RCAT: From PEPA to Product form

Jeremy T. Bradley*

Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2BZ, United Kingdom
Web: http://www.doc.ic.ac.uk/ jb/

1 Introduction

The motivation behind RCAT [1] is very simple: finding steady-state solution of large Markov processes is computationally very hard, is it possible to break up the calculation into more manageable pieces and recombine the smaller calculations to regain the full steady-state solution.

This is also the idea behind product form processes; processes whose global steady state distribution is proportional to the product of the steady-state distributions of the smaller components in the system. The simplest of these product forms is Jackson's theorem [2] that applies to open queueing networks.

Whereas product forms such as Jackson's theorem operate over queueing networks, RCAT operates over the stochastic process algebra components defined in PEPA [3]. In the same way that Jackson's theorem relates the global steady-state probability to the steady-state distributions of the constituent queueing nodes, so RCAT relates the steady state distribution of the global state space of the PEPA model to that of the individual PEPA components.

2 Reversed Processes

The reversed process of a stationary Markov process $\{X_t : t \geq 0\}$ with state space S, generator matrix Q and stationary probabilities $\vec{\pi}$ is a stationary Markov process with generator matrix Q' defined by:

$$q'_{ij} = \frac{\pi_j q_{ji}}{\pi_i} \qquad : i, j \in S \tag{1}$$

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

*With help from Peter Harrison and Ashok Argent-Katwala. Produced for 336 Performance Analysis course 16.03.2007.

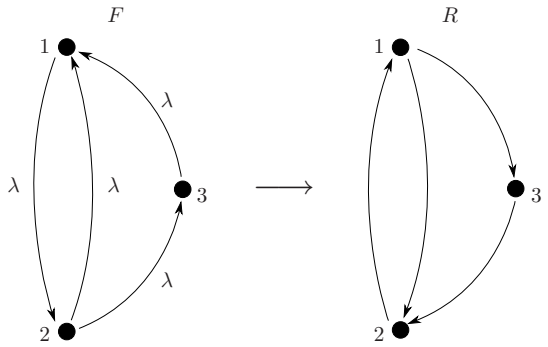


Fig. 1. A forward Markov process, F, and its reversed counterpart, R

This means that, if we have both the forward and reversed processes, we can calculate the equilibrium or steady-state distribution. This is the primary motivation for RCAT. By providing a straightforward technique for constructing the reversed process of a PEPA model, we can more easily find the equilibrium distribution of the original PEPA model.

Theorem 1. *(Kolmogorov's Generalised Criteria) A stationary Markov process with state space S and generator matrix Q has reversed process with generator matrix Q' if and only if:*

1. $q'_i = q_i$ *for every state* $i \in S$
2. *For every finite sequence of states* $i_1, i_2, ..., i_n \in S$,

$$q_{i_1 i_2} q_{i_2 i_3} \cdots q_{i_{n-1} i_n} q_{i_n i_1} = q'_{i_1 i_n} q'_{i_n i_{n-1}} \cdots q'_{i_3 i_2} q'_{i_2 i_1}$$

where $q_i = -q_{ii} = \sum_{j\,:\,j \neq i} q_{ij}$ *is the total exit rate from state* i.

In Figure 1, we show a simple forward process, F, with generator matrix, Q. In the reversed process, R, we can use the Kolmogorov Generalised Criteria to discover the reversed rates in its generator matrix, Q'. Thus comparing total outbound rates: $q_1 = \lambda, q_2 = 2\lambda, q_3 = \lambda$ gives, $q'_1 = q'_{12} + q'_{13} = \lambda, q'_2 = q'_{21} = 2\lambda, q'_3 = q'_{32} = \lambda$. There are two minimal cycles in F, $\{1 \to 2, 2 \to 3, 3 \to 1\}$ and $\{1 \to 2, 2 \to 1\}$, gives the following cycle equations:

$$q'_{13} q'_{32} q'_{21} = \lambda^3 \qquad q'_{12} q'_{21} = \lambda^2$$

Resolving these gives: $q'_{12} = \lambda/2, q'_{13} = \lambda/2, q'_{21} = 2\lambda, q'_{32} = \lambda$.

We will use this style of reasoning to determine the reversed rates of PEPA actions in sequential PEPA components later on.

