Stochastic Bounding of PEPA Models

Michael J. A. Smith[*][†]

Abstract

Stochastic bounds are a valuable tool for analysing large Markov chains. If the chain is too large to solve, we can construct upper and lower bounding chains that are easier to solve, and whose steady state solutions will bound that of the original chain. In [Fourneau *et al*, 2004], an algorithm is presented to construct lumpable bounding chains, which can be aggregated and solved.

In a stochastic process algebra, models are constructed compositionally, and so a natural extension of the above work is to construct bounds compositionally. We show how to do this for a class of models in the Performance Evaluation Process Algebra (PEPA). We present the necessary conditions for component bounds to be preserved under composition, and outline how to algorithmically compute these.

1   Introduction

The Performance Evaluation Process Algebra (PEPA) [6] is a stochastic process algebra that allows continuous time Markov chains (CTMCs) to be expressed in a compositional way. We are nearly always interested in modelling systems that contain many components operating in parallel, and this quickly leads to an explosion in the size of the state space. Hence it is very easy to specify a system whose underlying Markov chain is simply too large to solve.

Working at the level of Markov chains, there has been a lot of work in *aggregation*-based approximation techniques. The idea is to look for states that exhibit similar behaviour, and then approximate them by a single state. This is called an *aggregate model*. In this way, we create a smaller Markov chain that *can* be solved, and whose solution is 'close' to what we would obtain by aggregating the solution of the original chain.

What do we mean by the solutions being 'close'? This is a difficult question, and indeed there are very few approximation techniques for which the error can be bounded. In this paper, we will consider just one aggregation technique; that of *ordinary lumpability* [7]. In its exact form, we require that every state in an aggregate has *precisely the same* probability of moving to each of the other aggregates; a very strict requirement indeed. We can relax this somewhat by moving to *quasi-lumpability* [3], but there is currently no way of bounding the error.

If we return to our problem for a moment, there is another way of thinking about it. Essentially, we wish to approximate a Markov chain, but in general this will not be lumpable. Instead, we can modify the original chain so that it *becomes lumpable* but is related to the original chain. The relation that we want is a *stochastic order*. In essence, we find an *upper bound* of our Markov chain, meaning loosely that the steady state solution is an upper bound of the original. We will define this formally in Section 2.1. If we do the same for a *lower bound* of the original chain, we can find an exact bound for the actual solution. An algorithm to do this is described in [5].

Now, up until this point we have discussed only Markov chains. What of the stochastic process algebra models? It is possible to derive the Markov chain for the entire model, and apply the existing algorithm. However, this itself may be too large to represent, and does not take advantage of the structure provided by the process algebra. Taking a different approach, the work in [4] applies stochastic bounds to PEPA models directly, dealing with a restricted class of models. In this paper, we describe how this can be extended to a more general class of model, where we construct a bound for each sequential component, such that the bound is preserved when they are composed.

2   Preliminaries

2.1   Stochastic Comparison

There exist a number of stochastic orderings, for which Stoyan's book [8] is a detailed reference, but for the purposes of this paper we will consider only the *strong stochastic order*, which we will denote $<_{\text{st}}$.

Definition 2.1 *Let X and Y are random variables on a partially ordered space $(S, \prec)$. We say that X is less than Y in the strong stochastic sense, namely that $X <_{\text{st}} Y$, if for all non-decreasing functions f, $E[f(x)] \leq E[f(y)]$.*

This definition is equivalent to saying that $\mathbf{Pr}(X \succ k) \leq \mathbf{Pr}(Y \succ k)$, for all $k \in S$.

The strong stochastic order extends naturally to Markov chains. The following definition applies to both discrete and continuous-time chains.

