Eager Markov Chains

Parosh Aziz Abdulla
Uppsala University, Sweden.
parosh@it.uu.se

Noomene Ben Henda
Uppsala University, Sweden.
Noomene.BenHenda@it.uu.se

Richard Mayr
NC State University, USA.
mayr@csc.ncsu.edu

Sven Sandberg
Uppsala University, Sweden.
svens@it.uu.se

Abstract

We consider infinite-state discrete Markov chains which are eager*: the probability of avoiding a defined set of final states for more than n steps decreases exponentially in n. We study the problem of computing the expected reward (or cost) of runs until reaching the final states, where rewards are assigned to individual runs by computable reward functions. We present a path exploration scheme, based on forward reachability analysis, to approximate the expected reward up-to an arbitrarily small error, and show that the scheme is guaranteed to terminate in the case of eager Markov chains. We show that eager Markov chains include those induced by Probabilistic Vector Addition Systems with States, Noisy Turing Machines, and Probabilistic Lossy Channel Systems.*

1 Introduction

A lot of research effort has been devoted to developing methods for specification and analysis of stochastic programs [26, 23, 14, 29]. The motivation is to capture the behaviors of systems with uncertainty, such as programs with unreliable channels, randomized algorithms, and fault-tolerant systems; and to be able to analyze quantitative aspects such as performance and dependability. The underlying semantics of such a program is usually defined as a *finite-state* Markov chain. Then, techniques based on extensions of finite-state model checking can be used to carry out verification [15, 6, 10, 25].

One limitation of such methods is the fact that many systems that arise in computer applications can only be faithfully modeled as Markov chains which have *infinite* state spaces. A number of recent works have therefore considered the challenge of extending model checking to systems which induce infinite-state Markov chains. Examples include *probabilistic pushdown automata* (recursive state machines) which are natural models for probabilistic sequential programs with recursive procedures [17, 18, 20, 19, 16, 21]; and *probabilistic lossy channel systems* which consist of finite-state processes communicating through unreliable and unbounded channels in which messages are lost with a certain probability [1, 4, 7, 8, 11, 24, 27].

In a recent paper [3], we considered a class of infinite-state Markov chains which are *confluent*: any computation from which the set F of *final* states is reachable, will almost certainly reach F. We presented generic algorithms for analyzing both qualitative properties (checking whether F is reached with probability one), and quantitative properties (approximating the probability by which F is reached from a given initial state).

Besides computing probabilities, one is often also interested in *expectations*, e.g., the expected reward (or cost) for runs until they reach F, and of variances and moments of random variables. In this paper, we address this problem by considering *eager Markov chains*, where the probability of reaching F in n or more steps decreases exponentially with n. In other words, computations that reach F are likely to do so in "few" steps. Thus, eagerness is a strengthening of confluence, and eager Markov chains are a natural subclass of confluent ones. We consider the *approximate expectation problem*: for a given reward function, approximate the reward gained before a state in F is reached. We present an exploration scheme, based on forward reachability analysis, to approximate the expected reward up-to an arbitrarily small error $\epsilon > 0$; and show that the scheme is guaranteed to terminate in the case of eager Markov chains.

All finite-state Markov chains are trivially eager, but also several classes of infinite-state systems induce eager Markov chains:

- Markov chains which are *boundedly coarse*: there is a K such that if F is reachable then F will be reached within K steps with a probability which is bounded from below. We give two examples of boundedly coarse Markov chains, namely those induced by *Probabilistic Vector Addition Systems with States (PVASS)* and *Noisy Turing Machines (NTM)*.

- Markov chains which contain a finite *eager attractor*. An *attractor* is a set of states which is reached with probability one from each state in the Markov chain. An attractor is *eager*, if the probability of returning to it in more than n steps decreases exponentially with n. Examples of such Markov chains are those induced by *probabilistic lossy channel systems (PLCS)*. This is shown in two steps. First, we consider systems that contain *GR-attractors*, defined as generalizations of the classical *gambler's ruin* problem, and show that each GR-attractor is eager. Then, we show that each PLCS induces a Markov chain which contains a GR-attractor.

Related work. There has been an extensive work on model checking of finite-state Markov chains [15, 9, 6, 10, 25].

Recently, several works have considered probabilistic pushdown automata and probabilistic recursive state machines [17, 18, 20, 19, 16, 21]. However, all the decidability results in these papers are based on translating the relevant properties into formulas in the first-order theory of reals. Using results from [3], it is straightforward to show that such a translation is impossible to achieve for the classes of Markov chains we consider in this paper.

The works in [1, 4, 8, 11, 27, 7] consider model checking of PLCS. In particular, [3] gives a generic theory for verification of infinite-state Markov chains including PLCS and PVASS. However, all these works concentrate on computing probabilities, and do not give algorithms for analysis of expectation properties.

The work closest to ours is a recent paper by Brázdil and Kučera [12] which considers the problem of computing approximations of the accumulated reward (and gain) for some classes of infinite-state Markov chains which satisfy certain preconditions (e.g., PLCS). However, their technique is quite different from ours and their preconditions are incomparable to our eagerness condition. The main idea in [12] is to approximate an infinite-state Markov chain by a sequence of effectively constructible finite-state Markov chains such that the obtained solutions for the finite-state Markov chains converge toward the solution for the original infinite-state Markov chain. Their preconditions [12] include one that ensures that this type of approximation converges, which is not satisfied by, e.g., PVASS. Furthermore, they require decidability of model checking for certain path formulas in the underlying transition system.

In contrast, our method is a converging path exploration scheme for infinite-state Markov chains, which only requires the eagerness condition. It is applicable not only to PLCS but also to other classes like PVASS and noisy Turing machines. We also do not assume that reachability is decidable in the underlying transition system. Finally, we solve a somewhat more general problem. We compute approximations for the *conditional* expected reward, consider possibly infinite sets of final states (rather than just a single final state) and our reward functions can be arbitrary (exponentially bounded) functions on runs (instead of cumulative state-based linear-bounded functions in [12]).

2 Preliminaries

Transition Systems. A *transition system* is a triple $\mathcal{T} = (S, \longrightarrow, F)$ where S is a countable set of *states*, $\longrightarrow \subseteq S \times S$ is the *transition relation*, and $F \subseteq S$ is the set of *final states*. We write $s \longrightarrow s'$ to denote that $(s, s') \in \longrightarrow$. We assume that transition systems are deadlock-free, i.e., each state has at least one successor. If this condition is not satisfied, we add a self-loop to states without successors – this does not affect the properties of transition systems considered in this paper.

A *run* ρ is an infinite sequence $s_0 s_1 \ldots$ of states satisfying $s_i \longrightarrow s_{i+1}$ for all $i \geq 0$. We use $\rho(i)$ to denote s_i and say that ρ is an s-run if $\rho(0) = s$. We assume familiarity with the syntax and semantics of the temporal logic CTL^* [13]. We use $(s \models \phi)$ to denote the set of s-runs that satisfy the CTL^* formula ϕ. For instance, $(s \models \bigcirc F)$ and $(s \models \Diamond F)$ are the sets of s-runs that visit F in the next state resp. eventually reach F. For a natural number n, $\bigcirc^{=n} F$ denotes a formula which is satisfied by a run ρ iff $\rho(n) \in F$. We use $\Diamond^{=n} F$ to denote a formula which is satisfied by ρ iff ρ reaches F first in its n^{th} step, i.e., $\rho(n) \in F$ and $\rho(i) \notin F$ when $0 \leq i < n$. Similarly, for $\sim \in \{<, \leq, \geq, >\}$, $\Diamond^{\sim n} F$ holds for a run ρ if there is an $m \in \mathbb{N}$ with $m \sim n$ such that $\Diamond^{=m} F$ holds.

A *path* π is a finite sequence $s_0, \ldots, s_n$ of states such that $s_i \longrightarrow s_{i+1}$ for all $i : 0 \leq i < n$. We let $|\pi| := n$ denote the number of transitions in a path. Note that a path is a prefix of a run. We use ρ^n for the path $\rho(0)\rho(1)\cdots\rho(n)$ and $Path_F^{=n}(s)$ for the set $\{\rho^n | \, \rho \in (s \models \Diamond^{=n} F)\}$. In other words, $Path_F^{=n}(s)$ is the set of paths of length n starting from s and reaching F first in the last state.

A transition system $\mathcal{T} = (S, \longrightarrow, F)$ is said to be *effective* if it is finitely branching and for each $s \in S$, we can explicitly compute all successors, and check if $s \in F$.

Reward Functions. A *reward function* (with respect to a state s) is a mapping $f : (s \models \Diamond F) \rightarrow \mathbb{R}$ which assigns a *reward* $f(\rho)$ to any s-run that visits F. A reward function is *tail-independent* if its value only depends on the prefix of the run up-to the first state in F, i.e., if

$\rho_1, \rho_2 \in (s \models \Diamond^{=n} F)$ and $\rho_1^n = \rho_2^n$ then $f(\rho_1) = f(\rho_2)$. In such a case (abusing notation), we write $f(\pi)$ to denote $f(\rho)$ where $\pi = \rho^n$. We say that f is *computable* if we can compute $f(\pi)$.

