# Compositional Calculation of Steady-State Probabilities*

Eugene W. Stark**

Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794 USA

**Abstract.** We describe a method for converting the problem of computing steady-state probabilities of a continuous-time Markov chain into the problem of computing absorption probabilities for a related chain. The construction enables us to apply compositional techniques for computing absorption probabilities that we have previously developed for *probabilistic I/O automata* to be used for computing certain steady-state quantities as well. The method has been implemented, and we consider the issue of its practicality in the context of its application to an an example problem.

## 1 Introduction

In previous work [WSS94,WSS97] we introduced *probabilistic I/O automata* (PIOA) as a formal model for systems that exhibit concurrent and probabilistic behavior. PIOA are similar in many respects to *stochastic automata* [Buc99,Pla85,PA91,PF91], and like stochastic automata, PIOA are associated with *continuous-time Markov chains (CTMCs)*. PIOA are also equipped with a *composition operation* by which a complex automaton can be constructed from simpler components. Both PIOA and stochastic automata can thus be seen as a formalism for describing large CTMC system models from simpler components. The composition operation for PIOA is defined in essentially the same way as for stochastic automata, however, the PIOA model draws a distinction between *input* (passive) and *output* (active) actions, and in forming the composition of automata only input/input or input/output synchronization is permitted — the output/output case is prohibited.

In [SS98,SP99] we presented algorithms for calculating certain kinds of performance parameters for systems modeled in terms of PIOA. These algorithms work in a *compositional* fashion; that is, by treating the components of a composite system in succession rather than all at once. The compositional approach can

* This research was supported in part by the National Science Foundation under Grant CCR-9988155 and the Army Research Office under Grants DAAD190110003 and DAAD190110019. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, the Army Research Office, or other sponsors.

** Author's E-mail address: `stark@cs.sunysb.edu`

help avoid the state-space explosion problem in such calculations by affording the opportunity to perform state-space reduction as each successive component is treated. Our algorithms lend themselves most naturally to *transient* analyses; for example, computing the probability that a particular action will eventually be performed if the system is started in an initial state, or computing the expected time until a particular action occurs. However, one typically also wants to perform *steady-state* analyses, which concern the long-run probability that the system will be in a particular state or set of states. Traditional methods for performing steady-state analysis on Markov chains [Ste94] involve first constructing a description of the global state space and then solving a system of linear equations having dimension equal to the number of global states. The amount of memory required to store the solution vector becomes the limiting factor in how large a system can be treated in the traditional way.

Though there has recently been significant progress in using techniques such as tensor (or Kronecker) product representations for compact representation of large Markov chains, we became interested in whether there was a way to make use of our compositional techniques for PIOA to perform steady-state as well as transient analyses. This paper describes a technique by which this can be accomplished, gives a theoretical justification for the technique, and considers the question of the practicality of the technique in the context of an example.

The idea underlying our technique is a simple one. Suppose we wish to compute the steady-state probability that a system described by a CTMC is in particular set of states. For example, in Section 6 we will consider, for a resource allocation protocol, the steady-state probability that the resource is free. We modify the original system description by adding to it an additional "random alarm clock" component, which runs in parallel with the rest of the system. Formally, the alarm clock is a two-state CTMC which waits in its initial state for a random time drawn from an exponential distribution with mean $1/\epsilon$, at which time the alarm goes off and the alarm clock takes a transition to its final state.

Now, suppose further that the alarm clock interacts with the original system in such a way that at the instant the alarm goes off, the state of the original system becomes "frozen." Then the modified system consisting of system plus alarm clock is an absorbing CTMC, having one absorbing state for each of the states of the original system. The probability of absorption in any particular absorbing state will be equal to the probability that the system was in the corresponding original state at the time the alarm went off. If we now let $\epsilon$ go to zero, so that the alarm clock takes longer and longer before going off, then the probability of absorption in any particular absorbing state approaches the steady-state probability of the system being in the corresponding original state. Thus, the alarm clock construction provides a way to convert a problem of computing steady-state probabilities for a given system into a problem of computing absorption probabilities for a related system.

The rest of the paper is organized as follows. In Section 2, we give a theoretical justification for the alarm-clock method. In Section 3, we define probabilistic I/O automata and the associated composition operation. In Section 4, we define the alarm clock construction on PIOA. In Section 5, we show that in case the sets of states for which steady-state probabilities are desired can be defined in terms of an external *observer*, then the alarm clock construction need only be performed on the observer, rather than on the entire system description. This facilitates the use of compositional techniques. In Section 6, we consider the application of the the technique to a simple resource-allocation protocol, and we discuss issues of practicality. Section 7 contains some concluding remarks.

## 2 Steady-State Probabilities via Absorption

In this section, we give a theoretical justification for the use of the alarm clock construction to express steady-state probabilities as limits of absorption probabilities. Lemma 1 gives the basic result in a succinct form. This result, which seems intuitively obvious yet still requires some proving to establish, will likely not come as a surprise to probabilists. However, a complete proof has been included since it was not clear where to find a reference that gave the result in the form we need. We assume that the reader is familiar with the basics of continuous-time Markov chains. Background on this topic may be found, for example, in [Bré99].