Definition 2.2 *Let $\{X_t\}$ and $\{Y_t\}$ be Markov chains. Then $\{X_t\} <_{st} \{Y_t\}$ if $X_t <_{st} Y_t$ for all t.*

To get a more useful definition, we need to consider the transition matrices of the Markov chains. In particular, we are interested in two properties of such stochastic matrices: *comparability* and *monotonicity*. We shall assume that the state space of a matrix P (i.e. its row and column indices) is a partially ordered set $(S_P, \prec_P)$, and we shall omit the subscript when it is clear from context. Furthermore, we use the notation $P_{i,*}$ for the row vector containing row i of matrix P.

Definition 2.3 *A stochastic matrix P is monotone if for all row vectors u, v, $u <_{st} v$ implies that $uP <_{st} vP$. Equivalently, for all $x, y \in S$ such that $x \prec y$, $P_{x,*} <_{st} P_{y,*}$.*

Definition 2.4 *Stochastic matrices P and P' are comparable, such that $P <_{st} P'$, if they share the same state space $(S, \prec)$, and for all $x \in S$, $P_{x,*} <_{st} P'_{x,*}$.*

This allows us to give the following conditions for the strong stochastic comparison of DTMCs, as shown in [8].

Proposition 2.5 *Let $\{X_t\}$ and $\{Y_t\}$ be DTMCs with transition matrices P_X and P_Y respectively, and state space $(S, \prec)$. Then $\{X_t\} <_{st} \{Y_t\}$ if:*

1. *$X_0 <_{st} Y_0$*

2. *$P_X <_{st} P_Y$*

3. *At least one of P_X and P_Y is monotone.*

The move to continuous-time is straightforward, since every CTMC has an underlying probability transition matrix. In particular, we apply *uniformisation* to the infinitesimal generator matrix Q of the CTMC. That is to say, we define a stochastic matrix $P = I + \frac{Q}{\lambda}$, where $\lambda \geq \max_i\{|q_{i,i}|\}$ and I is the identity matrix. We can then work directly on the stochastic matrix P, and revert back to Q when we are finished.

2.2 The PEPA Language

In the PEPA language, a *system* is a set of concurrent *components*, which are capable of performing *activities*. An activity $a \in \mathcal{Act}$ is a pair (α, r), where $\alpha \in \mathcal{A}$ is its action type, and $r \in \mathbb{R}^+ \cup \{\top\}$ is the rate of the activity. This rate parameterises an exponential distribution, and if unspecified (denoted $\top$), the activity is said to be *passive*. This requires another component in cooperation to actively drive the rate of this action. PEPA terms have the following syntax:

$$P := (\alpha, r).P \mid P_1 + P_2 \mid P_1 \bowtie_L P_2 \mid P/L \mid A$$

We briefly describe these combinators as follows. For more detail, we refer the reader to [6].

- *Prefix $((\alpha, r).P)$*: the component can carry out an activity of type α at rate r to become the component P.

- *Choice $(P_1 + P_2)$*: the system may behave either as component P_1 or P_2. The current activities of both components are enabled, and the first activity to complete determines which component proceeds. The other component is discarded.

- *Cooperation $(P_1 \bowtie_L P_2)$*: the components P_1 and P_2 synchronise over the cooperation set L. For activities whose type is not in L, the two components proceed independently. Otherwise, they must perform the activity together, at the rate of the slowest component. At the level of the system equation, we require that at least one of the components must be active with respect to this action type.

- *Hiding (P/L)*: the component behaves as P, except that activities whose type is in L are hidden, and appear externally as the unknown type τ.

- *Constant $(A \stackrel{def}{=} P)$*: the name A is assigned to component P.

If a PEPA component can perform many activities of the same action type α, then to an observer it appears to be capable of performing α at the sum of the rates of these activities. We therefore need a notion of *apparent rate*.

