NUMERICAL SOLUTION of MARKOV CHAINS, p. 289–304

Reversed processses, product forms, non-product forms and a new proof of the BCMP theorem

* Peter G. Harrison

Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ, UK

KEY WORDS: Reversed Markov Process, Stochastic Proces Algebra, Product-form

ABSTRACT

The equilibrium joint state probabilities of interacting Markov processes are obtained in a hierarchical way, by finding the *reversed process* of the interaction in terms of the reversed processes of the components. From a reversed process, a product-form solution for the joint state probabilities follows directly. The method uses a Markovian process algebra formalism and generalizes the recent Reversed Compound Agent Theorem (RCAT) to solve a diverse class of concurrent systems. This class includes processes with shared, exclusive resources, a customer-oriented specification of a last-come-first-served (LCFS) queue with Coxian service times and an extended PS queue with a non-product form solution. From these results, a new, very short proof of the BCMP theorem ensues. The principal advantage of the methodology is its potential for mechanisation and symbolic implementation. Indeed, many non-standard product-forms have emerged directly from the compositional approach.

1. Introduction

Using the recent Reversed Compound Agent Theorem (RCAT) [5], product-forms for the equilibrium state probabilies of interacting Markov processes have been found in a unified way. These include Jackson queueing networks [10], networks with negative customers [3] and triggers [4] and extensions to chains of negative triggers [7, 6]. The approach taken to finding product-forms is entirely different to the usual one of solving a process's steady state Kolmogorov (balance) equations. In a hierarchical way, it seeks the *reversed process* of the Markov chain in terms of the reversed processes of its sub-chains. From a reversed process, a product-form is easy to obtain.

The formalism we use for this hierarchical analysis is PEPA [9], a Markovian Process Algebra (MPA), which has an appropriate recursive structure. We generalise RCAT in two ways, making it much more widely applicable and unifying disparate existing product-forms. In Section 2 we

*Correspondence to: P.G. Harrison, Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ, UK, pgh@doc.ic.ac.uk, Phone: +44 20-7594-8363, Fax: +44 20-7581-8024

briefly review the salient properties of reversed processes and define our MPA-based formalism. The methodology used to determine the reversed process of a certain type of cooperation between two agents, based on RCAT, is described in section 3. The first two conditions of this theorem are relaxed to yield a more general result that applies to a wider class of concurrent systems, including Boucherie's product-form [2] and a customer-oriented specification of a last-come-first-served (LCFS) queue with stages of (exponential) service. In section 5, another generalisation of RCAT is obtained and used to derive the equilibrium state probabilities in a similar staged queue with processor sharing (PS) queueing discipline. In fact this leads to a more general, *non-product form* solution for a class of queueing networks with global state-dependence. These results are used to define new aggregate processes with known reversed processes. These in turn are then utilized in a further application of RCAT at a higher level of description to give a new, very short proof of the BCMP theorem [1].

The methodology unifies many existing product-forms derived elsewhere over many years in diverse ways. Moreover, it generates new (to the author's best knowledge) ones, including the generalised PS queueing network referred to above, as well as others in preparation. The paper concludes in Section 7 where we assess the significance of this work and outline some directions for further research.

2. Reversed Markov processes and an MPA

A stochastic process $\{X_t \mid -\infty < t < \infty\}$ is *stationary* if $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ and $(X_{t_1+\tau}, X_{t_2+\tau}, \ldots, X_{t_n+\tau})$ have the same probability distribution for all times $t_1, t_2, \ldots, t_n$ and τ. The reversed process of $\{X_t\}$ is the (necessarily) stationary process $\{X_{\tau-t}\}$ for any real number τ. It is straightforward to find the reversed process of a stationary Markov process if the stationary state probabilities are known.

Proposition 1. *The reversed process of a stationary Markov process $\{X_t\}$ with state space S, generator matrix Q and stationary probabilities $\boldsymbol{\pi}$ is a stationary Markov process with generator matrix Q' defined by*

$$q'_{ij} = \pi_j q_{ji} / \pi_i \quad (i, j \in S)$$

and with the same stationary probabilities $\boldsymbol{\pi}$.

This proposition is standard, see for example [11], and immediately yields a product-form solution for $\boldsymbol{\pi}$. This is because, in an irreducible Markov process, we may choose a reference state 0 arbitrarily, find a sequence of connected states, in either the forward or reversed process, $0, \ldots, j$ (i.e. with either $q_{i,i+1} > 0$ or $q'_{i,i+1} > 0$ for $0 \leq i \leq j-1$) for any state j and calculate

$$\pi_j = \pi_0 \prod_{i=0}^{j-1} \frac{q_{i,i+1}}{q'_{i+1,i}} = \pi_0 \prod_{i=0}^{j-1} \frac{q'_{i,i+1}}{q_{i+1,i}}$$

We use a Markovian process algebra to describe agents (isomorphic to Markov chains when all time delays are exponential random variables) at a higher level than state transition graphs. In particular, the *cooperation* combinator of the MPA defines precisely how agents interact in a concise manner, using generic descriptions of their actions' rates.

In fact we will only need the *prefix* and *cooperation* combinators of the MPA PEPA (Performance Evaluation Process Algebra) [9]:

Definition 1.

1. *The prefix combinator defines an agent* $(a, \lambda).P$ *that carries out action* (a, λ) *of type (or 'name')* a *at rate* λ *and subsequently behaves as agent* P*.*
2. *The agent describing the cooperation of two agents* P *and* Q *which synchronise over actions with types in a specified set* L *is written* $P \underset{L}{\bowtie} Q$ *. Every action type in* L *is active in exactly one of the agents* P*,* Q *and passive in (i.e. 'waits' in) the other.*

The prefix combinator can describe every instantaneous transition rate between any two states of a Markov chain, and hence is sufficient alone to define *any* Markov chain. However, such specifications are usually no simpler than state transition matrices or graphs, and so we use the cooperation construct to facilitate hierarchical specifications. To clarify its usage, suppose an action with type $a \in L$ is active in P and passive in Q. It may be enabled in any of a subset of the states of P and Q; possibly in *every* state of P and/or Q. For example, if P and Q represent queues, a might represent a departure from queue-P (enabled in every state of P with non-zero queue length) and an arrival to queue-Q (enabled in every state of Q).

In addition, we name new agents using an *assignment combinator*, $A = P$, and use the *relabeling* $P\{y \leftarrow x\}$ to denote the process P in which all occurences of the symbol y are changed to x, which may be an expression. Thus, for example, $((a, \lambda).P)\{\lambda \leftarrow \mu\}$ denotes the agent $(a, \mu).P\{\lambda \leftarrow \mu\}$.

