Symbolically Quantifying Response Time in Stochastic Models using Moments and Semirings

Hugo Bazille, Eric Fabre, Blaise Genest

Symbolically Quantifying Response Time in Stochastic Models using Moments and Semirings.

Hugo Bazille[1], Eric Fabre[1], and Blaise Genest[2]

[1] Univ Rennes, Inria, SUMO Team, Rennes, France
[2] Univ Rennes, CNRS, IRISA, Rennes, France

Abstract. We study quantitative properties of the response time in stochastic models. For instance, we are interested in quantifying bounds such that a high percentage of the runs answers a query within these bounds. To study such problems, computing probabilities on a state-space blown-up by a factor depending on the bound could be used, but this solution is not satisfactory when the bound is large.
In this paper, we propose a new *symbolic* method to quantify bounds on the response time, using the moments of the distribution of simple stochastic systems. We prove that the distribution (and hence the bounds) is uniquely defined given its moments. We provide *optimal* bounds for the response time over all distributions having a pair of these moments. We explain how to *symbolically* compute in polynomial time any moment of the distribution of response times using adequately-defined semirings. This allows us to compute optimal bounds in parametric models and to reduce complexity for computing optimal bounds in hierarchical models.

1 Introduction

Response time has been considered lately as an important property of systems [8, 15,21]. In this context, one does not simply want a query to be answered eventually, but to be answered in a reasonable amount of time. In the model-checking community, problems on response time have been studied mainly *qualitatively*, in the context of (pure, that is non stochastic) two-player games [8, 21]. There, one looks for a strategy ensuring that the lim-sup of response time is finite. It ensures that under this strategy, there will be a bound on the response time to any query. This has been extended in [15] to a quantitative setting, where one wants to optimize the mean response time in a pure two-player game.

In this paper, we consider stochastic systems. In such systems, the response time is a random variable, unlikely to be bounded as even a single probabilistic loop on a reachable state will make the response time longer than T for a set of runs of small but positive probability, no matter T. Instead, we propose to quantify such response times. One way to do that is to obtain the distribution of response times. Another way is to compute, for a probability $0 < p < 1$, the bound T that is satisfied (by a set of runs) with probability at least $1 - p$. In this paper, we tackle both problems. For that, we use the concept of *moments* of the distribution of response times, as described next.

The *moment of order* r of a probability distribution δ over $\mathbb{R}$ or $\mathbb{R}^+$ is defined as the integral of $x^r\delta(x)$ over the support of δ, when defined (that is if $x^r\delta(x)$ is measurable and the integral is defined). For instance, the moment of order 1 is the expected value of δ, while the moment of order 2 allows one to compute the standard deviation of δ. Inspired by the computation of entropy for automata [10] (see also [1] for the computation of entropy for (non-Zeno) timed-automata), we design new semirings in which each moment corresponds to the sum of weights of runs reaching a state. This construction can be applied to probabilistic automata (that is, labeled discrete time Markov chains), as well as labeled *continuous time Markov chains*, where time is continuous and is drawn according to some rate. Adapting the Floyd-Warshall algorithm provides a *symbolic* way to perform the computation of the n first moments in time cubic in the number of states of the Markov Chain, and quadratic in n. For any n, we can thus compute the value of the first n moments. In some sense, we extend the approach of [12, 16] from computing probabilities to computing any moments. This allows us to evaluate the distribution of response times in two ways:

Firstly, thanks to the symbolic expression of moments, we prove that there is a unique distribution having the moments of a distribution of response times of a probabilistic automaton. We can then build a sequence of distributions matching the first n moments, for instance the maximal entropy one [11]. Here, maximal entropy means assuming the least information besides these moments. This sequence of distributions is then ensured to converge in law towards the distribution of response times.

Secondly, we study optimal symbolic bounds on the time to answer a high percentage of queries, obtained from moments. The Tchebychev inequality provides optimal symbolic bounds when considering the space of distributions having one given moment, of any order i. We obtain bounds optimal in the space of distributions having two given moments, of any orders i, j. We show how this improves Tchebychev bounds on some example. Having symbolic methods allows for instance to deal with parametric systems where the parameters represent uncertain probabilities. In this case, we can compute optimal bounds satisfying all valuations of parameters. For hierarchical systems [3], which are compact representations of large systems, our symbolic method allows to design a much more efficient algorithm (e.g. it does not consider twice the same component) to compute the moments, and thus the bounds. Missing proofs can be found in [5].

Related work: Response times in stochastic systems have been studied for a long time by the perf.eval. community under the name "first passage times", e.g. in [22]. Techniques used in this community to compute moments of Markov chains are mostly based on numerical methods, e.g. [13]. While [13] has the same complexity as our symbolic technique, it is very efficient on explicit models. However, these numerical methods are less adaptable than our symbolic algorithm, in particular concerning parametric or hierarchical systems.

Concerning the determinacy of the distribution given moments, it is known [20] that phase-type distributions of order n are determined by their first $2n-1$ moments. First passage distribution time in Markov chains with n states are

phase type distribution of order n. However, [20] does not help characterizing bounds as it does not ensure that a non-phase type distribution cannot have the exact same moments as a phase type distribution, unlike our result.

Bounding the response time has also been studied in the perf.eval. community. Again, methods used there are mostly numerical [6, 19]. In [19](p.68-69), a symbolic bound is also provided in the particular case of moments of order 1,2 and 3. In [2], it is shown how to use the two first moments of response time across various components to compute general bounds, using techniques close to ours, but restricted to moments of order 1 and 2. In our paper, we provide *optimal* bounds for any order $(i, j) \in \mathbb{N}^2$. Taking into account moments of order $i, j > 3$ is important when the proportion of runs to answer is close to 1.

Last, computing moments find other applications. For instance, in [4, 7, 14], complex functions describing the evolution of molecular species are approximated using the first k moments, for some k.

2 Probabilistic Automata

We first introduce a simple class of models, namely *probabilistic automata* (also called *labeled discrete time Markov chains*), on which we can demonstrate our techniques. Later, we will extend our results to handle continuous time, considering Continuous-Time Markov Chains (CTMC), as well as parametric and hierarchical systems.

Definition 1. *A probabilistic automaton A over a finite alphabet Σ is a tuple (S, Pr, δ_0) where:*

- *S is a finite set of states,*
- *$Pr : S \times \Sigma \times S \to [0, 1]$ is a stochastic transition function such that for all $s \in S$, $\sum_{a \in \Sigma, t \in S} Pr(s, a, t) = 1$: the weights of paths leaving s sum to 1,*
- *$\delta_0 : S \to [0, 1]$ is the initial distribution over states such that $\sum_{s \in S} \delta_0(s) = 1$.*

Example 1. For instance, the model depicted on Fig. 1 is a probabilistic automaton with 3 states $\{1, 2, 3\}$. There is a transition between 1 and 2 labeled **query** with probability 1. From state 2, with probability .9 we stay in state 2 with a transition labeled **wait**, and with probability .1 we go to state 3 with a transition labeled **response**. We loop in state 3 with probability 1.

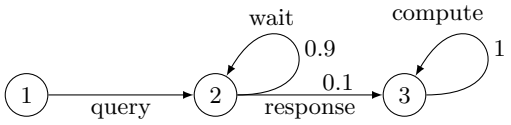

Fig. 1. A simple example of a query-response model

A finite sequence $\pi = s_0, a_1, s_1, \ldots, a_n, s_n \in (S\Sigma)^n S$ is called a *finite path* starting from s_0 and ending in s_n, and a transition $t \in \pi$ if $t = s_i a_{i+1} s_{i+1}$ for some i. We denote $|\pi| = n$ the length of the path π. For a path π_1 ending in s_n and a path π_2 starting from s_n, we can define the concatenated path $\pi_1 \cdot \pi_2$ where the last node of π_1 and the first node of π_2 are merged. A path π_1 is a *prefix* of π if there exists a path π_2 such that $\pi_1 \cdot \pi_2 = \pi$.

For a path π starting in a state s_0, we define $\mathbb{P}(\pi) = \prod_{t \in \pi} Pr(t)$ the probability that a path with prefix π is executed from s_0. A path π is realizable if $\mathbb{P}(\pi) > 0$.

Let s be a state, and Π be a set of finite paths starting from s such that no path in Π is a prefix of another path in Π. Then the probability that a path starting from s has a prefix in Π is $\mathbb{P}(\Pi) = \sum_{\rho \in \Pi} \mathbb{P}(\rho)$. We say that Π is *disjoint* if no path ρ of Π is a prefix of another path $\rho' \neq \rho$ of Π or similarly, $Cyl(\rho) \cap Cyl(\rho') = \emptyset$ with $Cyl(\rho) = \{\pi, \rho \text{ prefix of } \pi\}$.

Some labels of an automaton will be of particular interest concerning response time. Let $\Sigma_Q \subseteq \Sigma$ be a subset of labels standing for queries, and $\Sigma_R \subseteq \Sigma$ be a subset of labels standing for responses. For simplicity, we will assume that there is a unique query type $\Sigma_Q = \{q\}$ and a unique response type $\Sigma_R = \{r\}$, with $q \neq r$. We will also assume that there is no path with two (similar) queries q. To handle cases with several query/response types, it suffices for each type to consider only queries and answers of that type and disregard other types.