**Lemma 1.** *Let $G$ be the generator matrix for a finite-state CTMC $\mathcal{M}$ with $n$ states. For $\epsilon > 0$, define the $2n \times 2n$ block matrix $G_\epsilon$ as follows:*

$$G_\epsilon = \begin{pmatrix} 0 & 0 \\ \epsilon I & G - \epsilon I \end{pmatrix}$$

*Then $G_\epsilon$ is the generator of an absorbing CTMC. Let $\mathrm{abs}_\epsilon(\mu)$ denote the $n$-vector of absorption probabilities when the chain is started at $t = 0$ with initial distribution $(0 \;\; \mu)$. Then $\lim_{\epsilon \to 0^+} \mathrm{abs}_\epsilon(\mu)$ exists and is a stationary distribution for $\mathcal{M}$.*

*Proof.* Since $G$ is finite-dimensional it can be uniformized; that is, we may write:

$$G = \lambda(K - I)$$

where $\lambda = \max_i |G(i,i)|$ and $K$ is a transition matrix. Then for $\epsilon > 0$ we have

$$G_\epsilon = (\lambda + \epsilon)(K_\epsilon - I)$$

where

$$K_\epsilon = \begin{pmatrix} I & 0 \\ \frac{\epsilon}{\lambda+\epsilon} I & \frac{\lambda}{\lambda+\epsilon} K \end{pmatrix}$$

Clearly, $K_\epsilon$ is the transition matrix of an absorbing Markov chain. If $\mathrm{abs}_\epsilon(\mu)$ denotes the absorption probability vector when the chain is started with initial distribution $(0 \;\; \mu)$, then $\mathrm{abs}_\epsilon(\mu)$ is given explicitly by the formula (see, *e.g.* [Bré99], Section 6.2):

$$\mathrm{abs}_\epsilon(\mu) \;\; = \;\; \mu(\frac{\lambda}{\lambda+\epsilon}K)^*(\frac{\epsilon}{\lambda+\epsilon}I)$$

where

$$(\frac{\lambda}{\lambda+\epsilon}K)^* \;\; = \;\; \sum_{n=0}^{\infty}(\frac{\lambda}{\lambda+\epsilon}K)^n$$

is the so-called "fundamental matrix" of the absorbing chain.

We claim that:

1. $\hat{\mu} = \lim_{\epsilon\to 0+} \mathrm{abs}_\epsilon(\mu)$ exists.
2. $\hat{\mu}$ is a stationary distribution for $K$.

To see that $\lim_{\epsilon\to 0+} \mathrm{abs}_\epsilon(\mu)$ exists, we first observe that when $K$ is a transition matrix, and $\lambda > 0$ and $\epsilon > 0$, then the summation defining $(\frac{\lambda}{\lambda+\epsilon}K)^*$ is guaranteed to converge and we then have

$$(\frac{\lambda}{\lambda+\epsilon}K)^* \;\; = \;\; (I - \frac{\lambda}{\lambda+\epsilon}K)^{-1}.$$

Let

$$M_\epsilon \;\; = \;\; (\frac{\lambda}{\lambda+\epsilon}K)^*(\frac{\epsilon}{\lambda+\epsilon}I),$$

then $M_\epsilon$ may be regarded as a matrix whose entries are rational functions of $\epsilon$. It is easy to prove that for any fixed $\epsilon > 0$ the matrix $M_\epsilon$ is a transition matrix, hence its entries lie in the interval $[0,1]$. Now, an arbitrary rational function $r(\epsilon)$ either has a finite right limit $r(0^+)$ or else $r(\epsilon)$ diverges to $\infty$ or $-\infty$ as $\epsilon \to 0^+$. Since the matrix $M_\epsilon$ is a transition matrix for all fixed $\epsilon > 0$, it is impossible for the value of any individual entry to diverge to $\infty$ or $-\infty$ as $\epsilon \to 0^+$. Hence

$$\hat{M} \;\; = \;\; \lim_{\epsilon\to 0+}(\frac{\lambda}{\lambda+\epsilon}K)^*(\frac{\epsilon}{\lambda+\epsilon}I)$$

exists, as does $\hat{\mu} = \mu\hat{M}$.