We will place an exponential limit on the growth of reward functions: A reward function is said to be *exponentially bounded* if there are $\alpha \in \mathbb{R}_{>0}$ and $N \in \mathbb{N}$ such that $|f(\rho)| \leq \alpha^n$ for each $n \geq N$ and $\rho \in (s \models \Diamond^{=n} F)$. We call (α, N) the *parameter* of f.

Markov Chains. A *Markov chain* is a tuple $\mathcal{M} = (S, P, F)$ where S is a countable set of *states*, $P : S \times S \rightarrow [0, 1]$ is the *probability distribution*, satisfying $\forall s \in S. \sum_{s' \in S} P(s, s') = 1$, and $F \subseteq S$ is the set of *final states*.

A Markov chain induces a transition system, where the transition relation consists of pairs of states related by a positive probability. Formally, the *underlying transition system* of $\mathcal{M}$ is $(S, \longrightarrow, F)$ where $s_1 \longrightarrow s_2$ iff $P(s_1, s_2) > 0$. In this manner, concepts defined for transition systems can be lifted to Markov chains. For instance, a run or a reward function in a Markov chain $\mathcal{M}$ is a run or reward function in the underlying transition system, and $\mathcal{M}$ is effective, etc, if the underlying transition system is so.

A Markov chain $\mathcal{M} = (S, P, F)$ and a state s induce a probability space on the set of runs that start at s. The probability space $(\Omega, \Delta, \mathcal{P}_\mathcal{M})$ is defined as follows: $\Omega = sS^\omega$ is the set of all infinite sequences of states starting from s and Δ is the σ-algebra generated by the basic cylindric sets $\{D_u = uS^\omega : u \in sS^*\}$. The probability measure $\mathcal{P}_\mathcal{M}$ is first defined on finite sequences of states $u = s_0 \ldots s_n \in sS^*$ by $\mathcal{P}_\mathcal{M}(u) = \prod_{i=0}^{n-1} P(s_i, s_{i+1})$ and then extended to cylindric sets by $\mathcal{P}_\mathcal{M}(D_u) = \mathcal{P}_\mathcal{M}(u)$; it is well-known that this measure is extended in a unique way to the entire σ-algebra. We use $\mathcal{P}_\mathcal{M}(s \models \phi)$ to denote the measure of the set $(s \models \phi)$.

Given a Markov chain $\mathcal{M} = (S, P, F)$, a state $s \in S$, and a reward function f on the underlying transition system, define the random variable $X_f : \Omega \rightarrow \mathbb{R}$ as follows: $X_f(\rho) = 0$ if $\rho \notin (s \models \Diamond F)$, and $X_f(\rho) = f(\rho)$ if $\rho \in (s \models \Diamond F)$. Then $E(X_f | s \models \Diamond F)$ is the conditional expectation of the reward from s to F, under the condition that F is reached.

Eager Markov Chains. A Markov chain $\mathcal{M}$ is said to be *eager with respect to* $s \in S$ if there are $\alpha < 1$ and $N \in \mathbb{N}$ such that $\mathcal{P}_\mathcal{M}(s \models \Diamond^{\geq n} F) \leq \alpha^n$ for each $n \geq N$. Intuitively, $\mathcal{M}$ is eager with respect to s if the probability of avoiding F in n or more steps (starting from the initial state s) decreases exponentially with n. We call (α, N) the *parameter* of $(\mathcal{M}, s)$.

3 The Expectation Problem

In this Section, we consider the *approximate conditional expectation problem* defined as follows:

APPROX_EXPECT
Instance

- An effective Markov chain $\mathcal{M} = (S, P, F)$, a state $s \in S$ such that $s \models \exists\Diamond F$, $\mathcal{M}$ is eager w.r.t. s, and $(\mathcal{M}, s)$ has parameter (α_1, N_1).
- An exponentially bounded and computable tail-independent reward function f with parameter (α_2, N_2) such that $\alpha_1 \cdot \alpha_2 < 1$.
- An error tolerance $\epsilon \in \mathbb{R}_{>0}$

Task Compute a number $r \in \mathbb{R}$ such that
$r \leq E(X_f | s \models \Diamond F) \leq r + \epsilon$.

Observe that the instance of the problem assumes that F is reachable from s. This is because the expected value is undefined otherwise. We observe that the condition $\alpha_1 \cdot \alpha_2 < 1$ can always be fulfilled if the reward function f is bounded by a polynomial, since $\alpha_2 > 1$ can then be chosen arbitrarily close to 1. Many natural reward functions are in fact polynomial. For instance, it is common to assign a reward $g(s)$ to each state and consider the reward of a run to be the sum of state rewards up to F: if $\rho \models \Diamond^{=n} F$ then $f(\rho) = \sum_{i=0}^{n} g(\rho(i))$. If there is a bound on the state reward, i.e., $\exists M \in \mathbb{R}. \forall \rho. \forall i. |g(\rho(i))| < M$, then such a reward function is linearly bounded in the length of the run. Another important case is state rewards that depend on the size of the state which can grow at most by a constant in every step, e.g., values of counters in a Petri net (or VASS) or the number of messages in an unbounded communication channel. In this case, the reward function is at most quadratic in the length of the run.

Remark. If $\alpha_1 \cdot \alpha_2^k < 1$, the k^{th} moment X_f^k can also be computed as it satisfies the conditions above. In particular, all moments can be approximated for polynomially bounded reward functions. Using the formula $V(X_f) = E(X_f^2) - E(X_f)^2$, we can also approximate the variance.

4 Algorithm

We present a path enumeration algorithm (Algorithm 1) for solving APPROX_EXPECT (defined in the previous Section), and then show that it terminates and computes the correct value of r.

In Algorithm 1, since $s \models \exists\Diamond F$ by assumption, we know that $\mathcal{P}_{\mathcal{M}}(s \models \Diamond F) > 0$, and therefore:

$$E(X_f | s \models \Diamond F) = \frac{E(X_f)}{\mathcal{P}_{\mathcal{M}}(s \models \Diamond F)} = \frac{E(X_f)}{E(X_R)},$$

where $R(\rho) = 1$ if $\rho\,(s \models \Diamond F)$, and $R(\rho) = 0$ otherwise. The algorithm tries to approximate the values of $E(X_f)$ and $E(X_R)$ based on the observation that $E(X_f) = \sum_{i=0}^{\infty} \sum_{\pi \in Path_F^{=i}(s)} \mathcal{P}_{\mathcal{M}}(\pi) \cdot f(\pi)$ and $E(X_R) = \sum_{i=0}^{\infty} \sum_{\pi \in Path_F^{=i}(s)} \mathcal{P}_{\mathcal{M}}(\pi)$.

The algorithm maintains four variables: E_f and E_R which contain approximations of the values of $E(X_f)$ and $E(X_R)$; and ε_f and ε_R which are bounds on the errors in the current approximations. During the n^{th} iteration the values of E_f and E_R are modified by $\sum_{\pi \in Path_F^{=n}(s)} \mathcal{P}_{\mathcal{M}}(\pi) \cdot f(\pi)$ and $\sum_{\pi \in Path_F^{=n}(s)} \mathcal{P}_{\mathcal{M}}(\pi)$. This maintains the invariant that

$$E_f = \sum_{i=0}^{n} \sum_{\pi \in Path_F^{=n}(s)} \mathcal{P}_{\mathcal{M}}(\pi) \cdot f(\pi) \qquad (1)$$

$$E_R = \sum_{i=0}^{n} \sum_{\pi \in Path_F^{=n}(s)} \mathcal{P}_{\mathcal{M}}(\pi) \qquad (2)$$

each time we arrive at line 7. The algorithm terminates in case three conditions are satisfied:

- $n \geq \max(N_1, N_2)$. The error approximations depend on α_1 and α_2, so they are valid only if $n \geq N_1, N_2$.
- F is reached, i.e., $E_R > 0$.
- The difference between the upper and lower bounds $\frac{E_f + \varepsilon_f}{E_R}$ and $\frac{E_f - \varepsilon_f}{E_R + \varepsilon_R}$ on the conditional expectation (given in the proof of Theorem 4.1), is below the error tolerance ϵ.

Algorithm 1 – APPROX_EXPECT
Input
An instance of the problem as described in Section 3.
Variables
$E_f, E_R, \varepsilon_f, \varepsilon_R$: $\mathbb{R}$
begin
1. $n \leftarrow 0 \quad E_f \leftarrow 0 \quad E_R \leftarrow 0$
2. **repeat**
3. $\quad E_f \leftarrow E_f + \sum_{\pi \in Path_F^{=n}(s)} \mathcal{P}_{\mathcal{M}}(\pi) \cdot f(\pi)$
4. $\quad E_R \leftarrow E_R + \sum_{\pi \in Path_F^{=n}(s)} \mathcal{P}_{\mathcal{M}}(\pi)$
5. $\quad \varepsilon_f \leftarrow (\alpha_1 \cdot \alpha_2)^{n+1} / (1 - \alpha_1 \cdot \alpha_2)$
6. $\quad \varepsilon_R \leftarrow \alpha_2^{n+1} / (1 - \alpha_2)$
7. $\quad n \leftarrow n + 1$
8. **until** $(n \geq \max(N_1, N_2)) \ \wedge \ (E_R > 0) \ \wedge$ $\left(\frac{E_f + \varepsilon_f}{E_R} - \frac{E_f - \varepsilon_f}{E_R + \varepsilon_R} < \epsilon \right)$
9. **return** $\left(\frac{E_f - \varepsilon_f}{E_R + \varepsilon_R} \right)$

end

Observe that the parameters (α_1, N_1) and (α_2, N_2) are required by Algorithm 1, and hence they should be computable for the Markov chains to be analyzed by the algorithm. This is possible for the classes of Markov chains we consider in this paper.