Problem statement: We are interested in quantifying the time between queries and responses, called the *response time*, which is a random variable. A way to quantify it is to produce the distribution of response times, either for each transition labeled by a query, or averaged on these transitions, weighted by the probability to see each of these transitions. Another way is to answer model-checking questions such as: what is the smallest delay T such that the mass of paths unanswered after T units of time is smaller than some probability p?

To compute both the distribution and the delay T, we will use the so called *moments of the distribution of response times*. The moment of order 1 is the mean value, and the moment of order 2 allows to compute the standard deviation.

3 *Symbolically* computing moments using semirings

In this section, we define moments and explain how to compute them *symbolically* using appropriately-defined semirings.

Let X be the random variable of the response time. If all queries are answered, then X takes values in N_{max}, else X takes values in $N_{max} \cup \{\infty\}$. Let $p(x)$ be the probability that the response is obtained x units of time after the query, that is, the probability that $X = x$. Variable p is a distribution over response time, with $\sum_x p(x) = 1$.

Definition 2. *For $p : \mathbb{N} \to [0, 1]$ and $n \in \mathbb{N}$, we define the n-th* moment *of p by $\sum_{x \in \mathbb{N}} p(x) \cdot x^n = E(X^n)$, that is the expected value of X^n.*

3.1 Semirings associated with moments

We will compute moments of the distribution of response times by considering each query individually. We can then take e.g. the average over all queries (as we assumed that there are no two queries on the same path). Thus, we first fix a state q, target of a transition labeled by a query. State q symbolizes that a query has just been asked. We then let R be the set of target states of transitions labeled by a response. A state is in R if a response to this query has just been given. For instance, on Fig. 1, we have $q = 2$ and $R = \{3\}$.

We introduce a set of semirings that will allow us to compute symbolically the moment of order n of the distribution of response times to the query associated with state q, for all $n \in \mathbb{N}$. We will compute the moment inductively on a disjoint subset Π of paths of A from q to R. For an integer n, we denote $\mu_n(\Pi) = \sum_{\rho \in \Pi} \mathbb{P}(\rho)|\rho|^n$. Let $\mathbf{Path}_q^R$ be the set of paths in the automaton A between q and the first occurrence of R. Notice that $\mathbf{Path}_q^R$ is disjoint. Thus, we have that $\mu_n(\mathbf{Path}_q^R)$ is the moment of order n of the distribution of response times to the query associated with state q. To avoid some heavy notations, when R is reduced to one state t, let $\mu_n(\mathbf{Path}_s^t)$ be the set of paths between s to the first occurence of t and we denote $\mu_n(s,t) = \mu_n(\mathbf{Path}_s^t)$.

We now give some properties of μ. Let Π_1 be a set of paths ending in some state s and let Π_2 be a set of paths starting from s. We denote by $\Pi_1 \cdot \Pi_2$ the set of paths $\rho_1\rho_2$ with $\rho_1 \in \Pi_1$ and $\rho_2 \in \Pi_2$.

Proposition 1. *For all n, we have* $\mu_n(\Pi_1 \cdot \Pi_2) = \sum_{i=0}^{n} \binom{n}{i} \mu_i(\Pi_1) \cdot \mu_{n-i}(\Pi_2)$

This property hints to a set of semirings $(\mathbb{R}, \oplus_n, \otimes_n, \overline{0}_n, \overline{1}_n)$ with good properties to compute moments. For $(n+1)$-tuples $(x_0, \ldots, x_n)$ and $(y_0, \ldots, y_n)$, we define operations $\oplus_n$ and $\otimes_n$:

- $(x_0, \ldots, x_n) \oplus_n (y_0, \ldots, y_n) = (x_0 + y_0, \ldots, x_n + y_n)$
- $(x_0, \ldots, x_n) \otimes_n (y_0, \ldots, y_n) = (z_0, \ldots, z_n)$ with $z_i = \sum_{j=0}^{i} \binom{i}{j} x_j y_{i-j}$

The neutral element for $\oplus_n$ is $\overline{0}_n = (0, \ldots, 0)$. $\overline{0}_n$ is an annihilator for $\otimes_n$. The neutral element for $\otimes_n$ is $\overline{1}_n = (1, 0, \ldots, 0)$. In the following, we will denote the different laws and elements by $\oplus$, $\otimes$, $\overline{0}$ and $\overline{1}$.

Proposition 2. *For $n \geq 0$, $(\mathbb{R}^{n+1}, \oplus, \otimes, \overline{0}, \overline{1})$ defines a commutative semiring.*

Notice that if for all $i \leq n$, we have $x_i = \mu_i(\Pi_1)$ and $y_i = \mu_i(\Pi_2)$, denoting $(z_0, \ldots, z_n) = (x_0, \ldots, x_n) \otimes_n (y_0, \ldots, y_n)$, we get $\mu_i(\Pi_1 \cdot \Pi_2) = z_i$. Further, if both Π_1, Π_2 are disjoint, and if no path of Π_1 (resp. Π_2) is a prefix of a path of Π_2 (resp. Π_1), then $\mu_i(\Pi_1 \cup \Pi_2) = x_i + y_i$.

3.2 Computations in a semiring

Following the Floyd-Warshall algorithm to sum weights of paths reaching a state, we will decompose inductively $\mathbf{Path}_q^R$ using operations $\cup$ and $\cdot$. We will then use the semiring $(\mathbb{R}^{n+1}, \oplus, \otimes, \overline{0}, \overline{1})$to perform these computations inductively. The induction will be over the number of states in S. Let G be a subset of S disjoint with R: $G \cap R = \emptyset$. For all state $s \in S \setminus R$, we define $\mathbf{Path}_s^t(G) = \{s_0 \cdots s_n \mid s_0 = s, s_n = t, \forall 1 \leq i \leq n-1, s_i \in G\}$ the set of paths from state s to state t using only states G, except for the initial state, which is s and for the last state which is t, even if $s, t \in R$ or $s, t \notin G$.

For a set of paths Π, we define $w_n(\Pi) = (\mathbb{P}(\Pi), \mu_1(\Pi), \ldots, \mu_n(\Pi))$. Let $g \in G$ be a state of G. A path ρ in $\mathbf{Path}_s^t(G)$ has two possibilities: either it does not use g, or it uses g one or several times. We deduce the inductive formula:

Proposition 3. $w_n(\boldsymbol{Path}_s^t(G)) = \quad w_n(\boldsymbol{Path}_s^t(G \setminus \{g\})) \oplus$
$w_n(\boldsymbol{Path}_s^g(G \setminus \{g\})) \otimes \big(\bigoplus_{k=1}^{\infty} w_n(\boldsymbol{Path}_g^g(G \setminus \{g\}))^{\otimes k}\big) \otimes w_n(\boldsymbol{Path}_g^t(G \setminus \{g\}))$

Proof (Sketch of). If ρ does not use g, we have ρ is in $\mathbf{Path}_s^t(G \setminus \{g\})$. Otherwise, ρ can be expressed as $\rho_0 \ldots \rho_k$ with:

- ρ_0 is in $\mathbf{Path}_s^g(G \setminus \{g\})$,
- ρ_k is in $\mathbf{Path}_g^t(G \setminus \{g\})$,
- and for all $0 < j < k$, $\rho_j \in \mathbf{Path}_g^g(G \setminus \{g\})$.

We can then write an inductive formula satisfied by $\mathbf{Path}_s^t(G)$:

$$\mathbf{Path}_s^t(\emptyset) = \{(s, a, t) \mid Pr(s, a, t) \neq 0\}$$

$$\mathbf{Path}_s^t(G) = \mathbf{Path}_s^t(G \setminus \{g\}) \cup \bigcup_{k=1}^{\infty} \{\rho_0 ... \rho_k \mid \rho_0 \in \mathbf{Path}_s^g(G \setminus \{g\}), \rho_k \in \mathbf{Path}_g^t(G \setminus \{g\}), \forall j \in [1, k-1], \rho_j \in \mathbf{Path}_g^g(G \setminus \{g\})\}$$

□

In order to use this formula, we need to compute $\bigoplus_{k=1}^{\infty} w_n(\mathbf{Path}_g^g(G \setminus \{g\}))^{\otimes k} = w_n(\mathbf{Path}_g^g(G))$, which represents what happens along a cycle from g to g. Let (g, Π) a pair with g a state and Π a set of paths (cycles) using g exactly twice: the first state and the last states are g. The pair $(g, \mathbf{Path}_g^g(G \setminus \{g\}))$ satisfies this property. We define $w_n^*(\Pi) = \bigoplus_{k=1}^{\infty} w_n(\Pi)^{\otimes k}$. The restriction on (r, Π) ensures that $\bigcup_{k=1}^{\infty} \Pi^{\otimes k}$ is disjoint. We show that $w_n^*(\Pi)$ is defined in most cases, namely when $\mathbb{P}(\Pi) < 1$.

Proposition 4. *Let Π be a set of paths using state g exactly twice, as first and last state. If $\mathbb{P}(\Pi) < 1$, then*

$$w_n^*(\Pi)[0] = w_0^*(\Pi) = \mathbb{P}(\bigcup_{k=1}^{\infty} \Pi^{\otimes k}) = \frac{1}{1 - \mathbb{P}(\Pi)}, \text{ and for } i > 0$$

$$w_n^*(\Pi)[i] = \mu_i(\bigcup_{k=1}^{\infty} \Pi^{\otimes k}) = \frac{1}{1-\mathbb{P}(\Pi)} \sum_{j=0}^{i-1} \binom{i}{j} w_n(\Pi)[i-j] \times w_n^*(\Pi)[j]$$

Notice that $P(\Pi) = 1$ describes cases where s cannot reach t (as $t \notin G$, if $\mathbb{P}(w_n(\mathbf{Path}_g^g(G)) = 1$, it would mean that every path reaching g stays in G forever, and in particular never meets t). Thus, we first compute the set of states S_1 from which there exists a path to R. Notice that for each set Π of paths ending in $g \in S_1 \setminus R$, we have $\mathbb{P}(\Pi) < 1$, because there is a positive probability to reach R from g, which is not captured by paths in Π.