We denote reversed entities (agents, actions, action types, action rates) with an overbar. Thus, in the above example, $\overline{a}$ denotes the type of the reversed action with type a, indicated by a set of reversed arrows (corresponding to instances of a) in the Markov state transition graph. Similarly, $\overline{\lambda}$ denotes the rates of such reversed actions. Notice that in general these rates need not be the same for all instances of $\overline{a}$.

3. Reversed Compound Agent Theorems

Under appropriate conditions, the reversed agent of a cooperation $P \underset{L}{\bowtie} Q$ between two agents P and Q is a cooperation between the reversed agents of P and Q, after some re-parameterisation [5]. The proof of this result (RCAT) uses an extension of *Kolmogorov's criteria*, originally established for reversible processes, see [11], for example. This states that X and Y are reversed processes of each other if and only if (a) the sum of the outgoing rates from every state (reciprocal of the mean state holding time) is the same in both X and Y; (b) for every cycle in X, the product of the rates around it is equal to the corresponding product of (reversed) rates in Y (in the opposite direction). The original RCAT required that every passive action be enabled in every derivative of both the forward and reversed cooperating agents, i.e. in every state of their underlying Markov processes. This condition guarantees that the total outgoing rate from any state is the same in the processes underlying the agent $P \underset{L}{\bowtie} Q$ and its claimed reversed agent. Although these (sufficient) conditions are satisfied by many queueing network and other cooperations, they are not necessary. Here we relax them and give a more general RCAT. The third condition remains unchanged, as does the proof that the products of the rates around cycles in the forward and reversed processes are equal.

When considering the actions that participate in a cooperation $P \underset{L}{\bowtie} Q$ – with types that are members of the cooperation set L – we have to account for the possibility that certain action types of L might not be present in every derivative, i.e. in every state of the underlying, irreducible Markov chain (G in the theorem's statement). We define the following subsets of action types in L, in which the agent A is P or Q:

$\mathcal{P}_A(L)$ denotes the subset that are passive in A (with form $(a, \top)$ in A);

$\mathcal{A}_A(L) = L \setminus \mathcal{P}_A(L)$ denotes the subset that are active in A;

$\mathcal{P}_A^{i\rightarrow}$ denotes the subset that are passive in A and correspond to transitions *out of* state i in the Markov process of A;

$\mathcal{P}_A^{i\leftarrow}$ denotes the subset that are passive in A and correspond to transitions *into* state i in the Markov process of A;

$\mathcal{A}_A^{i\rightarrow}$ denotes the subset that are active in A and correspond to transitions *out of* state i in the Markov process of A;

$\mathcal{A}_A^{i\leftarrow}$ denotes the subset that are active in A and correspond to transitions *into* state i in the Markov process of A;

$\mathcal{P}^{(i,j)\rightarrow} = \mathcal{P}_P^{i\rightarrow} \cup \mathcal{P}_Q^{j\rightarrow}$ and $\mathcal{A}^{(i,j)\rightarrow} = \mathcal{A}_P^{i\rightarrow} \cup \mathcal{A}_Q^{j\rightarrow}$;

$\mathcal{P}^{(i,j)\leftarrow} = \mathcal{P}_P^{i\leftarrow} \cup \mathcal{P}_Q^{j\leftarrow}$ and $\mathcal{A}^{(i,j)\leftarrow} = \mathcal{A}_P^{i\leftarrow} \cup \mathcal{A}_Q^{j\leftarrow}$;

$\alpha_a^{(i,j)}$ denotes the instantaneous transition rate *out of* (joint) state (i, j) in the Markov process of $P \underset{L}{\bowtie} Q$ corresponding to *active* action type $a \in L$.

$\overline{\beta_a^{(i,j)}}$ denotes the instantaneous transition rate *out of* state (i, j) in the *reversed* Markov process of $P \underset{L}{\bowtie} Q$ corresponding to *passive* action type $a \in L$; note that a is *incoming* to state (i, j) in the forwards process.

Notice that it is possible for an action type to appear more than once in the sets defined above, which are therefore *multisets*. We define the set difference operator $\setminus$ on multisets so as to remove all occurrences of subtracted elements, i.e.

$$A \setminus B = \{x \mid x \in A \wedge x \notin B\}$$

so that $B \cap (A \setminus B) = \emptyset$.

Theorem 1.
Suppose that the cooperation $P \underset{L}{\bowtie} Q$ *has a derivation graph with an irreducible subgraph* G. *Given that every occurrence of a reversed action of an active action type in* $\mathcal{A}_P(L)$ *(respectively* $\mathcal{A}_Q(L)$*) has the same rate in* $\overline{P}$ *(respectively* $\overline{Q}$*), the reversed subgraph* $\overline{G}$ *is defined by the derivation graph of the reversed agent* $\overline{P \underset{L}{\bowtie} Q} =$

$$\overline{R}\{(\overline{a}, \overline{p_a}) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_P(L)\} \underset{\overline{L}}{\bowtie} \overline{S}\{(\overline{a}, \overline{q_a}) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_Q(L)\}$$

where

$$\begin{aligned} R &= P\{\top_a \leftarrow x_a \mid a \in \mathcal{P}_P(L)\} \\ S &= Q\{\top_a \leftarrow x_a \mid a \in \mathcal{P}_Q(L)\} \end{aligned}$$

$\{x_a\}$ are the solutions (for $\{\top_a\}$) of the equations

$$\top_a = \overline{q_a} \qquad a \in \mathcal{A}_Q(L)$$

$$\top_a = \overline{p_a} \qquad a \in \mathcal{A}_P(L)$$

and $\overline{p_a}$ (respectively $\overline{q_a}$) is the symbolic rate of action type $\overline{a}$ in $\overline{P}$ (respectively $\overline{Q}$), provided that *the underlying Markov chain is ergodic (has a steady state) and*

$$\sum_{a \in \mathcal{P}^{(i,j)\rightarrow}} x_a - \sum_{a \in \mathcal{A}^{(i,j)\leftarrow}} x_a = \sum_{a \in \mathcal{P}^{(i,j)\leftarrow} \setminus \mathcal{A}^{(i,j)\leftarrow}} \overline{\beta_a^{(i,j)}} - \sum_{a \in \mathcal{A}^{(i,j)\rightarrow} \setminus \mathcal{P}^{(i,j)\rightarrow}} \alpha_a^{(i,j)}$$

Proof The part of the proof of the original RCAT pertaining to cycles goes through unchanged for the present generalisation. It therefore remains to show that Kolmogorov's criteria (a) holds.