Theorem 4.1 *Algorithm 1 terminates and returns a correct value of r.*

Proof. Clearly, each time the algorithm is about to execute line 7, the values of E_f and E_R are described by equations (1) and (2). The error in E_f as an approximation to $E(X_f)$ when $n \geq \max(N_1, N_2)$ is thus

$$|E(X_f) - E_f| = \left| \sum_{i=n+1}^{\infty} \sum_{\pi \in Path_F^{=i}(s)} \mathcal{P}_{\mathcal{M}}(\pi) \cdot f(\pi) \right|$$
$$\leq \left| \sum_{i=n+1}^{\infty} \alpha_1^i \sum_{\pi \in Path_F^{=i}(s)} \mathcal{P}_{\mathcal{M}}(\pi) \right|$$
$$\leq \left| \sum_{i=n+1}^{\infty} \alpha_1^i \cdot \alpha_2^i \right| = (\alpha_1 \cdot \alpha_2)^{n+1} / (1 - \alpha_1 \cdot \alpha_2) = \varepsilon_f$$

Here, the first equality follows by definition, and the inequalities follow from the fact that f is exponentially bounded and $\mathcal{M}$ is eager.

The inequality $|E(X_R) - E_R| \leq \varepsilon_R$ is obtained similarly. By assumption, $\alpha_1 \cdot \alpha_2 < 1$ and $\alpha_2 < 1$, and hence $\lim_{n \to \infty} \varepsilon_f = \lim_{n \to \infty} \varepsilon_R = 0$, which implies that the algorithm terminates.

Now, we show correctness of the algorithm. It is clear that $0 \leq E_R \leq E(X_R)$ since E_R increases each iteration. Hence, we have the two inequalities $E_f - \varepsilon_f \leq E(X_f) \leq E_f + \varepsilon_f$ and $E_R \leq E(X_R) \leq E_R + \varepsilon_R$. If $E_R > 0$, we can invert the second inequality and multiply it with the first to obtain

$$\frac{E_f - \varepsilon_f}{E_R + \varepsilon_R} \leq \frac{E(X_f)}{E(X_R)} \leq \frac{E_f + \varepsilon_f}{E_R}.$$

Hence, when the algorithm terminates, $\frac{E_f - \varepsilon_f}{E_R + \varepsilon_R}$ is a correct value of r. $\square$

Remarks. If reachability is decidable in the underlying transition system (as is the case in the classes of Markov chains we consider in this paper), we can explicitly check whether the condition $s \models \exists \Diamond F$ is satisfied before running the algorithm.

When computing the sums over $Path_F^{=i}(s)$ on lines 3 and 4, the algorithm can use either breadth-first search or depth-first search to find the paths in the transition system. Breadth-first search has the advantage that it computes $Path_F^{=i}(s)$ explicitly, which can be reused in the next iteration to compute $Path_F^{=i+1}(s)$. With depth-first search, on the other hand, the search has to be restarted from s in each iteration, but it only requires memory linear in n.

5 Bounded Coarseness

In this section, we consider the class of Markov chains that are *boundedly coarse*. We will first give definitions and a proof that boundedly coarse Markov chains are eager with respect to any state, and then give examples of models that are boundedly coarse.

Consider a Markov chain $\mathcal{M} = (S, P, F)$. A state s of $\mathcal{M}$ is *boundedly coarse* with parameter (β, K) if either $s \not\models \Diamond F$ or $P(s \models \Diamond^{\leq K} F) \geq \beta$.

A Markov chain is *boundedly coarse* with *parameter* (β, K) if all states are boundedly coarse with parameter (β, K) .

Lemma 5.1 *If a Markov Chain $\mathcal{M}$ is boundedly coarse then it is eager with respect to all states in $\mathcal{M}$.*

Proof. Given a Markov chain $\mathcal{M} = (S, P, F)$ that is boundedly coarse with parameter (β, K), we first show that for each $s \in S$ we have $\mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{>nK} F\right) \leq (1-\beta)^n$. We use induction on n. The base case (with $n = 0$) is trivial. We consider the induction step (when $n \geq 1$).

$$
\begin{aligned}
&\mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{>(n+1)K} F\right) \\
&= \sum_{s' \in S-F} \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc^{=nK} s' \wedge \Diamond^{>nK} F\right) \cdot \mathcal{P}_{\mathcal{M}}\left(s' \models \Diamond^{>K} F\right) \\
&\leq (1-\beta) \cdot \sum_{s' \in S-F} \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc^{=nK} s' \wedge \Diamond^{>nK} F\right) \\
&\leq (1-\beta) \cdot (1-\beta)^n = (1-\beta)^{(n+1)},
\end{aligned}
$$

where the first inequality follows from the definition of bounded coarseness and the second from the induction hypothesis. This concludes the induction proof. Now, given $n \geq 1$,

$$
\begin{aligned}
\mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{\geq n} F\right) &= \mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{>n-1} F\right) \\
&\leq \mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{>\lfloor \frac{n-1}{K} \rfloor \cdot K} F\right) \leq (1-\beta)^{\lfloor \frac{n-1}{K} \rfloor} \\
&\leq (1-\beta)^{-\frac{K+1}{K}} ((1-\beta)^{\frac{1}{K}})^n
\end{aligned}
$$

Consequently, $\mathcal{M}$ is eager with parameter (α, N) chosen such that $(1-\beta)^{1/K} < \alpha < 1$ and $(1-\beta)^{-\frac{K+1}{K}} \cdot ((1-\beta)^{\frac{1}{K}})^N \leq \alpha^N$, i.e., $N \geq \frac{(K+1)\log(1-\beta)}{\log(1-\beta) - K \log \alpha}$. $\square$

Sufficient Condition. We give a sufficient condition for bounded coarseness. A state s is said to be of *coarseness* β if, for each $s' \in S$, $P(s, s') > 0$ implies $P(s, s') \geq \beta$. We say that $\mathcal{M}$ is of *coarseness* β if each state is of coarseness β, and $\mathcal{M}$ is *coarse* if it is of coarseness β, for some $\beta > 0$. Notice that if $\mathcal{M}$ is coarse then the underlying transition system is finitely branching; however, the converse is not necessarily true.

A transition system is *of span* K if for each $s \in S$, either $s \not\models \exists\Diamond F$ or $s \models \exists\Diamond^{\leq K} F$, i.e., either F is unreachable or it is reachable in at most K steps. A transition system is finitely spanning if it is of span K for some K and a Markov chain is finitely spanning (of span K) if its underlying transition system is so. The following result is immediate.

Lemma 5.2 *If a Markov chain is coarse (of coarseness β), and finitely spanning (of span K), then it is boundedly coarse with parameter (β^K, K).*

Probabilistic VASS. A *Probabilistic Vector Addition System with States (PVASS)* (see [3] for details) is an extended finite-state automaton which operates on a finite set of variables ranging over the natural numbers. The variables behave as weak counters (weak in the sense that they are not compared for equality with 0). Furthermore, each transition has a *weight* defined by a natural number. A PVASS $\mathcal{V}$ induces an (infinite-state) Markov chain $\mathcal{M}$ in a natural way where the states of $\mathcal{M}$ are configurations (the local state of the automaton together with the counter values) of $\mathcal{V}$, and the probability of performing a transition from a given configuration is defined by the weight of the transition relative to the weights of other transitions enabled from the same configuration.

It was shown in [3] that each Markov chain induced by a PVASS where the set F is *upward closed* (with respect to the standard ordering on configurations) is effective, coarse, and finitely spanning (with the span being computable). This, together with Lemmas 5.2 and 5.1, yields the following theorem.

Theorem 5.3 APPROX_EXPECT *is solvable for* PVASS *with an upward closed set of final configurations.*

Noisy Turing Machines. Noisy Turing Machines (NTMs) were recently introduced by Asarin and Collins [5]. They study NTMs from a theoretical point of view, considering the computational power as the noise level tends to zero, but motivate them by practical applications such as computers operating in a hostile environment where arbitrary memory bits can change with some small probability. We show that NTMs with a fixed noise level are boundedly coarse, so by Lemma 5.1, they induce eager Markov chains.

An NTM is like an M-tape Turing Machine (with a finite control part and a given final control state), except that prior to a transition, for each cell on each tape, with probability λ it is *subjected to noise*. In this case, it is changed to one of the symbols in the alphabet (possibly the same as before) uniformly at random.

An NTM induces a Markov chain $\mathcal{M} = (S, P, F)$ as follows. A state in S is a triple: the current time, the current control state, and an M-tuple of *tape configurations*. A tape configuration is represented as a triple: the head position; a finite word w over the alphabet representing the contents of all cells visited by the head so far; and a $|w|$-tuple of natural numbers, each representing the last point in time when the head visited the corresponding cell.