3.3 A symbolic algorithm

From the inductive formulae to compute set of paths from subsets of paths and to compute $w_n^*(\Pi)[i]$ from $w_n^*(\Pi)[j]$ for $j < i$, we deduce Algorithm 1, following the ideas of Floyd-Warshall, incrementally adding non response states from $S_1 \setminus R$, which can be used as intermediate states. Notice that states in $S \setminus S_1$ cannot reach R anyway. This algorithm is *symbolic* (or *algebraic*) in that every constant (e.g. $Pr(s, a, t)$) can be replaced by a variable (see e.g. Section 4.2).

Theorem 1. *Let $A = (S, \delta, \delta_0)$ be a probabilistic automaton. One can compute $\mu_i(s, t)$ for all $i \leq n$ and $s, t \in S$ in time $O(n^2 \times |S|^3)$.*

Proof. In Algorithm 1, after running the outer **for**-loop on $g_1, \ldots, g_j$, we have $w_n(s,t)[n] = \mu_n(\mathbf{Path}_s^t(\{g_1, \ldots, g_j\}))$. At the end of Algorithm 1, we obtain $w_n(s,t)[n] = \mu_n(\mathbf{Path}_s^t) = \mu_n(s,t)$.

To obtain $\mu_i(s,t)$ for all $i \leq n$, it suffices to run Algorithm 1 inductively on moment of order $1, \ldots, n$. Computing $w_n^*[i](s,t)$ in the inner **for**-loop takes time $O(i)$ as $w_n[j](s,t) = w_j[j](s,t)$ has already been computed inductively for all $j < i$. This yields the complexity of $O(\sum_{j=1}^{n} i \times |S|^3) = O(n^2 \times |S|^3)$. □

Algorithm 1: Algorithm computing the moment of order n

```
for s ∈ S do
    for t ∈ S do
        %Initialization
        w := Σ_{a∈Σ} Pr(s, a, t)
        w_n(s, t) := (w, w, ..., w)
    end
end
for g ∈ S_1 \ R do
    for s ∈ S do
        for t ∈ S do
            w_n(s, t) := w_n(s, t) ⊕ w_n(s, g) ⊗ w_n^*(g, g) ⊗ w_n(g, t)
        end
    end
end
```

Now, for each query q, we have $\mu_i(\mathbf{Path}_q^R) = \sum_{r \in R} \mu_i(q, r)$, as $\mathbf{Path}_q^{r_1}$ and $\mathbf{Path}_q^{r_2}$ have no path prefix of each other for $r_1 \neq r_2, r_1, r_2 \in R$. Now, the moment of order n of the distribution of response times of q is formally either ∞ if $\mu_0(\mathbf{Path}_q^R) < 1$ (there is positive probability to never answer q, that is have infinite response time), and $\mu_n(\mathbf{Path}_q^R)$ otherwise.

Example 2. For the example of figure 1, unfolding the algorithm for $n = 2$ (that is for probability, and moments of order 1 and 2) gives after initialization:

$w(1,2) = (1,1,1)$, $w(2,2) = (0.9, 0.9, 0.9)$, $w(2,3) = (0.1, 0.1, 0.1)$, and $w(1,3) = (0,0,0)$, as there is no direct transition from state 1 to state 3.

There are no paths with intermediary states 1 or 3, so $g = 1$ or $g = 3$ does not have any impact. For paths with intermediary states $g = 2$, the algorithm gives:

- $w(2,2) \leftarrow w(2,2) \oplus w(2,2) \otimes w(2,2)^* \otimes w(2,2) = w(2,2) \otimes w(2,2)^*$
- $w(2,3) \leftarrow w(2,3) \oplus w(2,2) \otimes w(2,2)^* \otimes w(2,3) = w(2,3) \otimes w(2,2)^*$
- $w(1,3) \leftarrow w(1,3) \oplus w(1,2) \otimes w(2,2)^* \otimes w(2,3)$

We have $w(2,2)^* = (\frac{1}{1-0.9}, \frac{0.9}{(1-0.9)^2}, \frac{0.9}{(1-0.9)^2} + \frac{2 \times 0.9^2}{(1-0.9)^3}) = (10, 90, 1710)$

At the end of the algorithm, we obtain $\mu_i(2,3) = \mu_i(\mathbf{Path}_2^{\{2\}}) = w(2,3) = (0.1, 0.1, 0.1) \otimes (10, 90, 1710) = (1, 10, 190)$. Hence, in this probabilistic automata, the probability of responding to the query is 1, in a mean time of 10, with a standard deviation of $\sqrt{190 - 10^2} = 9.5$.

3.4 Extension to continuous time

We now extend the symbolic computation of moments to *continuous time Markov Chains (CTMCs)*. In order to be as close as possible to the setting of probabilistic automata, we use the sojourn time representation of CTMCs. This representation is fully equivalent with the more usual representation of CTMCs with transition rates, see chapter 7.3 of [9].

Definition 3. *A CTMC is a tuple $(S, Pr, \delta_0, (\lambda_s)_{s \in S})$ with:*

- *(S, Pr, δ_0) is a probabilistic automata, and*
- *for all s, λ_s is the sojourn parameter associated with state s. That is, the PDF function of the sojourn time is $X_s(t) = \lambda_s e^{-\lambda_s \cdot t}$ and the probability to stay in s at least t units of time is $e^{-\lambda_s \cdot t}$.*

In this continuous context, we need integrals instead of sums to define the i-th moment of a variable X: $\mu_i(X) = \int_0^\infty X(t) t^i dt = 1$. For every state $s \in S$, let $X_s(t) = \lambda_s e^{-\lambda_s \cdot t}$. For all i, for all s, $\mu_i(X_s)$ is well defined and $\mu_i(X_s) = \frac{i!}{\lambda_s^i}$

We can easily extend the computation of moments for CTMCs. The inductive formulas for probabilities and moments of the reaching time distribution remain unchanged. We only need to change the definition of moments for every transition, which is input at the initialization phase of the algorithm 1: for all $s, t \in S$, we set $w_n(s,t)$ to be $(w^0(s,t), w^1(s,t), \ldots, w^n(s,t))$, where $w^0(s,t) = \sum_{a \in \Sigma} Pr(s,a,t)$ and $w^i(s,t) = \sum_{a \in \Sigma} Pr(s,a,t) \frac{i!}{\lambda_s^i}$ for all $i \in [1,n]$.

Theorem 2. *Let $A = (S, Pr, \delta_0, (\lambda_s)_{s\in S})$ be a CTMC. One can compute $\mu_i(s,t)$ for all $i \leq n$ and $s,t \in S$ in time $O(n^2 \times |S|^3)$.*

4 Uniqueness of distribution, parameters and hierarchy

In this section, we present cases where having a symbolic algorithm allows efficient techniques, compared to numerical methods. We start with hierarchical systems which are a way to compactly describe systems. Then, we present the possibility to work on systems with parameters. Finally, thanks to the symbolic expression of moments, we prove that there is a unique distribution having the moments of a distribution of reaching times of a (continuous-time) Markov chain.

4.1 Hierarchical Probabilistic Automata

We use notations mainly from [3] to describe hierarchical structures:

Definition 4. *A hierarchical probabilistic automaton (HPA) A over a finite alphabet Σ is a tuple of n modules $(S_i, Pr_i, \lambda_i, s_i^0, s_i^f)_{1\leq i\leq n}$ where for all i,*

- *S_i is the finite set of states of module i,*
- *$s_i^0 \in S_i$ is the initial state of module i, and s_i^f the finial state of module i,*
- *$Pr_i : S_i \setminus \{s_i^f\} \times \Sigma \times S_i \to [0,1]$ is a stochastic transition function such that for all $s \in S_i \setminus \{s_f\}$ (resp. $s \in S_1$ for $i = 1$), $\sum_{a\in\Sigma, t\in S_i} Pr_i(s,a,t) = 1$,*
- *$\lambda_i : S_i \to \{i+1, \ldots, n\}$ is a* partial *mapping associating some states of S_i from module i to deeper modules.*

Intuitively, the system starts in module 1, in state s_1^0. Each time a state $s \in S_i$ associated with a module $j > i$, that is $\lambda_i(s) = j$, is entered by a transition $t \to s$, the system goes to state s_j^0 and stays in S_j till s_j^f is seen, in which case it comes back to state s and takes a transition $s \to t'$ (according to the probability distribution from s). This process can be repeated from any state in a module i to any module j as long as $j > i$.

To define the semantics of $(S_i, Pr_i, \lambda_i, s_i^0, s_i^f)_{1\leq i\leq n}$ formally, we inductively replace states associated with the deepest module by their definition. Indeed, nodes from the deepest module are not associated with any module by definition. Once every module has been replaced, a (flat) probabilistic automaton is obtained with the intended semantics.