To see that $\hat{\mu}$ is a stationary distribution for $K$, note that from

$$\begin{aligned}\mathrm{abs}_\epsilon(\mu) &= \mu(\frac{\lambda}{\lambda+\epsilon}K)^*(\frac{\epsilon}{\lambda+\epsilon}I)\\ &= \mu(I - \frac{\lambda}{\lambda+\epsilon}K)^{-1}(\frac{\epsilon}{\lambda+\epsilon}I)\end{aligned}$$

we may derive the relation

$$\mathrm{abs}_\epsilon(\mu)(I - \frac{\lambda}{\lambda+\epsilon}K) = \mu(\frac{\epsilon}{\lambda+\epsilon}I).$$

Taking limits on both sides as $\epsilon \to 0$ yields:

$$\hat{\mu}(I - K) = 0,$$

from which we conclude $\hat{\mu} = \hat{\mu}K$; that is, $\hat{\mu}$ is a stationary vector for $K$. Since $K$ is the embedded chain of the CTMC $\mathcal{M}$ any stationary vector for $K$ is also a stationary vector for $\mathcal{M}$, proving the desired result. □

The previous result extends easily to the case where there are nontrivial classes of absorbing states, rather than singletons. In the next result we use the notation $\otimes$ for tensor (or Kronecker) product of matrices, and we use $I_S$ to denote an identity matrix whose rows and columns are indexed by elements of the finite set $S$. Thus the tensor product $I_S \otimes G$ denotes a block matrix consisting of $|S|$ copies of the matrix $G$ on the main diagonal. The *characteristic matrix* of a function $f : S \to S'$, where $S$ and $S'$ are finite sets, is the $|S| \times |S'|$ matrix $F$, whose rows are indexed by $S$ and whose columns are indexed by $S'$, such that $F_{s,s'} = 1$ if $s' = f(s)$ and such that $F_{s,s'} = 0$ otherwise.

**Lemma 2.** *Let $G$ be the infinitesimal generator of a finite-state CTMC $\mathcal{M}$ with state set $Q$. Let $h : Q \to S$ be a surjective function, let $H$ be the $|Q| \times |S|$ characteristic matrix of the function $h$, and let $L$ be the $|Q| \times |S| \cdot |Q|$ characteristic matrix of the function that takes $q$ to $(h(q), q)$. For $\epsilon > 0$, define the block matrix $G_\epsilon$ as follows:*

$$G_\epsilon = \begin{pmatrix} I_S \otimes G & 0 \\ \\ \epsilon L & G - \epsilon I_Q \end{pmatrix}$$

*Then $G_\epsilon$ is the generator of a CTMC with absorbing classes indexed by elements of $S$. Let $\mathrm{abs}_\epsilon(\mu)$ denote the $S$-indexed vector whose entries are the absorption probabilities if the chain is started with initial distribution $(0 \quad \mu)$. Then $\hat{\mu} = \lim_{\epsilon \to 0^+} \mathrm{abs}_\epsilon(\mu)$ exists, and has the form $\hat{\mu} = \hat{\nu} H$, where $\hat{\nu}$ is a stationary vector for $\mathcal{M}$. Thus, for each $s \in S$ we have $\hat{\mu}(s) = \sum_{q \in h^{-1}(s)} \hat{\nu}(q)$.*

*Proof.* Omitted from this abstract. ◻

## 3 Probabilistic Input/Output Automata

A *probabilistic input/output automaton* (PIOA) is a triple:

$$A = (E, Q, \{\tau_e : e \in E\})$$

where[1]:

1. $E = E_{\mathrm{in}} \cup E_{\mathrm{out}}$, where $E_{\mathrm{in}}$ is a set of *input* actions and $E_{\mathrm{out}}$ is a set of *output* actions, such that $E_{\mathrm{in}} \cap E_{\mathrm{out}} = \emptyset$.
2. $Q$ is a finite set of *states*.
3. For each $e \in E$, $\tau_e : Q \times Q \to [0, \infty)$. If $e \in E_{\mathrm{in}}$, then we regard $\tau_e(q, q')$ as a *transition probability*. If $e \in E_{\mathrm{out}}$, then we regard $\tau_e(q, q')$ as a *transition rate*. In accordance with this interpretation, we require that for all $e \in E_{\mathrm{in}}$ and $q \in Q$ the following *stochastic condition* holds:

$$\sum_{q' \in Q} \tau_e(q, q') = 1.$$

[1] In our previous work, we have studied probabilistic I/O automata having *internal* actions as well as input and output actions. In the present paper, however, we do not consider internal actions.

It will often be convenient to regard the functions $\tau_e$ as *transition matrices* whose rows and columns are indexed by $Q$. The stochastic condition above may then be seen as a condition on the row sums of these matrices. If $A$ is a PIOA, then the *total output rate* of a state $q$ is defined to be the quantity

$$\mathrm{rt}(q) = \sum_{e \in E_{\mathrm{out}}} \sum_{q' \in Q} \tau_e(q, q').$$

It represents the overall rate of occurrence of output actions from state $q$.

For the purposes of this abstract, we shall assume that if $e \in E_{\mathrm{out}}$, then the "diagonal" rate $\tau_e(q, q)$ is always zero. We will comment further on this issue in the full version.

