Compositional Stochastic Bounding of PEPA Models

Michael J. A. Smith †
Department of Informatics and Mathematical Modelling
Danmarks Tekniske Universitet
Lyngby, Denmark
Email: mjas@imm.dtu.dk

***Abstract*—Stochastic process algebras such as PEPA allow complex stochastic models to be described in a compositional way, but this leads to state space explosion problems. There has been a great deal of work on abstraction techniques for Markovian formalisms, and stochastic bounds in particular can be used to analyse monotone properties such as steady state distributions. Most techniques, however, operate at the level of the underlying Markov chain, whose state space may be too large to construct explicitly.**

In this paper, we present a compositional application of stochastic bounds to PEPA, in relation to steady state probabilities. We first introduce a new Kronecker representation for PEPA models, and then develop the necessary conditions for generating a compositional and lumpable stochastic bound of a PEPA model. We then show how to algorithmically construct such bounds, by means of an extension to the algorithm by Fourneau *et al* for partially-ordered state spaces. Finally, we present some results from our implementation, as part of the PEPA plug-in for Eclipse.

I. Introduction

The primary aim of stochastic modelling is to gain insight and understanding of systems that arise in practice, but the models we create of such 'real' systems are often much too large to analyse explicitly. We therefore need techniques for *abstracting* these models — specifically, reducing their size so that they become small enough to analyse. Given that we are interested in some particular properties of the model, we need to ensure that the abstract model gives us accurate results, but this usually comes at the expense of lower precision. For example, if the property is a probability, the abstract model could give the interval $[0.1, 0.2]$, which is accurate, but less precise than that actual answer of 0.18.

In this paper, we will consider steady state properties of Markovian models — in the long run of the model, what is the probability of being in a particular set of states? By answering this question, we can calculate important performance characteristics such as throughput, utilisation and identification of bottleneck components, under stable load conditions. To allow us to obtain steady state measures for models that are too large to solve explicitly, one approach is to combine, or *aggregate*, states that have similar behaviour. Unfortunately, this leads in general to non-Markovian models, which cannot easily be solved numerically.

† This work was funded by a Microsoft Research European Scholarship.

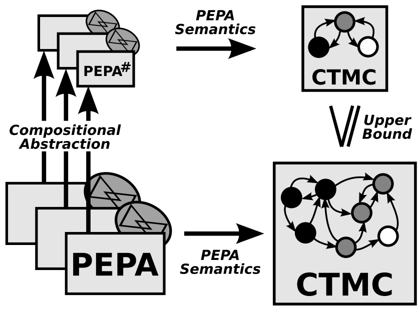

Fig. 1. Compositional abstraction of PEPA models

To mitigate this problem, without resorting to approximations, the technique of *stochastic bounds* has been proposed [13]. The idea is to construct an upper bound of the Markov chain that *can* be aggregated to another Markov chain — a condition known as lumpability. The aggregated Markov chain can then be solved to give an upper bound on the steady state distribution of the original, and by constructing a lower bound in a similar fashion, we can obtain probability intervals for steady state properties. An algorithm for doing this was presented by Fourneau *et al*. in [7].

A limitation with this approach, however, is that it is applied at the level of a Markov chain, as opposed to a higher level formalism. In particular, stochastic process algebras such as PEPA [8] are commonly used, to describe complex systems in a more compact and *compositional* manner. In such a language, it is easy to describe Markov chains that are not only too large to solve, but have state spaces that are too large to construct explicitly. Since we cannot apply the algorithm in [7] unless we can construct the state space of the Markov chain, we need to look at how this technique can be lifted to the level of the PEPA model itself.

The purpose of this paper is to extend the notion of a stochastic bound so that it can be applied to PEPA models compositionally. This is illustrated in Figure 1 — we bound and aggregate each component in a PEPA model separately, such that the aggregated model induces a Markov chain that is an upper bound of the original. This approach uses a variant of the Kronecker representation for the generator matrix of a PEPA model in [9], and there are no restrictions on the structure of a model. In addition to developing a method for compositionally bounding PEPA components, we extend Fourneau *et al*'s algorithm so that it can be applied to a class of partially ordered state spaces — this contribution is independent of PEPA. We have implemented the work in this paper as part of the PEPA plug-in for Eclipse [1].

There has been some other work in relation to compositional applications of stochastic bounds [3], [14]. The work in [3] in particular uses a weaker constraint than stochastic monotonicity, to obtain tighter bounds. Our approach is to work with stochastic monotonicity over partially ordered state spaces.

A summary of this paper is as follows. We begin in Section II by introducing Markov chains and lumpability, and in Section III with a description of the existing stochastic bounding algorithms for Markov chains. In Section IV we introduce PEPA along with its Kronecker representation. We then show how to compositionally apply stochastic bounds to PEPA models in Section V, and present an algorithm for computing bounds from a partially ordered state space in Section VI. Finally, we demonstrate this technique on an example model in Section VII, before concluding in Section VIII. Proofs of the theorems in this paper can be found in the attached Appendix for the purposes of the reviewers.

II. Abstraction of Markov Chains

Before we discuss stochastic bounding of Markov chains, let us first introduce some notation.

Definition 1. *A Discrete Time Markov Chain (DTMC) is a tuple* $(S, \boldsymbol{P})$*, and a Continuous Time Markov Chain (CTMC) is a tuple* $(S, \boldsymbol{P}, r)$*.* S *is a finite non-empty set of states,* $\boldsymbol{P} : S \times S \to [0, 1]$ *is a stochastic matrix, and* $r : S \to \mathbb{R}_{\geq 0}$ *is a function describing the rate of exit for each state. For a CTMC when* $r(s) = 0$ *— i.e. no transitions are possible from state* s *— we set* $\boldsymbol{P}(s, s) = 1$*, and* $\boldsymbol{P}(s, s') = 0$ *for all* $s' \neq s$*.*

The matrix $\boldsymbol{P}$ describes the probability $\boldsymbol{P}(s_1, s_2)$ of transitioning between two states s_1 and s_2 of the Markov chain in a single time step. In a DTMC, the duration of this time step is not specified, whereas for a CTMC it is determined by a random variable X, such that $\mathbf{Pr}(X < t) = 1 - e^{-rt}$ when the state has an exit rate of r.

Often, a CTMC is described in terms of its infinitesimal generator matrix $\boldsymbol{Q}$, whose elements $\boldsymbol{Q}(i, j)$ (where $i \neq j$) define the rate of transitioning between states i and j — with diagonal elements given by $\boldsymbol{Q}(i, i) = -\sum_{j \neq i} \boldsymbol{Q}(i, j)$. This can be calculated from the rate function r and the probability transition matrix $\boldsymbol{P}$ as follows:

$$\boldsymbol{Q} = r(\boldsymbol{P} - \boldsymbol{I})$$

Here, we define the multiplication of a matrix $\boldsymbol{M}$ by a function r as $(r\boldsymbol{M})(i, j) = r(i)\boldsymbol{M}(i, j)$. The steady state of an ergodic CTMC with generator matrix $\boldsymbol{Q}$ is a row vector $\boldsymbol{\pi}$, such that $\boldsymbol{\pi}\boldsymbol{e} = 1$ (where $\boldsymbol{e}$ is a column vector of 1s) and $\boldsymbol{\pi}\boldsymbol{Q} = \boldsymbol{0}$.

If a CTMC has a non-zero exit rate for every state, then we obtain its *embedded DTMC* by simply discarding these rates:

Definition 2. *For a CTMC* $\mathcal{M} = (S, \boldsymbol{P}, r)$*, its embedded DTMC is defined as* $\mathit{Embed}(\mathcal{M}) = (S, \boldsymbol{P})$*.*

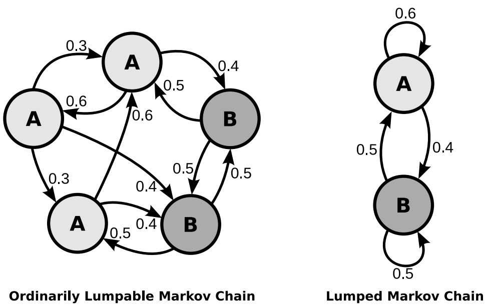

Fig. 2. Ordinary lumpability of a Markov chain

This, however, alters the behaviour of the Markov chain by throwing away the relative timing information of its states. In particular, the steady-state solution of the embedded DTMC will be different from that of the CTMC. We can avoid this problem by first *uniformising* the CTMC.