Definition 2.6 *The apparent rate $r_\alpha(P)$ of action type α in component P is defined as follows:*

1. $r_\alpha((\beta, r).P) = \begin{cases} r & \text{if } \beta = \alpha \\ 0 & \text{if } \beta \neq \alpha \end{cases}$

2. $r_\alpha(P + Q) = r_\alpha(P) + r_\alpha(Q)$

3. $r_\alpha(P/L) = \begin{cases} r_\alpha(P) & \text{if } \alpha \in L \\ 0 & \text{if } \alpha \notin L \end{cases}$

4. $r_\alpha(P \underset{L}{\bowtie} Q) = \begin{cases} \min\{r_\alpha(P), r_\alpha(Q)\} & \text{if } \alpha \in L \\ r_\alpha(P) + r_\alpha(Q) & \text{if } \alpha \notin L \end{cases}$

The operational semantics of PEPA defines a labelled multi-transition system. If a component P can evolve to component P' by a series of transitions, namely $P \xrightarrow{(\alpha_1, r_1)} \ldots \xrightarrow{(\alpha_n, r_n)} P'$, then P' is a *derivative* of P. The *derivative set* $\mathrm{ds}(P)$ is the set of all derivatives of P. These derivatives, and the transitions between them, form a *derivation graph*. Since the duration of a transition in this graph is given by an exponentially distributed random variable, this corresponds to a CTMC.

2.3 Lumpability

If we can partition the states of a CTMC such that states in the same partition have the same transition rates to each partition, then that partitioning is *ordinarily lumpable* [2]. If we solve the aggregated CTMC, then the steady state probability of being in a partition will be equal to the sum of the probabilities of being in its constituent states. We define this formally as follows:

Definition 2.7 *An ordinary lumping of a Markov chain with generator matrix Q is a partitioning of its states, $A_1, \ldots, A_K$, such that for every pair of states s_1 and s_2 in the same partition A, $\forall A_k$. $\sum_{s \in A_k} q_{s_1, s} = \sum_{s \in A_k} q_{s_2, s}$.*

Since a PEPA model maps onto a CTMC, we can lift this definition to PEPA directly. This is only useful, however, when we do not care about the *action types* of the transitions. Since we will be dealing with PEPA models compositionally, we need a stronger definition of lumpability to account for these actions. For example, if two states have the same transition rate to a partition, but one performs an α action and the other a β, we should not allow them to be aggregated.

Definition 2.8 *An ordinary lumping of a PEPA component P is a partitioning of its states $s \in \mathrm{ds}(P)$, $A_1, \ldots, A_K$, such that for every pair of states s_1 and s_2 in the same partition A:*

1. *For all action types α, $r_\alpha(s_1) = r_\alpha(s_2)$*

2. *For all action types α, $\forall A_k$:*

$$\sum_{s \in A_k} \sum_{s_1 \xrightarrow{(\alpha, r)} s} r = \sum_{s \in A_k} \sum_{s_2 \xrightarrow{(\alpha, r)} s} r$$

This corresponds to the *strong equivalence* relation defined in [6].

3 Compositional Bounds

If we wish to construct a lumpable bound of a sequential component in a PEPA model, we need to ensure that this bound is preserved when the component cooperates with others. This means that we need to look at how the generator matrices of the underlying Markov chain interact when they are composed. We can view the generator matrix of a component as a sum of matrices; one for each action α that the component can perform:

$$[\![C]\!] = \sum_\alpha Q_\alpha = \sum_\alpha r_\alpha (P_\alpha - I)$$

Here, we have written the generator matrices Q_α in the form $r_\alpha(P_\alpha - I)$, where I is the identity matrix. r_α is the apparent rate function, which we multiply each row of $P_\alpha - I$ by. P_α is deceptively *not* a probabilistic transition matrix. Each row i of P is a probability distribution (giving the next state) if $r_\alpha(i) > 0$, and is zero otherwise (i.e. there is no next state if α cannot be performed).