Let the instantaneous transition rate in the Markov chain of P (respectively Q, R, S) out of state k corresponding to action type a be p_{ka} (respectively q_{ka}, r_{ka}, s_{ka}) and let $p_k = \sum_{a \in \mathcal{O}_k} p_{ka}$ (q_k, r_k, s_k similarly), where $\mathcal{O}_k$ is the set of all outgoing action types in the derivative corresponding to state k. Thus p_k is the total outgoing rate from state k in the Markov chain of P, and $\alpha_a^{(i,j)} = p_{ia}$ if a is active in P and q_{ja} if a is active in Q.

Every node in the derivation graph can be identified as $(i, j) \in G$, where i, j are states in the chains corresponding to P, Q respectively.

In $P \underset{L}{\bowtie} Q$, the total rate out of any node $(i, j) \in G$ is

$$p_i\{\top \leftarrow 0\} + q_j\{\top \leftarrow 0\} - \sum_{a \in \mathcal{A}_P^{i\rightarrow} \setminus \mathcal{P}_Q^{j\rightarrow}} p_{ia} - \sum_{a \in \mathcal{A}_Q^{j\rightarrow} \setminus \mathcal{P}_P^{i\rightarrow}} q_{ja}$$

where the relabeling $\{\top \leftarrow 0\}$ is an abbreviation for $\{\top_a \leftarrow 0 \mid a \in L\}$; i.e. every occurrence of an unspecified rate correponding to action types in L is set to zero. The subtracted terms correspond to active actions a that are disabled, i.e. do not have passive actions to synchronise with in state (i, j). Since $\mathcal{A}_P^{i\rightarrow}$ and $\mathcal{A}_Q^{j\rightarrow}$ are disjoint, as are $\mathcal{P}_Q^{j\rightarrow}$ and $\mathcal{P}_P^{i\rightarrow}$, the total rate out of node (i, j) in the forward cooperation can be simplified to:

$$p_i\{\top \leftarrow 0\} + q_j\{\top \leftarrow 0\} - \sum_{a \in \mathcal{A}^{(i,j)\rightarrow} \setminus \mathcal{P}^{(i,j)\rightarrow}} \alpha_a^{(i,j)}$$

Now,

$$\begin{aligned} r_i &= p_i\{\top_a \leftarrow x_a \mid a \in \mathcal{P}_P(L)\} \\ s_j &= q_j\{\top_a \leftarrow x_a \mid a \in \mathcal{P}_Q(L)\} \end{aligned}$$

are the total rates out of states i and j in R and S respectively and hence, by definition, in $\overline{R}$ and $\overline{S}$ respectively. Thus,

$$\begin{aligned} r_i &= p_i\{\top \leftarrow 0\} + \sum_{a \in \mathcal{P}_P^{i\rightarrow}} x_a \\ s_j &= q_j\{\top \leftarrow 0\} + \sum_{a \in \mathcal{P}_Q^{j\rightarrow}} x_a \end{aligned}$$

Hence, the total rate out of state (i, j) in

$$\overline{R}\{(\overline{a}, \overline{p_a}) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_P(L)\} \underset{\overline{L}}{\bowtie} \overline{S}\{(\overline{a}, \overline{q_a}) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_Q(L)\}$$

is

$$\begin{aligned} p_i\{\top \leftarrow 0\} + q_j\{\top \leftarrow 0\} &+ \sum_{a \in \mathcal{P}_P^{i\rightarrow}} x_a + \sum_{a \in \mathcal{P}_Q^{j\rightarrow}} x_a \\ &- \sum_{a \in \mathcal{A}_P^{i\leftarrow}} x_a - \sum_{a \in \mathcal{A}_Q^{j\leftarrow}} x_a \\ &- \sum_{a \in \mathcal{P}_P^{i\leftarrow} \setminus \mathcal{A}_Q^{j\leftarrow}} \overline{p_{ia}} - \sum_{a \in \mathcal{P}_Q^{j\leftarrow} \setminus \mathcal{A}_P^{i\leftarrow}} \overline{q_{ja}} \end{aligned}$$

since, in the reversed cooperation,

(a) the (outgoing) passive action types $\overline{a}$ in $\overline{L}$ are the reversed actions of the (incoming) active action types a in the forwards cooperation set L. By definition, these have rates x_a and must be subtracted out;
(b) the (outgoing) active actions in $\overline{L}$ are the reversed actions of the (incoming) passive actions in the forwards cooperation set L – the rates of the disabled ones of these must also be subtracted out.

The total outgoing rate in the above reversed process therefore becomes:

$$p_i\{\top \leftarrow 0\} + q_j\{\top \leftarrow 0\} + \sum_{a \in \mathcal{P}^{(i,j)\rightarrow}} x_a - \sum_{a \in \mathcal{A}^{(i,j)\leftarrow}} x_a - \sum_{a \in \mathcal{P}^{(i,j)\leftarrow} \setminus \mathcal{A}^{(i,j)\leftarrow}} \overline{\beta_a^{(i,j)}}$$

The theorem's last condition therefore gives Kolmogorov's criteria (a). ♠

Although expressed in process-algebraic terms, this theorem is easily applied without explicit knowledge of process algebra. All that is necessary is the generic grouping of actions (transitions) of the same type, e.g. arrivals. The reversed agents of these agents are assumed known and so the reversed rate associated with each instance of an active action type (in its own participating agent) can be determined and checked if it is a constant (first condition of the theorem). The equations for the x_a can then be posed and the theorem applied. The last condition, relating to the presence of action types and their rates, can also be checked in the cooperating agents individually in each state. Special cases (including the original RCAT [5]) are considered below.

In an agent with a bundle of multiple transitions between two states, the reversed rate of the aggregate (summed) transition is distributed amongst the reversed arcs in proportion to the forward transition rates. This definition is needed to handle components that can either proceed independently or cooperate. For example, a service completion at a queue can cause either an external departure or the transfer of a customer to another queue.

3.1. The original RCAT

In the original RCAT of [5], all passive actions are enabled in every state, in both the forward and reversed processses. Hence, $\mathcal{P}^{(i,j)\rightarrow} = \mathcal{A}^{(i,j)\leftarrow} = L$ so the last condition becomes

$$\sum_{a \in L} x_a - \sum_{a \in L} x_a = \sum_{a \in \emptyset} \overline{\beta_a^{(i,j)}} - \sum_{a \in \emptyset} \alpha_a^{(i,j)}$$

which is true trivially.