Definition 3. *The uniformisation of a CTMC* $\mathcal{M} = (S, \boldsymbol{P}, r)$*, with uniformisation rate* $\lambda \geq \max_{s \in S} r(s)$ *is given by* $\mathit{Unif}_\lambda(\mathcal{M}) = (S, \overline{\boldsymbol{P}}, \overline{r})$*, where* $\overline{r}(s) = \lambda$ *for all* $s \in S$*, and:*

$$\begin{array}{lll} \overline{\boldsymbol{P}}(s, s') & = \frac{r(s)}{\lambda}\boldsymbol{P}(s, s') & \text{if } s \neq s' \\ \overline{\boldsymbol{P}}(s, s) & = 1 - \sum_{s' \neq s} \overline{\boldsymbol{P}}(s, s') & \text{otherwise} \end{array}$$

Essentially, uniformisation adjusts the CTMC by inserting self-loops, so that the exit rate of every state is the same.

Consider a Markov chain with a state space S. The basic idea of state space abstraction is to reduce S to an abstract state space $S^\sharp$, which should be smaller than S ($|S^\sharp| < |S|$). To define an abstraction, we need a mapping between the concrete and abstract states:

Definition 4. *An abstraction of a state space* S *is a pair* $(S^\sharp, \alpha)$*, where* $\alpha : S \to S^\sharp$ *is a surjective function that maps every concrete state to an abstract state.*

Aggregating a Markov chain according to an abstraction α means that we cannot distinguish between states that map to the same abstract state. Therefore, in order to still have a Markov chain, the rate of transition between two abstract states must be independent of the particular concrete state we are in. This condition is called *ordinary lumpability* [10]:

Definition 5. *An ordinary lumping of a CTMC* $\mathcal{M} = (S, \boldsymbol{P}, r)$ *is an abstraction* $(S^\sharp, \alpha)$ *such that for all states* $s, s' \in S$*, if* $\alpha(s) = \alpha(s')$ *then for all states* $s^\sharp \in S^\sharp$*:*

$$\sum_{\{t \ | \ \alpha(t) = s^\sharp\}} r(s)\boldsymbol{P}(s, t) = \sum_{\{t \ | \ \alpha(t) = s^\sharp\}} r(s')\boldsymbol{P}(s', t)$$

An ordinary lumping $(S^\sharp, \alpha)$ *induces* a new CTMC, in that it completely defines the transition rates between abstract states.

An example of a lumpable Markov chain is shown in Figure 2, which can be viewed as a uniformised CTMC with $\lambda = 1$. Solving this Markov chain, the steady state probability of being in an abstract state (A or B) is equal to the sum of the probabilities of being in each of its constituent states. In other words, solving the aggregated CTMC is equivalent to aggregating the solution of the original CTMC.

III. Stochastic Bounding of Markov Chains

Stochastic bounding allows us to construct upper and lower bounds for monotone properties of Markov chains. The focus of this paper is to bound steady state probabilities, which we do by constructing upper and lower bounds of the Markov chain that preserve a bound on the steady state distribution. The important point is to ensure that these bounds are lumpable, so that we can reduce the size of the Markov chain to solve. To do this, we need to define what we mean by a *stochastic ordering* on probability distributions.

There are a number of different stochastic orderings [13], but for our purposes we will use only the *strong stochastic order*, which we denote $\leq_{\mathrm{st}}$. This is defined for comparing random variables in general, but for our purposes we will consider just discrete random variables, which can be represented as a vector of probabilities, summing to one:

Definition 6. *Let X be a random variable on a partially ordered state space $(S, \prec)$, and $\boldsymbol{x}$ be a vector such that $\mathbf{Pr}(X \succ s) = \sum_{s' \succ s} \boldsymbol{x}(s')$. Let $\boldsymbol{y}$ be defined similarly for a random variable Y over $(S, \prec)$. We say that $\boldsymbol{x} \leq_{\mathrm{st}} \boldsymbol{y}$ if for all $s \in S$:*

$$\sum_{s' \succ s} \boldsymbol{x}(s') \leq \sum_{s' \succ s} \boldsymbol{y}(s')$$

We can extend this ordering to Markov chains in both discrete and continuous time, by looking at their transition matrices. In particular, there are two important properties of such stochastic matrices: *comparability* and *monotonicity*. We shall assume that the state space of a matrix $\boldsymbol{P}$ (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 $\boldsymbol{P}(i,*)$ for row i of matrix $\boldsymbol{P}$, which is itself a row vector.

Definition 7. *A stochastic matrix $\boldsymbol{P}$ is monotone if for all row vectors $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{u} \leq_{\mathrm{st}} \boldsymbol{v}$ implies that $\boldsymbol{uP} \leq_{\mathrm{st}} \boldsymbol{vP}$. Equivalently, for all $s, s' \in S$ such that $s \prec s'$, $\boldsymbol{P}(s,*) \leq_{\mathrm{st}} \boldsymbol{P}(s',*)$.*

Definition 8. *The stochastic matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$ are comparable, namely that $\boldsymbol{P} \leq_{\mathrm{st}} \boldsymbol{P}'$, if they share the same state space $(S, \prec)$, and for all $s \in S$, $\boldsymbol{P}(s,*) \leq_{\mathrm{st}} \boldsymbol{P}'(s,*)$.*

Using these notions of stochastic comparison and monotonicity, the strong stochastic ordering can be defined for DTMCs as follows:

Definition 9. *Consider the DTMCs $\mathcal{M}_1 = (S, \boldsymbol{P}_1)$ and $\mathcal{M}_2 = (S, \boldsymbol{P}_2)$. We say that $\mathcal{M}_1 \leq_{\mathrm{st}} \mathcal{M}_2$ if:*

1) *For the initial distributions $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, $\boldsymbol{x}_1 \leq_{\mathrm{st}} \boldsymbol{x}_2$.*
2) *For the probability transition matrices, $\boldsymbol{P}_1 \leq_{\mathrm{st}} \boldsymbol{P}_2$.*
3) *At least one of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ is monotone.*

We can similarly compare two CTMCs, in terms of their embedded DTMCs, if they are first uniformised:

Definition 10. *Two CTMCs $\mathcal{M}_1 = (S, \boldsymbol{P}_1, r_1)$ and $\mathcal{M}_2 = (S, \boldsymbol{P}_2, r_2)$ are comparable such that $\mathcal{M}_1 \leq_{\mathrm{st}} \mathcal{M}_2$ if for all valid uniformisation constants λ:*

$$\mathit{Embed}(\mathit{Unif}_{\lambda}(\mathcal{M}_1)) \leq_{\mathrm{st}} \mathit{Embed}(\mathit{Unif}_{\lambda}(\mathcal{M}_2))$$

Since stochastic comparison of CTMCs is defined in terms of stochastic comparison of DTMCs, we can consider only the latter, without loss of generality, for the remainder of this section. The need to uniformise CTMCs, however, will become important when we apply these techniques compositionally to PEPA models in Section V.

For stochastic bounds to be of practical use, we need algorithms to construct monotone upper and lower bounding matrices, given the probability transition matrix of a DTMC. Furthermore, not only do we need them to be bounding, but they must be *lumpable* with respect to the desired abstraction.

In [7], an algorithm is given to derive an irreducible and lumpable bounding matrix for a discrete-time Markov chain (DTMC), assuming that the state space is *totally ordered*. The algorithm is an extension of that by Abu-Amsha and Vincent [2], which finds a monotone upper bounding probability transition matrix for a DTMC. This algorithm arose from observing that, for $\boldsymbol{P} \leq_{\mathrm{st}} \boldsymbol{R}$, the following inequalities must hold for the $(n \times n)$ stochastic matrix $\boldsymbol{R}$ to be monotone:

1) For all $1 \leq i \leq n$, $\boldsymbol{P}(i,*) \leq_{\mathrm{st}} \boldsymbol{R}(i,*)$.
2) For all $1 \leq i \leq n-1$, $\boldsymbol{R}(i,*) \leq_{\mathrm{st}} \boldsymbol{R}(i+1,*)$.