If we compose n PEPA components, $C_1, \ldots, C_n$, such that action types $\alpha \in \mathcal{Z}$ are synchronised[1], and $\alpha \notin \mathcal{Z}$ are performed independently, then we can write the generator matrix Q of the resulting PEPA component as follows:

$$Q = \sum_{\alpha \notin \mathcal{Z}} \bigoplus_{i=1}^{n} Q_{i,\alpha} + \sum_{\alpha \in \mathcal{Z}} \min\{r_{1,\alpha}, \ldots, r_{n,\alpha}\} \left(\bigotimes_{i=1}^{n} P_{i,\alpha} - I \right)$$

where, for $c_i \in \mathrm{ds}(C_i)$, $\min\{r_{1,\alpha}, \ldots, r_{n,\alpha}\}(c_1, \ldots, c_n) = \min\{r_{1,\alpha}(c_1), \ldots, r_{N,\alpha}(c_n)\}$.

Here, $\oplus$ and $\otimes$ denote Kronecker sum and product respectively. To find an upper or lower bound for Q compositionally, we need to bound the matrices $Q_{i,\alpha} = r_{i,\alpha}(P_{i,\alpha} - I)$ in such a way that stochastic order, monotonicity and lumpability are preserved when they are composed. Preservation of lumpability naturally follows as a consequence of strong equivalence [6], so we will concern ourselves with just stochastic ordering and monotonicity. Furthermore, these are both trivially preserved by the Kronecker sum, so we need only consider the second term of the above.

The problem in preserving these properties can be seen by looking at the term $r_\alpha(P_\alpha - I)$. Let us consider the property of monotonicity. The first thing we might try is to say that r_α must be monotone (in the normal sense of a function), and so must P_α (in the stochastic sense). However, this does not ensure that $r_\alpha(P_\alpha - I)$ is also monotone. The problem is the term $-r_\alpha I$, where we subtract an *increasing* value from the diagonal of each row. Consider the following simple example, where we write the apparent rate function as a matrix:

$$Q = r_\alpha(P_\alpha - I) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \left(\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{pmatrix} - I$$

It is clear here that Q is not monotone. The same problem occurs with stochastic ordering.

The solution is to strengthen then definition of stochastic ordering and monotonicity, by adding an extra constraint. We call these the *rate-wise stochastic ordering* and *rate-wise monotonicity* respectively. The definitions are as follows:

Definition 3.1 *For generator matrices $Q_\alpha = r_\alpha(P_\alpha - I)$ and $Q'_\alpha = r'_\alpha(P'_\alpha - I)$, we say that $Q_\alpha <_{\mathrm{rst}} Q'_\alpha$ under the rate-wise stochastic ordering, if:*

1. $P_\alpha <_{\mathrm{st}} P'_\alpha$

2. For all states c:

$$1 \leq \frac{r'_\alpha(c)}{r_\alpha(c)} \leq \min_{c' \prec c} \left\{ \frac{\sum_{d \succ c'} P_\alpha(c,d) - 1}{\sum_{d \succ c'} P'_\alpha(c,d) - 1} \right\}$$

Here, and in the following, we use the notation $P(i,j)$, rather than $P_{i,j}$, to index a matrix P, since we are already using subscripts to identify the matrices. Note that the indices will be *states*, so that $P_{i,\alpha}(c, c')$ gives the probability of a transition from state $c \in \mathrm{ds}(C_i)$ to $c' \in \mathrm{ds}(C_i)$, when an α action takes place.

Definition 3.2 *A generator matrix $Q_\alpha = r_\alpha(P_\alpha - I)$ is rate-wise monotone if:*

1. P_α is monotone.

2. For all states c, c' such that $c' \succ c$:

$$1 \leq \frac{r_\alpha(c')}{r_\alpha(c)} \leq \min_{c'' \prec c} \left\{ \frac{\sum_{d \succ c''} P_\alpha(c,d) - 1}{\sum_{d \succ c''} P_\alpha(c',d) - 1} \right\}$$

Unfortunately, it is still not the case that rate-wise monotonicity and rate-wise stochastic ordering are preserved in general when two components cooperate. The problem arises due to the minimum operator, which is applied to the rate vectors. If we take monotonicity, for example, the ratio between successive rates places constraints on the probabilistic transition matrix. When we compose two monotone components, it is possible for one to be completely bounded by the other in terms of its ability to perform an action α. That is to say, the rate of performing α in each state of one component is less than the rate of α in *any* state of the other. Hence the minimum of the two rate vectors, and the resulting constraint on the composed probabilistic transition matrix, depends on only one of the components. The required constraint on the composed matrix may therefore be tighter than that of one of the components.