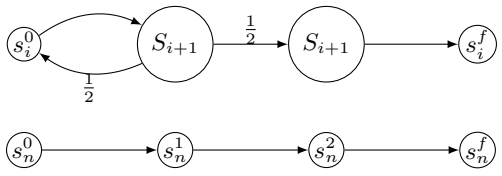

Fig. 2. An HPA with an exponential number of states.

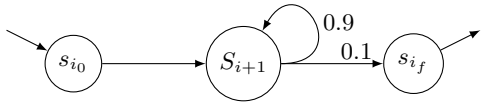

Fig. 3. An HPA without redundancy

Hence, HPA have the same expressive power as probabilistic automata. Yet, they may be much more compact: we denote by $|A|$ the size of the description of the hierarchical automaton and by $\|A\|$ the size of the unfolded automaton. The interest of such a description is that it may be exponentially smaller than the size of the unfolded automaton, as depicted in figure 2: here, every module contains two copies of the next module, with the exception of the last one. While the number of states in the description is linear ($4n$), the number of states in the unfolded automaton is equal to $3 \cdot 2^n - 2$.

The symbolic Algorithm 1 is naturally modular, in that computations on a module used several times can be performed only once by considering states of the deepest module first. Indeed, one module can be summarized by three information items: the probability (and moments) to answer the query in this module, the probability (and moments) to leave this module without answering the query in the module and the probability to stay forever in this module without answering the query. Then the information can be used for shallower modules: every time a state s in a module i is associated with the deepest module, it can be replaced by this small set of states containing all the relevant information about the deepest module (and computed only once). Then, this process can be repeated to eliminate modules recursively. This leads to a complexity in the small size $|A|$ of the compact HPA representation rather than in the large size $||A||$ of the unfolded PA:

Theorem 3. *Let A be an HPA with k modules of size at most m. The n first moments of the distribution associated with A can be computed in time $O(n^2km^3)$.*

Not only does Theorem 3 reduces the complexity for hierarchical representations with redundancy ($O(n^2k)$ for the example in Fig. 2 instead of $O(n^2 2^{3k})$ when running the algorithm in [13] on the equivalent flat PA), it also gives a better complexity on structure without redundancy. Consider the example in figure 3, without redundancy, with an unfolded PA with $3k+1$ states. Theorem 3 takes time $O(n^2k3^3)$, while the algorithm in [13] on the equivalent flat PA would take time $O(n^2(3k)^3)$.

4.2 Parametric systems

Another case where having a symbolic algorithm is helpful is when the system has parameters standing for probability values (see for instance Fig. 4, where p is such a parameter). We illustrate two cases here.

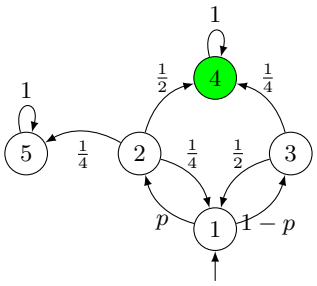

Fig. 4. Example of a parametric system with set of parameters $\{p\}$

The first case is when parameters help with redundancy. Often, stochastic systems reuse the same constructions, but with different probability values. This would be naturally encoded as a module M of a hierarchical system using a set of parameters P. This module M would be used several times, with different values of parameters specified in each module using it.

In this case, one can run Algorithm 1 on M, using the parameter values literally in the equations. This yields rational functions $f_n : [0,1]^P \to (0,1]$ of the parameters expressing the moments of order n for module M, for all n. For instance with the example of Fig. 4, the probability to reach state 4 from state 1 is equal to $\frac{2p+4}{5p+4}$, and the mean time is equal to $\frac{112+44p-12p^2}{(5p+4)(2p+4)}$. Each time module M is used, f_n can be evaluated using the value of the parameters P for this particular usage.

Another possible use of parameters is to model uncertainty of values. In the example of Fig. 4, we may not know exactly the value of parameter p, but only know that it is above 0.8. In this case, one may be interested of synthesizing the largest (resp. smallest) moment of order n which is smaller (resp. larger) than the moment of any system realizing the parametric system, that is where p is replaced by any value above 0.8. This will be particularly interesting in the next section discussing bounds. To do so, one can use the rational function f_n to compute its minimal and maximal values (e.g. deriving it and looking for 0 with Euler's method). In this way, we also obtain the best/worst value for p.

4.3 Uniqueness of the distribution

Last, we use the symbolic expression of moments obtained in Section 3 in order to prove the uniqueness of the distribution having moments of first passage times of (continuous-time) Markov chains. Thus this distribution is the distribution of response times of the system considered.

Notice that in general, there may be several distributions that correspond to a given sequence of moments $(\mu_n)_{n\in\mathbb{N}}$. This would compromise approximating the distribution using moments, as there would not be a unique such distribution.

Example 3. Let us consider a distribution δ on $\mathbb{R}^+$. If δ has the sequence of moments $\{\mu_n = n! \mid n \in \mathbb{N}\}$, then δ is the exponential distribution with parameter 1. Similarly, the sequence of moments $\{\mu_n = (2n)! \mid n \in \mathbb{N}\}$ for a distribution on $\mathbb{R}^+$ is characteristic of the square of the exponential distribution of parameter 1.

Now, consider the cube of the exponential distribution of parameter 1. Its sequence of moments is $\{\mu_n = (3n!) \mid n \in \mathbb{N}\}$. However, there exist an infinite number of distributions with this sequence of moments [18]

We now prove answer positevely to the Stieljes moment problem for the case of the distribution of response time in a (continuous-time) Markov chain, that is its sequence of moments respects the Carleman's condition from year 1922, that guarantees the uniqueness of the distribution. The condition is that $\sum_{n\in\mathbb{N}} \mu_n(\delta)^{-\frac{1}{2n}} = \infty$.

Theorem 4. *Let A be a probabilistic automaton or a CTMC. For all $n \in \mathbb{N}$, let μ_n be the moment of order n of the times of first passage in a set of state R of A. Then there exists a unique distribution δ such that $\mu_n(\delta) = \mu_n$ for all $n \in \mathbb{N}$.*

Sketch of proof: We first consider CTMC where all states have the same sojourn time λ. Then, a path that uses i transitions to answer a query will follow the gamma distribution with parameters (i, λ). We have a symbolic expression for moments of this distribution thanks to Section 3. This can be used to minimize $\sum_{n=0}^{\infty} \mu_n(\delta)^{-\frac{1}{2n}}$ by a diverging sum.

For general CTMCs, we use the fact that $\mathbb{E}(\Gamma(i, \lambda_1)^n) \leq \mathbb{E}((E(\lambda_1) + \cdots + E(\lambda_i))^n)$ iff $\lambda_1 = min(\lambda_j)_{j=1}^{i}$. It allows us to minimize the Carleman's sum of the CTMC considered by the Carleman's sum of the CTMC where all sojourn times are replaced by the smallest sojourn time λ, hence the divergence.

The case of probabilistic automaton is simpler. □

We show how this theorem allows to approximate distribution δ in the next subsection.

4.4 A sequence of distributions converging towards δ

Since we have unicity of the distribution corresponding to the sequence of moments of the distribution of response time of a probabilistic automaton, we obtain the following convergence in law:

Proposition 5. *[17] Let δ be the distribution of response times of a probabilistic automaton. Let $(\delta_i)_{i\in\mathbb{N}}$ be a sequence of distributions on $\mathbb{R}^+$ such that for all n, $\lim_{i\to\infty} \mu_n(\delta_i) = \mu_n(\delta)$. Then, if C_i is the cumulative distribution function of δ_i and C the cumulative distribution function of δ, then for all x $\lim_{i\to\infty} C_i(x) = C(x)$.*

Thus, C can be approximated by taking a sequence $(\delta_n)_{n\in\mathbb{N}}$ of distribution such that for all $i \leq n$, $\mu_i(\delta_n) = \mu_i(\delta)$. A reasonable choice for δ_n is to consider the distribution of maximal entropy corresponding to the moments $\mu_1, \ldots, \mu_n$, as presented in [11]. The distribution of maximal entropy can be understood as the

distribution that assume the least information. It can be approximated as close as desired, for instance $\frac{1}{n}$ close to the distribution of maximal entropy having moments $(\mu_1(\delta), \ldots, \mu_n(\delta))$. Applying Prop. 5, we thus obtain that the cumulative distribution function associated with δ_i converges towards the cumulative distribution function associated with δ.

5 Bounding the response time

We now explain how to use moments in order to obtain optimal bounds on the response time. First, notice that as soon as there exists a loop between a query and a response (as in Fig.1), then there will be runs with arbitrarily long response times, although there might be probability 1 to eventually answer every query (which is the case for Fig.1). We thus turn to a more quantitative evaluation of the response time.

Let $0 < p < 1$. We are interested in a bound T on the delay between a query and a response such that more than $1 - p$ of the queries are answered before this bound. For a distribution $\delta : \mathbb{R}^+ \to \mathbb{R}^+$ of response times, we denote by $B(\delta, p)$ the lowest T such that the probability to have a response time above T is lower than p. Equivalently, we look for the highest T such that the probability of a response time above T is at least p.

We place ourselves in the general setting of continuous distributions, where Dirac delta functions are allowed for simplicity. Discrete distributions form a special case, with delta functions at integer values. One could get rid of Dirac delta functions by ϵ-approximating them without changing the moments, obtaining the same bounds as we prove here.