In the basic algorithm, we set the first row of $\boldsymbol{R}$ equal to that of $\boldsymbol{P}$ ($\boldsymbol{R}(1,*) = \boldsymbol{P}(1,*)$), and then set each subsequent row according to the maximum of the left-hand sides of the above inequalities. The condition $\boldsymbol{P}(i,*) \leq_{\mathrm{st}} \boldsymbol{R}(i,*)$ means that $\sum_{k=j}^{n} \boldsymbol{P}(i,k) \leq \sum_{k=j}^{n} \boldsymbol{R}(i,k)$, for all j, since the ordering is total. This leads to the following definition of $\boldsymbol{R}$:

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

We can iteratively construct the matrix $\boldsymbol{R}$ based on the above equation, starting with the first row and the last column, and iterating down the rows before moving onto the next column. Since the algorithm for computing a lower bounding matrix is very similar, we will only consider the construction of upper bounds throughout this paper.

Unfortunately, this does not guarantee that if $\boldsymbol{P}$ is irreducible then $\boldsymbol{R}$ will also be, since it is possible to delete transitions. Fourneau *et al*. [7] address this by a slight modification to the algorithm, to avoid unnecessarily deleting transitions. Furthermore, they produce an upper bounding matrix that is not only monotone and irreducible, but *lumpable* with respect to a given partition. To achieve this, they structure the matrix so that states in the same partition have contiguous indices, and make the matrix lumpable by setting the next state distribution in each partition to the maximum that occurs within that partition. This ensures that the matrix remains monotone.

The main disadvantage of this algorithm is that it needs the state space to be totally ordered. It is possible to extend any partial order to a total order, but this will result in a stronger

$$C_1 = (a, r_a).C_2 + (b, r_b).C_3 \qquad D_1 = (a, r_D).D_2$$
$$C_2 = (\tau, r_2).C_1 + (\tau, r_2).C_3 \qquad D_2 = (b, r_D).D_1$$
$$C_3 = (\tau, r_3).C_2 + (\tau, r_3).C_1$$

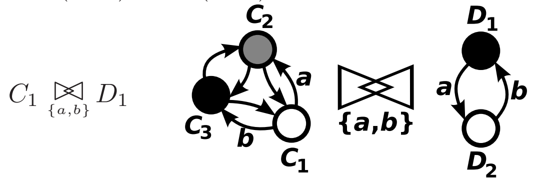

Fig. 3. An example PEPA model and its graphical representation

order than we need, and so give lower precision in the bounds. We will see later, in Section VI, how to modify this algorithm to work with a simple class of partially ordered state spaces.

IV. The Performance Evaluation Process Algebra

So far we have looked only at bounding Markov chains, but the aim of this paper is to present a compositional approach to bounding stochastic process algebra models. To this end, let us introduce the Performance Evaluation Process Algebra (PEPA) [8] — a compositional formalism with CTMC semantics. In PEPA, a *system* is a set of concurrent *components*, which are capable of performing *activities*. An activity $a \in \mathcal{A}ct$ is a pair (a, r), where $a \in \mathcal{A}$ is its action type, and $r \in \mathbb{R}_{\geq 0} \cup \{\top\}$ is the rate of the activity. This rate parameterises an exponential distribution, and if unspecified (denoted $\top$), the activity is said to be *passive*. In this case, another component is needed to actively drive the rate of this action. PEPA terms have the following syntax:

$$C := (a, r).C \mid C_1 + C_2 \mid C_1 \underset{L}{\bowtie} C_2 \mid C/L \mid A$$

We briefly describe these combinators as follows:

- *Prefix* ($(a, r).C$): the component can carry out an activity of type a at rate r to become the component C.
- *Choice* ($C_1 + C_2$): the system may behave either as component C_1 or C_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* ($C_1 \underset{L}{\bowtie} C_2$): the components C_1 and C_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 most one of the components may be passive with respect to this action type.
- *Hiding* (C/L): the component behaves as C, except that activities whose type is in L are hidden, and appear externally as the unknown type τ.
- *Constant* ($A \overset{def}{=} C$): component C has the name A.

The operational semantics of PEPA defines a labelled multi-transition system, which induces a *derivation graph* for a given component. Since the duration of a transition in this graph is given by an exponentially distributed random variable, this corresponds to a CTMC. An example PEPA model with two components is shown in Figure 3.

To apply stochastic bounds to a PEPA model, we need to consider the structure of its underlying CTMC. It was shown in [9] how the generator matrix of this Markov chain can be represented in a compositional, *Kronecker* form. Here, we develop a slight variation to avoid the need for functional rates. If we consider the system equation of a PEPA model, it has the following form:

$$C_1 \underset{L_1}{\bowtie} \cdots \underset{L_{N-1}}{\bowtie} C_N \qquad (1)$$

We can ignore the hiding operator C/L without loss of generality, since it is always possible to rename action types to avoid name conflicts between components.

Considering a sequential component C_i in such a model, we can view it as inducing a local generator matrix $\boldsymbol{Q}_i$ — though strictly speaking, this is not a generator matrix, since the component might perform passive activities. If we write $\boldsymbol{Q}_i$ in terms of a rate function r and a probabilistic transition matrix $\boldsymbol{P}$, however, we can retain the passive rates in r. If the component has a state space S_i — namely, $S_i = \mathrm{ds}(C_i)$ (the derivation set of C_i) — then we can write $\boldsymbol{Q}_i$ as an $|S_i| \times |S_i|$ matrix as follows:

$$\boldsymbol{Q}_i = \sum_a \boldsymbol{Q}_{i,a} = \sum_a r_{i,a} \left(\boldsymbol{P}_{i,a} - \boldsymbol{I}_{|S_i|} \right) \qquad (2)$$

Here $r_{i,a} : S_i \to \mathbb{R}_{\geq 0} \cup \{\top\}$ gives the rate of action type a for each state in S_i, $\boldsymbol{P}_{i,a}$ gives the next-state transition probabilities conditional on performing an activity of type a, and $\boldsymbol{I}_{|S_i|}$ is the $|S_i| \times |S_i|$ identity matrix. If, for a state s, $r_{i,a}(s) = 0$, we write $\boldsymbol{P}_{i,a}(s, s) = 1$ and $\boldsymbol{P}_{i,a}(s, s') = 0$ for $s' \neq s$. Since the rate is zero, we could effectively have chosen any values for this row, but this choice is convenient since it encodes the fact that we remain in the same state.

To build a compositional representation of the generator matrix $\boldsymbol{Q}$ of an arbitrary PEPA model, we need to combine the individual generator matrices $\boldsymbol{Q}_{i,a}$ in an appropriate way. More precisely, the compositional representation of $\boldsymbol{Q}$ has to describe the same CTMC as induced by the semantics of the PEPA model. Because cooperation between two PEPA components uses the *minimum* of two rates, we need to be especially careful that this leads to the correct apparent rate for each state and action type.

To describe the generator matrix term $\boldsymbol{Q}_{i,a}$ for activities of type a in a component C_i, we will use the shorthand $(r_{i,a}, \boldsymbol{P}_{i,a})$, which we define as follows:

$$(r_{i,a}, \boldsymbol{P}_{i,a}) = r_{i,a} \left(\boldsymbol{P}_{i,a} - \boldsymbol{I}_{|S_i|} \right) = \boldsymbol{Q}_{i,a}$$

where S_i is the state space of C_i. If C_i cannot perform any activities of action type a, we define its generator matrix term to be $\boldsymbol{Q}_{i,a} = (r_\perp, \boldsymbol{I}_{|S_i|})$, where $r_\perp(s) = 0$ for all $s \in S_i$.