---

[1]This restriction to only one cooperation set disallows certain models (for example, those of the form $C_1 \bowtie_{\{\alpha\}} (C_2 \parallel C_3)$, where all components can perform α), but many can be refactored to this form by appropriate renaming of actions.

It is therefore not possible for us to construct a bound for a sequential component, without considering the *context* in which it occurs (unless we assume a 'worst case' context). To define this context, we need a measure on components, to indicate the extent to which the rate vector increases. For monotonicity, we are concerned with the ratio between successive rates, and in particular the maximum of these. This is because, when taking the Kronecker product, we consider all possible state combinations. Hence the maximum increase will actually occur, and gives a bound on how a component can affect those that it cooperates with.

Definition 3.3 *The internal rate measure of action α of a component C, with corresponding generator matrix $Q_\alpha = r_\alpha(P_\alpha - I)$, is defined as:*

$$\|C\|_\alpha = \max_c \left\{ \frac{c'}{r_\alpha(c)} \,\middle|\, c' \in \mathrm{succ}(c) \right\}$$

Here, $\mathrm{succ}(c)$ denotes the set of immediate successors of the state c (i.e. $c' \in \mathrm{succ}(c)$ iff $c' \succ c \wedge \forall c'' \succ c.\ c' \not\succ c''$). For the stochastic ordering, we need to compare the rate vectors of two components (the original and the bound), but otherwise the same principle applies.

Definition 3.4 *The comparative rate measure of action α of components C and C', with matrices $Q_\alpha = r_\alpha(P_\alpha - I)$ and $Q'_\alpha = r'_\alpha(P'_\alpha - I)$ respectively, is defined as:*

$$\|C, C'\|_\alpha = \max_c \left\{ \frac{r'_\alpha(c)}{r_\alpha(c)} \right\}$$

In the above definitions, note that the ratio may be undefined (i.e. $r_\alpha(c) = 0$). In this case, we define the ratio to have the value $\top$, which dominates all the reals.

We can now define precisely what we mean by a context, which is slightly different to a conventional process algebra definition, since we discard the cooperation structure.

Definition 3.5 *The context $\mathcal{C}$ of a component C is the set of all components that it can cooperate with, as defined by the system equation. For all contexts $\mathcal{C}' = \{C'_i >_{\mathrm{st}} C_i | C_i \in \mathcal{C}\}$, we say that $\mathcal{C}' >_{\mathrm{st}} \mathcal{C}$.*

We say that $\mathcal{C}$ is internally bounded by A, with respect to action α, if:

$$\forall C_i \in \mathcal{C},\ \|C_i\|_\alpha \leq A$$

Furthermore, $\mathcal{C}$ and $\mathcal{C}'$ are comparatively bounded by B, with respect to action α, if:

$$\forall C_i \in \mathcal{C},\ \|C_i, C'_i\|_\alpha \leq B$$

This then leads to our final extension to the definitions of stochastic ordering and monotonicity. We call these the *context-bounded rate-wise stochastic ordering* and *context-bounded rate-wise monotonicity* respectively. Intuitively, the definition requires that the rate vector $r_{\alpha,i}$ of component i does not increase faster than is allowed for by the matrices $P_{\alpha,j}$, $j \neq i$ of all the components with which it cooperates.