To each PIOA $A = (E, Q, \{\tau_e : e \in E\})$ we associate a continuous-time Markov chain $\mathcal{M}_{\mathrm{out}}(A)$ whose infinitesimal generator is the matrix $G$ defined as follows:

$$G(q, q') = \begin{cases} -\mathrm{rt}(q), & \text{if } q = q' \\ \sum_{e \in E_{\mathrm{out}}} \tau_e(q, q'), & \text{otherwise.} \end{cases}$$

This generator describes a stochastic process whose transitions from state $q$ occur with rate $\mathrm{rt}(q)$, such that with probability one each transition is associated with the occurrence of a single output action $e \in E_{\mathrm{out}}$. The selection between the various output actions available from state $q$ is the result of competition among independent exponentially distributed random variables, each associated with a distinct output action, such that the exponential distribution associated with action $e$ in state $q$ has (rate) parameter $\sum_{q' \in Q} \tau_e(q, q')$. We call this Markov chain the *output process* associated with the PIOA $A$.

A finite collection $\{A_i : i \in I\}$ of probabilistic I/O automata, where $A_i = (E^{A_i}, Q^{A_i}, \{\tau_e^{A_i} : e \in E_i\})$, is called *compatible* if for all $i, j \in I$ with $i \neq j$, we have $E_{\mathrm{out}}^{A_i} \cap E_{\mathrm{out}}^{A_j} = \emptyset$. The *composition* of such a collection is a PIOA $(E, Q, \tau)$ where

- $E_{\mathrm{out}} = \bigcup_{i \in I} E_{\mathrm{out}}^{A_i}$ and $E_{\mathrm{in}} = (\bigcup_{i \in I} E_{\mathrm{out}}^{A_i}) \setminus E_{\mathrm{out}}$.
- The state set $Q$ is the cartesian product $\prod_{i \in I} Q_i$.
- For each $e \in E$, the transition matrix $\tau_e$ is the tensor (or Kronecker) product $\bigotimes_{i \in I} \tau_e^{A_i}$ where we adopt the convention that $\tau_e^{A_i}$ is an identity matrix if $e \notin E_{A_i}$.

One can verify using the above definition of composition, that if PIOA $A$ is the composition of the compatible collection $\{A_i : i \in I\}$, then the output rate of $e \in E_{\mathrm{out}}$ in state $(q_i : i \in I)$ of $A$ is precisely equal to the output rate of $e$ in $A_i$ for the unique $i$ such that $e \in E_{\mathrm{out}}^{A_i}$. In addition, the total output rate for state $(q_i : i \in I)$ of $A$ is the sum over all $i \in I$ of the total output rate for state $q_i$ in $A_i$. The above definition of composition for PIOA therefore corresponds to the intuitive idea that in a state $(q_i : i \in I)$ of $A$ the component automata $A_i$ are in a "race" to control the next output action. Upon entering the state $(q_i : i \in I)$, each $A_i$ draws a random time from an exponential distribution with mean $1/\mathrm{rt}(q_i)$. The component $i$ that chooses the smallest time "wins" the race and gets to select the next output action, say $e \in E_{\mathrm{out}}^{A_i}$. A "losing" component $j$ either does nothing, if $e \notin E_{\mathrm{in}}^{A_j}$, or else (if $e \in E_{\mathrm{in}}^{A_j}$) it performs an input transition that is synchronized with the output transition performed by component $i$.

The above notion of composition was introduced for PIOA in [WSS94]. Similar or identical notions of composition have also been defined and studied in other literature on stochastic process algebras. Hillston [Hil94] compares the merits of a number of such notions of composition that have appeared in the context of work on the stochastic process algebras TIPP [GHR93] PEPA [Hil96], MPA [Buc94], and MPA/EMPA [BDG98]. The use of tensor notation provides a succinct way to define composition of automata date back at least to Plateau [Pla85] for stochastic automata.

## 4 "Alarm Clock" Construction on PIOA

In this section, we define a construction on PIOA that corresponds to the CTMC construction of Lemma 2. We call this construction the "alarm clock" construction, following the intuition set out in Section 1. The idea of the "alarm clock" construction is as follows: given a PIOA $A = (E, Q, \{\tau_e : e \in E\})$ and a surjective function $h : Q \to S$, where the set $S$ is assumed to be disjoint from $E$, construct a PIOA $\mathrm{AC}_\epsilon(A, h)$ that behaves like $A$ until an exponentially distributed random "alarm" with rate $\epsilon$ expires, at which time the image $h(q)$ of of the state $q$ of $A$ that was the current state at the time of the alarm is "frozen," and the system takes a transition to an absorbing class of states determined by $h(q)$. The system continues to evolve after the alarm, but the "frozen" value $h(q)$ is remembered forever.