5.1 Tchebychev bounds associated with one moment

Let $i \in \mathbb{N}$ and $\mu_i > 0$. We let Δ_{i,μ_i} be the set of distributions of response time which have μ_i as moment of order i. We are interested in bounding $B(\delta, p)$ for all $\delta \in \Delta_{i,\mu_i}$, that is for all distributions with μ_i as moment of order i. Such a bound is provided by *Tchebychev inequality*, and it is optimal:

Proposition 6. *Let $i \in \mathbb{N}$ and μ_i. Let $\alpha_i(\mu_i, p) = \sqrt[i]{\frac{\mu_i}{p}}$. Then for all $\delta \in \Delta_{i,\mu_i}$, we have $B(\delta, p) \leq \alpha_i(\mu_i, p)$. Further, $\exists \delta \in \Delta_{i,\mu_i}$ such that $B(\delta, p) = \alpha_i(\mu_i, p)$.*

Proof. It suffices to remark that $\mu_i > pb^i$ for b the bound we want to reach. Further, this bound is trivially optimal: it suffices to consider a distribution with a Dirac of mass $(1 - p)$ at 0 and a Dirac of mass p at $\alpha_i(\mu_i, p)$. □

Given a probabilistic automaton, let δ be its associated distribution of response time. We can compute its associated moments μ_i using Algorithm 1, described in the previous section. We thus know that $\delta \in \Delta_{i,\mu_i}$. Given different values of i, one can compute the different moments and apply for each of the Tchebychev bound and use the minimal bound obtained.

Understanding the relationship between the α_i is thus important. For $i < j$, one can use Jensen's inequality for the convex function $f : x \to x^{\frac{j}{i}}$ over $\mathbb{R}^+$, and obtain: $(\mu_i)^j \leq (\mu_j)^i$. For instance, $\mu_1^2 < \mu_2$.

For $p = 1$, this gives $\alpha_i(p = 1) < \alpha_j(p = 1)$. On the other hand, for p sufficiently close to 0, we have $\alpha_j(p) < \alpha_i(p)$. That is, when p is very small, moments of high orders will give better bounds than moments of lower order. On the other hand, if p is not that small, moments of small order will suffice.

5.2 Optimal bounds for a pair of moments

We now explain how to extend Tchebychev bounds to pairs of moments: We consider the set of distributions where two moments are fixed. Let $i < j$ be two orders of moments and $\mu_i, \mu_j > 0$. We denote by $\Delta_{i,\mu_i}^{j,\mu_j}$ the set of distributions with μ_i, μ_j as moments of order i, j respectively. As $\Delta_{i,\mu_i}^{j,\mu_j}$ is strictly included into Δ_{i,μ_i} and in Δ_{j,μ_j}, $\min(\alpha_i(p), \alpha_j(p))$ is a bound for any $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$. However, it may be the case that $\min(\alpha_i(p), \alpha_j(p))$ is not optimal. We now provide *optimal* bounds $\alpha_i^j(p)$ for any pair $i < j$ of order of moments and probability p:

Theorem 5. *Let $i < j$ be natural integers, $p \in (0, 1)$, and let $\mu_i, \mu_j > 0$. Let $\alpha_i = (\frac{\mu_i}{p})^{\frac{1}{i}}$ and $\alpha_j = (\frac{\mu_j}{p})^{\frac{1}{j}}$. We define $\alpha_i^j(p)$ to be:*

- *α_i if $\alpha_i \leq \alpha_j$,*
- *$(\frac{\mu_j - M}{p})^{\frac{1}{j}}$ otherwise, where $0 \leq M \leq \mu_j$ is the smallest positive real root of:*

$$\mu_i = (1-p)^{\frac{j-i}{j}} M^{\frac{i}{j}} + p^{\frac{j-i}{j}} (\mu_j - M)^{\frac{i}{j}}.$$

For all $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$, we have $B(\delta, p) \leq \alpha_i^j$, and $\exists \delta \in \Delta_{i,\mu_i}^{j,\mu_j}$ with $B(\delta, p) = \alpha_i^j$

To obtain a value for M, one can use for instance Newton's method. For $i = 1, j = 2$, we can compute explicitly M and obtain:

$$\alpha_1^2 = \mu_1 + \sqrt{\frac{(1-p)}{p}(\mu_2 - \mu_1^2)}.$$

Example 4. Consider the distribution associated with the system of Fig.1. We obtain the following bounds $\alpha_i(p), \alpha_i^{i-1}(p)$ considering different values of p and i:

i	μ_i	$\alpha_i(0.1)$	$\alpha_i^{i-1}(0.1)$	$\alpha_i(0.01)$	$\alpha_i^{i-1}(0.01)$
1	10	100	100	1000	1000
2	190	43.6	38.5	137.8	104.9
3	5410	37.8	**36.8**	81.5	73.9
4	205390	37.9	37.8	67.4	63.8
5	9747010	39.6	37.9	64.2	**61.43**
6	555066190	42.1	39.6	62.8	61.47

For $p = 0.1$, it is not useful to consider moments of order higher than 3. On the other hand, for $p = 0.01$, the moment of order 5 provides better bounds than moment of lower orders.

For hierarchical systems, one can compute moments in an efficient way using Theorem 3, and then use Theorem 5 to obtain the associated optimal bounds. In order to handle parametric systems, we use the following result which allows to underapproximate the value of M, and thus overapproximate the optimal bound, by iterating the following operator f from $x = 0$:

$$f : x \mapsto \frac{(\mu_i - [\mu_j - x]^{\frac{i}{j}} p^{\frac{j-i}{j}})^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}}$$

Lemma 1. *$(f^n(0))_{n\in\mathbb{N}}$ is strictly increasing and converges towards M.*

We show how to ϵ-approximate the *optimal* bound B of a *parametric* probabilistic automaton A with set of parameters P, that is such that for all $val \in V^P$, the probabilistic automaton A with valuation val for parameter values has a bound $b(val) \leq B$ and there exists a $val \in V^P$ such that $b(val) = B$. First, we obtain the moments as symbolic functions of the parameters using Section 4.2. Then, we compute $M_1 = f(0)$ as a function of the parameters, using Lemma 1 and replacing μ_i, μ_j by their expression. One can then compute the minimum m_1 of function M_1 over all the parameters. We then proceed with $M_2 = f(m_1)$, and so on till obtaining a value m. This allows to obtain a lower bound m over values of M for all parameter values. Computing the largest μ_j over all parameters allows to obtain an upper bound B_{up}: $B \leq B_{up} = (\frac{\mu_j - m}{p})^{\frac{1}{j}}$. A lower bound B_{lw} is easily obtained by considering the value $\geq m$ of M for the parameters maximizing μ_j. If the distance between B_{up} and B_{lw} is larger than ϵ, one can partition the space of parameter values in zones and proceed in the same way on each zone, forgetting zones for which B_{up} is lower than the B_{lw} of another zone, till the distance between $\max(B_{lw})$ and $\max(B_{up})$ over zones is smaller than ϵ.

6 Conclusion

In this paper, we have shown how to compute moments symbolically for probabilistic automata and CTMCs, using adequately defined semirings. This method has the same complexity as the efficient numerical methods already known [13]. The proof of this symbolic computation allows proving that there is a unique distribution of response time corresponding to a probabilistic automaton or a CTMC. This allows obtaining simple approximated distributions scheme converging in law towards the distribution of response time. The symbolic computation of moments also allows computing moments in compact (hierarchical) models faster, as well as finding lowest/highest value of moments in parametric systems.

We also provide optimal bounds on the delay after which very few queries stay unanswered. It is optimal when considering distribution displaying a given pair of moments, and we showed on a simple example how this improves Tchebychev bounds. This can be used efficiently to obtain bounds for compact (hierarchical) models or to compute an optimal bound which fulfills the response of almost all queries even for systems where some parameter values are not known exactly.