These last-visit times allow us to add noise "lazily": cells not under the head are not modified. Since it is known when the head last visited each cell, we compensate for the missing noise by a higher noise probability for the cell under the head. If the cell was last visited k time units ago, we increase the probability of noise to $1-(1-\lambda)^k$, which is the probability that the cell is subject to noise in any of k steps. Then the last-visit time for the cell under the head is updated to contain the current time, and the next configuration is selected according to the behavior of the control part. The final states F are those where the control state is final.

Lemma 5.4 *The Markov chain induced by a Noisy Turing Machine is coarse and finitely spanning.*

Proof Sketch. For any state $s \in S$, if $s \models \exists\Diamond F$, there must be some path in the control part that goes from the control state of s to the final control state. Hence there must be such a path of length bounded by the number N of control states. It is possible that the symbol under the head will be subject to noise for the next N steps in such a way that this path is taken. Thus, the Markov chain has span N. Since only M cells are subject to noise and each happens with probability $\geq \lambda$, each successor has probability $\geq (\lambda/K)^M$, where K is the size of the alphabet. Hence, the Markov chain has coarseness $(\lambda/K)^M$. $\square$

By Lemmas 5.2, 5.1, and 5.4 NTMs are eager, and we have:

Theorem 5.5 APPROX_EXPECT *is solvable for NTMs.*

Remark. A somewhat simpler way to generate a Markov chain from an NTM avoids the need for a counter per tape cell. Instead, all cells ever visited by a head are subject to noise in each step. When a cell is visited for the first time, say after k steps, the probability of noise is increased to $1-(1-\lambda)^k$. This is an example of a Markov chain that is boundedly coarse but not coarse (the probability of a successor obtained by changing n tape cells is λ^n).

6 Eager Attractors

We consider Markov chains that contain a *finite attractor*, and prove that certain weak conditions on the attractor imply eagerness of the Markov chain.

Consider a Markov chain $\mathcal{M} = (S, P, F)$. A set $A \subseteq S$ is said to be an *attractor* if $\mathcal{P}_\mathcal{M}(s \models \Diamond A) = 1$ for each $s \in S$. In other words, a run from any state will almost certainly return back to A. We will only work with attractors that are *finite*; therefore we assume finiteness (even when not explicitly mentioned) for all the attractors in the sequel.

Eager Attractors. We say that an attractor $A \subseteq S$ is *eager* if there is a $\beta < 1$ such that for each $s \in A$ and $n \geq 0$ it is the case that $\mathcal{P}_\mathcal{M}\left(s \models \bigcirc\left(\Diamond^{\geq n} A\right)\right) \leq \beta^n$. In other words, for every state $s \in A$, the probability of first returning to A in $n+1$ (or more) steps is exponentially bounded in n. We call β the *parameter* of A. Notice that it is not a restriction to have β independent of s, since A is finite.

Theorem 6.1 *Let $\mathcal{M} = (S, P, F)$ be a Markov chain that contains an eager attractor $A \subseteq S$. Then $\mathcal{M}$ is eager with respect to any $s \in A$. Furthermore, we can compute the parameters (α, N) of $\mathcal{M}$.*

We devote the rest of this section to the proof of Theorem 6.1. Fix a state $s \in A$, let $n \geq 1$, and define

$$U_s(n) := \mathcal{P}_\mathcal{M}\left(s \models \Diamond^{=n} F\right)$$

We will compute an upper bound on $U_s(n)$, where the upper bound decreases exponentially with n. To do that, we partition the set of runs in $(s \models \Diamond^{=n} F)$ into two subsets R_1 and R_2, and show that both have "low" probability measures:

- R_1: the set of runs that visit A "seldom" in the first n steps. Such runs are not probable since A is eager. In our proof, we use the eagerness of A to compute an upper bound $U_s^1(n)$ on the measure of R_1, where $U_s^1(n)$ decreases exponentially with n.

- R_2: the set of runs that visit A "often" in the first n steps. Each time a run enters a state in A, it will visit F with a probability, which is bounded from below, before it returns back to A. The runs of R_2 are not probable, since the probability of avoiding F between the "many" re-visits of A is low. We use this observation to compute an upper bound $U_s^2(n)$ on the measure of R_2, that also decreases exponentially with n.

A crucial aspect here is to define the border between R_1 and R_2. We consider a run to re-visit A often (i.e., belong to the set R_2) if the number of re-visits is at least n/c, where c is a constant defined later that only depends on β.

To formalize the above reasoning, we need the following definition. For natural numbers $n, t : 1 \leq t \leq n$, we define the formula $A^{\#}_{n,t}$, which is satisfied by an s-run ρ iff ρ^n contains exactly t occurrences of elements in A before the last state in ρ^n, i.e., the very last state $\rho(n)$ does not count toward t even if it is in A. Then:

$$\begin{aligned} U_s(n) =& \mathcal{P}_\mathcal{M}\left(s \models \Diamond^{=n} F\right) \\ &= \sum_{t=1}^{n} \mathcal{P}_\mathcal{M}\left(s \models \Diamond^{=n} F \wedge A^{\#}_{n,t}\right) = U_s^1(n) + U_s^2(n), \end{aligned}$$

where

$$U^1_s(n) := \sum_{t=1}^{\lfloor \frac{n}{c} \rfloor} \mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{=n} F \wedge A^{\#}_{n,t}\right),$$

$$U^2_s(n) := \sum_{t=\lfloor \frac{n}{c} \rfloor + 1}^{n} \mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{=n} F \wedge A^{\#}_{n,t}\right).$$

Below, we derive our bounds on $U^1_s(n)$ and $U^2_s(n)$.

Bound on $U^1_s(n)$. The proof is based on the following idea. Each run $\rho \in R_1$ makes a number of visits (say t visits) to A before reaching F. We can thus partition ρ into t segments, each representing a part of ρ between two re-visits of A. To reason about the segments of ρ, we need a number of definitions.

For natural numbers $1 \leq t \leq n$, let $n \oplus t$ be the set of vectors of positive natural numbers of the form $(x_1, \ldots, x_t)$ such that $x_1 + \cdots + x_t = n$. Intuitively, the number x_i represents the length of the i^{th} segment of ρ. Observe that the set $n \oplus t$ contains $\binom{n-1}{t-1}$ elements.

For paths $\pi = s_0 s_1 \cdots s_m$ and $\pi' = s'_0 s'_1 \cdots s'_n$ with $s_m = s'_0$, let $\pi \bullet \pi'$ denote the path $\pi = s_0 s_1 \cdots s_m s'_1 \cdots s'_n$. For a set $A \subseteq S$ and $v = (x_1, \ldots, x_t) \in (n \oplus t)$, a run ρ satisfies $A^{\#}_{n,v}$ if $\rho^n = \pi_1 \bullet \pi_2 \bullet \cdots \bullet \pi_t$ and for each $i : 1 \leq i \leq t$: (i) $|\pi_i| = x_i$, (ii) $\pi_i(0) \in A$, and (iii) $\pi_i(j) \notin A$, for each $j : 0 < j < |\pi_i|$. By eagerness of $\mathcal{M}$ we get the following bound on the measure of runs satisfying $A^{\#}_{n,v}$.

Lemma 6.2 *For each $n, t : 1 \leq t \leq n$, $v \in (n \oplus t)$, and $s \in A$, it is the case that $\mathcal{P}_{\mathcal{M}}\left(s \models A^{\#}_{n,v}\right) \leq \beta^{n-t}$.*

Recalling the definition of $U^1_s(n)$ and using Lemma 6.2:

$$U^1_s(n) \leq \sum_{t=1}^{\lfloor \frac{n}{c} \rfloor} \mathcal{P}_{\mathcal{M}}\left(s \models A^{\#}_{n,t}\right) = \sum_{t=1}^{\lfloor \frac{n}{c} \rfloor} \sum_{v \in (n \oplus t)} \mathcal{P}_{\mathcal{M}}\left(s \models A^{\#}_{n,v}\right)$$

$$= \sum_{t=1}^{\lfloor \frac{n}{c} \rfloor} \sum_{v \in (n \oplus t)} \beta^{n-t} = \sum_{t=1}^{\lfloor \frac{n}{c} \rfloor} \binom{n-1}{t-1} \beta^{n-t}.$$

To bound the last sum, we use the following lemma.

Lemma 6.3 *For all $n \geq 2c$ and $c \geq 2$,*

$$\sum_{t=1}^{\lfloor n/c \rfloor} \binom{n-1}{t-1} \beta^{n-t} \leq \left(\left(\frac{c}{c-1} \right) (2c)^{1/c} \left(\frac{1}{c} + \frac{1}{\beta} \right)^{1/c} \cdot \beta \right)^n.$$

Choose $c > \max\left(1 + \frac{1}{\beta^{-1/3} - 1}, 7, \frac{9}{\log^2 \beta}, \frac{-3\log(\frac{1}{7} + \beta^{-1})}{\log \beta}\right)$. Define $\alpha_1 = \left(\frac{c}{c-1}\right) \cdot (2c)^{1/c} \cdot \left(\frac{1}{c} + \frac{1}{\beta}\right)^{1/c} \cdot \beta$, and $N_1 = 2c$. It is not difficult to prove that we have $\alpha_1 < 1$. By Lemma 6.3, $U^1_s(n) \leq \alpha_1^n$ for each $n \geq N_1$.