Formally, suppose $A$ is a PIOA and $h : Q \to S$ is a surjective function. Let $\mathrm{AC}_\epsilon(A, h) = (E, (S \times Q) \cup Q, \{\tau_e : e \in E\})$, where $E_{\mathrm{in}} = E_{\mathrm{in}}^A$, $E_{\mathrm{out}} = E_{\mathrm{out}}^A \cup S$, and the transition matrices $\tau_e$ are defined in block form as follows:

– For $e \in E$:

$$\tau_e = \begin{pmatrix} I_S \otimes \tau_e & 0 \\ 0 & \tau_e \end{pmatrix}$$

– For $e \in S$:

$$\tau_e = \begin{pmatrix} 0 & 0 \\ \epsilon H_e & 0 \end{pmatrix}$$

where $H_e$ is the $|Q| \times |S| \cdot |Q|$ matrix such that $H_e(q, (e, q)) = 1$ whenever $h(q) = e$, and all other entries of $H_e$ are zero.

**Lemma 3.** *Given a PIOA $A$ and a surjective function $h : Q \to S$, the output process of the PIOA $\mathrm{AC}_\epsilon(A, h)$ has the following for its infinitesimal generator:*

$$\begin{pmatrix} I_S \otimes G & 0 \\ \epsilon H & G - \epsilon I_Q \end{pmatrix}$$

*where $G$ is the infinitesimal generator of the output process of $A$ and $H$ is the $|Q| \times |S| \cdot |Q|$ characteristic matrix of the function that takes $q$ to $(h(q), q)$.*

*Proof.* Direct from the definition of output process and $\mathrm{AC}_\epsilon(A, h)$. □

**Theorem 1.** *Let $A = (E, Q, \{\tau_e : e \in E\})$ be a PIOA and let $h : Q \to S$ be a surjective function. Suppose further that the output process of $A$ is irreducible, so that a unique stationary distribution exists. Let $\mu$ be an arbitrarily chosen distribution for $A$, and for each $\epsilon > 0$ let $\mathrm{abs}_\epsilon(\mu)$ denote the $S$-indexed vector of absorption probabilities for $\mathrm{AC}_\epsilon(A, h)$ when started with distribution $(0 \;\; \mu)$. Then $\hat{\mu} = \lim_{\epsilon \to 0^+} \mathrm{abs}_\epsilon(\mu)$ exists, and the entries of $\hat{\mu}$ are the steady-state probabilities for $\mathcal{M}_{\mathrm{out}}(A)$ of the sets $h^{-1}(s) \subseteq Q$.*

*Proof.* Immediate from Lemma 2 and Lemma 3. □

## 5 Compositional Calculation

Theorem 1 could be used directly on a PIOA $A$ to calculate the full steady-state distribution. However, this would require the explicit enumeration of the entire state space of $A$ in order to construct $\mathrm{AC}_\epsilon(A, h)$. There would not be much point in doing this, since we might just as well have calculated a stationary distribution for $\mathcal{M}_{\mathrm{out}}(A)$ directly without bothering with $\mathrm{AC}_\epsilon(A, h)$ at all. However, in case we are not interested in the full steady-state distribution but rather only in certain kinds of partial information, then we can exploit the $\mathrm{AC}_\epsilon(A, h)$ construction to calculate steady-state probabilities. This can be done when the type of partial information we are interested in is that which can be discerned by an "observer" of $A$, which is a PIOA $O$ that interacts with $A$ simply by observing what output actions $A$ performs.

Formally, define an *observer* for a PIOA $A$ to be a PIOA $O$, such that $E^O_{\mathrm{out}} = \emptyset$ and $E^O \subseteq E^A$. Note that if $O$ is an observer for $A$, then $O$ is automatically compatible with $A$. The following Lemma says that if we are only interested in the observer's component of the state, then an alarm clock construction on the composite $O|A$ can be accomplished by first performing an alarm clock construction on $O$ alone, and then composing the result with $A$.

**Lemma 4.** *Let $A$ be a PIOA. Let $O$ be an observer for $A$, and let $h : Q^{O|A} \to Q^O$ be the projection from states of $O|A$ to states of $O$. Then we have the following isomorphism of PIOA:*

$$\mathrm{AC}_\epsilon(O|A, h) \;\simeq\; \mathrm{AC}_\epsilon(O, \mathrm{id}_{Q^O}) \mid A.$$

*Proof.* Omitted from this abstract. □

**Lemma 5.** *Let $A$ be a* closed *PIOA (*i.e. *$E^A_{\mathrm{in}} = 0$) and let $h : Q^A \to S$, where $S$ is a finite set. Given a probability distribution $\mu$ on $Q^A$, the probability that $\mathrm{AC}_\epsilon(A, h)$ reaches absorbing class $\{s\} \times Q^A$ when started with distribution $(0 \;\; \mu)$ equals the probability that $\mathrm{AC}_\epsilon(A, h)$ performs output action $s$ when started with distribution $(0 \;\; \mu)$.*

*Proof.* By inspection of the definition of $\mathrm{AC}_\epsilon(A,h)$ it is easy to see that the set of trajectories in which absorbing class $\{s\} \times Q^A$ is reached is exactly the same as the set of trajectories in which output action $s$ is performed. □