Definition 3.6 *For generator matrices $Q_\alpha = r_\alpha(P_\alpha - I)$ and $Q'_\alpha = r'_\alpha(P'_\alpha - I)$, we say that $Q_\alpha <^A_{\mathrm{rst}} Q'_\alpha$ under the context-bounded rate-wise stochastic ordering, if:*

1. *$P_\alpha <_{\mathrm{st}} P'_\alpha$*

2. *For all states c:*

$$1 \leq \max \left\{ \frac{r'_\alpha(c)}{r_\alpha(c)}, A \right\} \leq \min_{c' \prec c} \left\{ \frac{\sum\limits_{d \succ c'} P_\alpha(c, d) - 1}{\sum\limits_{d \succ c'} P'_\alpha(c, d) - 1} \right\}$$

We extend the definition of rate-wise monotonicity similarly.

Definition 3.7 *A generator matrix $Q_\alpha = r_\alpha(P_\alpha - I)$ is context-bounded rate-wise monotone with respect to the bound A, if:*

1. *P_α is monotone.*

2. *For all states c, c' such that $c' \succ c$:*

$$1 \leq \max \left\{ \frac{r_\alpha(c')}{r_\alpha(c)}, A \right\} \leq \min_{c'' \prec c} \left\{ \frac{\sum\limits_{d \succ c''} P_\alpha(c, d) - 1}{\sum\limits_{d \succ c''} P_\alpha(c', d) - 1} \right\}$$

To keep this paper short, we omit the proofs that context-bounded rate-wise stochastic ordering and monotonicity are indeed preserved by cooperation. They are available on request.

4   Algorithmically Constructing Bounds

In [5], an algorithm is given to derive an irreducible and lumpable bounding matrix for a discrete-time Markov chain (DTMC). It is actually quite simple to extend this to PEPA components, using the results of the previous section. The algorithms, however, assume that the ordering on the state space for each sequential component is total, as this makes the matrix computations much simpler. This is therefore a specialisation of the results of the previous section, which assumed an arbitrary partial order on states.

The basic idea behind the algorithm comes from Vincent *et al* [1], who observed that for monotonicity and stochastic ordering to hold, the following inequalities must be satisfied:

1. For all i, $P(i, *) <_{\text{st}} R(i, *)$.

2. For all i, $R(i - 1, *) <_{\text{st}} R(i, *)$.

Here, P is the original probabilistic transition matrix, and R is its monotone upper bound. By $P(i, *)$, we denote row i of matrix P. To arrive at the algorithm, we set the first row of R equal to that of P ($R(1, *) = P(1, *)$), and then set each subsequent row according to the maximum of the left-hand sides of the above inequalities. Since the ordering on states is total, $P(i, *) <_{\text{st}} R(i, *)$ means that $\sum_{k=j}^{n} P(i, k) \leq \sum_{k=j}^{n} R(i, k)$, for all j. This leads to the following definition of R:

$$R(i, j) = \max \left\{ \sum_{k=j}^{n} R(i - 1, k), \sum_{k=j}^{n} P(i, k) \right\} - \sum_{k=j+1}^{n} R(i, k)$$

This allows R to be computed by iterating over its elements. A lower-bounding matrix can be constructed similarly.

Unfortunately, this does not guarantee that if P is irreducible then R will also be, since it is possible to delete transitions. Fourneau *et al.* address this by a slight modification to the algorithm, so that we avoid unnecessarily deleting transitions. To produce an upper-bounding matrix that is not only monotone and irreducible, but *lumpable* with respect to a given partition, the algorithm has a further step. Essentially, after iterating through the columns for a given partition, we fix the matrix so that the row sums are the same, as required by lumpability. This is done while preserving monotonicity.

We can now modify this algorithm so that, given a generator matrix $Q_\alpha = r_\alpha(P_\alpha - I)$, an internal context bound A, and a comparative context bound B, we produce a context-bounded rate-wise monotone upper bounding matrix $Q'_\alpha = r'_\alpha(P'_\alpha - I)$, lumpable with respect to a partitioning $\mathcal{L}$.