Once the reversed process rates in Q' have been found, these can be used to extract the equilibrium distribution, $\vec{\pi}$. This can be done directly from Equation (1) or by using a modified version that relates any π_j entry in the steady

state vector to a base value, π_0.

$$\pi_j = \pi_0 \prod_{i=0}^{j-1} \frac{q_{i,i+1}}{q'_{i+1,i}} = \pi_0 \prod_{i=0}^{j-1} \frac{q'_{i,i+1}}{q_{i+1,i}} \tag{2}$$

This holds for an irreducible Markov process, where we choose a reference state 0 arbitrarily and find a sequence of connected states, in either the forward or reversed process, connecting state 0 to state j. Specifically, we require either $q_{i,i+1} > 0$ or $q'_{i,i+1} > 0$ for $0 \leq i \leq j-1$ for any state j.

3 PEPA

3.1 The PEPA language

RCAT is based on the stochastic process algebra PEPA. PEPA [3] is a parsimonious stochastic process algebra that can describe compositional stochastic models. These models consist of components whose actions incorporate random exponential delays.

The syntax of a PEPA component, P, is represented by:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A \tag{3}$$

$(a, \lambda).P$ is a prefix operation. It represents a process which does an action, a, and then becomes a new process, P. The time taken to perform a is described by an exponentially distributed random variable with parameter λ. The rate parameter may also take the value $\top$, which makes the action passive in a cooperation (see below).

$P_1 + P_2$ is a choice operation. A race is entered into between components P_1 and P_2. If P_1 evolves first then any behaviour of P_2 is discarded and vice-versa.

$P_1 \underset{L}{\bowtie} P_2$ is the cooperation operator. P_1 and P_2 run in parallel and synchronise over the set of actions in the set L. If P_1 is to evolve with an action $a \in L$, then it must first wait for P_2 to reach a point where it is also capable of producing an a-action, and vice-versa. In an active cooperation, the two components then jointly produce an a-action with a rate that reflects the slower of the two components (usually the minimum of the two individual a-rates). In a passive cooperation, where P_1, say, can evolve with an $(a, \top)$-transition, the joint a-action inherits its rate from the P_2 component alone.

P/L is a hiding operator where actions in the set L that emanate from the component P are rewritten as silent τ-actions (with the same appropriate delays). The actions in L can no longer be used in cooperation with other components.

A is a constant label and allows, amongst other things, recursive definitions to be constructed.

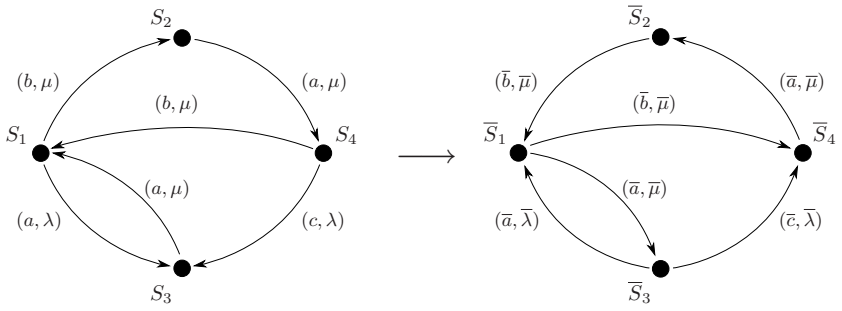


Fig. 2. A sequential PEPA component and its reversed form

3.2 Reversing a sequential component

Reversing a sequential PEPA component, S, is straightforward. The RCAT tackles the reversal of a parallel PEPA component, $P \underset{L}{\bowtie} Q$, and makes use of reversing sequential components in its definition.

Definition 1. *For all states, S, in a sequential component:*

$$\overline{S} \stackrel{\text{def}}{=} \sum_{i\,:\,R_i \xrightarrow{(a_i,\lambda_i)} S} (\overline{a}_i, \overline{\lambda}_i).\overline{R}_i$$

The above definition, states that a reversed sequential component, $\overline{S}$, is defined to be a choice between all of the states that have S as a 1-step successor in the forward process.

In simple terms, this literally means reversing the direction of the transitions in the local state space of the component, as shown in Figure 2.