**Theorem 2.** *Let $A$ be a closed PIOA and let $O$ be an observer for $A$, such that the output process of $A|O$ is irreducible. Let $h : Q^{O|A} \to Q^O$ denote the projection from states of $O|A$ to states of $O$. For each $\epsilon > 0$, let $\mathrm{abs}_\epsilon(\mu)$ denote the probability that $\mathrm{AC}_\epsilon(0, \mathrm{id}_{Q^0}) \mid A$ outputs $s \in Q^O$ when started with distribution $\mu$, where $\mu$ is an arbitrary distribution that assigns zero probability mass to the absorbing classes. Then $\nu = \lim_{\epsilon \to 0^+} \mathrm{abs}_\epsilon(\mu)$ exists, and the entries of $\nu$ are the steady-state probabilities of the sets $h^{-1}(s) \subseteq Q^O \times Q^A$.*

*Proof.* Immediate from Theorem 1, Lemma 4, and Lemma 5. □

## 6 An Example

Theorem 2 shows that steady-state probabilities for sets of states of $O|A$ can in principle be calculated as limits as $\epsilon \to 0^+$ of probabilities that certain outputs are produced by $\mathrm{AC}_\epsilon(0, \mathrm{id}_{Q_0}) \mid A$. In previous work [SS98,SP99], we have given a method for calculating such probabilities which has the feature of being *compositional*: if $A$ is a composite PIOA $A_1|A_2|\ldots|A_n$ then the probability that the PIOA $\mathrm{AC}_\epsilon(0, \mathrm{id}_{Q_0}) \mid A$ produces a given output can be obtained by treating, in sequence, first the observer plus alarm clock $\mathrm{AC}_\epsilon(O, \mathrm{id}_{Q_O})$, then the PIOA $A_1$, then the PIOA $A_2$, *etc.* until finally $A_n$ is treated. As each component is treated, a minimization step is performed in order to reduce the size of the data that need to be carried along to the next step.

As an example of the application of the above technique, we now use it to compute certain steady-state characteristics of a simple resource allocation protocol.

### 6.1 Description

Consider a system in which $N$ user processes compete for exclusive access to a single resource. In order to ensure mutual exclusion, an additional *arbiter* process is introduced to handle user requests and manage the allocation of the resource. When user $i$ requires access to the resource, it executes a handshake protocol in which it first performs a $req_i$ action to inform the arbiter of its desire to use the resource, and then it awaits a corresponding $grant_i$ response from the arbiter. Once user $i$ has received the $grant_i$ response from the arbiter, user $i$ has exclusive access to the resource until it explicitly relinquishes it by performing a $rel_i$ action. Upon receiving the $rel_i$ action, the arbiter resets the protocol by performing a *reset* action.

The arbiter works as follows. If a $req_i$ arrives when the resource is free, then the resource is tentatively allocated to user $i$ and a $grant_i$ action is performed. By performing the $grant_i$ action, the arbiter commits to the allocation. If requests arrive from other users after the arbiter has issued the $grant_i$, then these additional requests are ignored. Later, once the user with the resource has relinquished it via a $rel_i$ action, the arbiter will perform a *reset*, at which point the protocol reverts to its initial state and any user still interested in the resource must re-issue its request. Between the arrival of a $req_i$ until the time a corresponding $grant_i$ is issued, there is a period of ambiguity concerning what happens if other requests arrive at the arbiter. In the version of the protocol we consider, the arbiter in fact commits to the allocation as soon as the first $req_i$ arrives, and ignores any other $req_j$ that arrive in the period between $req_i$ and the subsequent $grant_i$.

The full version of the paper will include state diagrams for the user and the arbiter. For brevity, we have omitted these from this abstract.

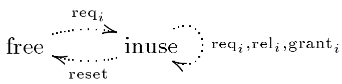


**Fig. 1.** Observer for Calculating Steady-State Probability of Resource Free

### 6.2 Analysis

We used the techniques described in this paper to analyze the resource allocation protocol for various numbers of users $N$. For a given value of $N$, each user process has 5 states and the arbiter has $2N + 3$ states, so that the total number of global states of an $N$-user system is $5^N \cdot (2N + 3)$. The analysis algorithm was implemented in the programming language Standard ML, and all the computation times reported below were measured using the Standard ML of New Jersey system version 110.0.7 running on a 1.8GHz Xeon processor.

Table 1 shows the results of computing the probability of eventual occurrence of an *err* or $err_i$ action if the protocol is started with the arbiter and all users in the idle state. This calculation is not a steady-state calculation, and

**Table 1.** Probability of Protocol Error

| $N$ | GlobalStates | Error Prob. | Time (sec.) |
|---|---|---|---|
| 1 | 25 | 0 | 0.013 |
| 2 | 175 | 0 | 0.027 |
| 3 | 1125 | 0 | 0.061 |
| 4 | 6875 | 0 | 0.103 |
| 5 | 40625 | 0 | 0.162 |
| 10 | 224564375 | 0 | 0.821 |

thus does not require the alarm clock construction described this paper. Instead the calculation was performed using the compositional methods detailed in our earlier papers. Note that the impossibility of a protocol error can be established extremely quickly, and that the compositional technique clearly does not require the enumeration of the entire global state space to obtain the answer.

