mathsf{Term}_1 \cdot \mathsf{Term}_2$, and by Lemma 9 and Lemma 10 we have Term_2 is constant and Term_1 is exponentially small in n, where $x \le (d_{\max} - d_{\min} + 1)$. The exponentially small Term_1 overrides the polynomial factor $(n+1)^x$ and the constant Term_2, and ensures that $a_{k_1,k_2,\ldots,k_x} \le n^{-3x}$. By Lemma 3 it follows that $\alpha_k \le n^{-2x} \le \frac{1}{n^2}$. ∎

3.3 Very large k: $(1-1/e^2)n$ to $n - d_{\min} - 1$

In this subsection we consider the case when the size k of the reverse reachable set is between $(1-\frac{1}{e^2}) \cdot n$ and $n - d_{\min} - 1$. Note that if the reverse reachable set has size at least $n - d_{\min}$, then the reverse reachable set must be the set of all vertices, as otherwise the remaining vertices cannot have enough edges among themselves. Take $\ell = n - k$. Hence $d_{\min} + 1 \le \ell \le n/e^2$. As stated earlier, in this case $a_{k_1,k_2,\ldots,k_x}$ becomes small since we require that the ℓ vertices outside the reverse reachable set must have all their edges within themselves; this corresponds to the factor of $\left(\binom{n-k}{d_i}/\binom{n}{d_i}\right)^{a_i-k_i}$. Since ℓ is very small, this has a very low probability. With this intuition, we proceed to show the following bound on $a_{k_1,k_2,\ldots,k_x}$.

▶ **Lemma 12.** *We have $a_{k_1,k_2,\ldots,k_x} \le \left(x \cdot e \cdot \frac{\ell}{n}\right)^{\ell}$.*

We see that $\left(x \cdot e \cdot \frac{\ell}{n}\right)^{\ell}$ is a convex function in ℓ and its maximum is attained at one of the endpoints. For $\ell = n/e^2$, the bound is exponentially decreasing with n where as for constant ℓ, the bound is polynomially decreasing in n. Hence, the maximum is attained at left endpoint of the interval (constant value of ℓ). However, the bound we get is not sufficient to apply Lemma 3 directly. An important observation is that as ℓ becomes smaller and smaller, the number of combinations $\sum k_i = k$, where $t_i \le k_i \le a_i$ in the expression of α_k also decrease. Thus, we break this case into two sub-cases.

▶ **Lemma 13.** *For $d_{\max} + 1 < \ell \le n/e^2$, we have $a_{k_1,k_2,\ldots,k_x} < n^{-(2+x)}$ and $\alpha_k \le 1/n^2$.*

▶ **Lemma 14.** *There exists a constant $h > 0$ such that for $d_{\min} + 1 \le \ell \le d_{\max} + 1$, we have $a_{k_1,k_2,\ldots,k_x} < h \cdot n^{-\ell}$ and $\alpha_k \le \frac{h}{n^2}$.*

▶ **Lemma 15** (Main lemma for very large k)**.** *For all t, for all $(1-\frac{1}{e^2}) \cdot n \le k \le n-1$, the probability that the size of the reverse reachable set S is k is at most $O(\frac{1}{n^2})$.*

Proof. By Lemma 13 and Lemma 14 we obtain the result for all $(1-\frac{1}{e^2}) \cdot n \le k \le n - d_{\min} - 1$. Since the reverse reachable set must contain all vertices if it has size at least $n - d_{\min}$, the result follows. ∎

3.4 Expected Number of Iterations and Running Time

From Lemma 7, Lemma 11, and Lemma 15, we obtain that there exists a constant h such that (i) $\alpha_k \le \frac{1}{n^2}$, for $30 \cdot x \cdot \log(n) \le k < n - d_{\max} - 1$; (ii) $\alpha_k \le \frac{h}{n^2}$, for $n - d_{\max} - 1 \le k \le n - d_{\min} - 1$; and (iii) $\alpha_k = 0$, for $n - d_{\min} \le k \le n - 1$. Hence using the union bound we get the following result $\mathbb{P}(|S| < 30 \cdot x \cdot \log(n) \text{ or } |S| = n) \geq 1 - \frac{h}{n}$, where S is the reverse reachable set of target set (i.e., with probability at least $1 - \frac{h}{n}$ either at most $30 \cdot x \cdot \log(n)$ vertices reach the target set or all the vertices reach the target set). Let $I(n)$ and $T(n)$ denote the expected number of iterations and the expected running time of the classical algorithm for MDPs on random graphs with n vertices and constant out-degree. Then from above we have: $I(n) \leq \left(1 - \frac{h}{n}\right) \cdot 30 \cdot x \cdot \log(n) + \frac{h}{n} \cdot n$. It follows that $I(n) = O(\log(n))$. For the expected running time we have: $T(n) \leq \left(1 - \frac{h}{n}\right) \cdot (30 \cdot x \cdot \log(n))^2 + \frac{h}{n} \cdot n^2$. It follows that $T(n) = O(n)$. Hence we have the following theorem.