Actions a become $\overline{a}$ in the reversed component; rates λ become $\overline{\lambda}$ in the reversed component. $\overline{\lambda}$ may be a function of many forward rates and it can be calculated using Kolmogorov's Generalised Criteria, as has been shown in Section 2.

3.3 PEPA activity substitution

The substitution function allows an activity $\alpha = (a, r)$ in a PEPA model to be syntactically replaced with the activity $\alpha' = (a', r')$:

Definition 2. *The activity substitution function is defined as follows:*

$$\begin{aligned}
(\beta.P)\{\alpha \leftarrow \alpha'\} &= \begin{cases} \alpha'.(P\{\alpha \leftarrow \alpha'\}) & : \textit{if } \alpha = \beta \\ \beta.(P\{\alpha \leftarrow \alpha'\}) & : \textit{otherwise} \end{cases} \\
(P + Q)\{\alpha \leftarrow \alpha'\} &= P\{\alpha \leftarrow \alpha'\} + Q\{\alpha \leftarrow \alpha'\} \\
(P \underset{L}{\bowtie} Q)\{\alpha \leftarrow \alpha'\} &= P\{\alpha \leftarrow \alpha'\} \underset{L\{\alpha \leftarrow \alpha'\}}{\bowtie} Q\{\alpha \leftarrow \alpha'\}
\end{aligned}$$

$$\textit{where } L\{(a,\lambda) \leftarrow (a',\lambda')\} = \begin{cases} (L \setminus \{a\}) \cup \{a'\} & : \textit{if } a \in L \\ L & : \textit{otherwise} \end{cases}$$

It is used by RCAT to replace passive activities with active ones (and active activities with passive ones) when reversing a parallel component. This is necessary, since one of the side-effects of applying reversal to a parallel component, $P \underset{L}{\bowtie} Q$, is that cooperating actions that were previously active in the component P and passive in Q become passive in $\overline{P}$ and active in $\overline{Q}$ (and vice versa).

4 Reversed Compound Agent Theorem (RCAT)

RCAT defines the reversed PEPA process $\overline{P \underset{L}{\bowtie} Q}$ in terms of slight modifications of the reversed processes of P and Q.

First we restrict the PEPA model that RCAT operates over. An action is assumed not to relate to both an active and passive activity within the same component, i.e. if an action is active in a component, all its instances are active in that component, if it is passive then all instances of that activity are passive.

Definition 3. *The subset of action types in a set L which are* passive *with respect to a process P (i.e. are of the form $(a, \top)$ in P) is denoted by $\mathcal{P}_P(L)$. The set of corresponding active action types is denoted $\mathcal{A}_P(L) = L \setminus \mathcal{P}_P(L)$.*

Henceforth, when referring to a reversed agent, we mean an agent that satisfies Kolmogorov's criteria: the agent will define a reversed process if and only if the original process was stationary.

Theorem 2. *(Reversed Compound Agent Theorem)*
Suppose that the cooperation $P \underset{L}{\bowtie} Q$ has a derivation graph with an irreducible subgraph G. Given that:

1. *every passive action type in $\mathcal{P}_P(L)$ or $\mathcal{P}_Q(L)$ is always enabled in P or Q respectively (i.e. enabled in all states of the transition graph);*
2. *every reversed action of an active action type in $\mathcal{A}_P(L)$ or $\mathcal{A}_Q(L)$ is always enabled in $\overline{P}$ or $\overline{Q}$ respectively;*
3. *every occurrence of a reversed action of an active action type in $\mathcal{A}_P(L)$ or $\mathcal{A}_Q(L)$ has the same rate in $\overline{P}$ or $\overline{Q}$ respectively.*

the reversed agent $\overline{P \underset{L}{\bowtie} Q}$, with derivation graph containing the reversed subgraph $\overline{G}$, is:

$$R^* \underset{\overline{L}}{\bowtie} S^*$$

where:

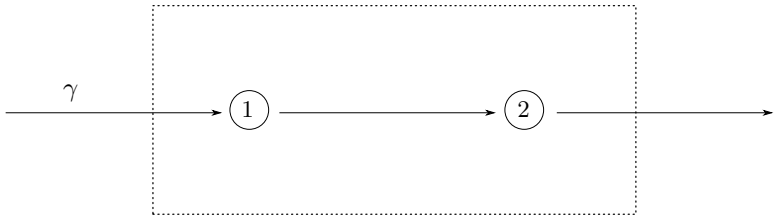
$$\begin{aligned} R^* &= \overline{R}\{(\overline{a}, \overline{p}_a) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_P(L)\} \\ S^* &= \overline{S}\{(\overline{a}, \overline{q}_a) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_Q(L)\} \\ R &= P\{(a, \top) \leftarrow (a, x_a) \mid a \in \mathcal{P}_P(L)\} \\ S &= Q\{(a, \top) \leftarrow (a, x_a) \mid a \in \mathcal{P}_Q(L)\} \end{aligned}$$

where the symbolic rates $\{x_a\}$ *are given by:*

$$x_a = \begin{cases} \overline{q}_a & : \textit{if } a \in \mathcal{P}_P(L) \\ \overline{p}_a & : \textit{if } a \in \mathcal{P}_Q(L) \end{cases}$$

and $\overline{p}_a$, $\overline{q}_a$ *are the symbolic rates of action types* $\overline{a}$ *in* $\overline{P}$ *and* $\overline{Q}$ *respectively.*

5 Example: Tandem queues



A tandem queue system has two M/M/1 queueing nodes with the output from queue 1 becoming the input to queue 2. There is an external input at rate γ into queue 1. The service rate at node i is μ_i for $1 \leq i \leq 2$.

This can be modelled in PEPA with e representing an external arrival, a representing an internal transfer between queues and d representing a departure from the network.

$$\begin{aligned}
Sys &\stackrel{def}{=} P_0 \underset{\{a\}}{\bowtie} Q_0 \\
P_0 &\stackrel{def}{=} (e,\gamma).P_1 \\
P_n &\stackrel{def}{=} (e,\gamma).P_{n+1} + (a,\mu_1).P_{n-1} \qquad : n > 0 \\
Q_0 &\stackrel{def}{=} (a,\top).Q_1 \\
Q_n &\stackrel{def}{=} (a,\top).Q_{n+1} + (d,\mu_2).Q_{n-1} \qquad : n > 0
\end{aligned}$$

Constructing the subprocesses R and S in the theorem leaves P unchanged. Substituting the passive activity, $(a,\top)$ in Q gives S.

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

This gives:

$$\begin{aligned}
R_0 &\stackrel{def}{=} (e,\gamma).R_1 \\
R_n &\stackrel{def}{=} (e,\gamma).R_{n+1} + (a,\mu_1).R_{n-1} \qquad : n > 0 \\
S_0 &\stackrel{def}{=} (a,x_a).S_1 \\
S_n &\stackrel{def}{=} (a,x_a).S_{n+1} + (d,\mu_2).S_{n-1} \qquad : n > 0
\end{aligned}$$

Reversing the sequential components R and S using the definition of Section 3.2 gives us:

$$\begin{array}{lcll}
\overline{R}_0 & \stackrel{def}{=} & (\overline{a}, \overline{\mu}_1).\overline{R}_1 & \\
\overline{R}_n & \stackrel{def}{=} & (\overline{a}, \overline{\mu}_1).\overline{R}_{n+1} + (\overline{e}, \overline{\gamma}).\overline{R}_{n-1} & : n > 0 \\
\overline{S}_0 & \stackrel{def}{=} & (\overline{d}, \overline{\mu}_2).\overline{S}_1 & \\
\overline{S}_n & \stackrel{def}{=} & (\overline{d}, \overline{\mu}_2).\overline{S}_{n+1} + (\overline{a}, \overline{x}_a).\overline{S}_{n-1} & : n > 0
\end{array}$$

At this stage we can calculate the reverse rates, $\overline{\mu}_1, \overline{\mu}_2, \overline{x}_a, \overline{\gamma}$ in terms of the forward rates, by using the Kolmogorov Generalised Criteria as used in Section 2.

Comparing exit rates from R_0, S_0 with those from $\overline{R}_0, \overline{S}_0$ gives $\overline{\mu}_1 = \gamma$, $\overline{\mu}_2 = x_a$. At this stage we don't worry that some of the forward rates are undefined.