3.2. Invisible passive transitions

Suppose a passive action a_0 is invisible, leading from some state, $i_0 \in P$ say, to itself. Then we have $\overline{\beta_{a_0}^{(i_0,j)}} = x_{a_0}$ for all j. If all passive actions are invisible, $\mathcal{P}^{(i,j)\rightarrow} = \mathcal{P}^{(i,j)\leftarrow} = \mathcal{P}^{(i,j)}$, say, and the last condition becomes:

$$\sum_{a \in \mathcal{P}^{(i,j)\rightarrow}} x_a = \sum_{a \in \mathcal{P}^{(i,j)\leftarrow} \cup \mathcal{A}^{(i,j)\leftarrow}} x_a - \sum_{a \in \mathcal{A}^{(i,j)\rightarrow} \setminus \mathcal{P}^{(i,j)\rightarrow}} \alpha_a^{(i,j)}$$

i.e.

$$\sum_{a \in \mathcal{A}^{(i,j)\leftarrow} \setminus \mathcal{P}^{(i,j)}} x_a = \sum_{a \in \mathcal{A}^{(i,j)\rightarrow} \setminus \mathcal{P}^{(i,j)}} \alpha_a^{(i,j)} \tag{1}$$

Note that this equation is satisfied vacuously if $\mathcal{P}^{(i,j)} = L$.

3.2.1. Blocking via invisible transitions Consider now a pair of passive invisible action types of the above kind, a_0 and a_1 say. Equation 1 may be satisfied in any state (i,j) in which $\mathcal{P}^{(i,j)} \neq \{a_0, a_1\}$ (when it is not satisfied vaccuously) if both a_0 and a_1 satisfy (a) or (b) separately, *or*, alternatively, a_0 is incoming and outgoing with rate x_{a_1} and a_1 is incoming and outgoing with rate x_{a_0}.

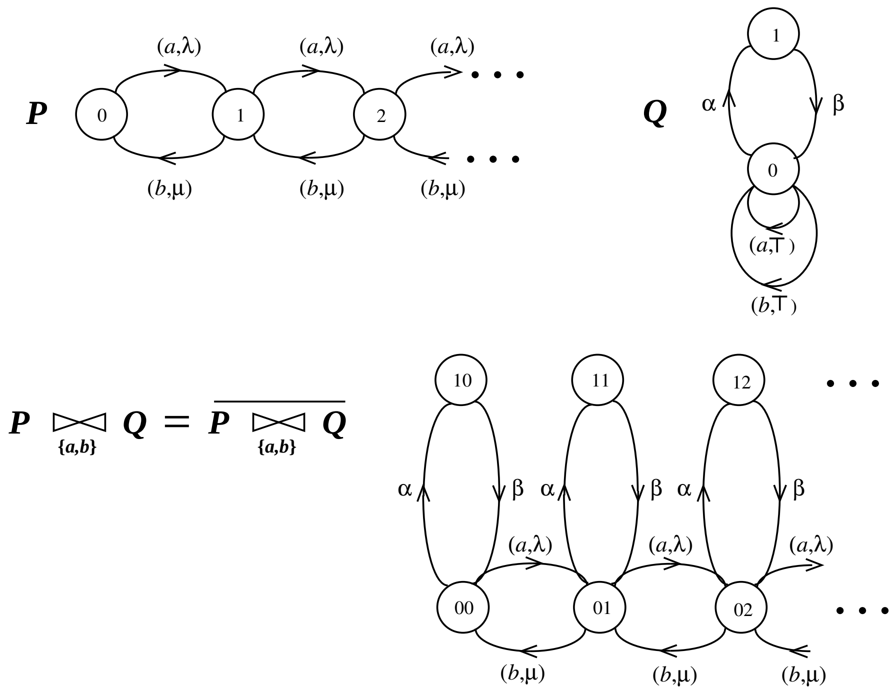

Figure 1. Two invisible passive actions controlling an M/M/1 queue

In other words, the reversed rates of the *active* actions (in the other cooperating component) a_0 and a_1 are the forward rates of each other. This happens in many systems like M/M/1 queues where a_0 and a_1 can represent arrivals and departures respectively. Such a queue can therefore be blocked by an independent Markov chain, namely when this chain is in a state without passive invisible transitions, see figure 1. Theorem 1 is satisfied and here it can be seen that the cooperation is a *reversible* process. Clearly we could consider sets of more than two invisible action types similarly.

3.2.2. Processes with shared resources Now consider a pair of passive invisible action types, $a_0 \in \mathcal{P}_P(L)$ and $a_1 \in \mathcal{P}_Q(L)$. Here some states are 'invalid' and occur as transient or absorbing states outside the irreducible subchains of interest (G of theorem 1). It is then possible in certain models to satisfy equation 1 vaccuously for all states in these irreducible subchains.

Consider first the cooperation depicted in figure 2. Here the respective states 1 are 'exclusive' in the sense that when process P is in state 1, process Q cannot perform action a to enter its state 1, and *vice vera*. Thus, state 11 is invalid in the cooperation and cannot be entered from the irreducible subchain containing the other states 00, 10, 01. Condition 1 is satisfied vaccuously since state 11 is not in the irreducible subchain and so the transitions from 11 to 10 and 11 to 01 are absent; they are transient in the cooperation. The reversed process is now given by theorem 1, followed by a product-form solution for the equilibrium state probabilities.

More generally, two processes P and Q can each have a subset of states, E_P, E_Q respectively, designated as mutually exclusive in the sense that when the state of P is in E_P, Q cannot enter E_Q, and *vice versa*. This can be implemented in a cooperation $P \underset{L}{\bowtie} Q$ by including in L the set of all (active) action types of P, T_P (respectively, of Q, T_Q) leading into E_P (respectively E_Q). Correspondingly, an invisible passive action is introduced at every state of Q not in E_Q (respectively at every state of P not in E_P) for every action type in T_P (respectively, T_Q). In figure 2, $E_P = \{1\}, E_Q = \{1\}, L = \{a, b\}, T_P = \{a\}, T_Q = \{b\}$. The invisible actions then implicitly 'guard' access to the exclusive area in the other process by not being enabled in their own process's exclusive area.

There are no joint changes of state in the cooperation because a passive action leaves its component in the same state. The state transition graph of the cooperation is therefore the Cartesian product of those of the components P and Q with prohibited states removed, together with all arcs leading into and out of them. In the reversed process of the cooperation, an active action, $\overline{a}$ say, is now invisible and causes the corresponding passive action to proceed with rate x_a, when enabled. But x_a is precisely the rate of the reversed active action a of the forward process. Consequently, the reversed cooperation is equivalent to the cooperation of the reversed processes $\overline{P}$ and $\overline{Q}$ with passive actions *remaining* the invisible ones of the forward process and active actions reversed (and still active) in their own, independently reversed component.