Using this notation, we will introduce two Kronecker operators, $\circledast$ and $\odot$, which correspond to cooperating and independent activities. If C_1 and C_2 cooperate over an action type a, we will use the operator $\circledast$, defined as follows:

$$(r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a}) = (\min\{r_{1,a}, r_{2,a}\}, \boldsymbol{P}_{1,a} \otimes \boldsymbol{P}_{2,a}) \quad (3)$$

where $\min\{r_{1,a}, r_{2,a}\}(s_1, s_2) = \min\{r_{1,a}(s_1), r_{2,a}(s_2)\}$ for all $s_1 \in S_1$ and $s_2 \in S_2$. The operator $\otimes$ denotes the Kronecker product of two matrices [11].

If, on the other hand, C_1 and C_2 independently perform activities of type a, we will use the operator $\odot$, which we define in terms of $\circledast$:

$$(r_{1,a}, \boldsymbol{P}_{1,a}) \odot (r_{2,a}, \boldsymbol{P}_{2,a}) = (r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_\top, \boldsymbol{I}_{|S_2|}) + (r_\top, \boldsymbol{I}_{|S_1|}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a}) \quad (4)$$

where $r_\top(s) = \top$ for all s. Here, '$+$' is the normal matrix addition at the level of the generator matrices, but it can be expressed compositionally as follows:

Theorem 11. *Consider two generator matrices $\boldsymbol{Q}_1 = (r_1, \boldsymbol{P}_1)$ and $\boldsymbol{Q}_2 = (r_2, \boldsymbol{P}_2)$, corresponding to the same state space S ($\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ are both $|S| \times |S|$ matrices). Then $\boldsymbol{Q}_1 + \boldsymbol{Q}_2$ can be written as follows:*

$$(r_1, \boldsymbol{P}_1) + (r_2, \boldsymbol{P}_2) = \left(r_1 + r_2, \frac{r_1}{r_1 + r_2}\boldsymbol{P}_1 + \frac{r_2}{r_1 + r_2}\boldsymbol{P}_2\right)$$

where $(r_1 + r_2)(s) = r_1(s) + r_2(s)$, and $\frac{r_i}{r_1+r_2}(s) = \frac{r_i(s)}{r_1(s)+r_2(s)}$, $i \in \{1, 2\}$, for all $s \in S$.

The coefficients of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ describe the relative probability of taking a transition corresponding to $\boldsymbol{Q}_1$ or $\boldsymbol{Q}_2$. Note that they are functions, in that each row of the matrix is multiplied by a different value — this is because the relative apparent rate can differ between states.

We can now define our Kronecker representation for PEPA models, using the $\circledast$ and $\odot$ operators.

Definition 12. *The* Kronecker form *$\mathcal{Q}(C)$ of a PEPA model $C = C_1 \underset{L_1}{\bowtie} \cdots \underset{L_{N-1}}{\bowtie} C_N$ is defined as:*

$$\mathcal{Q}(C) = \sum_{a \in \mathcal{A}ct(C)} \mathcal{Q}_a(C)$$

where $\mathcal{A}ct(C)$ is the set of all action types that occur in C (both synchronised and independent), and $\mathcal{Q}_a$ is defined inductively as follows (S.C. stands for Sequential Component):

$$\begin{aligned} \mathcal{Q}_a(C_i) &= (r_{i,a}, \boldsymbol{P}_{i,a}) \quad \text{if } C_i \text{ is an S.C.} \\ \mathcal{Q}_a(C_i \underset{L}{\bowtie} C_j) &= \begin{cases} \mathcal{Q}_a(C_i) \circledast \mathcal{Q}_a(C_j) & \text{if } a \in L \\ \mathcal{Q}_a(C_i) \odot \mathcal{Q}_a(C_j) & \text{if } a \notin L \end{cases} \end{aligned}$$

Theorem 13. *For all well-formed[1] PEPA models C, the Markov chain given by the generator matrix $\mathcal{Q}(C)$ and the Markov chain induced by the semantics of PEPA have consistent rates of transition between states.*

[1]A well-formed PEPA model is one in which cooperation occurs only at the level of the system equation.

As an example of how the Kronecker form is applied, let us take the PEPA model from Figure 3. Applying our Kronecker form to $\boldsymbol{Q} = \mathcal{Q}(C_1 \underset{\{a,b\}}{\bowtie} D_1)$:

$$\begin{aligned} \mathcal{Q}(C_1 \underset{\{a,b\}}{\bowtie} D_1) = \; & \mathcal{Q}_\tau(C_1) \odot \mathcal{Q}_\tau(D_1) \\ & + \mathcal{Q}_a(C_1) \circledast \mathcal{Q}_a(D_1) \\ & + \mathcal{Q}_b(C_1) \circledast \mathcal{Q}_b(D_1) \end{aligned}$$

If we expand out the $\circledast$ and $\odot$ operators, showing the tensor products $\otimes$, we arrive at the following:

$$\begin{aligned} \boldsymbol{Q} = \; & \min\left\{\begin{bmatrix} 0 \\ 2r_2 \\ 2r_3 \end{bmatrix}, \begin{bmatrix} \top \\ \top \end{bmatrix}\right\}\left(\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \boldsymbol{I}_6\right) \\ & + \min\left\{\begin{bmatrix} \top \\ \top \\ \top \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right\}\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \boldsymbol{I}_6\right) \\ & + \min\left\{\begin{bmatrix} r_a \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} r_D \\ 0 \end{bmatrix}\right\}\left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} - \boldsymbol{I}_6\right) \\ & + \min\left\{\begin{bmatrix} r_b \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ r_D \end{bmatrix}\right\}\left(\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} - \boldsymbol{I}_6\right) \end{aligned} \quad (5)$$

where $\boldsymbol{I}_6 = \boldsymbol{I}_3 \otimes \boldsymbol{I}_2$. Note that the second term in the above evaluates to zero, because the D component does not perform any internal τ activities.

V. Stochastic Bounding of PEPA Models

Now that we have introduced the Kronecker form for PEPA models, we can turn to applying stochastic bounds compositionally. In this section, we will extend the definitions of stochastic ordering and monotonicity, so that when we compose the bounds of two PEPA components, the resulting CTMC is an bound of that induced by the original components. We will then present an algorithm for constructing these bounds in Section VI.

The work we present here is general in the sense that it applies to *all* PEPA models. This is in contrast to previous work, which considers the application of stochastic bounds to particular classes of PEPA model, such as passage time properties of workflow-structured models [6]. There is an advantage to looking at specific classes of models, in that it may be possible to obtain more precise bounds in light of the additional information that is available. However, generality is also important, in that we can analyse models without needing to assume anything about their particular structure.

Before bounding a PEPA model, we need to decide upon two things — an *ordering*, and a *partitioning* of its state space. Since the idea is to produce the bound compositionally, these must also be defined compositionally. We can lift the definition of an abstraction $(S^\sharp, \alpha)$ over a Markov chain (Definition 4), to a PEPA model with N components, $C_1, \ldots, C_N$, by defining an abstraction $(S_i^\sharp, \alpha_i)$ over the state space S_i of each component C_i. Together, these induce an abstraction over the state space $S_1 \times \cdots \times S_N$ of the system. This defines a unique

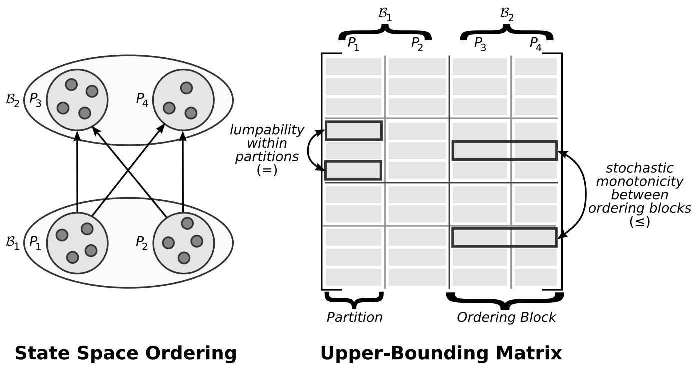

Fig. 4. State space ordering and lumpability constraints

partitioning of the state space according to which concrete states map to the same abstract state.