Comparing the minimal cycle, $\{R_0 \to R_1, R_1 \to R_0\}$ with $\{\overline{R}_0 \to \overline{R}_1, \overline{R}_1 \to \overline{R}_0\}$ and $\{S_0 \to S_1, S_1 \to S_0\}$ with $\{\overline{S}_0 \to \overline{S}_1, \overline{S}_1 \to \overline{S}_0\}$, gives $\overline{\mu}_1\overline{\gamma} = \gamma\mu_1$, $\overline{\mu}_2\overline{x}_a = x_a\mu_2$.

Solving these gives us $\overline{x}_a = \mu_2$, $\overline{\gamma} = \mu_1$, straightforwardly.

Solving for x_a is given by the last part of RCAT:

$$\begin{array}{rcl}
x_a & = & \left\{ \begin{array}{ll} \overline{q}_a & : \text{if } a \in \mathcal{P}_P(L) \\ \overline{p}_a & : \text{if } a \in \mathcal{P}_Q(L) \end{array} \right. \\
& = & \overline{p}_a \quad : \text{since } a \in \mathcal{P}_Q(L) \\
& = & \overline{\mu}_1 \\
& = & \gamma
\end{array}$$

Constructing R^* and S^* is a matter of substituting $\top$ rates into the relevant component.

$$\begin{array}{rcl}
R^* & = & \overline{R}\{(\overline{a}, \overline{p}_a) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_P(L)\} \\
& = & \overline{R}\{(\overline{a}, \overline{\mu}_1) \leftarrow (\overline{a}, \top)\} \\
S^* & = & \overline{S}\{(\overline{a}, \overline{q}_a) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_Q(L)\} \\
& = & \overline{S}
\end{array}$$

Giving, the reversed process:

$$\begin{array}{lcll}
Sys & \stackrel{def}{=} & R^* \underset{\{\overline{a}\}}{\bowtie} S^* & \\
R^*_0 & \stackrel{def}{=} & (\overline{a}, \top).R^*_1 & \\
R^*_n & \stackrel{def}{=} & (\overline{a}, \top).R^*_{n+1} + (\overline{e}, \mu_1).R^*_{n-1} & : n > 0 \\
S^*_0 & \stackrel{def}{=} & (\overline{d}, \gamma).S^*_1 & \\
S^*_n & \stackrel{def}{=} & (\overline{d}, \gamma).S^*_{n+1} + (\overline{a}, \mu_2).S^*_{n-1} & : n > 0
\end{array}$$

To construct the product form from the forward and reversed processes, we have to look for a path from a start state (e.g. (P_0, Q_0)) to a generic state (P_m, Q_n). By inspection this can be achieved by queue P seeing $m+n$ arrivals or e-actions followed by n a-actions to transfer n jobs from queue P to queue Q.

Using Equation (2), we get:

$$\begin{aligned}
\pi(P_m, Q_n) &= \pi(P_0, Q_0) \prod_{i=0}^{j-1} \frac{q_{i,i+1}}{q'_{i+1,i}} \\
&= \pi(P_0, Q_0) \prod_{i=0}^{m+n-1} \frac{\gamma}{\mu_1} \times \prod_{i=0}^{n-1} \frac{\mu_1}{\mu_2} \\
&= \pi(P_0, Q_0) \left(\frac{\gamma}{\mu_1}\right)^{m+n} \left(\frac{\mu_1}{\mu_2}\right)^{n} \\
&= \pi(P_0, Q_0) \rho_1^m \rho_2^n
\end{aligned}$$

where $\rho_i = \gamma/\mu_i$ is the utilisation of node P for $i = 1$ and Q for $i = 2$. This agrees with the product form obtained from Jackson's Theorem for open queueing networks.

References

[1] P. G. Harrison, "Turning back time in Markovian process algebra," *Theoretical Computer Science*, vol. 290, pp. 1947–1986, January 2003.

[2] J. R. Jackson, "Jobshop-like queueing systems," *Management Science*, vol. 10, no. 1, pp. 131–142, 1963.

[3] J. Hillston, *A Compositional Approach to Performance Modelling*, vol. 12 of *Distinguished Dissertations in Computer Science*. Cambridge University Press, 1996.