In other words, the rate from state (i, j) to (i', j) in the reversed cooperation's transition graph is the rate from i to i' in $\overline{P}$, i.e. $\overline{p_{ii'}}$. Similarly, the rate from (i, j) to (i, j') in the reversed cooperation is the rate from j to j' in $\overline{Q}$, i.e. $\overline{q_{jj'}}$. Hence, the state transition graph of the reversed cooperation is simply the Cartesian product of those of the reversed components with prohibited states and arcs removed. Consequently, by proposition 1, the equilibrium probability of a valid state (i, j) is proportional to $\pi_P(i)\pi_Q(j)$ where $\pi_P(.)$ and $\pi_Q(.)$ are the equilibrium probability mass functions of P and Q respectively.

The method extends inductively to arbitrarily many (n) cooperating processes,

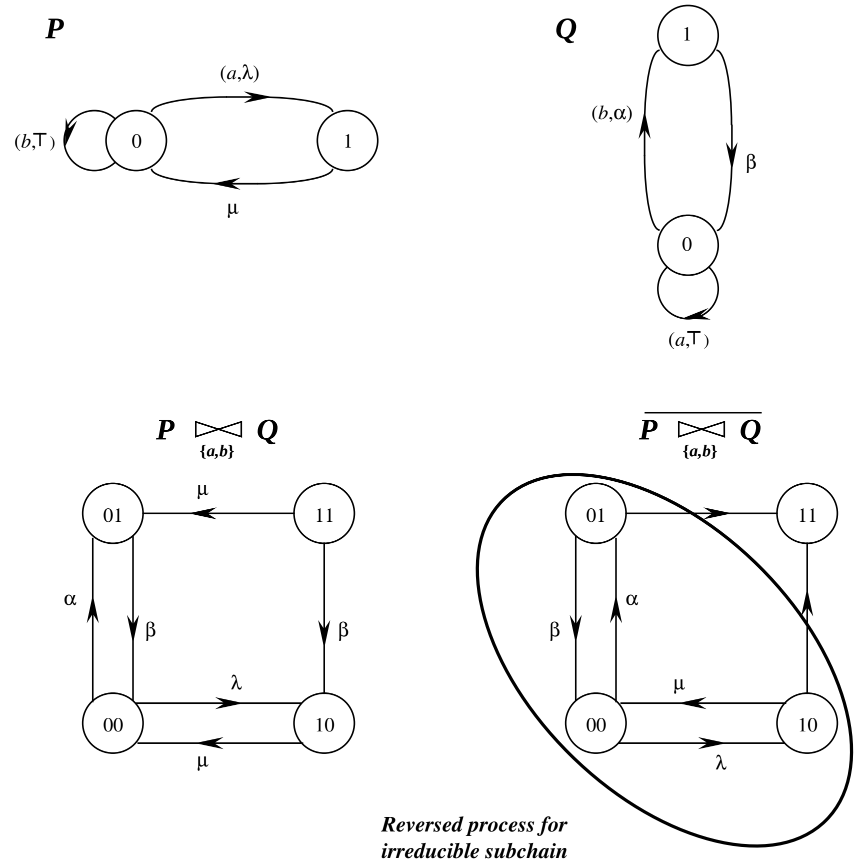

Figure 2. Mutual exclusion implemented by two invisible passive actions

$P_1 \underset{L_1}{\bowtie} P_2 \underset{L_2}{\bowtie} P_3 \underset{L_3}{\bowtie} \ldots \underset{L_{n-1}}{\bowtie} P_n$, giving the result that the equilibrium probability of valid state vector $\vec{n}$ is proportional to $\prod_{i=1}^{n} \pi_{P_i}(n_i)$. Note that renormalisation is necessary because of the omitted, invalid states. This is very similar to the result of Boucherie [2], which states that the same product-form solution holds when a process P in its critical section E_P *blocks* the other processes; i.e. the other processes cannot undergo *any* transitions.

This system can be modelled by labelling every action in a cooperating process Q as active, with types $a_1, a_2, \ldots, a_n$ say, and including the corresponding types as invisible passive actions on every state of P outside E_P. Equation 1 (and so RCAT, since there is only one instance of each action type) is satisfied in a two-component cooperation if

$$\sum_{a \in \mathcal{A}^{(i,j)\leftarrow} \setminus \mathcal{P}^{(i,j)}} x_a = \sum_{a \in \mathcal{A}^{(i,j)\rightarrow} \setminus \mathcal{P}^{(i,j)}} \alpha_a^{(i,j)} \tag{2}$$

for all valid states (i, j). When $i \notin E_P$ and $j \notin E_Q$, all passive actions are enabled and the equality is true vacuously. When $i \in E_P$ and $j \notin E_Q$, all passive actions in Q are enabled and all those in P are not, and the equation states

$$\sum_{a \in \mathcal{A}_Q^{j\leftarrow}} x_a = \sum_{a \in \mathcal{A}_Q^{j\rightarrow}} p_{ia}$$

This is true by the first of Kolmogorov's extended criteria (applied to process R in RCAT) since the reversed rate of each active action type a is x_a. The case $i \notin E_P$ and $j \in E_Q$ is similar and $i \in E_P$, $j \in E_Q$ yields an invalid state. This argument extends trivially to an arbitrary number of cooperating processes, whereupon the result of [2] follows.

4. Last-come-first-served (LCFS) queues

4.1. The server view

Last-come-first-served queues may be described using a *resource-oriented view* in which the state of the queue is a vector, the ith component of which is the current status of the ith highest priority customer in the queue. Thus, pre-emptive/non-pre-emptive arrivals to the queue enter at component 1/2, shifting the other components to the right. Similarly, departures cause a state transition in which the first component is deleted, the other components shift to the left and the new first component (old second component) receives service. The status of a customer describes the service so far received. At a server with *Coxian* service times of the type shown in figure 3, a customer's status is the stage of service to be entered on resumption, an integer s, $1 \leq s \leq S$ where S is the number of stages at the server. The delay at stage s is an exponential random variable with parameter μ_s, and so the remaining time in a stage is an identical exponential random variable by the memoryless property (see [8], for example). The quantities a_s are the probabilities that, after passing through stage s, a customer passes to the next stage $s + 1$, where $a_S = 0$. A customer completing service at stage s therefore immediately departs with probability $1 - a_s$.