Bound on $U^2_s(n)$. Let B be the subset of A from which F is reachable, i.e., $B := \{s \in A | \; s \models \exists \Diamond F\}$. If $s \in A - B$ then trivially $U^2_s(n) = 0$. In the following we consider the case when $s \in B$. Let $w := |B|$.

The bound on $U^2_s(n)$ is computed based on the observation that runs in R_2 visit A many times before reaching F. To formalize this, we need a definition. For a natural number k and sets of states S_1, S_2, we define $\left(s \models S_1^k \; \underline{\textit{Before}} \; S_2\right)$ to be the set of s-runs ρ that make at least k visits to S_1 before visiting S_2 for the first time. Formally, an s-run satisfies the formula if there are $0 \leq i_1 < i_2 < \cdots < i_k \leq n$ such that $\rho(i_j) \in S_1$ for each $j : 1 \leq j \leq k$, and $\rho(i) \notin S_2$ for each $i : 0 \leq i \leq n$. We write $S_1 \; \underline{\textit{Before}} \; S_2$ instead of $S_1^1 \; \underline{\textit{Before}} \; S_2$, $S_1^k \; \underline{\textit{Before}} \; s_2$ instead of $S_1^k \; \underline{\textit{Before}} \; \{s_2\}$, and $s_1^k \; \underline{\textit{Before}} \; S_2$ instead of $\{s_1\}^k \; \underline{\textit{Before}} \; S_2$.

Notice that $(s \models \Diamond^{=n} F \wedge A^{\#}_{n,t}) = (s \models \Diamond^{=n} F \wedge B^{\#}_{n,t}) \subseteq (s \models B^t \; \underline{\textit{Before}} \; F)$. It follows that $U^2_s(n) \leq \sum_{t=\lfloor \frac{n}{c} \rfloor + 1}^{n} \mathcal{P}_{\mathcal{M}}\left(s \models B^t \; \underline{\textit{Before}} \; F\right)$.

Any run from s that makes t visits to B before visiting F must have the following property. By the Pigeonhole principle there exists at least one state $s_B \in B$ that is visited at least $\lceil t/w \rceil$ times before visiting F. This means that

$$\left(s \models B^t \; \underline{\textit{Before}} \; F\right) \subseteq \bigcup_{s_B \in B} \left(s \models s_B^{\lceil t/w \rceil} \; \underline{\textit{Before}} \; F\right),$$

and hence

$$U^2_s(n) \leq \sum_{t=\lfloor \frac{n}{c} \rfloor + 1}^{n} \sum_{s_B \in B} \mathcal{P}_{\mathcal{M}}\left(s \models s_B^{\lceil t/w \rceil} \; \underline{\textit{Before}} \; F\right).$$

By cutting runs at the first occurrence of s_B, we see that $\mathcal{P}_{\mathcal{M}}(s \models s_B^{\lceil t/w \rceil} \; \underline{\textit{Before}} \; F) = \mathcal{P}_{\mathcal{M}}(s \models s_B \; \underline{\textit{Before}} \; F) \cdot \mathcal{P}_{\mathcal{M}}(s_B \models s_B^{\lceil t/w \rceil} \; \underline{\textit{Before}} \; F)$ and in particular $\mathcal{P}_{\mathcal{M}}(s \models s_B^{\lceil t/w \rceil} \; \underline{\textit{Before}} \; F) \leq \mathcal{P}_{\mathcal{M}}(s_B \models s_B^{\lceil t/w \rceil} \; \underline{\textit{Before}} \; F)$. Consider the runs in the set $(s_B \models s_B^{\lceil t/w \rceil} \; \underline{\textit{Before}} \; F)$. In such a run, there are $\lceil t/w \rceil$ parts that go from s_B to s_B and avoid F. The following lemma gives an upper bound on such runs. To capture this upper bound, we introduce the parameter μ which is defined to be positive and smaller than the minimal probability, when starting from some $s \in B$, of visiting F before returning to s. In other words, $0 < \mu \leq \min_{s \in B} \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc(F \; \underline{\textit{Before}} \; s)\right)$. Note that μ is well-defined since F is reachable from all $s \in B$.

Lemma 6.4 *$\mathcal{P}_{\mathcal{M}}\left(s_B \models s_B^x \; \underline{\textit{Before}} \; F\right) \leq (1-\mu)^{x-1}$, for each $s_B \in B$.*

Since μ only needs to be a lower bound, we can assume $\mu < 1$. From Lemma 6.4 it follows that

$$
\begin{aligned}
U_s^2(n) &\le \sum_{t=\lfloor \frac{n}{c} \rfloor+1}^{n} \sum_{s_B\in B} (1-\mu)^{\lceil t/w \rceil -1} \\
&\le \frac{w}{1-\mu} \cdot \sum_{t=\lfloor \frac{n}{c} \rfloor+1}^{n} (1-\mu)^{t/w} \\
&= \frac{w}{1-\mu} \cdot \frac{(1-\mu)^{(\lfloor \frac{n}{c} \rfloor+1)/w} - (1-\mu)^{(n+1)/w}}{1-(1-\mu)^{1/w}} \\
&< \frac{w}{(1-\mu)(1-(1-\mu)^{1/w})} \cdot \left((1-\mu)^{\frac{1}{cw}}\right)^n .
\end{aligned}
$$

Define α_2 such that $(1-\mu)^{\frac{1}{cw}} < \alpha_2 < 1$ and $N_2 \ge \frac{\log w - \log(1-\mu) - \log(1-(1-\mu)^{1/w})}{\log \alpha_2 - \frac{1}{cw}\log(1-\mu)}$. It is then easy to see that $U_s^2(n) \le \alpha_2^n$ for each $n \ge N_2$.

Remark. The reason why we do not use *equality* in the definition of μ, i.e., define $\mu = \min_{s\in B} \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc(F \underline{\mathit{Before}}\ s)\right)$, is that (as it will later be explained for PLCS) it is in general hard to compute $\min_{s\in B} \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc(F \underline{\mathit{Before}}\ s)\right)$ exactly. However, we can compute a non-zero lower bound, which is sufficient for the applicability of our algorithm,

Eagerness of $\mathcal{M}$ with respect to $s \in A$. From the bounds on $U_s^1(n)$ and $U_s^2(n)$, we derive the parameters of $(\mathcal{M}, s)$ as follows. Take α_3 such that $\max(\alpha_1, \alpha_2) < \alpha_3 < 1$ and $N_3 \ge \max\left(N_1, N_2, \frac{\log 2}{\log \alpha_3 - \log \max(\alpha_1,\alpha_2)}\right)$. Then $U_s(n) \le U_s^1(n) + U_s^2(n) \le \alpha_1^n + \alpha_2^n \le \alpha_3^n$ for each $n \ge N_3$. Finally,

$$
\mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{\ge n} F\right) = \sum_{i=n}^{\infty} U_s(i) \le \frac{\alpha_3^n}{1-\alpha_3}
$$

for each $n \ge N_3$. Choose α such that $\alpha_3 < \alpha < 1$ and $N \ge \max\left(N_3, \frac{\log(1-\alpha_3)}{\log \alpha_3 - \log \alpha}\right)$. It follows that $\mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{\ge n} F\right) \le \alpha^n$ for each $n \ge N$. This concludes the proof of Theorem 6.1. □

7 GR-Attractors

We define the class of *gambler's ruin-like attractors* or *GR-attractors* for short, show that any GR-attractor is also eager (Lemma 7.1), and that any PLCS contains a GR-attractor (Lemma 7.4). Finally, we conclude that APPROX_EXPECT is solvable for PLCS (Theorem 7.5).

7.1 GR-Attractors

Let $\mathcal{M} = (S, P, F)$ be a Markov chain that contains a finite attractor $A \subseteq S$. Then A is called a *GR-attractor*, if there exists a "distance" function $h : S \to \mathbb{N}$ and a constant $q > 1/2$ such that for any state $s \in S$ the following conditions hold.

1. $h(s) = 0 \iff s \in A$.
2. $\sum_{h(s')<h(s)} P(s, s') \ge q$, for all $h(s) \ge 1$.
3. $P(s, s') = 0$ if $h(s') > h(s) + 1$.

Let $p := 1 - q$. We call (p, q) the parameter of A. Intuitively, h describes the distance from A. This condition means that, in every step, the distance to A does not increase by more than 1, and it decreases with probability uniformly $> 1/2$. In particular, this implies that A is an attractor, i.e., $\forall s \in S. \mathcal{P}_{\mathcal{M}}(s \models \Diamond A) = 1$, but not every attractor has the distance function. As we will see below, a Markov chain with a GR-attractor generalizes the classical "gambler's ruin" problem [22], but converges at least as quickly. We devote the rest of Section 7.1 to show the following Lemma.

Lemma 7.1 *Let $\mathcal{M}$ be a Markov chain. Every finite GR-attractor with parameter (p, q) is an eager attractor with parameter $\beta = \sqrt{4pq}$.*

To prove this, we need several auxiliary constructions.