References

1. E. Asarin, N. Basset, A. Degorre. Entropy of regular timed languages. In *Information and Computation 241*, p.142-176, Elsevier, 2015.
2. R. Angrish, S. Chakraborty. Probabilistic Timing Analysis of Asynchronous Systems with Moments of Delay. ASYNC'02, IEEE, 2002.
3. R. Alur. Formal analysis of hierarchical state machines. In *Verification: Theory and Practice*, p.42-66, 2002.
4. M. Backenköhler, L. Bortolussi, V. Wolf. Generalized Method of Moments for Stochastic Reaction Networks in Equilibrium. CMSB'16, LNCS 9859, 2016.
5. H. Bazille, E. Fabre, B. Genest. Symbolically Quantifying Response Time in Stochastic Models using Moments and Semirings. `https://perso.crans.org/~genest/BFG18.pdf`
6. J. Bradley, N. Dingle, U. Harder, P. Harrison, W. Knottenbelt. Response Time Densities and Quantiles in Large Markov and Semi-Markov Models. In *Performance Evaluation of Parallel, Distributed and Emergent Systems 1*, 2006.
7. S. Bogomolov, T. Henzinger, A. Podelski, J. Ruess, C. Schilling. Adaptive Moment Closure for Parameter Inference of Biochemical Reaction Networks. CMSB'15, LNCS 9308, 2015.
8. K.Charterjee, T. Henzinger, F. Horn. The complexity of request-response games. LATA'11, LNCS 6638, 2011.
9. C. Cassandras, S. Lafortune. *Introduction to Discrete Event Systems*. Springer, 2007.
10. C. Cortes, M. Mohri, A. Rastogi, M. Riley. On the computation of the Relative Entropy of Probabilistic Automata. In *International Journal of Foundations of Computer Science (IJFCS)*, p.219-242 , 2006.
11. T. Cover, J. Thomas. Elements of Information Theory. Wiley, 2006.
12. C. Daws. Symbolic and Parametric Model Checking of Discrete-Time Markov Chains. ICTACT'04, LNCS 3407, p. 280-294, 2004.
13. T. Dayar, N. Akar. Computing moments of first passage times to a subset of states in Markov chains. In *SIAM Journal on Matrix Analysis and Applications*, p.396-412, 2005.
14. A.M. Gonzalez, J. Uhlendorf, J. Schaul, E. Cinquemani, G. Batt, G. Ferrari-Trecate. Identification of biological models from single-cell data: a comparison between mixed-effects and moment-based inference. ECC'13, IEEE, p.3652-3657, 2013.
15. F. Horn, W. Thomas, N. Wallmeier, M. Zimmerman. Optimal Strategy Synthesis for Request-Response Games. In *RAIRO* 49(3), p.179-203, 2015.
16. N. Jansen, F. Corzilius, M. Volk, R. Wimmer, E. Abrahm, J.-P. Katoen, B. Becker. Accelerating Parametric Probabilistic Verification. In *QEST'14*, LNCS 8657, p. 404-420, 2014.
17. Y. Prohorov, Y. Rozanov. *Probability Theory*, Basic Concepts · Limit Theorems Random Processes. Translated from Russian, Springer, 1969.
18. J. Stoyanov. Determinacy of Distributions by Their Moments. ICMSM'06, 2006.
19. Á. Tari. Moments based bounds in stochastic models, PhD Thesis. Budapesti Műszaki és Gazdaságtudományi Egyetem, 2005.
20. M. Telek, G. Horvéth. A minimal representation of Markov arrival processes and a moments matching method. In *Performance Evaluation*, p.1153-1168, 2007.
21. N. Wallmeier, P. Hütten, W. Thomas. Symbolic Synthesis of Finite-State Controllers for Request-Response Specifications. CIAA'03, 2003.
22. D. Yao. First-passage-time moments of Markov processes. In *Journal of Applied Probability*, p.939-945, 1985.

Appendix

Proofs for Section 3

Proposition 1. *For all n, we have $\mu_n(\Pi_1 \cdot \Pi_2) = \sum_{i=0}^{n} \binom{n}{i} \mu_i(\Pi_1) \cdot \mu_{n-i}(\Pi_2)$*

Proof.
$$\begin{aligned}
\mu_n(\Pi_1.\Pi_2) &= \sum_{\pi_1 \in \Pi_1} \sum_{\pi_2 \in \Pi_2} \mathbb{P}(\pi_1\pi_2)|\pi_1\pi_2|^n \\
&= \sum_{\pi_1 \in \Pi_1} \sum_{\pi_2 \in \Pi_2} \mathbb{P}(\pi_1)\mathbb{P}(\pi_2) \sum_{i=0}^{n} \binom{n}{i} |\pi_1|^i |\pi_1|^{n-i} \\
&= \sum_{i=0}^{n} \binom{n}{i} \sum_{\pi_1 \in \Pi_1} \mathbb{P}(\pi_1)|\pi_1|^i \sum_{\pi_2 \in \Pi_2} \mathbb{P}(\pi_2)|\pi_2|^{n-i} \\
&= \sum_{i=0}^{n} \binom{n}{i} \mu_i(\Pi_1)\mu_{n-i}(\Pi_2)
\end{aligned}$$
□

Proposition 2. *For $n \geq 0$, $(\mathbb{R}^{n+1}, \oplus, \otimes, \overline{0}, \overline{1})$ defines a commutative semiring.*

Proof. It is clear that $(\mathbb{R}_+^{n+1}, \oplus, \overline{0})$ is a commutative monoid. Associativity and commutativity in $(\mathbb{R}_+^{n+1}, \otimes, \overline{1})$ come from the symmetric role of the x_i and y_i in $\otimes$. Thus, we have to prove that $\otimes$ is distributive over $\oplus$. Since $\otimes$ is commutative, we only have to prove that for all $x, y, z \in R_+^{n+1}$, $(x \otimes y) \oplus (x \otimes z) = x \otimes (y \oplus z)$. For $i \geq 0$, we check the i-th component:

$$\begin{aligned}
((x \otimes y) \oplus (x \otimes z))_i &= \sum_{j=0}^{i} \binom{j}{i} x_j y_{i-j} + \sum_{j=0}^{i} \binom{j}{i} x_j z_{i-j} \\
&= \sum_{j=0}^{i} \binom{j}{i} x_j (y_{i-j} + z_{i-j}) \\
&= \big(x \otimes (y \oplus z)\big)_i
\end{aligned}$$

which completes the proof. □

Proposition 3. $w_n(\boldsymbol{Path}_s^t(G)) = w_n(\boldsymbol{Path}_s^t(G \setminus \{g\})) \oplus$
$w_n(\boldsymbol{Path}_s^g(G \setminus \{g\})) \otimes \big(\bigoplus_{k=1}^{\infty} w_n(\boldsymbol{Path}_g^g(G \setminus \{g\}))^{\otimes k}\big) \otimes w_n(\boldsymbol{Path}_g^t(G \setminus \{g\}))$

Proof. We have the inductive formula:

$$\begin{aligned}
\mathbf{Path}_s^t(\emptyset) &= \{(s, a, t) \mid Pr(s, a, t) \neq 0\} \\
\mathbf{Path}_s^t(G) &= \mathbf{Path}_s^t(G \setminus \{g\}) \cup \bigcup_{k=1}^{\infty} \{\rho_0 ... \rho_k \mid \rho_0 \in \mathbf{Path}_s^g(G \setminus \{g\}), \\
&\qquad \rho_k \in \mathbf{Path}_g^t(G \setminus \{g\}), \forall j \in [1, k-1], \rho_j \in \mathbf{Path}_g^g(G \setminus \{g\})\}
\end{aligned}$$

Thus, we get $\forall n, \mu_n(\mathbf{Path}_s^t(G)) = \mu_n(\mathbf{Path}_s^t(G \setminus \{g\})) +$
$\sum_{k=1}^{\infty} \mu_n\big((\mathbf{Path}_s^g(G \setminus \{g\})) \cdot ((\mathbf{Path}_g^g(G \setminus \{g\})))^k \cdot (\mathbf{Path}_g^t(G \setminus \{g\}))\big)$. By using proposition 1, we can deduce that $w_n(\mathbf{Path}_s^t(G)) = w_n(\mathbf{Path}_s^t(G \setminus \{g\})) \oplus$

$\bigoplus_{k=1}^{\infty}\big(w_n(\mathbf{Path}_s^g(G\setminus\{g\}))\otimes w_n(\mathbf{Path}_g^g(G\setminus\{g\}))^{\otimes k}\otimes w_n(\mathbf{Path}_g^t(G\setminus\{g\}))\big)$.
Notice that by associativity, the second part of the sum is equal to:
$w_n(\mathbf{Path}_s^g(G\setminus\{g\}))\otimes\big(\bigoplus_{k=1}^{\infty}w_n(\mathbf{Path}_g^g(G\setminus\{g\}))^{\otimes k}\big)\otimes w_n(\mathbf{Path}_g^t(G\setminus\{g\}))$.

Proposition 4. *Let Π be a set of paths using state g exactly twice, as first and last state. If $\mathbb{P}(\Pi)<1$, then*

$$w_n^*(\Pi)[0]=\mathbb{P}(\bigcup_{k=1}^{\infty}\Pi^{\otimes k})=\frac{1}{1-\mathbb{P}(\Pi)},\ \textit{and}$$

$$w_n^*(\Pi)[i]=\mu_i(\bigcup_{k=1}^{\infty}\Pi^{\otimes k})=\frac{1}{1-\mathbb{P}(\Pi)}\sum_{j=0}^{i-1}\binom{i}{j}w_n(\Pi)[i-j]\times w_n^*(\Pi)[j]$$

Proof. For $k>0$, define $R_k=(x_0,\dots,x_n)\otimes\cdots\otimes(x_0,\dots,x_n)$ with $x_0=\mathbb{P}(\Pi)<1$ and $x_i=w_i(\Pi)$ for all i. We denote $(x_{(k,0)},\dots,x_{(k,n)})=R_k$. For $k>0$, we have $x_{(k,i)}=\sum_{j=0}^{i}\binom{i}{j}x_{(1,j)}x_{(k-1,i-j)}$. In particular, for all i, $x_{(1,i)}=x_i$.

First, let us prove by induction on i that $x_{(m,i)}\underset{m\to\infty}{\to}0$. It is true for $i=0$, as $x_0^m\to 0$ since $0\le x_0<1$. Assume by induction that it is true for all $j<i$. Then as $x_{(m,i)}=\sum_{j=0}^{i}\binom{i}{j}x_jx_{(m-1,i-j)}$ is a linear combination of the $x_{(m-1,i-j)}$, and all $x_{(m-1,i-j)}$ converges towards 0, then so does $x_{(m,i)}$ converge to 0.