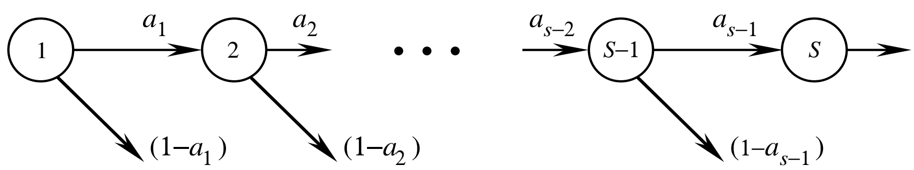

Figure 3. Coxian server

4.2. The customer view

Alternatively, the queue can be described from the *customer viewpoint.* Each customer at a Coxian server is in one of the following three classes of states:

1. Not at the server, represented by state 0;

2. In service at stage s, $1 \leq s \leq S$;
3. Suspended from service at stage s since a new customer has arrived, $1 \leq s \leq S$.

Such a customer, C say, can be described by the process depicted in figure 4. Here, a Poisson (rate λ) arrival source process P, defined by $P = (b, \lambda).P$ synchronises with the transition $0 \to 1$ in the customer process C. C then passes though a subsequence of the states $1, 2, \ldots, S$ before returning to state 0 on completion of service. To enable the 'exit' action e, we add the invisible passive action e to the definition of P, $P = (e, \top).P$. Intervening arrivals take C to state s' from its current state s by synchronising with action type b. P has a single state and b is an invisible active transition in P with rate λ. The passive action b is enabled in every state of C and, preparing for a later application of RCAT, $x_b = \lambda$.

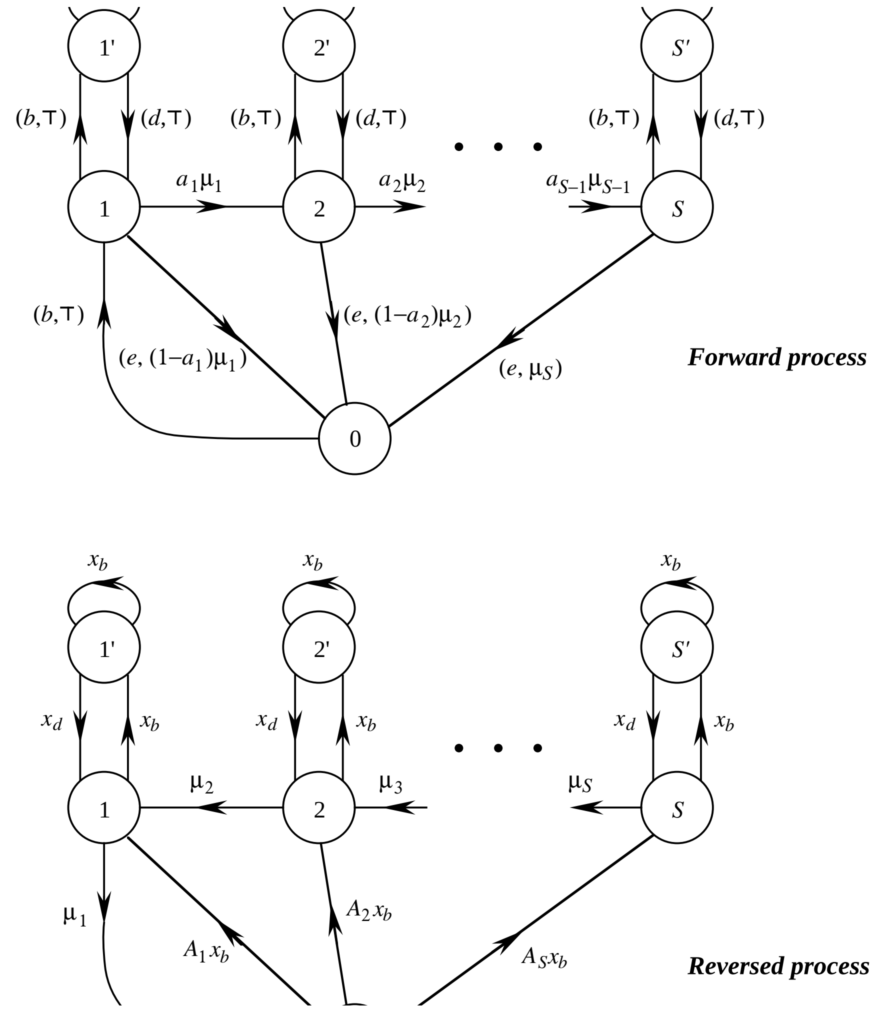

Figure 4. Last-come-first-served customer process, C

4.3. The set of customers at a LCFS queue

The LCFS queue can now be simply described as the cooperation:

$$P\{e \leftarrow d_1\} \underset{\{b,d_1\}}{\bowtie} Q_L$$

where $Q_L = C_1 \underset{\{b,d_2\}}{\bowtie} C_2 \underset{\{b,d_3\}}{\bowtie} \dots$ and $C_i = C\{d \leftarrow d_{i+1}, e \leftarrow d_i\}$. This synchronisation is unbounded but finite for finite queue lengths; there is one instance of C_i for each customer in the queue. Arrivals synchronise with C_1 and propagate to C_2, if present, and beyond, leaving all customers in suspended states unchanged. The customer being served (following the last suspended customer) then changes to a suspended state and its action b propagates to the next customer, which must be in state 0 and passes to state 1, whereupon the propagation stops. The propagation of synchronisations between processes is discussed in [6] by considering 'inserted states'. RCAT remains satisfied if each pairwise cooperation in the chain satisfies it. A synchronisation over $\{b\}$ with a customer process in state 0 is defined *not* to propagate.

The passive actions d_i are not enabled in every state and so the original RCAT cannot be used. However, the conditions of the extended version, theorem 1, are satisfied since certain states of the Cartesian product are invalid, $x_{d_i} = x_a = \lambda$ and $\overline{\beta_b^s} = x_b$ for invisible instances of action type b at states s. In particular, whenever an active action d_n is enabled (in process C_n, the last to arrive), passive action d_n is always enabled in process C_{n-1}, if $n > 1$, or else d_1 is enabled in the source component P. The resulting reversed process is shown in figure 4 and leads to the following product-form for the equilibrium probability that there are n customers in the queue at stages $\mathbf{s} = s_1, \dots, s_n$:

$$\pi_{\mathbf{s}} \propto \prod_{i=1}^{n} \frac{A_i \lambda}{(1 - a_i)\mu_i}$$

where $A_i = (1 - a_i) \prod_{j=1}^{i-1} a_j$.