For a state $s \in S$ with $h(s) = k$, we want to derive an upper bound for the probability of reaching A in n or more steps. Formally, $f(k, n) := \sup_{h(s)=k} \mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{\ge n} A\right)$.

To obtain an upper bound on $f(k, n)$, we relate our Markov chain $\mathcal{M}$ to the Markov chain $\mathcal{M}^G$ from the gambler's ruin problem [22], defined as $\mathcal{M}^G = (\mathbb{N}, P_G, \{0\})$ with $P_G(x, x-1) = q, P_G(x, x+1) = p := 1-q$ for $x \ge 1$ and $P_G(0, 0) = 1$. Let $g(k, n) := \mathcal{P}_{\mathcal{M}^G}\left(k \models \Diamond^{\ge n}\ 0\right)$.

The following Lemma shows that f is bounded by g, so that any upper bound for the gambler's ruin problem also applies to a GR-attractor.

Lemma 7.2 *If $0 \le k \le n$ then $f(k, n) \le g(k, n)$.*

Next, we give an upper bound for the gambler's ruin problem.

Lemma 7.3 *For all $n \ge 2$, $g(1, n) \le \frac{3q}{\sqrt{\pi}}(4pq)^{\lfloor \frac{n}{2} \rfloor}$.*

Proof. **(of Lemma 7.1)** Let $\beta := \sqrt{4pq}$. For $n = 0$, we have $\mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc\left(\Diamond^{\ge n} A\right)\right) \le 1 = \beta^0$. For $n = 1$, we have $\mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc\left(\Diamond^{\ge n} A\right)\right) \le p \le \beta^1$. For $n \ge 2$, Lemma 7.2 gives $\mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc\left(\Diamond^{\ge n} A\right)\right) \le p \cdot g(1, n)$, so by Lemma 7.3, $\mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc\left(\Diamond^{\ge n} A\right)\right) \le \frac{3pq}{\sqrt{\pi}}\ (4pq)^{\lfloor \frac{n}{2} \rfloor} = \frac{3}{4\sqrt{\pi}}\ (4pq)^{\lfloor \frac{n}{2} \rfloor+1} \le \frac{3}{4\sqrt{\pi}}(4pq)^{\frac{n}{2}} \le \left(\sqrt{4pq}\right)^n = \beta^n$. □

7.2 Probabilistic Lossy Channel Systems

As an example of systems with finite GR-attractors, we consider *Probabilistic lossy channel systems (PLCS)*. These are probabilistic processes with a finite control unit and a finite set of channels, each of which behaves as a FIFO buffer which is unbounded and unreliable in the sense that it can spontaneously lose messages. There exist several variants of PLCS which differ in how many messages can be lost, with which probabilities, and in which situations. We consider the relatively realistic PLCS model from [4, 11, 27] where each message in transit independently has the probability $\lambda > 0$ of being lost in every step, and the transitions themselves are subject to probabilistic choice. In [4, 11, 7], it was shown that each Markov chain induced by a PLCS contains a finite attractor. Here we show a stronger result.

Lemma 7.4 *Each Markov chain induced by a PLCS contains a GR-attractor.*

Proof. For any configuration c, let $\#c$ be the number of messages in transit in c. We define the attractor A as the set of all configurations that contain at most k messages in transit, for a sufficiently high number k (to be determined). $A := \{c \mid \#c \leq k\}$. Since there are only finitely many different messages and a finite number of control-states, A is finite for every fixed k. The distance function h is defined by $h(c) := \max\{0, \#c - k\}$. Now we show that h satisfies the requirements for a GR-attractor. The first condition, $h(c) = 0 \iff c \in A$, holds by definition of h and A. The third condition holds, because, by definition of PLCS, at most one new message can be added in every single step. Consider now a configuration c with at least k messages. For the second condition it suffices to show that, for sufficiently large k, the probability of losing at least two messages in transit is at least $q > 1/2$ (and thus the new configuration contains at least one message less than the previous one, since at most one new message is added). The probability q of losing at least 2 messages (of at least $k+1$) satisfies $q \geq 1 - ((1-\lambda)^{k+1} + (k+1)\lambda(1-\lambda)^k) = 1 - (1-\lambda)^k(1+\lambda k))$. Since $\lambda > 0$, we can choose k s.t. $q > 1/2$. It suffices to take $k \geq \frac{2}{\lambda}$. □

Theorem 7.5 *The problem* APPROX_EXPECT *is computable for PLCS.*

Proof. By Lemma 7.4 the Markov chain induced by a PLCS contains a GR-attractor, which is an eager attractor by Lemma 7.1. Then, by Theorem 6.1 the Markov chain is eager and Algorithm 1 can in principle solve the problem APPROX_EXPECT. However, to apply the algorithm, we first need to know (i.e., compute) the parameters, or at least sufficient bounds on them.

Given the parameter λ for message loss in the PLCS, we choose the parameter k and the GR-attractor A such that $q > 1/2$, as in the proof of Lemma 7.4. This attractor is also eager with parameter $\beta = \sqrt{4(1-q)q} < 1$ by Lemma 7.1. For any effectively representable set of target states F of a PLCS, the set $\{s \in S : s \models \exists\Diamond F\}$ is upward-closed and effectively computable [2]. Thus we can compute $B = A \cap \{s \in S : s \models \exists\Diamond F\}$ and obtain the parameter $w = |B|$. Since B is known and finite, we can compute (in finite time) an appropriate μ, i.e., a μ such that $0 < \mu \leq \min_{s\in B} \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc(F\ \underline{Before}\ s)\right)$, by path exploration. When A, w, μ, and β are known, we can compute, in turn, c, α_1, N_1, α_2, N_2, α_3, N_3 and finally α and N, according to Section 6. □

Remark. Choosing a larger k (and thus larger attractor A) has advantages and disadvantages. The advantage is that a larger k yields a larger q and thus a smaller parameter $\beta = \sqrt{4pq}$ and thus possibly faster convergence. The disadvantage is that a larger attractor A possibly yields a smaller parameter μ and a larger parameter w (see Section 6) and both these effects cause slower convergence.

8 Conclusion, Discussion, and Future Work

We have presented an algorithm for approximating expected rewards up-to an arbitrarily small error. We have described a class of infinite-state Markov chains, namely *eager Markov chains*, for which the algorithm is guaranteed to terminate. Eager Markov chains represent a subclass of *confluent Markov chains* for which qualitative and quantitative probability analysis algorithms were presented in [3]. Several classes of systems induce eager Markov chains. One example are those which contain *eager attractors*. Such systems are a subclass of the class of systems containing finite attractors, for which several probability analysis algorithms have been presented in the literature [3, 4, 7, 11, 27]. Another example is Markov chains which are *boundedly coarse*. These, in turn, represent a subclass of globally coarse Markov chains which were analyzed in [3] and defined as those where the set of final states is either unreachable or reachable with a probability bounded from below. The classes are related as in Figure 1, where the left and right parts are "orthogonal": except for the "Eager" and "Confluent" circles, every circle in the left part of the figure intersects every circle in the right part without one being contained in the other. Here, all attractors are finite, and *NTM'* is the alternative encoding of NTMs into Markov chains suggested in the remark near the end of Section 5.

Our (weak) assumption of exponentially bounded reward functions allows us to analyze many common aspects of systems. In fact, many natural reward functions are polynomial. For instance, the reward function that computes the running time by simply returning the length of a run is linear, as well as and the reward function for NTMs that computes the space usage by summing over all tapes the difference between the maximum and minimum reading head positions. For PLCS, the reward function that computes the sum of all channel sizes of all states in the run is quadratically bounded, as is the one that computes the sum of all elements of all vectors in a run of a PVASS.

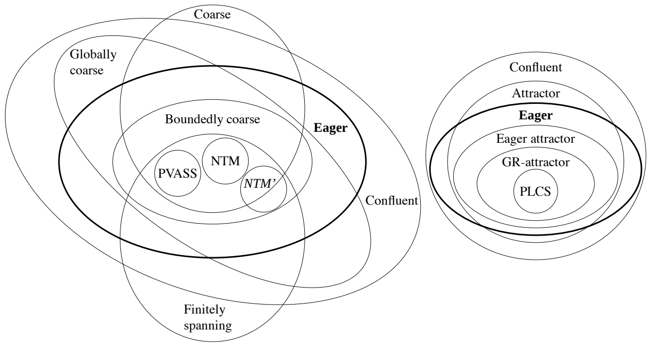

Figure 1. The inclusion relations between classes of Markov chains.

Directions for future work include the analysis of steady-state properties like limiting average expected rewards for the classes of Markov chains we have considered in this paper. Intuitively, this quantity expresses the average reward per step in the long run. We also consider extending our results to Markov decision processes, where both non-deterministic and stochastic transitions are allowed, and to stochastic games (competitive Markov chains) where, in addition, universal transitions are possible.

References

[1] P. A. Abdulla, C. Baier, P. Iyer, and B. Jonsson. Reasoning about probabilistic lossy channel systems. In *Proc. CONCUR 2000*, 2000.

[2] P. A. Abdulla, K. Čerāns, B. Jonsson, and T. Yih-Kuen. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160:109–127, 2000.