▶ **Theorem 16.** *The expected number of iterations and the expected running time of the classical algorithm for MDPs with Büchi objectives over graphs with constant out-degree are at most $O(\log(n))$ and $O(n)$, respectively.*

▶ Remark. For Theorem 16, we considered the model where the out-degree of each vertex v is fixed as d_v and there exist constants $d_{\min}$ and $d_{\max}$ such that $d_{\min} \leq d_v \leq d_{\max}$ for every vertex v. We discuss the implication of Theorem 16 for related models. First, when the out-degrees of all vertices are same and constant (say d^*), Theorem 16 can be applied with the special case of $d_{\min} = d_{\max} = d^*$. A second possible alternative model is when the outdegree of every vertex is a distribution over the range $[d_{\min}, d_{\max}]$. Since we proved that the average case is linear for every possible value of the outdegree d_v in $[d_{\min}, d_{\max}]$ for every vertex v (i.e., for all possible combinations), it implies that the average case is also linear when the outdegree is a distribution over $[d_{\min}, d_{\max}]$.

4 Average Case Analysis in Erdös-Rényi Model

In this section we consider the classical Erdös-Rényi model of random graphs $\mathcal{G}_{n,p}$, with n vertices, where each edge is chosen to be in the graph independently with probability p [11] (we consider directed graphs and then $\mathcal{G}_{n,p}$ is also referred as $\mathcal{D}_{n,p}$ in literature). First, in Section 4.1 we consider the case when p is $\Omega\left(\frac{\log(n)}{n}\right)$, and then we consider the case when $p = \frac{1}{2}$ (that generates the uniform distribution over all graphs). We will show two results: (1) if $p \geq \frac{c \cdot \log(n)}{n}$, for any constant $c > 2$, then the expected number of iterations is constant and the expected running time is linear; and (2) if $p = \frac{1}{2}$ (with $p = \frac{1}{2}$ we consider all graphs to be equally likely), then the probability that the number of iterations is more than one falls exponentially in n (in other words, graphs where the running time is more than linear are exponentially rare).

4.1 $\mathcal{G}_{n,p}$ with $p = \Omega\left(\frac{\log(n)}{n}\right)$

In this subsection we will show that given $p \geq \frac{c \cdot \log(n)}{n}$, for any constant $c > 2$, the probability that not all vertices can reach the given target set is at most $O(1/n)$. Hence the expected number of iterations of the classical algorithm for MDPs with Büchi objectives is constant and hence the algorithm works in average time linear in the size of the graph. Observe that to show the result the worst possible case is when the size of the target set is 1, as otherwise the chance that all vertices reach the target set is higher. Thus from here onwards, we assume that the target set has exactly 1 vertex.

The probability $R(n, p)$. For a random graph in $\mathcal{G}_{n,p}$ and a given target vertex, we denote by $R(n, p)$ the probability that each vertex in the graph has a path along the directed edges to the target vertex. Our goal is to obtain a lower bound on $R(n, p)$.

The key recurrence. Consider a random graph G with n vertices, with a given target vertex, and edge probability p. For a set K of vertices with size k (i.e., $|K| = k$), which contains the target vertex, $R(k, p)$ is the probability that each vertex in the set K, has a path to the target vertex, that lies within the set K (i.e., the path only visits vertices in K). The probability $R(k, p)$ depends only on k and p, due to the symmetry among vertices.

Consider the subset S of all vertices in V, which have a path to the target vertex. In that case, for all vertices v in $V \setminus S$, there is no edge going from v to a vertex in S (otherwise there would have been a path from v to the target vertex). Thus there are no incoming edges from $V \setminus S$ to S. Let $|S| = i$. Then the $i \cdot (n - i)$ edges from $V \setminus S$ to S should be absent, and each edge is absent with probability $(1 - p)$. The probability that each vertex in S can reach the target is $R(i, p)$. So the probability of S being the reverse reachable set is given by:

$$(1 - p)^{i\cdot(n-i)} \cdot R(i, p). \tag{2}$$

There are $\binom{n-1}{i-1}$ possible subsets of i vertices that include the given target vertex, and i can range from 1 to n. Exactly one subset S of V will be the reverse reachable set. So the sum of probabilities of the events that S is reverse reachable set is 1. Hence we have: $1 = \sum_{i=1}^{n} \binom{n-1}{i-1} \cdot (1 - p)^{i\cdot(n-i)} \cdot R(i, p)$. Moving all but the last term (with $i = n$) to the other side, we get the following recurrence relation:

$$R(n, p) = 1 - \sum_{i=1}^{n-1} \binom{n-1}{i-1} \cdot (1 - p)^{i\cdot(n-i)} \cdot R(i, p). \tag{3}$$

