Constructing a Residual Density Operator*

Jeremy Bradley†

Last updated: 26 July 2000

1 The Problem

The problem occurs when we try and write down the state space of two or more concurrent random events, as found in queueing problems. Typically we wish to assign a state to moments when one or other of the random events completes and restarts. In the queueing analogy this would be when either a service or arrival occurs.

The issue is that at these restarting moments the partner process which has not completed has a residual amount of distribution left to run. If the partner process is Markovian, we can use the nice memoryless property and we have no need to calculate the residual distribution explicitly. In general we do not have this luxury. We have devised an operator to extract this residual distribution, thus allowing us to write down the state spaces of some interesting flavours of finite queue.

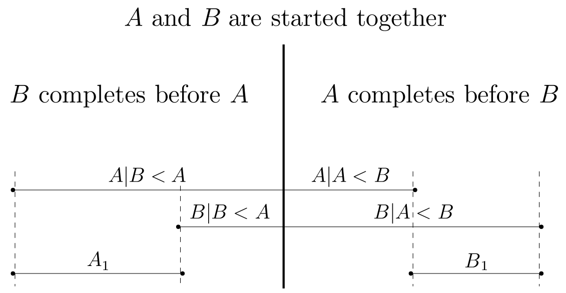
As will be seen the current definition of the operator is not wholly satisfactory and has some odd side-effects when used in simple calculations. On the other hand we as yet can see no better way of defining the operator and we would especially like to know when it is legal (formally, well defined I suppose) to use it.

We call these state spaces—where a state is generated for every restart of a process—a stochastic transition system [1]. In probabilistic analysis, it is analogous to a semi-Markov process [2].

*Thanks to Peter Harrison and Will Knottenbelt for clarifying some ideas.
†Department of Computer Science, University of Durham, South Road, Durham DH1 3LE

2 Defining the Residual Operator

Let us start with two random processes A and B. These are both started (sampled) simultaneously and allowed to run to completion once, as in figure 1.



$$A_1 = (A|B < A) \ominus (B|B < A)$$

$$B_1 = (B|A < B) \ominus (A|A < B)$$

Fig. 1. The residual distributions A_1 and B_1 in the event that B finishes first and A finishes first, respectively.

In figure 1 we have conditioned the A and B events on the premise that either A completes first, or B completes first. This is because we want our residual to be a positive random variable. As can be seen from the diagram, if A finishes first then there is a residual amount of B still left to run—it is this amount that we wish to express in terms of a pdf.

It is also convenient to condition the variables like this from a queue modelling point of view because, in the event that an arrival happens before a service (or vice-versa), we know that this means a change in the state of the buffer.

Originally in [1], we define the residual operator, $\ominus$, based on random variable convolution. If $X + Y = Z$ we define $Y = Z \ominus X$, and since in terms of Laplace functions, $L_Z(z) = L_X(z)L_Y(z)$, we can also provide a method of reproducing Y, namely[1]:

$$L_Y(z) = L_{Z\ominus X}(z) = \frac{L_Z(z)}{L_X(z)} \tag{1}$$

So we can define A_1 and B_1 appropriately:

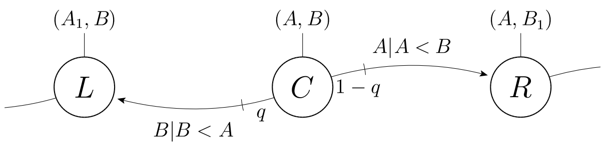
$$B_1 = (B|A < B) \ominus (A|A < B) \quad \text{and} \quad A_1 = (A|B < A) \ominus (B|B < A) \tag{2}$$

[1] Maybe this is the issue, there is no notion of conditioning in the general definition of $\ominus$: maybe there needs to be for this to be well-defined?

3 Using the Residual Operator

We start with an abstract notion of two repeatedly resampling processes, A and B running concurrently: there is as yet no notion of a buffer.

In figure 2, we have three states, C, L and R. We start two processes, A and B off in state C and therefore attribute the state the pair (A, B), meaning that there is A time to run before the first process is resampled and B time to run before the second process renews.