This is the familiar result for a single LCFS queue used in the proof of the BCMP theorem [1]. We will later use the process $C_1 \underset{\{b,d_2\}}{\bowtie} C_2 \underset{\{b,d_3\}}{\bowtie} \cdots \underset{\{b,d_n\}}{\bowtie} C_n$, call it Q_L^n, to represent a LCFS queue with n customers in a network of queues, noting that its arrival action type b is always enabled. Furthermore, its departure action type d_n synchronises with arrivals at other queues and has reversed (passive) action that is enabled in all valid states of the cooperation Q_L^n. The conditions of the original RCAT are therefore satisfied in the network, treating Q_L^n as a primitive component with known reversed process.

5. Processor sharing queues

In a processor sharing (PS), Coxian queue, all customers receive service concurrently at a reduced rate; any number of customers can be in each stage at the same time. There is therefore no blocking of customers as in first-come-first-served (FCFS) and LCFS queues, where only one customer can be served at any given time. The PS queue with Coxian service times can be modelled as a queueing network in which the customers at stage s receive (combined) service at rate $f(\mathbf{n})\mu_s(n_s)$, where $f(\mathbf{n}) = 1/(n_1 + \ldots + n_S)$, $\mu_s(n_s) = n_s \mu_s$ and $\mathbf{n} = (n_1, \dots, n_S)$ is the state of the server, i.e. n_s is the number of customers at stage s, $1 \le s \le S$. In other words, each customer at stage s receives service at rate $\mu_s/(n_1 + \ldots + n_S)$. More precisely, at time

t, some customer at stage s will complete service at that stage in the interval $(t, t+h]$ with probability $f(\mathbf{n})\mu_s(n_s)h + o(h)$.

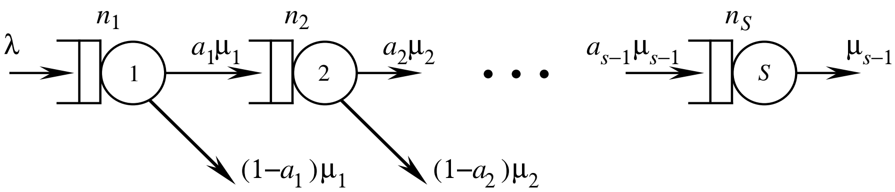

Figure 5. Processor sharing (PS), Coxian queue

This queueing network interpretation of a PS, Coxian server is depicted in figure 5. Because of the non-local state dependence in the service rates, RCAT does not apply and it is not even obvious how to define the cooperations between queues. Nevertheless, we can define a cooperation $P \underset{L}{\bowtie} Q$ to have underlying Markov process with transition rate from state (i, j) to (i', j') equal to:

- The rate from i to i', $p_{ii'}$, in the Markov process of P if either $j = j'$ or P is active in the sychronised transition $(i, j) \to (i', j')$;
- The rate from j to j', $q_{jj'}$, in the Markov process of Q if either $i = i'$ or Q is active in the sychronised transition $(i, j) \to (i', j')$.

In the case of a 2-stage Coxian server, suppose we apply theorem 1 regardless, to pose a possible reversed process, noting that there is still a solution for the constant reversed rates, viz. $x_a = \lambda$, the component queues being reversible. This conjectured reversed process is shown in figure 6.

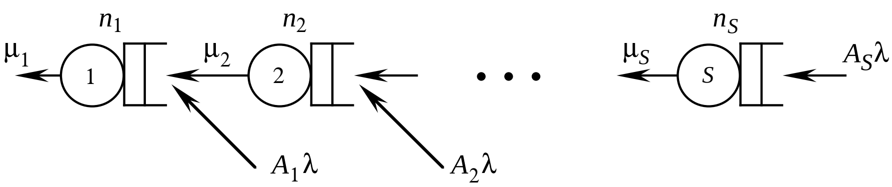

Figure 6. Reversed PS, Coxian queue

The same proof stands that the conditions of theorem 1 (or, the three conditions of the original RCAT) imply that the outgoing rate from all states is the same in the forward and reversed processes. Consequently, we only need to check that products of the rates around cycles is the same in both processes in order to establish Kolmogorov's criteria and so verify the conjecture. In fact it is only necessary to consider *minimal cycles*, i.e. cycles such that a circuit around any cycle can be constructed as a path over arcs comprising the union of a set of complete minimal cycles.

In general, the prospect of checking every minimal cycle is awe-inspiring. However, any minimal cycle in the cooperation must be formed from a (possibly synchronising) pair of minimal cycles in the component processes. In the case of two queues, the component minimal cycles are of length two (an arrival and a departure). Hence it is easy to construct the three possible classes of minimal cycle in the synchronisation:

1. Minimal cycles in either of the components (horizontal or vertical), which do not include synchronisations;
2. Minimal cycles in which both arcs of the components' cycles synchronise, giving a 'diagonal' cycle of length two;
3. Minimal cycles in which exactly one arc from each of the components' cycles synchronise, giving two 'triangles' for each possible cooperation between the two pairs of arcs.

The second of Kolmogorov's criteria obviously holds in the first class, by the hypothesis that the reversed cooperation comprises a cooperation between the reversed components. In the case of the Coxian-based queueing network above, there is only one synchronising action, leading to the two triangular minimal cycles A and B shown in figure 7.