Bound on p for lower bound on $R(n, p)$. We will prove a lower bound on p in terms of n such that the probability that not all n vertices can reach the target vertex is less than $O(1/n)$. In other words, we require $R(n, p) \geq 1 - O\left(\frac{1}{n}\right)$. Since $R(i, p)$ is a probability value, it is at most 1. Hence from Equation 3 it follows that it suffices to show that

$$\sum_{i=1}^{n-1} \binom{n-1}{i-1} \cdot (1 - p)^{i\cdot(n-i)} \cdot R(i, p) \leq \sum_{i=1}^{n-1} \binom{n-1}{i-1} \cdot (1 - p)^{i\cdot(n-i)} \leq O\left(\frac{1}{n}\right) \tag{4}$$

to show that $R(n, p) \geq 1 - O\left(\frac{1}{n}\right)$. We will prove a lower bound on p for achieving Equation 4. Let us denote by $t_i = \binom{n-1}{i-1} \cdot (1 - p)^{i\cdot(n-i)}$, for $1 \leq i \leq n - 1$. The following lemma establishes a relation of t_i and t_{n-i}.

▶ **Lemma 17.** *For $1 \leq i \leq n - 1$, we have $t_{n-i} = \frac{n-i}{i} \cdot t_i$.*

Define $g_i = t_i + t_{n-i}$, for $1 \leq i \leq \lfloor n/2 \rfloor$. From the previous lemma we have

$$g_i = t_{n-i} + t_i \quad = \quad \frac{n}{i} \cdot t_i = \frac{n}{i} \cdot \binom{n-1}{i-1} \cdot (1 - p)^{i\cdot(n-i)} = \binom{n}{i} \cdot (1 - p)^{i\cdot(n-i)}.$$

We observe that in the range of $[2, \lfloor \frac{n}{2} \rfloor]$, g_i attains its maximum value at one of the two endpoints. Then observing that $g_2 \leq t_1$ and $g_{\lfloor n/2 \rfloor} \leq t_1$, we conclude the following.

▶ **Lemma 18.** *For sufficiently large n, if $p \geq \frac{c\cdot\log(n)}{n}$ with $c > 2$, then $g_i \leq t_1$ for all $2 \leq i \leq \lfloor \frac{n}{2} \rfloor$.*

Now we simplify the expression of t_1 and prove the following using standard inequalities.

▶ **Lemma 19.** *For sufficiently large n, if $p \geq \frac{c\cdot\log(n)}{n}$ with $c > 2$, then $t_1 \leq \frac{1}{n^2}$.*

We are now ready to establish the main lemma that proves the upper bound on $R(n,p)$ and then the main result of the section.

▶ **Lemma 20.** *For sufficiently large n, for all $p \geq \frac{c \cdot \log(n)}{n}$ with $c > 2$, we have $R(n,p) \geq 1 - \frac{1.5}{n}$.*

▶ **Theorem 21.** *The expected number of iterations of the classical algorithm for MDPs with Büchi objectives for random graphs $\mathcal{G}_{n,p}$, with $p \geq \frac{c \cdot \log(n)}{n}$, where $c > 2$, is $O(1)$, and the average case running time is linear.*

Proof. By Lemma 20 it follows that $R(n,p) \geq 1 - \frac{1.5}{n}$, and if all vertices reach the target set, then the classical algorithm ends in one iteration. In the worst case the number of iterations of the classical algorithm is n. Hence the expected number of iterations is bounded by: $1 \cdot \left(1 - \frac{1.5}{n}\right) + n \cdot \frac{1.5}{n} = O(1)$. Since the expected number of iterations is $O(1)$ and every iteration takes linear time, it follows that the average case running time is linear. ◼

4.2 Average-case analysis over all graphs

In this section, we consider uniform distribution over all graphs, i.e., all possible different graphs are equally likely. This is equivalent to considering the Erdös-Rényi model such that each edge has probability $\frac{1}{2}$. Using $\frac{1}{2} \geq 3 \cdot \log(n)/n$ (for $n \geq 17$) and the results from Section 4.1, we already know that the average case running time for $\mathcal{G}_{n,1/2}$ is linear. In this section we show that in $\mathcal{G}_{n,\frac{1}{2}}$, the probability that not all vertices reach the target is in fact exponentially small in n. It will follow that MDPs where the classical algorithm takes more than constant iterations are exponentially rare. We consider the same recurrence $R(n,p)$ as in the previous subsection and consider t_k and g_k as defined before. The following theorem shows the desired result.