[3] P. A. Abdulla, N. B. Henda, and R. Mayr. Verifying infinite Markov chains with a finite attractor or the global coarseness property. In *Proc. LICS' 05*, pages 127–136, 2005.

[4] P. A. Abdulla and A. Rabinovich. Verification of probabilistic systems with faulty communication. In *Proc. FOSSACS03*, volume 2620 of *LNCS*, pages 39–53, 2003.

[5] E. Asarin and P. Collins. Noisy Turing machines. In *Proc. ICALP '05*, pages 1031–1042, 2005.

[6] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model-checking continuous-time Markov chains. *ACM Trans. on Computational Logic*, 1(1):162–170, 2000.

[7] C. Baier, N. Bertrand, and P. Schnoebelen. A note on the attractor-property of infinite-state Markov chains. *Information Processing Letters*, 97(2):58–63, 2006.

[8] C. Baier and B. Engelen. Establishing qualitative properties for probabilistic lossy channel systems. In Katoen, editor, *ARTS'99, Formal Methods for Real-Time and Probabilistic Systems, 5th Int. AMAST Workshop*, volume 1601 of *LNCS*, pages 34–52. Springer Verlag, 1999.

[9] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking meets performance evaluation. *ACM Performance Evaluation Review*, 32(2):10–15, 2005.

[10] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Automated performance and dependability evaluation using model checking. In *Proc. Performance 2002*, pages 261–289, 2002.

[11] N. Bertrand and P. Schnoebelen. Model checking lossy channels systems is probably decidable. In *Proc. FOSSACS03*, volume 2620 of *LNCS*, pages 120–135, 2003.

[12] T. Brázdil and A. Kučera. Computing the expected accumulated reward and gain for a subclass of infinite Markov chains. In *Proc. FSTTCS '05*, volume 3821 of *LNCS*, pages 372–383, 2005.

[13] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, Dec. 1999.

[14] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[15] L. de Alfaro, M. Z. Kwiatkowska, G. Norman, D. Parker, and R. Segala. Symbolic model checking of probabilistic processes using mtbdds and the Kronecker representation. In *Proc. TACAS '00*, volume 1785 of *LNCS*, pages 123–137, 2000.

[16] J. Esparza and K. Etessami. Verifying probabilistic procedural programs. In *Proc. FSTTCS '04*, pages 16–31, 2004.

[17] J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. In *Proc. LICS' 04*, pages 12–21, 2004.

[18] J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: Expectations and variances. In *Proc. LICS' 05*, pages 117–126, 2005.

[19] K. Etessami and M. Yannakakis. Algorithmic verification of recursive probabilistic state machines. In *Proc. TACAS '05*, volume 3440 of *LNCS*, pages 253–270, 2005.

[20] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proc. STACS'05*, volume 2996 of *LNCS*, pages 340–352, 2005.

[21] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *Proc. ICALP '05*, volume 3580 of *LNCS*, pages 891–903, 2005.

[22] W. Feller. *An Introduction to Probability Theory and Its Applica tions*, volume 1. Wiley & Sons, second edition, 1966.

[23] S. Hart and M. Sharir. Probabilistic temporal logics for finite and bounded models. In *Proc. STOC '84*, pages 1–13, 1984.

[24] P. Iyer and M. Narasimha. Probabilistic lossy channel systems. In *TAPSOFT '97: Theory and Practice of Software Development*, volume 1214 of *LNCS*, pages 667–681, 1997.

[25] M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic model checking in practice: Case studies with PRISM. *ACM Performance Evaluation Review*, 32(2):16–21, 2005.

[26] D. Lehmann and S. Shelah. Reasoning with time and chance. *Information and Control*, 53:165–198, 1982.

[27] A. Rabinovich. Quantitative analysis of probabilistic lossy channel systems. In *Proc. ICALP '03*, volume 2719 of *LNCS*, pages 1008–1021, 2003.

[28] P. Stănică. Good lower and upper bounds on binomial coefficients. *Journal of Inequalities in Pure and Applied Mathmatics*, 2(3), 2001.

[29] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. FOCS '85*, pages 327–338, 1985.

A Proofs of Some Lemmas

Lemma 6.2 By induction on t.

Base Case. Suppose that $v \in (n \oplus 1)$. Then $\mathcal{P}_{\mathcal{M}}\left(s \models A^{\#}_{n,v}\right) = \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc(\diamond^{\geq n-1} A)\right)$. By eagerness of A it follows that $\mathcal{P}_{\mathcal{M}}\left(s \models A^{\#}_{n,v}\right) \leq \beta^{n-1}$.

Induction Step. If $t > 1$, let $v = (x_1, \ldots, x_t)$ and let $v_1 = (x_2, \ldots, x_t)$. We know that

$$\mathcal{P}_{\mathcal{M}}\left(s \models A^{\#}_{n,v}\right) = \sum_{s_1 \in A} \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc\left(\diamond^{=x_1-1} s_1\right)\right) \cdot \mathcal{P}_{\mathcal{M}}\left(s_1 \models A^{\#}_{n-x_1,v_1}\right).$$

By the induction hypothesis it follows that $\mathcal{P}_{\mathcal{M}}\left(s_1 \models A^{\#}_{n-x_1,v_1}\right) \leq \beta^{n-x_1-(t-1)}$. This means that

$$\begin{aligned}
&\mathcal{P}_{\mathcal{M}}\left(s \models A^{\#}_{n,v}\right) \leq \\
&\quad \beta^{n-x_1-(t-1)} \cdot \sum_{s' \in A} \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc\left(\diamond^{=x_1-1} s'\right)\right) = \\
&\qquad \beta^{n-x_1-(t-1)} \cdot \mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc\left(\diamond^{=x_1-1} A\right)\right).
\end{aligned}$$

By eagerness of A it follows that $\mathcal{P}_{\mathcal{M}}\left(s \models \bigcirc\left(\diamond^{=x_1-1} A\right)\right) \leq \beta^{x_1-1-1}$, and hence

$$\mathcal{P}_{\mathcal{M}}\left(s \models A^{\#}_{n,v}\right) \leq \beta^{n-x_1-1-(t-1)} \cdot \beta^{x_1-1} = \beta^{n-t}. \quad \square$$

To prove Lemma 6.3, we need the following auxiliary lemma:

Lemma A.1 *For all $x \geq 2c$ and $c \geq 2$,*

$$\binom{x}{\lfloor x/c \rfloor} < \left(\left(\frac{c}{c-1}\right)(2c)^{1/c}\right)^x.$$

Proof. We apply Theorem 2.6. of [28] with $p := \lfloor x/c \rfloor$, $n := 1$ and $m := x$ and obtain

$$\begin{aligned}
&\binom{x}{\lfloor x/c \rfloor} < \frac{1}{\sqrt{2\pi}} \frac{x^{x+1/2}}{(x-\lfloor x/c \rfloor)^{x-\lfloor x/c \rfloor+1/2}(\lfloor x/c \rfloor)^{\lfloor x/c \rfloor+1/2}} \leq \\
&\left(\frac{x}{x-\lfloor x/c \rfloor}\right)^x \left(\frac{x-\lfloor x/c \rfloor}{\lfloor x/c \rfloor}\right)^{\lfloor x/c \rfloor} \sqrt{\frac{x}{2\pi(x-\lfloor x/c \rfloor)\lfloor x/c \rfloor}} \leq \\
&\left(\frac{x}{x-x/c}\right)^x (2c)^{x/c} \sqrt{\frac{x}{2\pi(x-x/c)(x/c-1)}} \leq \\
&\left(\left(\frac{c}{c-1}\right)(2c)^{1/c}\right)^x. \quad \square
\end{aligned}$$

Lemma 6.3

$$\begin{aligned}
&\sum_{t=1}^{\lfloor n/c \rfloor} \binom{n-1}{t-1} \beta^{n-t} \leq \\
&\beta^n \sum_{t=0}^{\lfloor n/c \rfloor} \binom{n}{t} \left(\frac{1}{\beta}\right)^t = \\
&\beta^n \sum_{t=0}^{\lfloor n/c \rfloor} \binom{n}{\lfloor n/c \rfloor}\binom{\lfloor n/c \rfloor}{t} \frac{(n-\lfloor n/c \rfloor)!\,(\lfloor n/c \rfloor - t)!}{(n-t)!} \left(\frac{1}{\beta}\right)^t = \\
&\beta^n \binom{n}{\lfloor n/c \rfloor} \sum_{t=0}^{\lfloor n/c \rfloor} \binom{\lfloor n/c \rfloor}{t} \left(\prod_{i=1}^{\lfloor n/c \rfloor - t} \frac{i}{n-\lfloor n/c \rfloor + i}\right) \left(\frac{1}{\beta}\right)^t \leq \\
&\beta^n \binom{n}{\lfloor n/c \rfloor} \sum_{t=0}^{\lfloor n/c \rfloor} \binom{\lfloor n/c \rfloor}{t} \left(\frac{1}{c}\right)^{\lfloor n/c \rfloor - t} \left(\frac{1}{\beta}\right)^t \leq \\
&\beta^n \binom{n}{\lfloor n/c \rfloor} \left(\frac{1}{c}\right)^{\lfloor n/c \rfloor} \left(1+\frac{c}{\beta}\right)^{\lfloor n/c \rfloor} \\
&\leq \{\text{lemma A.1 with the hypotheses } n \geq 2c \text{ and } c \geq 2\} \\
&\beta^n \left(\left(\frac{c}{c-1}\right)(2c)^{1/c}\right)^n \left(\frac{1}{c}+\frac{1}{\beta}\right)^{n/c} = \\
&\left(\left(\frac{c}{c-1}\right) \cdot (2c)^{1/c} \cdot \left(\frac{1}{c}+\frac{1}{\beta}\right)^{1/c} \cdot \beta\right)^n.
\end{aligned}$$