Step 1 – Bounding the Rate Vector

To construct an upper-bounding rate vector r_α, we use the following simple algorithm:

$$r'_\alpha(i) = \max \{r'_\alpha(i - 1), r_\alpha(i)\}$$

Step 2 – Bounding the Probability Transition Matrix

The basic algorithm is the same as that described previously, except that we have two additional constraints on the matrix, according to the internal and comparative bounds respectively. After applying the previous step to each component in the system equation, we can determine the internal bound A, and the comparative bound B, of each component's context. To compute the matrix P'_α, elements $P'_\alpha(i, j)$, where $i < j$ are given by:

$$P'_\alpha(i, j) = \max \left\{ \sum_{k=j}^{n} P'_\alpha(i - 1, k), \sum_{k=j}^{n} P_\alpha(i, k) \right\} - \sum_{k \succ \text{succ}(j)} P'_\alpha(i, k)$$

and when $i \geq j$ by:

$$P'_\alpha(i, j) = \max \begin{cases} \sum_{k=j}^{n} P'_\alpha(i - 1, k), \\ \sum_{k=j}^{n} P_\alpha(i, k), \\ 1 + \min\{\frac{1}{A}, \frac{r_\alpha(i)}{r'_\alpha(i)}\} \left(\sum_{k=j}^{n} P_\alpha(i, k) - 1 \right) \\ 1 + \min\{\frac{1}{B}, \frac{r_\alpha(i-1)}{r_\alpha(i)}\} \left(\sum_{k=j}^{n} P_\alpha(i - 1, k) - 1 \right) \end{cases}$$
$$- \sum_{k \succ \text{succ}(j)} P'_\alpha(i, k)$$

The extra two conditions here come from rearranging Definition 3.6 and Definition 3.7 respectively.

5   Conclusions and Future Work

In this paper, we have presented some initial results regarding compositionally constructing stochastic bounds for a class of PEPA models. While this will always give an exact bound for the solution of the model, it is not guaranteed that the bounds will be tight. This is a limitation of stochastic bounds in general, and is only made worse by our need for stricter conditions. Indeed, there are many open questions regarding how the choice of ordering and partitioning the state space can affect the accuracy of the bounds.

We do not present here a completed 'tool' that could be directly applied to the approximate solution of complex models. Instead, this represents the worst case situation, where we have zero semantic knowledge of the model. By 'semantic', we mean information that is available to the modeller, such as how states logically relate to one another, which can for instance be of great help when deciding on partitions. Without doubt there is a great deal of interesting work to be done in this area.

Successfully applying stochastic bounds to real models is still something of a black art. However, if we can move away from the low-level details of Markov chains, and utilise the structure and compositionality of stochastic process algebra, we can hope to make this process easier and more reliable. The results in this paper are the first tentative steps toward this goal.

References

[1] O. Abu-Amsha and J. M. Vincent. An algorithm to bound functionals of Markov chains with large state space. In *4th INFORMS Conference on Telecommunications*, 1998.

[2] P. Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, 31(1):59–75, 1994.

[3] T. Dayar and W. J. Stewart. Quasi lumpability, lower-bounding coupling matrices, and nearly completely decomposable markov chains. *SIAM J. Matrix Anal. Appl.*, 18(2):482–498, 1997.

[4] J. M. Fourneau and L. Kloul. A precedence PEPA model for performance and reliability analysis. In András Horváth and Miklós Telek, editors, *Formal Methods and Stochastic Models for Performance Evaluation: Third European Performance Engineering Workshop (EPEW 2006)*, number 4054 in LNCS, pages 1–15. Springer-Verlag, June 2006.

[5] J. M. Fourneau, M. Lecoz, and F. Quessette. Algorithms for an irreducible and lumpable strong stochastic bound. *Linear Algebra and its Applications*, 386:167–185, 2004.

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

[7] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Springer, 1976.

[8] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models*. Wiley & Sons, New York, 1983.