$q = \mathbb{P}(B < A)$

$A_1 = (A|B < A) \ominus (B|B < A)$

$B_1 = (B|A < B) \ominus (A|A < B)$

Fig. 2. A diagrammatic notation for explaining the stochastic transition system of a state in a queue.

The system then evolves to either of state L or state R according to which process finishes first. If the system goes to state R then $A < B$ and the observed event is the conditioned $A|A < B$ as discussed earlier. The A event is resampled and the B event takes on the residual taking into account the $A|A < B$. Thus the new state R has the pair (A, B_1) where $B_1 = (B|A < B) \ominus (A|A < B)$, again by the arguments of section 2.

4 Modelling Examples

If we are modelling a queue then a state is not just identified by its residual distribution pair, but by its pair and the state of the buffer. A G/G/1/K queue has an infinite representation but both M/G/1/K and G/M/1/K have finite representations using this technique.

Figures 3 and 4 show M/G and G/M[2] queues which have been expressed using the notation of figure 2.

[2]There is a subtlety with the G/M/1/3 representation in that we are assuming that the arrival process is turned off when the buffer is full—this is non-standard, but generating the transition system for a *normal* G/M/1/K is not hard either.

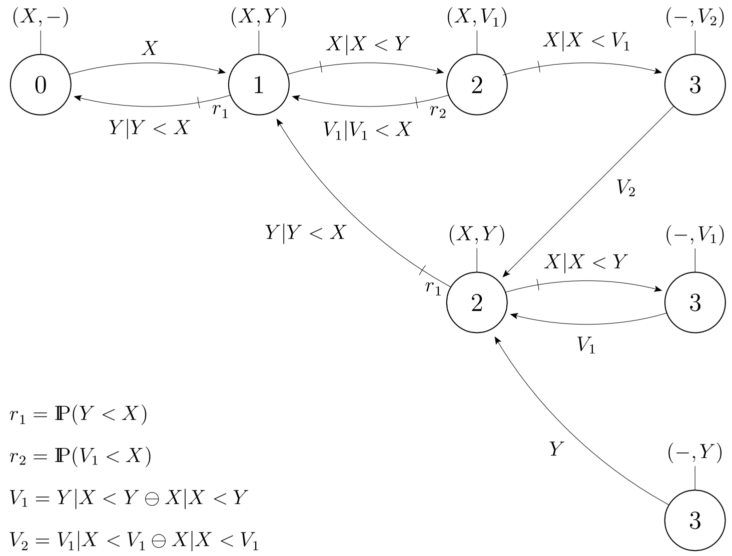


Fig. 3. An M/G/1/3 queue with arrival process X and service process Y.

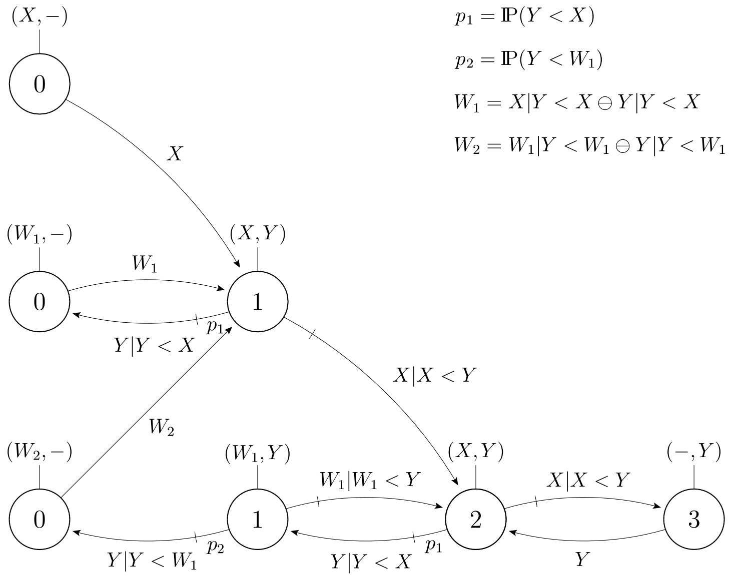


Fig. 4. A G/M/1/3 queue with arrival process X and service process Y.