For a sufficiently large c (which depends on β), the base is < 1. $\square$

Lemma 6.4 By induction on x. The base case (when $x = 1$) is trivial. For the induction step, we observe that

$$\begin{aligned}
&\mathcal{P}_{\mathcal{M}}\left(s_B \models s_B^x \ \underline{\mathit{Before}}\ F\right) \leq \\
&\mathcal{P}_{\mathcal{M}}\left(s_B \models s_B^2 \ \underline{\mathit{Before}}\ F\right) \cdot \mathcal{P}_{\mathcal{M}}\left(s_B \models s_B^{x-1} \ \underline{\mathit{Before}}\ F\right).
\end{aligned}$$

By definition, we know that

$$\begin{aligned}
&\mathcal{P}_{\mathcal{M}}\left(s_B \models s_B^2 \ \underline{\mathit{Before}}\ F\right) \\
&= \mathcal{P}_{\mathcal{M}}\left(s_B \models \bigcirc\left(s_B \ \underline{\mathit{Before}}\ F\right)\right) \leq (1-\mu).
\end{aligned}$$

By the induction hypothesis we know that

$$\mathcal{P}_{\mathcal{M}}\left(s_B \models s_B^{x-1} \ \underline{\mathit{Before}}\ F\right) \leq (1-\mu)^{x-2}.$$

The result now follows. $\square$

To prove Lemma 7.2, we first show that g increases in the first parameter:

Lemma A.2 $\forall j,k,n : 0 \leq j \leq k \leq n \implies g(j,n) \leq g(k,n)$.

Proof. We show that $\forall k, n : 1 \le k \le n : g(k-1,n) \le g(k,n)$ which implies the result. We use induction on n. The base case $n = 0$ holds because $P(k \models \Diamond^{\ge 0}\ 0) = 1$ for all k. In the induction step we assume $n \ge 1$ and consider two cases. If $k = 1$ then the result is trivial since $g(k-1, n+1) = 0$. If $k \ge 2$, then

$$\begin{aligned} g(k-1,n+1) &= q \cdot g(k-2,n) + p \cdot g(k,n) \\ &\le q \cdot g(k-1,n) + p \cdot g(k+1,n) = g(k,n+1), \end{aligned}$$

where the equalities follow from the definition of $\mathcal{M}^G$ and the inequality from the induction hypothesis. □

Lemma 7.2 By induction on n.

The base case is trivial, since $f(0,0) = g(0,0) = 1$. For the induction step, we consider two cases. The case when $k = 0$ is trivial since $f(0, n+1) = 0$. Now, we prove the case when $k \ge 1$. For any $\varepsilon > 0$, let s be a state such that $h(s) = k$ and $\mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{\ge n+1} A\right) + \varepsilon \ge f(k, n+1)$. Such an s exists by the definition of f. Then:

$$\begin{aligned}
& f(k,n+1) - \varepsilon \\
& \le \{\text{Definition of } s\} \\
& \mathcal{P}_{\mathcal{M}}\left(s \models \Diamond^{\ge n+1} A\right) \\
& = \{\text{Definition of GR-attractor, clause 3}\} \\
& \textstyle\sum_{j=0}^{k-1} \sum_{h(s')=j} P(s,s') \cdot \mathcal{P}_{\mathcal{M}}\left(s' \models \Diamond^{\ge n} A\right) + \\
& \textstyle\sum_{h(s')=k} P(s,s') \cdot \mathcal{P}_{\mathcal{M}}\left(s' \models \Diamond^{\ge n} A\right) + \\
& \textstyle\sum_{h(s')=k+1} P(s,s') \cdot \mathcal{P}_{\mathcal{M}}\left(s' \models \Diamond^{\ge n} A\right) \\
& \le \{\text{Definition of } f\} \\
& \textstyle\sum_{j=0}^{k-1} f(j,n) \cdot \left(\sum_{h(s')=j} P(s,s')\right) + \\
& f(k,n) \cdot \left(\textstyle\sum_{h(s')=k} P(s,s')\right) + \\
& f(k+1,n) \cdot \left(\textstyle\sum_{h(s')=k+1} P(s,s')\right) \\
& \le \{\text{Induction hypothesis and Lemma A.2}\} \\
& g(k-1,n) \cdot \left(\textstyle\sum_{j=0}^{k-1} \sum_{h(s')=j} P(s,s')\right) + \\
& g(k+1,n) \cdot \left(\textstyle\sum_{(h(s')=k) \vee (h(s')=k+1)} P(s,s')\right) \\
& \le \{\text{Definition of GR-attractor, clause 3}\} \\
& g(k-1,n) \cdot \left(\textstyle\sum_{j=0}^{k-1} \sum_{h(s')=j} P(s,s')\right) + \\
& g(k+1,n) \cdot \left(1 - \textstyle\sum_{j=0}^{k-1} \sum_{h(s')=j} P(s,s')\right) \\
& \le \{\text{Definition of GR-attractor, clause 2, and Lemma A.2}\} \\
& q \cdot g(k-1,n) + p \cdot g(k+1,n) \\
& = \{\text{Definition of } g \text{ and } \mathcal{M}^G\} \\
& g(k,n+1).
\end{aligned}$$

Since this holds for arbitrarily small $\varepsilon > 0$, we must have $f(k, n+1) \le g(k, n+1)$. □

Lemma 7.3 The case for $n = 1$ is trivial. In the following we assume $n \ge 2$. It follows from equation (5.9) in [22] (page 323) that

$$g(1,n) = \sum_{x=n}^{\infty} \frac{1}{x} \binom{x}{\frac{x-1}{2}} p^{\frac{x-1}{2}} q^{\frac{x+1}{2}},$$

where $p = 1 - q$ and the binomial coefficient is interpreted as zero if $(x-1)/2$ is not an integer. Substituting $2m+1$ for x gives

$$\begin{aligned} g(1,n) &= \sum_{m=\lfloor n/2 \rfloor}^{\infty} \frac{1}{2m+1} \binom{2m+1}{m} p^m q^{m+1} \\ &= \sum_{m=\lfloor n/2 \rfloor}^{\infty} \frac{1}{m+1} \binom{2m}{m} p^m q^{m+1}. \end{aligned}$$

Since $n \ge 2$, we can assume that $m \ge 1$. A bound on the binomial coefficient follows, e.g., from results in [28]:

$$\binom{2m}{m} \;<\; \frac{1}{\sqrt{\pi}}\, m^{-\frac{1}{2}}\, 2^{2m}.$$

It follows that

$$g(1,n) \le \frac{q}{\sqrt{\pi}} \sum_{m=\lfloor n/2 \rfloor}^{\infty} m^{-\frac{3}{2}}\, (4pq)^m.$$

Since $q > 1/2$ we have $4pq < 1$. Thus the summands are monotone decreasing in m and we can conservatively approximate the sum by the integral and obtain

$$g(1,n) \le \frac{q}{\sqrt{\pi}} \left(\left\lfloor \frac{n}{2} \right\rfloor^{-\frac{3}{2}} (4pq)^{\lfloor \frac{n}{2} \rfloor} + \int_{\lfloor \frac{n}{2} \rfloor}^{\infty} m^{-\frac{3}{2}} (4pq)^m \, dm \right).$$

Since $4pq < 1$ we have $\log(4pq) < 0$. Therefore, standard integration by parts gives the following upper bound on the integral.

$$\int_{\lfloor \frac{n}{2} \rfloor}^{\infty} m^{-\frac{3}{2}} (4pq)^m \, dm \;\le\; 2 \left\lfloor \frac{n}{2} \right\rfloor^{-\frac{1}{2}} (4pq)^{\lfloor \frac{n}{2} \rfloor}.$$

Thus,

$$g(1,n) \le \frac{q}{\sqrt{\pi}} \left(\left\lfloor \frac{n}{2} \right\rfloor^{-\frac{3}{2}} (4pq)^{\lfloor \frac{n}{2} \rfloor} + 2 \left\lfloor \frac{n}{2} \right\rfloor^{-\frac{1}{2}} (4pq)^{\lfloor \frac{n}{2} \rfloor} \right).$$

Since $n \ge 2$, we have $\lfloor \frac{n}{2} \rfloor^{-\frac{3}{2}} \le 1$ and $\lfloor \frac{n}{2} \rfloor^{-\frac{1}{2}} \le 1$ so

$$g(1,n) \le \frac{3q}{\sqrt{\pi}} (4pq)^{\lfloor \frac{n}{2} \rfloor}. \qquad \square$$