In addition to partitioning the state space, we need to provide an ordering. The ordering we choose will in general depend on the property we are interested in. For example, if are interested in the steady state probability of being in a particular set of states, it makes sense to place these at the 'top' of the ordering. This is so that we can directly compare the probability of being in this set. Furthermore, choosing a partial order can be advantageous, since it allows more flexibility when constructing the bound. The only constraint we have is to only allow entire partitions to be compared with other partitions, so that the abstraction can be applied.

The definitions and theorems in this section are applicable to any partial order, but in order to algorithmically construct the bound, we will restrict ourselves to the following class:

Definition 14. *A simple partial order over a state space S is given by a set of M disjoint sets, $\mathcal{B}_1, \ldots, \mathcal{B}_M \subseteq S$, where:*

$$s \prec s' \quad \textit{iff} \quad \exists i, j.\ s \in \mathcal{B}_i \wedge s' \in \mathcal{B}_j \wedge i < j$$

Each $\mathcal{B}_i$ may contain multiple partitions, but not vice versa. This is illustrated in Figure 4, which shows an example state space ordering and partitioning, and the resulting constraints on the upper bounding transition matrix.

Given an ordering and partitioning of a state space, we need to find a monotone CTMC that is both lumpable and an upper bound of the original CTMC. To do this compositionally, we must work at the level of $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$, bounding both the rate function r_a and the transition matrix $\boldsymbol{P}_a$ separately. This is so that when we construct the generator matrix of the entire model, by expanding out the Kronecker form of Definition 12, it remains upper bounding, monotone and lumpable.

Unfortunately, it is not the case that the monotonicity of r_a and $\boldsymbol{P}_a$ implies that of $\boldsymbol{Q}_a$. In order for $\boldsymbol{Q}_a$ to be monotone, its embedded DTMC after we uniformise it must be monotone, as per Definition 10. The effect of uniformisation is to add self loops (i.e. to add probability mass to the diagonal elements) so that each state has the same exit rate λ. Since r_a is *increasing*, however, this means that we add a *decreasing* amount to the diagonal element of each row (the '$-r_a\boldsymbol{I}$' term).

This is best illustrated by example. The following rate function r_a and probability transition matrix $\boldsymbol{P}_a$ are monotone, assuming a totally ordered state space, but the embedded DTMC after uniformisation is not:

$$\begin{aligned} r_a(\boldsymbol{P}_a - \boldsymbol{I}) &= \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \left(\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} - \boldsymbol{I} \right) \\ &= \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \left(\begin{bmatrix} \frac{3}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} - \boldsymbol{I} \right) \end{aligned} \tag{6}$$

To avoid this problem, we need to strengthen the definitions of stochastic ordering and monotonicity, by adding an extra constraint. We call these the *rate-wise stochastic ordering* and *rate-wise monotonicity* respectively. Their definitions are:

Definition 15. *For generator matrices $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$ and $\boldsymbol{Q}'_a = r'_a(\boldsymbol{P}'_a - \boldsymbol{I})$, we say that $\boldsymbol{Q}_a \leq_{\mathrm{rst}} \boldsymbol{Q}'_a$ under the* rate-wise stochastic ordering, *if:*

1) $\boldsymbol{P}_a \leq_{\mathrm{st}} \boldsymbol{P}'_a$
2) *For all states s:*

$$1 \leq \frac{r'_a(s)}{r_a(s)} \leq \min_{s' \prec s} \left\{ \frac{1 - \sum_{t \succ s'} \boldsymbol{P}_a(s,t)}{1 - \sum_{t \succ s'} \boldsymbol{P}'_a(s,t)} \right\}$$

Definition 16. *A generator matrix $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$ is rate-wise monotone if:*

1) *$\boldsymbol{P}_a$ is monotone.*
2) *For all states s, s' such that $s \prec s'$:*

$$1 \leq \frac{r_a(s')}{r_a(s)} \leq \min_{s'' \prec s} \left\{ \frac{1 - \sum_{t \succ s''} \boldsymbol{P}_a(s,t)}{1 - \sum_{t \succ s''} \boldsymbol{P}_a(s',t)} \right\}$$

Intuitively, in both cases, we ensure that the probability transition matrix increases faster than the rate function, so that after uniformisation we remain monotone and comparable. Note that this is not a condition on the original model — we just need to ensure that the condition holds of the upper bound.

It is important to comment on the above definitions in the case when $r_a(s) = 0$, since the ratio becomes infinite. In real terms, this means that we are effectively blocked from adding probability mass below the diagonal element for previously disabled activities, since the denominator on the right hand side must also be zero if the condition is to be satisfied.

We can show that strong stochastic comparison and monotonicity follow from rate-wise stochastic comparison and monotonicity. This means that the CTMC that we construct by this method is stochastically comparable in the usual sense. We state this in the following two theorems:

Theorem 17. *If $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I}) \leq_{\mathrm{rst}} \boldsymbol{Q}'_a = r'_a(\boldsymbol{P}'_a - \boldsymbol{I}')$ and for all $s \in S$, $r_a(s) \leq r'_a(s)$, then $\boldsymbol{Q}_a \leq_{\mathrm{st}} \boldsymbol{Q}'_a$.*

Theorem 18. *If $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$ is rate-wise monotone, and for all $s \prec s' \in S$, $r_a(s) \leq r_a(s')$, then $\boldsymbol{Q}_a$ is monotone.*

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 functions. 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 activity of type a. That is to say, the rate of performing a in each state of one component is less than the rate of a in *any* state of the other. Hence the minimum of the two rate functions, 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.

This problem is clearer if we look at a particular example:

$$r_a(\boldsymbol{P}_a - \boldsymbol{I}) = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} \left(\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} - \boldsymbol{I} \right)$$

Note that this is similar to Equation 6, which illustrated why we need rate-wise comparison and monotonicity. In this case, the component *is* rate-wise monotone, but if it were to synchronise with a component that has a rate function of, say, $[1, 1, 2]$, we would result in the same problem as before.

It is therefore not possible for us to construct a bound for a sequential component, without considering the *context* in which it occurs. To define this context, we need a measure on components, to indicate the extent to which the rate function 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 19. *The internal rate measure of a component C, with generator matrix $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$ for action type a, is:*

$$\|C\|_a = \max_s \left\{ \frac{r_a(s')}{r_a(s)} \;\middle|\; s' \in \text{succ}(s) \right\}$$

Here, $\text{succ}(s)$ denotes the set of immediate successors of the state s as defined by the simple partial order. More precisely, $s' \in \text{succ}(s)$ iff $s' \succ s \wedge \neg\exists s''.\ s' \succ s'' \succ s$. In the case of stochastic ordering, we need to compare the rate functions of two components (the original and the bound), but otherwise the same principle applies:

Definition 20. *The comparative rate measure of components C and C', with generator matrices $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$ and $\boldsymbol{Q}'_a = r'_a(\boldsymbol{P}'_a - \boldsymbol{I})$ respectively for action type a, is defined as:*

$$\|C, C'\|_a = \max_s \left\{ \frac{r'_a(s)}{r_a(s)} \right\}$$

In the above definitions, note that the ratio may be undefined (i.e. $r_a(s) = 0$). 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 care only about those components that can affect the rate at which we perform an activity.

Definition 21. *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. We say that $\mathcal{C}$ is internally bounded by B_{int}, with respect to action type a, if:*

$$\forall C_i \in \mathcal{C},\ \|C_i\|_a \leq B_{int}$$

Furthermore, $\mathcal{C}$ and $\mathcal{C}'$ are comparatively bounded by B_{comp}, with respect to action type a, if:

$$\forall C_i \in \mathcal{C},\ \|C_i, C'_i\|_a \leq B_{comp}$$

Since the internal and comparative bounds depend only on the rate functions, we have a simple algorithm for computing them. If we construct a monotone upper bound of each rate function before bounding the transition matrices, we can compute the internal and comparative bounds of a context as the maximum bounds of the components within the context.

This leads us to the final extension of our definitions — the *context-bounded rate-wise stochastic ordering* and *context-bounded rate-wise monotonicity*, which extend Definitions 15 and 16 respectively. Intuitively, they require the rate function $r_{i,a}$ of component i to not increase faster than the matrices $\boldsymbol{P}_{j,a}$, $j \neq i$, of all its cooperating components allow for.