Table 2 shows the results of computing the steady-state probability of the resource being free, for the immediate-allocation version of the algorithm. This was accomplished via the alarm clock construction using the two-state observer whose transition diagram is shown in Figure 1. This observer considers the resource as not free as soon as it has been tentatively allocated in response to a request. The resource remains not free until the next *reset* action occurs. Our implementation produces exact results (not floating point approximations), which are shown in the table as fractions. The results appear to follow the formula $1/(3N+1)$, which is intuitively justifiable on the basis of symmetry considerations (though we have not established this rigorously). The column labeled "Red. Dim." gives the dimension of the reduced system of equations that is produced after all components have been treated.

**Table 2.** Steady-State Probability of Resource Free

| $N$ | Global States | Red. Dim. | Free Prob. | Time (sec.) |
|---|---|---|---|---|
| 1 | 25 | 5 | 1/4 | 0.049 |
| 2 | 175 | 13 | 1/7 | 0.154 |
| 3 | 1125 | 33 | 1/10 | 0.247 |
| 4 | 6875 | 81 | 1/13 | 2.835 |
| 5 | 40625 | 193 | 1/16 | 3744. |

The $N = 5$ case is a good illustration of the problems of practicality of the technique, at least in the current implementation. The program very quickly reduces the problem of calculating the steady-state probability for the 40625-state system to the problem of solving a system of 193 linear equations in 193 unknowns. However, this system of equations is then solved in the field of rational functions over the unknown alarm clock rate $\epsilon$, to obtain a symbolic expression for the dependence on $\epsilon$ of the probability of the resource being free at the time the alarm goes off. The result is the following rational function:

$$\frac{\epsilon^3 + 3\epsilon^2 + 3\epsilon + 1}{\epsilon^3 + 8\epsilon^2 + 18\epsilon + 16}$$

whose value at $\epsilon = 0$ is the steady-state probability $1/16$. Nearly all the time is spent in this solution process, which is performed using exact rational arithmetic for the coefficients. Obviously, there is nothing "difficult" about the resulting function. However, it is the blowup in the amount of precision required for the coefficients in the intermediate steps that causes the calculation to take so long.

There are at least three possibilities for improving the practicality of the technique. One would be to discover a way in which the arithmetic required to solve the final system of equations could be carried out in floating point arithmetic, rather than exact rational arithmetic. Note that one cannot naively replace rational coefficients by floating point values, since the approximation leads to ambiguities concerning the degree of the rational functions being calculated. So some theoretical justification for such a replacement would be required. A second possibility for improving the practicality would be to find a way to compute the limiting value as $\epsilon \to 0$ of the rational function $r(\epsilon)$ that is the solution to the final system of equations, without having to actually determine the function completely. This would reduce solving equations in the field of rational functions to a calculation carried out in the coefficient field, and would likely result in a significant reduction in computation time.

A third possibility for reducing the computation time is to approximate the limit of $r(\epsilon)$ as $\epsilon \to 0$ by computing $r(\epsilon_i)$ for a specific sequence of values $\epsilon_i$ that converge to 0. Although some further theory is required here to determine the error that would result from a particular choice of $\epsilon$, this theory might not be too difficult to develop, since rational functions have simple convergence behavior. Table 3 gives an idea of what happens when we try this approach. The calculations are all performed using exact arithmetic, but the (scalar) solution of the final 193-dimensional system is performed in floating point, using LU decomposition. Each of the calculations took roughly 4.5 seconds to perform.

**Table 3.** Approximating Steady-State Probability of Resource Free ($N = 5$)

| $\epsilon$ | Free Prob. |
|---|---|
| 1 | 0.186046511628 |
| 1/10 | 0.0744365527655 |
| 1/100 | 0.0636742890541 |
| 1/1000 | 0.0626172118867 |
| 1/10000 | 0.0625117189941 |
| 1/100000 | 0.0625011718817 |
| 1/1000000 | 0.0625001171735 |

This technique seems to obtain good approximations to the answer fairly quickly, at least for $N = 5$. The same technique applied to the $N = 10$ case with $\epsilon = 1/1000000$ takes several hours to reduce the original model, which has nearly 225 million global states, to a system of 11265 equations. However, our LU-decomposition-based solution procedure got bogged down in solving this system, probably due to "fill-in" that affected the originally sparse matrix during the solution process. An iterative solution procedure would likely improve the situation.