Then, define $S_m=\bigoplus_{k=0}^{m}R_k$, with $R_0=\overline{1}$. We have $S_m=(z_{(m,0)},\dots,z_{(m,n)})$ with $z_{(m,i)}=\sum_{k=0}^{m}x_{(k,i)}=x_{(0,i)}+\sum_{k=1}^{m}\sum_{j=0}^{i}\binom{i}{j}x_{(1,j)}x_{(k-1,i-j)}=w_m^*(\Pi)[i]$.

Let us prove that the series $z_{m,i}$ converges by induction on i. For $i=0$, it is easy to see that $\lim_{m\to\infty}z_{m,0}=\frac{1}{(1-x_0)}$. Assume by induction that for all $j<i$ with $i>0$, the series $z_{(m,j)}$ converges. We denote its limit $x^*_{|j}$. We have:

$$\begin{aligned}z_{(m,i)}&=\sum_{k=1}^{m}\sum_{j=0}^{i}\binom{i}{j}x_{(1,j)}x_{(k-1,i-j)}\\&=\sum_{j=1}^{i}\binom{i}{j}x_{(1,j)}\sum_{k=1}^{m}x_{(k-1,i-j)}+x_{(1,0)}\sum_{k=1}^{m}x_{(k-1,i)}\end{aligned}$$

Then, $\sum_{k=0}^{m-1}x_{(k,i)}(1-x_{(1,0)})+x_{m,i}=\sum_{j=1}^{i}\binom{i}{j}x_{(1,j)}\sum_{k=1}^{m}x_{(k-1,i-j)}$. Furthermore, $x_{m,i}\underset{m\to\infty}{\to}0$ and $x_{1,0}=x_0$. Then we can conclude that

$\lim_{m\to\infty}\sum_{k=0}^{m}x_{(k,i)}=\frac{1}{(1-x_0)}\sum_{j=0}^{i-1}\binom{i}{j}x_{(i-j)}x^*_{|j}$

Hence we obtained the formulation of $w_m^*(\Pi)[i]=z_{(m,i)}$ by induction. □

Proofs of Section 4

Formal Semantics of a Hierarchical Probabilistic Automaton.

Let $A = (S_i, Pr_i, s_i^0, s_i^f)_{i \leq n}$ be a hierarchical automaton. We give now a formal semantics to A, defining the flat probabilistic automaton associated to it.

Let $r \in S_i$ such that $\lambda_i(r) = n$. We redefine module i as $(S'_i, Pr'_i, s'^0_i, s'^f_i)$ with:

- $S'_i = S_i \setminus \{r\} \cup S_n$.
- We define $Pr'_i(s, a, t)$ as follows for all $s \in S'_i \setminus s_i^f$, $t \in S'_i$ and $a \in \Sigma$:
  - for $s, t \in S_i \setminus \{r\}$, $Pr'_i(s, a, t) = Pr_i(s, a, t)$
  - for $s, t \in S_n$, $Pr'_i(s, a, t) = Pr_n(s, a, t)$
  - for $s \in S_i$ and $t = s_n^0$, we have $Pr'_i(s, a, t) = Pr_i(s, a, r)$,
  - for $r \neq s_i^f$, $s = s_n^f$ and t, we have $Pr'_i(s, a, t) = Pr_i(r, a, t)$,
  - Otherwise, $Pr'_i(s, a, t) = 0$,
- If $r = s_i^0$, then $s'^0_i = s_n^0$, else $s'^0_i = s_i^0$,
- If $r = s_i^f$, then $s'^f_i = s_n^f$, else $s'^f_i = s_i^f$,

We proceed by replacing inductively all nodes associated with the deepest module n till there is no more such node. Then we can remove module n altogether and proceed inductively with nodes associated with module $n - 1$.

Adaptation of the algorithm to the HPA model We now turn to the proof of Theorem 3.

Theorem 3. *Given an HPA A, with k modules of size at most m. Then one can compute the n first moments of the distribution associated with A in time $O(n^2 k m^3)$.*

In the model of probabilistic automata, sojourn time in each state takes exactly one time unit. We extend trivially the computation of moments to systems where the sojourn time in a state s depends on the output transition (s, a, t) and follows a distribution $\delta_{s,a,t}$: it suffices to replace in Algorithm 1 the initialization from $\sum_{a \in \Sigma} Pr(s, a, t)$ to $\sum_{a \in \Sigma}(Pr(s, a, t) \cdot \mu_n(\delta_{s,a,t}))$. Equivalently, we can specify $(\mu_i(\delta_{s,a,t}))_{i \leq n}$ on each transition.

Proof. Algorithm 1 is suitable for HPA because it can be performed be going through states module by module, deepest module first. We now show how to replace a module S_i by a set of four states $\tilde{S}_i$ carrying an equivalent information, thanks to transition characterized by probabilities and moments.

Assume that a query has been performed and not yet answered before entering a module. There are three possibilities. Firstly, it can be answered in the module. Secondly, it is not answered in the module but it the run leaves this module. The last possibility is the query is not answered and the run does not leave the module. Thus, the module can be summarized by four states,as presented in figure 5: the initial state, a final state, a self looping state representing paths

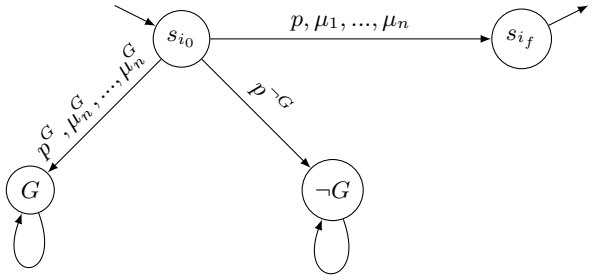

Fig. 5. Replacing a module by four states.

where the query is answered in the module, and another self looping state that represents paths where the query is neither answered nor leave the module.

We recall that in each module the final state has no transition to another state in the module. Now, we need to compute the following quantities: the probability (and moments) for reaching the goal from s_{i_0} and the ones for reaching s_{i_f} from s_{i_0} without having reached the goal before. These two quantities can be computed inductively by the algorithm 1 as presented before.

Then, the module s_i can be replaced by the new four states module in modules of higher rank. Instead of having a cubic complexity over the size of the automaton, this procedure allows us to have a complexity proportional to the sum of the cube of the sizes of the modules, which can lead to great improvement, especially in redundant systems.

□

Proofs for section 5

We now give a sequence of lemmas that will yield the proof of Theorem 5 and Lemma 1.

Let p such that $0 < p < 1$. Let μ_i, μ_j.

case $\alpha_i < \alpha_j$ We prove that in the case where $\alpha_i < \alpha_j$, α_i is actually optimal in $\Delta_{i,\mu_i}^{j,\mu_j}$. This is the first item in Theorem 5

As it is a bound for all $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$, we just need to show that it is optimal

Let $0 < \eta < 1$ and $0 < p < 1$. We let z be a positive real that we will choose big enough, bigger than α_i and actually larger, that will be set later.

Let δ be the distribution with mass $(1-p)$ in 0, mass p_1 in $\eta\alpha_i$ mass p_2 in α_j and mass p_3 in z, with $p_1 + p_2 + p_3 = p$.

We want to choose p_1, p_2, p_3 such that μ_i is the moment of order i and μ_j is the moment of order j, that is such that $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$. We thus have the following equations:

$$p_1 + p_2 + p_3 = p(1)$$
$$p_1(\eta\alpha_i)^i + p_2\alpha_j^i + p_3 z^i = \mu_i(2)$$
$$p_1(\eta\alpha_i)^j + p_2\alpha_j^j + p_3 z^j = \mu_j(3)$$

We denote $A = \alpha_i^i$, $B = \alpha_j^i$, $C = z^i$, $D = (\eta\alpha_i)^j$ and $F = z^j$.

Using (1) and (3), we obtain:

$$p_3 = (p - p_2)\frac{(\mu_j - p(\eta\alpha_i)^j)}{p(F - (\eta\alpha_i)^j)}(4)$$

Granted $p_2 < p$, for $F > (\eta\alpha_i)^j$ (that is $z > \alpha_i$ which we assumed), we get $p_3 > 0$.

Now, using (2), we obtain: $p_3(C-\eta^i A)+p_2(B-\eta^i A) = \mu_i - p\eta^i A$. As $\mu_i = pA$, we get $p_2(B - \eta^i A) + p_3(C - \eta^i A) = pA(1 - \eta^i)$.

Using equivalents for z going to ∞, we get $p_3(C - \eta^i A)$ equivalent to $(1 - p_2/p)C/F$. Notice that C/F tends to 0. We obtain $p2 = \frac{(1-\eta^i)pA-O(C/F)}{(B-\eta^i A)-O(C/F)}$. For z big enough (η being fixed), we get $p_2 > 0$.

Dividing terms by A, we get $p2 < p\frac{(1-\eta^i)}{(B/A-\eta^i)-O(C/AF)}$. We have $B/A > 1$. For z big enough, $O(C/AF) < B/A - 1$, and we get $p_2 < p$. That is $p_3 > 0$ as well.

Also, remark that in (4), we have $\frac{(\mu_j - p(\eta\alpha_i)^j)}{p(F-(\eta\alpha_i)^j)}$ tends to 0 when z tends to infinity. Hence for z big enough, $p_3 < (p - p_2)$. That is, $p_1 = p - p_2 - p_1 > 0$.

That is, for z big enough, we can chose p_1, p_2, p_3 positive and satisfying the equations we wanted to obtain. That is, $p_1, p_2, p_3 < p$ as $p = p_1 + p_2 + p_3$, and $\mu_1(\delta) = \mu_1$ and $\mu_2(\delta) = \mu_2$. Thus, $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$ Last, we have $B(\delta, p) = \eta\alpha_i$.