▶ **Theorem 22.** *In $\mathcal{G}_{n,\frac{1}{2}}$ with sufficiently large n the probability that the classical algorithm takes more than one iteration is less than $\left(\frac{3}{4}\right)^n$.*

Proof. We first observe that Equation 3 and Equation 4 holds for all probabilities. Next we observe that Lemma 18 holds for $p \geq \frac{c \cdot \log(n)}{n}$ with any constant $c > 2$, and hence also for $p = \frac{1}{2}$ for sufficiently large n. We have $\sum_{i=1}^{n-1} t_i \leq \frac{3 \cdot n}{2} \cdot t_1$. For $p = \frac{1}{2}$ we have $t_1 = \binom{n-1}{0} \cdot \left(1 - \frac{1}{2}\right)^{n-1} = \frac{1}{2^{n-1}}$. Hence we have $R(n,p) \geq 1 - \frac{3 \cdot n}{2 \cdot 2^{n-1}} > 1 - \frac{1.5^n}{2^n} = 1 - \left(\frac{3}{4}\right)^n$. The second inequality holds for sufficiently large n. It follows that the probability that the classical algorithm takes more than one iteration is less than $(\frac{3}{4})^n$. The desired result follows. ◼

5 Conclusion

In this work both for the general case and the important special case of MDPs with constant out-degree we establish that the average case running time of the classical algorithm is linear, as compared to the quadratic worst case complexity. Moreover, as for the improved algorithms it is known that they require at most linear time more than the classical algorithm, it also follows that the average case running time of all the improved algorithms is also linear. We considered models where all MDPs in the relevant class are equally likely. We are not aware of any work that characterizes more appropriate probability distributions over graphs to represent MDPs that arise in practice. Characterizing distributions over MDPs that arise in practice and studying the average case complexity under such distributions is beyond the scope of this work, and is a subject for future work.

References

1 C. Baier and J-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
2 A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *FSTTCS 95*, volume 1026 of *LNCS*, pages 499–513. Springer-Verlag, 1995.
3 K. Chatterjee and M. Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *SODA*. ACM-SIAM, 2011.
4 K. Chatterjee and M. Henzinger. An $O(n^2)$ algorithm for alternating Büchi games. In *SODA*. ACM-SIAM, 2012.
5 K. Chatterjee, M. Jurdziński, and T.A. Henzinger. Simple stochastic parity games. In *CSL'03*, volume 2803 of *LNCS*, pages 100–113. Springer, 2003.
6 K. Chatterjee, M. Jurdziński, and T.A. Henzinger. Quantitative stochastic parity games. In *SODA'04*, pages 121–130. SIAM, 2004.
7 C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.
8 L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997.
9 L. de Alfaro, M. Faella, R. Majumdar, and V. Raman. Code-aware resource management. In *EMSOFT 05*. ACM, 2005.
10 L. de Alfaro and P. Roy. Magnifying-lens abstraction for markov decision processes. In *CAV*, pages 325–338, 2007.
11 P. Erdös and A. Rényi. On the evolution of random graphs. *Math. Inst. of the Hungarian Acad. of Sciences*, pages 17–61, 1960.
12 J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
13 H. Howard. *Dynamic Programming and Markov Processes*. MIT Press, 1960.
14 M. Kwiatkowska, G. Norman, and D. Parker. Verifying randomized distributed algorithms with prism. In *Workshop on Advances in Verification (WAVE'00)*, 2000.
15 M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In *TOOLS' 02*, pages 200–204. LNCS 2324, Springer, 2002.
16 A. Pogosyants, R. Segala, and N. Lynch. Verification of the randomized consensus algorithm of Aspnes and Herlihy: a case study. *Distributed Computing*, 13(3):155–186, 2000.
17 M. L. Puterman. *Markov Decision Processes*. J. Wiley and Sons, 1994.
18 R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, 1995. Technical Report MIT/LCS/TR-676.
19 M.I.A. Stoelinga. Fun with FireWire: Experiments with verifying the IEEE1394 root contention protocol. In *Formal Aspects of Computing*, 2002.
20 W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.