It is also possible to use the alarm clock construction to compute other steady state parameters besides the probability of being in a particular set of states. For example, we might be interested in the expected delay from the time a user first issues a request from its idle state, until the time the matching grant is issued, assuming that the system is started in steady state. This can be done by using an alarm clock construction as follows: "wait for the alarm to go off, then wait for the first user request, wait for the matching grant, and tally the expected time between the request and the grant." As the alarm clock rate $\epsilon$ tends to zero, the result approaches the steady state value. Table 4 shows the results of using this approach to compute the exact steady-state expected delay from the time a user issues a request from its idle state until the time the matching grant is issued. Table 5 shows the results of an approximate calculation of the same quantities. A full explanation and justification for the specific construction used to perform these calculations is outside the scope of this paper.

**Table 4.** Exact Steady-State Request-to-Grant Expected Delay

| $N$ | Global States | Red. Dim. | Exp. Delay. | Time (sec.) |
|---|---|---|---|---|
| 1 | 25 | 5 | 1 | 0.147 |
| 2 | 175 | 13 | 241/56 | 0.945 |
| 3 | 1125 | 33 | 297/40 | 8.855 |
| 4 | 6875 | 81 | 1091/104 | 285.416 |

**Table 5.** Approx. Steady-State Request-to-Grant Expected Delay ($\epsilon = 1/1000000$)

| $N$ | Global States | Red. Dim. | Exp. Delay. | Time (sec.) |
|---|---|---|---|---|
| 1 | 25 | 5 | 0.9999999999 | 0.188 |
| 2 | 175 | 13 | 4.3035717430 | 0.802 |
| 3 | 1125 | 33 | 7.4250004769 | 3.520 |
| 4 | 6875 | 81 | 10.4903851886 | 16.725 |
| 5 | 40625 | 193 | 13.5312506364 | 87.273 |

## 7 Conclusion

We have described a method for converting the problem of computing steady-state probabilities of a CTMC into the problem of computing absorption probabilities for a related CTMC. This "alarm-clock" construction permits compositional techniques for computing absorption probabilities that we have previously described for probabilistic I/O automata to be used for computing certain steady-state quantities as well. There are significant impediments to the practicality of the technique if exact results are required, but with some additional theoretical work it seems likely that it could be used to produce approximate results.

## References


[BDG98] M. Bernardo, L. Donatiello, and R. Gorrieri. A formal approach to the integration of performance aspects in the modeling and analysis of concurrent systems. *Information and Computation*, 1998. To appear.

[Bré99] Pierre Brémaud. *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*, volume 31 of *Texts in Applied Mathematics*. Springer, New York, 1999.

[Buc94] Peter Buchholz. Markovian process algebra. In U. Herzog and M. Rettelbach, editors, *Proceedings of the 2nd Workshop on Process Algebra and Performance Modeling*, pages 11–30, University of Erlangen, July 1994.

[Buc99] Peter Buchholz. Exact performance equivalence: An equivalence relation for stochastic automata. *Theoretical Computer Science*, 215:263–287, 1999.

[GHR93] N. Götz, U. Herzog, and M. Rettelbach. Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebra. In L. Donatiello and R. Nelson, editors, *Performance '93*, volume 729 of *Lecture Notes in Computer Science*, pages 121–146, Berlin, 1993. Springer-Verlag.

[Hil94] J. Hillston. The nature of synchronization. In U. Herzog and M. Rettelbach, editors, *Proceedings of the 2nd Workshop on Process Algebra and Performance Modeling*, pages 51–70, University of Erlangen, July 1994.

[Hil96] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[PA91] Brigitte Plateau and Karim Atif. Stochastic automata networks for modeling parallel systems. *IEEE Transactions on Software Engineering*, 17:1093–1108, 1991.

[PF91] Brigitte Plateau and J. M. Fourneau. A methodology for solving Markov models of parallel systems. *Journal of Parallel and Distributed Computing*, 12:370–387, 1991.

[Pla85] Brigitte Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. *Performance Evaluation Review*, 13:147–154, 1985.

[SP99] E. Stark and G. Pemmasani. Implementation of a compositional performance analysis algorithm for probabilistic I/O automata. In *Proceedings of 1999 Workshop on Process Algebra and Performance Modeling (PAPM99)*. Prensas Universitarias de Zaragoza, September 1999.

[SS98] E. W. Stark and S. Smolka. Compositional analysis of expected delays in networks of probabilistic I/O automata. In *Proc. 13th Annual Symposium on Logic in Computer Science*, pages 466–477, Indianapolis, IN, June 1998. IEEE Computer Society Press.

[Ste94] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[WSS94] S.-H. Wu, S. A. Smolka, and E. W. Stark. Compositionality and full abstraction for probabilistic I/O automata. In *Proceedings of CONCUR '94 — Fifth International Conference on Concurrency Theory*, Uppsala, Sweden, August 1994.

[WSS97] S.-H. Wu, S. A. Smolka, and E. W. Stark. Composition and behaviors of probabilistic I/O automata. *Theoretical Computer Science*, 176(1-2):1–38, 1997.