Definition 22. *For generator matrices $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$ and $\boldsymbol{Q}'_a = r'_a(\boldsymbol{P}'_a - \boldsymbol{I})$, we say that $\boldsymbol{Q}_a \leq_{\text{rst}}^{B_{comp}} \boldsymbol{Q}'_a$ under the context-bounded rate-wise stochastic ordering, if:*

1) $\boldsymbol{P}_a \leq_{\text{st}} \boldsymbol{P}'_a$
2) *For all states s:*

$$1 \leq \max\left\{ \frac{r'_a(s)}{r_a(s)}, B_{comp} \right\} \leq \min_{s' \prec s} \left\{ \frac{1 - \sum_{t \succ s'} \boldsymbol{P}_a(s,t)}{1 - \sum_{t \succ s'} \boldsymbol{P}'_a(s,t)} \right\}$$

We extend the definition of rate-wise monotonicity similarly:

Definition 23. *A generator matrix $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$ is context-bounded rate-wise monotone with respect to B_{int}, if:*

1) *$\boldsymbol{P}_a$ is monotone.*
2) *For all states s, s' such that $s \prec s'$:*

$$1 \leq \max\left\{ \frac{r_a(s')}{r_a(s)}, B_{int} \right\} \leq \min_{s'' \prec s} \left\{ \frac{1 - \sum_{t \succ s''} \boldsymbol{P}_a(s,t)}{1 - \sum_{t \succ s''} \boldsymbol{P}_a(s',t)} \right\}$$

We can now prove that the CTMC of the system, after composing the individual components, is a monotone, lumpable upper bound of the concrete CTMC, with respect to the ordering on each component. For components C_1 and C_2 with state spaces $(S_1, \prec_1)$ and $(S_2, \prec_2)$ respectively, the $\circledast$ operator preserves stochastic comparison and monotonicity with respect to the lifted orders $(S_1 \times S_2, \prec_1^L)$ and $(S_1 \times S_2, \prec_2^L)$. We say

$(s_1, s_2) \prec_1^L (s_1', s_2')$ if $s_1 \prec_1 s_1'$, and $\neg \exists s_2''.\ s_2'' \prec_2 s_2$. We define $\prec_2^L$ similarly.

Theorem 24 (**Montonicity).** *Let two components, C_1 and C_2, occur in contexts $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, where $C_1 \in \mathcal{C}_2$ and $C_2 \in \mathcal{C}_1$. Let $\mathcal{C}_1$ be internally bounded by B^1_{int} and $\mathcal{C}_2$ by B^2_{int}, for action type a.*

If the matrices $\boldsymbol{Q}_{1,a} = r_{1,a}(\boldsymbol{P}_{1,a} - \boldsymbol{I})$ of C_1 and $\boldsymbol{Q}_{2,a} = r_{2,a}(\boldsymbol{P}_{2,a} - \boldsymbol{I})$ of C_2 are context-bounded rate-wise monotone by B^1_{int} and B^2_{int} respectively, then $(r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a})$ is context-bounded rate-wise monotone by the internal bound B^3_{int} of the context $\mathcal{C}_1 \cap \mathcal{C}_2$ of $C_1 \underset{L}{\bowtie} C_2$, for all action sets L.

Theorem 25 (**Lumpability).** *Let C_1 and C_2 be PEPA models with generator matrices $\boldsymbol{Q}_1 = \sum_a \boldsymbol{Q}_{1,a}$ and $\boldsymbol{Q}_2 = \sum_a \boldsymbol{Q}_{2,a}$, where $\boldsymbol{Q}_{1,a} = r_{1,a}(\boldsymbol{P}_{1,a} - \boldsymbol{I})$ and $\boldsymbol{Q}_{2,a} = r_{2,a}(\boldsymbol{P}'_{2,a} - \boldsymbol{I})$. Then for all action types a, if the terms $\boldsymbol{Q}_{1,a}$ in $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_{2,a}$ in $\boldsymbol{Q}_2$ are ordinarily lumpable according to the partitions $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively, then the term $\boldsymbol{Q}_a = (r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a})$ in $\boldsymbol{Q} = \sum_a \boldsymbol{Q}_a$ is ordinarily lumpable according to $\mathcal{L}_1 \times \mathcal{L}_2$.*

Theorem 26 (**Stochastic Order).** *Consider the components C_i and C_i', with generator matrices $\boldsymbol{Q}_{i,a} = r_{i,a}(\boldsymbol{P}_{i,a} - \boldsymbol{I})$ and $\boldsymbol{Q}'_{i,a} = r'_{i,a}(\boldsymbol{P}'_{i,a} - \boldsymbol{I})$, for $i \in \{1, 2\}$ and action type a. Let $\boldsymbol{Q}_{i,a} \leq_{\mathrm{rst}}^{B^i_{comp}} \boldsymbol{Q}'_{i,a}$, with contexts $\mathcal{C}_i \leq_{\mathrm{st}} \mathcal{C}_i'$, where B^i_{comp} is the comparative bound of $\mathcal{C}_i$ and $\mathcal{C}_i'$. If B^3_{comp} is the comparative bound of the contexts $\mathcal{C}_1 \cap \mathcal{C}_2$ and $\mathcal{C}'_1 \cap \mathcal{C}'_2$, we have $(r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a}) \leq_{\mathrm{rst}}^{B^3_{comp}} (r'_{1,a}, \boldsymbol{P}'_{1,a}) \circledast (r'_{2,a}, \boldsymbol{P}'_{2,a})$.*

The preservation of monotonicity and stochastic order is ensured by the definitions we have developed. Lumpability is preserved by the PEPA cooperation combinator, as a consequence of the strong equivalence congruence [8].

It is straightforward to show that corresponding theorems hold for compositional addition of matrices as per Theorem 11, if they have the same partially ordered state space. A consequence is that we just need to construct an upper bound for the rate function and probability transition matrix of each sequential component for each action type, so that when we expand the $\circledast$ and $\odot$ operators of the Kronecker form, we get an upper bound for the CTMC of the entire model.

VI. An Algorithm for Computing a Partially Ordered Stochastic Bound of a PEPA Component

Our algorithm for constructing an upper bound for a steady state property of a PEPA model is as follows. We assume n components, $C_1, \ldots, C_n$, in shared action cooperation form, with cooperation set L. For each action type $a \notin L$, we rename a to the local action type τ — i.e. we group all local transitions together. Each component C_i has an abstraction $(S_i^\sharp, \alpha_i)$, and a simple partial order specified by $\mathcal{B}_i = \{\mathcal{B}_{i,1}, \ldots \mathcal{B}_{i,m_i}\}$.

1) For each component C_i, we construct a mapping $\mathcal{I}_i$ from its state space S_i to matrix indices $\{1, \ldots, |S_i|\}$, so that states in the same partition have contiguous indices, which are ordered such that $s \prec s' \Rightarrow \mathcal{I}_i(s) < \mathcal{I}_i(s')$.
2) We compute a lumpable monotone upper bounding rate function $r'_{i,a}$ from the rate function $r_{i,a}$ of each component C_i and action type $a \in L$:

$$r'_{i,a}(s \in \mathcal{B}_{i,k}) = \max \left(\begin{array}{l} \{r_{i,a}(s') \mid \alpha_i(s') = \alpha_i(s)\} \cup \\ \bigcup_{j=k+1}^{m_i} \{r_{i,a}(s') \mid s' \in \mathcal{B}_{i,j}\} \end{array} \right)$$

3) We calculate the internal bound B_{int} and comparative bound B_{comp} for the context of each component and action type $a \in L$, using the bounded rate functions.
4) We compute an upper bounding probability transition matrix $\boldsymbol{R}_{i,a}$ from the transition matrix $\boldsymbol{P}_{i,a}$ of each component C_i and action type $a \in L$ (Algorithm 1).
5) For the internal action type τ, we uniformise the generator matrix $\boldsymbol{Q}_{i,\tau}$ of each component C_i, and apply Algorithm 1 with no context constraints. Since every state has the same exit rate, the algorithm reduces to that of Fourneau *et al*. over a partially ordered state space.
6) We construct and solve the generator matrix obtained by multiplying out the Kronecker representation of the upper bounding model[2].