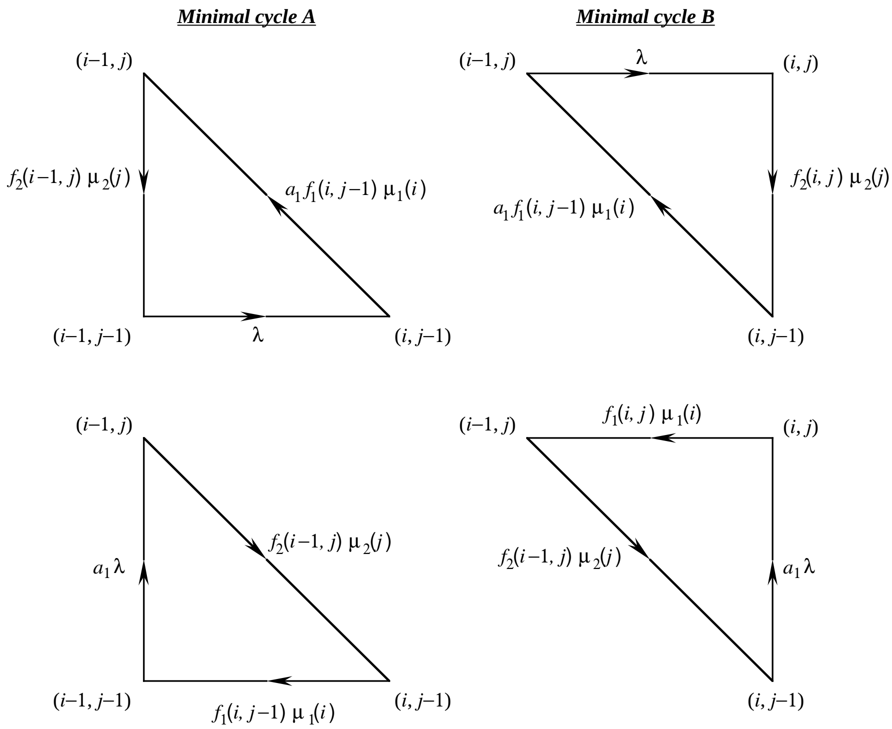

Figure 7. Minimal cycles in the PS coxian queue process

The product of rates around the cycles A is the same in the forward and reversed processes for any function f. This property holds for cycles B if and only if

$$\frac{f_1(i, j-1)}{f_1(i, j)} = \frac{f_2(i-1, j)}{f_2(i, j)}$$

for all states i, j. This condition holds when, for example, $f_1(i, j) = f_2(i, j) = g(i + j)$ for any function g; in particular in the PS model where $g(i, j) = 1/(i + j)$. Notice, however, that other global state dependencies yield a simple symbolic reversed process and equilibrium state probabilities: for example $f_1(i,j) = x^j$ and $f_2(i,j) = x^i$ for any real number x. This yields a new class of non-product-form solutions for the steady state probabilities in a globally state-dependent pair of queues:

$$\pi(i,j) \propto \frac{\lambda^{i+j} a_1^j}{\prod_{k=1}^{i+j} f_1(k,0)\mu_1(k)} \prod_{k=i+1}^{i+j} \frac{f_1(k, i+j+1-k)}{f_2(k, i+j+1-k)} \frac{\mu_1(k)}{\mu_2(i+j+1-k)}$$

The same result extends to networks with arbitrary interconnectivity, not just the feed-forward ones shown here. It is a more general result than that for a 2-stage PS Coxian server, which simplifies to:

$$\pi(i,j) \propto \frac{(i+j)!\lambda^{i+j} a_1^j}{i!j!\mu_1^i \mu_2^j}$$

This is the standard PS result, which extends inductively to S-stage servers to yield:

$$\pi(\mathbf{n}) \propto \frac{n!\lambda^n}{n_1! \dots n_S!} \prod_{i=1}^{S} \left(\frac{A_i}{(1-a_i)\mu_i} \right)^{n_i}$$

where $n = n_1 + \ldots + n_S$. Summing over $n_1, \ldots, n_S$ such that $\sum_{i=1}^S n_i = n$ yields the equilibrium queue length probability (by a routine application of the multinomial theorem)

$$\pi(n) \propto \rho^n$$

where $\rho = \lambda/\overline{\mu}$ and $\overline{\mu}^{-1} = \sum_{i=1}^S A_i \mu_i^{-1}/(1-a_i)$ is the mean service time of the coxian server.

Finally, note that in the PS Coxian queue cooperation, the external arrival actions are enabled in every state, as are the reversed external departure actions. Thus, as with the LCFS queue cooperation, the last condition of theorem 1 (or first two conditions of the original RCAT) will hold in a cooperation of these queue-cooperations.

6. The BCMP theorem

We can now apply the original RCAT [5] to obtain the BCMP product-form [1] for a network of queues which are either:

- FCFS with exponential service time; or
- LCFS, PS or infinite server (IS)† with Coxian service time.

In every case, all passive actions are enabled in every state of both the forward and reversed cooperations; as established above for LCFS and PS queues and clearly for FCFS queues. The required reversed rates x_a are given by the traffic equations. RCAT can therefore be applied, giving the known product-form. Extension to the multi-class case is straightforward, but with heavier notation.

†IS servers are analysed analogously to PS servers.

7. Conclusion

The compositionality of the RCAT, extended here, introduces an entirely new approach to deriving the equilibrium state probabilities in separable Markov processes. This approach does not require balance equations to be solved but instead determines the reversed process whence a simple solution ensues – typically, but not necessarily, of product-form. The origins of the RCAT and the methodology based on it lie in a combination of MPA and the theory of reversed stationary processes.

The methodology derives many product-forms, known and possibly unknown, in a uniform way. This is exemplified here by the derivation, from the same theorem, of the solutions of 'Boucherie' networks, with competition for resources, and BCMP networks. Moreover, significant new (to the author's best knowledge) product-forms have recently been obtained, including G-networks with generalised resets and certain queues with batches and Markov modulation, as well as non-queueing Markov models. The principal advantage of this new approach is that it can be implemented symbolically. By incorporating the methodology into a suitable support environment – possibly, but not necessarily, for process algebras – the derivation of many product-form theorems could be automated and new ones derived in a unified stochastic modelling framework.

REFERENCES

1. F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios. Open, closed and mixed networks of queues with different classes of customers. *Journal ACM*, 22(2):248-260, 1975.
2. R.J. Boucherie. A Characterisation of Independence for Competing Markov Chains with Applications to Stochastic Petri Nets. *IEEE Transactions on Software Engineering*, 20(7):536–544, July 1994.
3. E. Gelenbe. Queueing networks with negative and positive customers. *Journal of Applied Probability*, 28:656–663, 1991.
4. E. Gelenbe. G-networks with triggered customer movement. *Journal of Applied Probability*, 30:742–748, 1993.
5. Peter G Harrison. Turning Back Time in Markovian Process Algebra. *Theoretical Computer Science*, 2002.
6. P.G. Harrison. Mechanical Solution of G-networks via Markovian Process Algebra. In proc. International Conference on Stochastic Modelling and the IV International Workshop on Retrial Queues, Cochin, India, December 2002, *Notable Publications*, 2002.
7. Peter G Harrison. Product-forms from a CAT and DOG. *Performance Evaluation Review*, 2002.
8. P.G. Harrison and N.M. Patel. *Performance Modelling of Communication Networks and Computer Architectures.* Addison-Wesley, 1992.
9. J. Hillston. *A Compositional Approach to Performance Modelling.* PhD thesis, University of Edinburgh, 1994.
10. J.R. Jackson. Jobshop-like queueing systems. *Management Science*, 10(1):131–142, 1963.
11. F.P. Kelly. *Reversibility and Stochastic Networks.* Wiley, 1979.