Case $\alpha_j < \alpha_i$ We now consider the case where $\alpha_j < \alpha_i$, that is the second item of Theorem 5. We first prove that the α_i^j defined is a bound for all $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$. We take δ any distribution with μ_i, μ_j for moment of response time of order i, j. We let $b = B(\delta, p)$. We partition δ in 2 parts: δ_1 between 0 and b (and 0 elsewhere), and δ_2 after b (and 0 before). We denote $\mu_k(\delta_\ell) = \int_0^\infty \delta_\ell(t)t^k dt$, for $\ell \in \{1, 2\}$.

As δ_2 represents a proportion p of the distribution, and as all the mass is after b, we have the following:

$$\mu_j(\delta_2) = \mu_j - \mu_j(\delta_1) \geq pb^j$$

Lemma 2.

$$\mu_j(\delta_1) \geq \frac{(\mu_i - [\mu_j - \mu_j(\delta_1)]^{\frac{i}{j}} p^{\frac{j-i}{j}})^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}}$$

Proof. We apply Jensen inequality to both δ_1 and δ_2.

We obtain $\mu_j(\delta_1) \geq \frac{\mu_i(\delta_1)^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}}$ and $\mu_i(\delta_2) \leq \mu_j(\delta_2)^{\frac{i}{j}} p^{\frac{j-i}{j}}$.

As $\mu_i(\delta_1) = \mu_i - \mu_i(\delta_2)$, we obtain $\mu_i(\delta_1) \geq \mu_i - \mu_j(\delta_2)^{\frac{i}{j}} p^{\frac{j-i}{j}} = \mu_i - [\mu_j - \mu_j(\delta_1)]^{\frac{i}{j}} p^{\frac{j-i}{j}}$, which yields the statement. □

The M of Lemma 1 and Theorem 5 will be $\mu_j(\delta_1)$ for δ a distribution realizing $B(\delta, p) = \alpha_i^j(p)$. We now prove the second part of Theorem 5 and Lemma 1.

Lemma 3. *Let μ_i, μ_j and p such that $\alpha_j(p, \mu_j) < \alpha_i(p, \mu_i)$. Then for all $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$, we have:*

$$B(\delta, p) \leq (\frac{\mu_j - M}{p})^{\frac{1}{j}}$$

for $M \leq \mu_j$ the smallest positive real root of:

$$\mu_i = (1-p)^{\frac{j-i}{j}} M^{\frac{i}{j}} + p^{\frac{j-i}{j}} (\mu_j - M)^{\frac{i}{j}}.$$

For $i = 1, j = 2$, we can compute explicitly M and obtain:

$$B(\delta, p) \leq \mu_1 + \sqrt{\frac{(1-p)}{p}(\mu_2 - \mu_1^2)}$$

Proof. Let $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$. We denote $b = B(\delta, p)$. We decompose $\delta = \delta_1 + \delta_2$ with δ_1 on $[0, b)$ and δ_2 from $[b, \infty)$.

One can define the operator f with:

$$f(x) = \frac{(\mu_i - [\mu_j - x]^{\frac{i}{j}} p^{\frac{j-i}{j}})^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}}$$

By applying Lemma 2 with $\mu_j(\delta_1) \geq 0$ on the left hand side, we obtain $\mu_j(\delta_1) \geq f(0)$. Hence we can apply Lemma 2 with $\mu_j(\delta_1) \geq f(0)$ on the left hand side, yielding $\mu_j(\delta_1) \geq f(f(0))$. By a trivial induction, we obtain $\mu_j(\delta_1) \geq f^n(0)$ for all n.

Let us show that the iteration of f converges from 0. First, for all n, we have $f^n(0)$ is bounded: $0 \leq f^n(0) \leq \mu_j(\delta_1)$. Indeed, let x be a real such that $0 \leq x \leq \mu_j(\delta_1)$. Then, we got:

$$\frac{(\mu_i - \mu_j^{\frac{i}{j}} p^{\frac{j-i}{j}})^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}} \leq f(x) \leq \frac{(\mu_i - (\mu_j - \mu_j(\delta_1))^{\frac{i}{j}} p^{\frac{j-i}{j}})^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}}$$

Since $\alpha_2 \leq \alpha_1$, we have $(\mu_i - \mu_j^{\frac{i}{j}} p^{\frac{j-i}{j}})^{\frac{j}{i}} \geq 0$.

Also, thanks to Lemma 2, we have $\mu_j(\delta_1) \geq \frac{(\mu_i - [\mu_j - \mu_j(\delta_1)]^{\frac{i}{j}} p^{\frac{j-i}{j}})^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}}$.

Thus, by a trivial induction, we obtain $0 \leq f^n(0) \leq \mu_j(\delta_1)$ for all n.

Secondly, we show by induction on n that $f^n(0)$ is an increasing sequence. We already proved the first step of the induction: $f(0) \geq 0$.

Inductive step: let $n \in \mathbb{N}$ such that $f^n(0) \geq f^{n-1}(0)$. Then,

$$\frac{(\mu_i - [\mu_j - f^n(0)]^{\frac{i}{j}} p^{\frac{j-i}{j}})^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}} \geq \frac{(\mu_i - [\mu_j - f^{n-1}(0)]^{\frac{i}{j}} (p)^{\frac{j-i}{j}})^{\frac{j}{i}}}{(1-p)^{\frac{j-i}{i}}}$$

And so, $f^{n+1}(0) \geq f^n(0)$.

Then, the sequence $(f^n(0))$ converges. Let M be its convergence point. We thus proved Lemma 1. We also have $f(M) = M$. Thus, $M \leq \mu_j$ and it is the smallest positive real root of:

$$(1-p)^{\frac{j-i}{j}} M^{\frac{i}{j}} = \mu_i - p^{\frac{j-i}{j}} (\mu_j - M)^{\frac{i}{j}}.$$

Now, we know that $\mu_j(\delta_1) \geq M$. This gives $pB(\delta,p)^j \leq \mu_j - M$.

We now tackle the last item of the statement. Let $i = 1, j = 2$. We have $B(\delta,p) \leq \sqrt{\frac{\mu_2 - M}{p}} = \frac{\mu_1 - \sqrt{1-p}\sqrt{M}}{p}$.

We let $x = \sqrt{M}$. This x satisfies the equation

$$\sqrt{1-p}x = \mu_1 - \sqrt{p}\sqrt{(\mu_2 - x^2)}.$$

That is

$$\sqrt{p}\sqrt{(\mu_2 - x^2)} = \mu_1 - \sqrt{(1-p)}x$$

and hence:

$$p\mu_2 - px^2 = \mu_1^2 + (1-p)x^2 - 2\mu_1\sqrt{(1-p)}x$$

We have the second degree equation:

$$x^2 - 2\mu_1\sqrt{(1-p)}x + \mu_1^2 - p\mu_2 = 0$$

The smallest solution is $x = \mu_1\sqrt{1-p} - \sqrt{(1-p)\mu_1^2 + p\mu_2 - \mu_1^2} = \mu_1\sqrt{1-p} - \sqrt{p}\sqrt{\mu_2 - \mu_1^2}$.

This gives:

$$B(\delta,p) \leq \frac{\mu_1 - \sqrt{1-p}(\mu_1\sqrt{1-p} - \sqrt{p}\sqrt{\mu_2 - \mu_1^2})}{p} = \mu_1 + \sqrt{\frac{1-p}{p}(\mu_2 - \mu_1^2)}.$$

□

We end the proof of Theorem 5 by showing optimality of the bound for $\Delta_{i,\mu_i}^{j,\mu_j}$:

Lemma 4. *Let μ_i, μ_j and p such that $\alpha_j(\mu_j, p) < \alpha_i(\mu_i, p)$. Then there exists a distribution $\delta \in \Delta_{i,\mu_i}^{j,\mu_j}$ with $B(\delta,p) = \sqrt[j]{\frac{1}{p}(\mu_j - M_1)}$ for $M_1 \leq \mu_j$ the smallest positive real root of:*

$$\mu_i = (1-p)^{\frac{j-i}{j}} (M_1)^{\frac{i}{j}} + p^{\frac{j-i}{j}} (\mu_j - M_1)^{\frac{i}{j}}.$$

Proof. Let us consider the distribution δ with:

- $(1-p)$ of the mass at $\sqrt[j]{\frac{M_1}{1-p}}$ and
- p of the mass at $\sqrt[j]{\frac{(\mu_j - M_1)}{p}}$

It trivially satisfies $B(\delta, p) = \sqrt[j]{\frac{(\mu_j - M_1)}{p}}$. Also, we have easily $\mu_j(\delta) = (1 - p)\frac{M_1}{1-p} + p\frac{\mu_j - M_1}{p} = \mu_j$.

Now, consider $\mu_i(\delta) = (1-p)^{\frac{j-i}{j}}(M_1)^{\frac{i}{j}} + p^{\frac{j-i}{j}}(\mu_j - M_1)^{\frac{i}{j}}$. By definition of M_1 as a root of the equation $\mu_i = (1-p)^{\frac{j-i}{j}}(M_1)^{\frac{i}{j}} + p(\mu_j - M_1)^{\frac{i}{j}}$, we obtain $\mu_i(\delta) = \mu_i$. □