Algorithm 1 An algorithm for constructing a context-bounded rate-wise upper bounding probability transition matrix

$y' \leftarrow |S_i^\sharp|$
for $y \leftarrow |S_i^\sharp|$ to 1 **do**
  if $b(y) = b(\mathcal{B}_k)$ for some $\mathcal{B}_k$ **then**
    refresh_sum$(\boldsymbol{P}, \boldsymbol{R}, b(y), e(y'))$
    normalise$(\boldsymbol{R}, b(y), e(y'))$
    for $p \leftarrow y'$ to y **do**
      normalise_partition$(\boldsymbol{R}, b(p), e(p))$
    end for
    $y' \leftarrow y - 1$
  end if
end for

It is important to note that not all of the components in the model necessarily need to have ordering constraints on their state space. For example, if we are interested in a property of just one component — i.e. the projection from the state space of the system onto that of the component — then we have no particular constraints on the probability distributions of the other components. But what does this mean in terms of constructing a bound for that component? The theorems in the previous section only account for when we *need* to bound a component. If we wish to exclude one of the components, we have to assume the ‘worst’ case — that is to say, that the component does not have any effect on the rest of the system.

Intuitively, when we bound a component, we maximise the probability of moving into higher valued states in the ordering. Since cooperation in PEPA takes the minimum of two rates, it is possible for a component to limit, but not increase, the transition rates for a particular action type. Hence a *monotone*

[2] To implement this, we need to explore the transition system generated by the new model, rather than performing the Kronecker multiplications explicitly. This avoids including unreachable states, which would result in a singular generator matrix.

Algorithm 2 *refresh_sum*$(\boldsymbol{P}, \boldsymbol{R}, b, e)$

for $\mathcal{B}_k \leftarrow \mathcal{B}_1$ to $\mathcal{B}_m$ **do**
  $R_{max} \leftarrow \max_{s \in \mathcal{B}_{k-1}} \sum_{j=b}^{|S|} \boldsymbol{R}(s, j)$
  $P_{max} \leftarrow \max_{s \in \mathcal{B}_k} \sum_{j=b}^{|S|} \boldsymbol{P}(s, j)$
  $B_I \leftarrow 1 - \min \left\{ B_{int}, \frac{\max_{s \in \mathcal{B}_k} r(s)}{\max_{s \in \mathcal{B}_k} r'(s)} \right\} (1 - R_{max})$
  $B_C \leftarrow 1 - \min \left\{ B_{comp}, \frac{\max_{s \in \mathcal{B}_{k-1}} r'(s)}{\max_{s \in \mathcal{B}_k} r'(s)} \right\} (1 - P_{max})$
  for $i \leftarrow b(\mathcal{B}_k)$ to $e(\mathcal{B}_k)$ **do**
    if $i \geq b$ **then**
      $\Sigma_{new} \leftarrow \max\{R_{max}, P_{max}, B_I, B_C\}$
    else
      $\Sigma_{new} \leftarrow \max\{R_{max}, P_{max}\}$
    end if
    $P_{new} \leftarrow \Sigma_{new} - \sum_{j'=e+1}^{|S|} \boldsymbol{R}(i, j')$
    $P_{old} \leftarrow \sum_{j=b}^{e} \boldsymbol{P}(i, j)$
    for $j \leftarrow b$ to e **do**
      if $P_{old} > 0$ **then**
        $\boldsymbol{R}(i, j) \leftarrow \frac{\boldsymbol{P}(i,j)}{P_{old}} P_{new}$
      else
        $\boldsymbol{R}(i, j) \leftarrow \frac{1}{e-b+1} P_{new}$
      end if
    end for
  end for
end for

Algorithm 3 *normalise*$(\boldsymbol{R}, b, e)$

for $y \leftarrow 1$ to $|S^\sharp|$ **do**
  $R_{new} \leftarrow \max_{i=b(y)}^{e(y)} \sum_{j=b}^{e} \boldsymbol{R}(i, j)$
  for $i \leftarrow b(y)$ to $e(y)$ **do**
    $R_{old} \leftarrow \sum_{j=b}^{e} \boldsymbol{R}(i, j)$
    for $j \leftarrow b$ to e **do**
      if $R_{old} > 0$ **then**
        $\boldsymbol{R}(i, j) \leftarrow \frac{\boldsymbol{R}(i,j)}{R_{old}} R_{new}$
      else
        $\boldsymbol{R}(i, j) \leftarrow \frac{1}{e-b+1} R_{new}$
      end if
    end for
  end for
end for

Algorithm 4 *normalise_partition*$(\boldsymbol{R}, b, e)$

for $y \leftarrow 1$ to $|S^\sharp|$ **do**
  $R_{average} \leftarrow \frac{1}{e(y)-b(y)+1} \sum_{i=b(y)}^{e(y)} \sum_{j=b}^{e} \boldsymbol{R}(i, j)$
  for $i \leftarrow b(y)$ to $e(y)$ **do**
    for $j \leftarrow b$ to e **do**
      $\boldsymbol{R}(i, j) \leftarrow R_{average}$
    end for
  end for
end for

upper bound for a component is a true upper bound in the worst case context. This means that we can ignore other components and still obtain, locally, an upper bound.

Let us examine Algorithm 1 in more detail. This takes as input a probability transition matrix $\boldsymbol{P}$, and an empty matrix $\boldsymbol{R}$ (of the same dimensions) in which to construct the monotone and lumpable upper bound. We assume that the upper bound r' of the rate function r has already been constructed, along with the internal and comparative bounds, B_{int} and B_{comp}. We define $b(y)$ and $e(y)$ respectively as the minimum and maximum index in the set $\{\mathcal{I}(s) \mid \mathcal{I}^\sharp(\alpha(s)) = y\}$[3]. $b(\mathcal{B}_k)$ and $e(\mathcal{B}_k)$ are defined similarly for the set $\{\mathcal{I}(s) \mid s \in \mathcal{B}_k\}$.

Algorithm 1 makes use of three sub-procedures:

1) *refresh_sum*$(\boldsymbol{P}, \boldsymbol{R}, b, e)$ (Algorithm 2) ensures that for each ordering block, from indices b to e, the matrix $\boldsymbol{R}$ is context-bounded rate-wise monotone, and an upper bound of $\boldsymbol{P}$. The core of this algorithm is the computation of Σ_{new}, where the bounds B_I and B_C come directly from re-arranging the definitions of context-bounded rate-wise stochastic ordering and monotonicity respectively (Definitions 22 and 23). These additional constraints are only needed for elements on or below the diagonal ($i \geq b$), since the '$-r_a\boldsymbol{I}$' term — which is the reason for the rate-wise extension to stochastic ordering and monotonicity — only applies to sums that include the diagonal element.
   To achieve a new row sum of Σ_{new}, we adjust the individual entries in $\boldsymbol{R}$ so that the relative probabilities are preserved. This is a choice that we make, to minimise our impact on the matrix — because we do not have a total order, we can distribute the probability mass within an ordering block in any way.
2) *normalise*$(\boldsymbol{R}, b, e)$ (Algorithm 3) ensures that for each partition in the ordering block from indices b to e, states in the same partition have the same probability of moving to a different ordering block.
3) *normalise_partition*$(\boldsymbol{R}, b, e)$ (Algorithm 4) ensures that each state in the partition from indices b to e has the same probability of moving to another partition. We choose to assign the average transition probabilities.