What benefit do we gain from this expansion: well by letting V_i be shorthand for $V_i^{(X,Y)}$ and similarly for W_i as $W_i^{(X,Y)}$, also r_i as $r_i^{(X,Y)}$ and p_i as $p_i^{(X,Y)}$. By taking $V_0^{(X,Y)} = Y$ and $W_0^{(X,Y)} = X$:

$$V_{i+1}^{(X,Y)} = (V_i^{(X,Y)}|X < V_i^{(X,Y)}) \ominus (X|X < V_i^{(X,Y)}) \tag{3}$$

$$W_{i+1}^{(X,Y)} = (W_i^{(X,Y)} | Y < W_i^{(X,Y)}) \ominus (Y | Y < W_i^{(X,Y)}) \tag{4}$$

$$r_{i+1}^{(X,Y)} = \mathbb{P}(V_i^{(X,Y)} < X), \qquad p_{i+1}^{(X,Y)} = \mathbb{P}(Y < W_i^{(X,Y)}) \tag{5}$$

We get the rather pleasant symmetry that holds for any size of finite M/G queue and this particular flavour of G/M queue where:

$$V_i^{(X,Y)} = W_i^{(Y,X)} \tag{6}$$

and so:

$$p_i^{(X,Y)} = 1 - r_i^{(Y,X)} \tag{7}$$

By inspection of the transition systems we can say that any analytic (in terms of X and Y distributions) state-based metric (i.e. steady state, transient state, passage-time distributions) for a given state in a blocked arrival G/M/1/K queue is related to its dual state in an M/G/1/K queue, by simple interchanging of X and Y quantities. (State 0 maps to state K, state 1 to state $K-1$, state i to state $K-i$.)

5 Oddities of the Residual Operator

As a standard result we can demonstrate exponential memorylessness in the instance where an exponential distribution Y is pre-empted by an arbitrary distribution X and what remains is still exponential. This is shown (and easily proved) by the result:

$$Y | X < Y \ominus X | X < Y = Y \tag{8}$$

This gives us a small amount of confidence in the operator.

The difficulties arise when we consider variance. In the equation $X + Y = Z$, we know that $\mathbf{Var}(Z) \geq \mathbf{Var}(X)$ or $\mathbf{Var}(Y)$, which means that in the equation $Y = Z \ominus X$, we are in trouble if $\mathbf{Var}(X) > \mathbf{Var}(Z)$.

In the case of an M/D/1/K queue, where $V_1 = Y|X < Y \ominus X|X < Y$ and $Y \sim \delta(\mu)$. We know that $Y|X < Y \sim \delta(\mu)$ and so has variance 0 and for most distributions of X (i.e. any which are not deterministic), we have constructed a V_1 quantity with negative variance.

This, it has to be said, gives us less confidence in the operator!

However the really bizarre twist is that we can still use these negative variance distributions and get meaningful and correct results. Its been our experience that if we construct the Laplace distribution of these odd distributions and use the Laplacians in situations where we are not going to invert them i.e. finding the steady-state quantity P_i in [2, Prop.4.8.1,p.131], we can still get verifiably correct results.

It has been suggested that we just have to find a different derivation of the residual operator, but having tried quantities like $(Y - X)|X < Y$ and $(Y - \min(X, Y))|X < Y$, we are at a loss to know where to go from here. It is difficult to see how the operator could be constructed in a way other than it has and yet at the least its going to need some preconditions for use. Suggestions welcome[3].

References

[1] BRADLEY, J. T. *Towards Reliable Modelling with Stochastic Process Algebras.* PhD thesis, Department of Computer Science, University of Bristol, Bristol BS8 1UB, UK, October 1999.

[2] ROSS, S. M. *Stochastic processes.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1982.

[3]Email: jb@doc.ic.ac.uk from 1 August 2001