Essentially, the *normalise* procedure ensures lumpability of ordering blocks — by 'borrowing' probability mass from

[3] $\mathcal{I}^\sharp$ is defined such that $\mathcal{I}(s) < \mathcal{I}(s') \Rightarrow \mathcal{I}^\sharp(\alpha(s)) \leq \mathcal{I}^\sharp(\alpha(s'))$.

$$
\begin{aligned}
PC_i &= (arrive, \lambda_i).PC'_i + (walkon_{i\oplus 1}, \top).PC_i\\
PC'_i &= (serve_i, \top).PC_i\\
Server_i &= (walkon_{i\oplus 1}, \omega).Server_{i\oplus 1} + (serve_i, \mu).Server'_i\\
Server'_i &= (walk_{i\oplus 1}, \omega).Server_{i\oplus 1}
\end{aligned}
$$

$$(PC_0 \parallel \cdots \parallel PC_{n-1}) \underset{\{walkon, serve\}}{\bowtie} Server_0$$

Fig. 5. A PEPA model of a round-robin server

Aggregated States	State Space Size	Upper Bound
None	768	0.31184
$\{ Server'_{0\ldots5} \}$	7	0.33333
$\{ Server_{0\ldots5} \}$	7	0.75000
$\{ Server'_{2\ldots5}, Server_{3\ldots5} \}$	6	1.00000

Fig. 6. Steady state probability of being in a $Server'_i$ state

lower ordering blocks — which preserves monotonicity. The *normalise_partition* procedure then ensures lumpability of partitions, by re-distributing probability mass within the same ordering block.

VII. An Example

We have implemented the stochastic bounding algorithm presented in the previous section as part of the PEPA plug-in for Eclipse [1], and we will now consider its application to a PEPA model. Consider the model in Figure 5. This describes a set of n PCs, which are serviced in a round-robin fashion by a server. Jobs arrive at computer PC_i at rate λ_i, and the service rate of the server is μ. The server in state i moves to state $i \oplus 1 = (i+1) \bmod n$, after serving a job from PC i (with the *walk* activity) or not (with the *walkon* activity).

Consider this model when $n = 6$, in which case the concrete PEPA model has 768 states. To avoid any symmetry in the model that could allow a more exact aggregation, we will assume that every computer has a different arrival rate: $\lambda_i = i + 1$. One property of interest is the proportion of time the server spends in a $Server'_i$ state — in other words, it has just served a job but is not yet ready to serve another. Figure 6 shows the steady state probability of $\{Server'_i\}$ for different choices of states to aggregate in the $Server$ component. The first entry is the exact probability, and the others are upper bounds obtained by the simple partial order $\mathcal{B}_1 = \{Server_i\}$ and $\mathcal{B}_2 = \{Server'_i\}$. Note that this partial order is the best we can choose given the property of interest, as it is the *weakest* ordering, and therefore imposes the least number of constraints.

From these results, we see that aggregating all the $Server'$ states gives a good upper bound on the actual probability. The other choices of aggregation yield poor results, however, which illustrates how the choice of abstraction can dramatically affect the precision. In general, we cannot obtain precise bounds for all models and properties, but this technique has the advantage of being very fast, even if we obtain only limited information.

VIII. Conclusions

Stochastic bounding is a powerful technique for reducing the size of a Markov chain, and allows us to obtain bounds on steady state probabilities. We have applied this compositionally to PEPA models, allowing us to bound models where the underlying state space is too large to represent. The algorithm we have presented and implemented enables these bounds to be constructed automatically, given an ordering and partitioning of the state space of each component.

There are many other techniques for abstracting Markov chains that we did not mention — to name but a few, abstract Markov chains [5], disaggregation/aggregation [12] and quasi-lumpability [4]. Abstract Markov chains are useful for bounding transient properties, but not for steady state properties, since the abstraction does not result in a Markov chain. Disaggregation/aggregation and quasi-lumpability can be applied to steady state properties, but are approximate techniques — it is difficult to reason about the error introduced.

Whilst we have demonstrated the utility of compositional stochastic bounds, there remain many interesting future research directions. In particular, we intend to investigate the benefits of combining components before applying the abstraction. In summary, stochastic bounding is a useful technique for stochastic process algebra modellers to have at their disposal, and we feel that we have broadened its applicability.

References

[1] The PEPA plug-in for Eclipse. Available to download from: `http://www.dcs.ed.ac.uk/pepa/tools/plugin/`.

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

[3] D. Daly. *Bounded aggregation techniques to solve large Markov models*. PhD thesis, Champaign, IL, USA, 2005.

[4] T. Dayar and W.J. Stewart. Quasi-lumpability, lower-bounding coupling matrices, and nearly completely decomposable Markov chains. *SIAM Journal on Matrix Analysis and Applications*, 18(2):482–498, 1997.

[5] H. Fecher, M. Leucker, and V. Wolf. Don't know in probabilistic systems. In *Proceedings of SPIN'06*, number 3925 in LNCS, pages 71–88, 2006.

[6] J.-M. Fourneau and L. Kloul. A precedence PEPA model for performance and reliability analysis. In A. Horváth and M. 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, Jun 2006.

[7] 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.

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

[9] J. Hillston and L. Kloul. An efficient Kronecker representation for PEPA models. In *Proceedings of the Joint International Workshop, PAPM-PROBMIV '01*, number 2165 in LNCS, pages 120–135. Springer, 2001.

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

[11] B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. *SIGMETRICS Performance Evaluation Review*, 13(2):147–154, 1985.

[12] H.A. Simon and A. Ando. Aggregation of variables in dynamic systems. *Econometrica*, 29(2):111–138, 1961.

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

[14] M. Tremolieres, J.M. Vincent, and B. Plateau. Determination of the optimal stochastic upper bound of a Markovian generator. Technical Report Rapport de Recherche RR 906-I, 1992.

Appendix

Theorem 11. *Consider two generator matrices $\boldsymbol{Q}_1 = (r_1, \boldsymbol{P}_1)$ and $\boldsymbol{Q}_2 = (r_2, \boldsymbol{P}_2)$, corresponding to the same state space S ($\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ are both $|S| \times |S|$ matrices). Then $\boldsymbol{Q}_1 + \boldsymbol{Q}_2$ can be written as follows:*

$$(r_1, \boldsymbol{P}_1) + (r_2, \boldsymbol{P}_2) = \left(r_1 + r_2, \frac{r_1}{r_1 + r_2} \boldsymbol{P}_1 + \frac{r_2}{r_1 + r_2} \boldsymbol{P}_2 \right)$$

where $(r_1 + r_2)(s) = r_1(s) + r_2(s)$, and $\frac{r_i}{r_1+r_2}(s) = \frac{r_i(s)}{r_1(s)+r_2(s)}$, $i \in \{1, 2\}$, for all $s \in S$.

Proof: The proof is as follows. For an entry s, s', when $s \neq s'$, in $\boldsymbol{Q}_1 + \boldsymbol{Q}_2$, we have:

$$\begin{aligned} (\boldsymbol{Q}_1 + \boldsymbol{Q}_2)(s, s') &= ((r_1, \boldsymbol{P}_1) + (r_2, \boldsymbol{P}_2))(s, s') \\ &= (r_1, \boldsymbol{P}_1)(s, s') + (r_2, \boldsymbol{P}_2)(s, s') \\ &= r_1(s)\boldsymbol{P}_1(s, s') + r_2(s)\boldsymbol{P}_2(s, s') \\ &= (r_1(s) + r_2(s)) \Big(\tfrac{r_1(s)}{r_1(s)+r_2(s)} \boldsymbol{P}_1(s, s') + \\ &\qquad \tfrac{r_2(s)}{r_1(s)+r_2(s)} \boldsymbol{P}_2(s, s') \Big) \\ &= \left(r_1 + r_2, \tfrac{r_1}{r_1+r_2} \boldsymbol{P}_1 + \tfrac{r_2}{r_1+r_2} \boldsymbol{P}_2 \right) (s, s') \end{aligned}$$

When $s = s'$, we similarly have:

$$\begin{aligned} (\boldsymbol{Q}_1 + \boldsymbol{Q}_2)(s, s) &= ((r_1, \boldsymbol{P}_1) + (r_2, \boldsymbol{P}_2))(s, s) \\ &= (r_1, \boldsymbol{P}_1)(s, s) + (r_2, \boldsymbol{P}_2)(s, s) \\ &= r_1(s)(\boldsymbol{P}_1(s, s) - 1) + r_2(s)(\boldsymbol{P}_2(s, s) - 1) \\ &= (r_1(s) + r_2(s)) \Big(\tfrac{r_1(s)}{r_1(s)+r_2(s)} \boldsymbol{P}_1(s, s) + \\ &\qquad \tfrac{r_2(s)}{r_1(s)+r_2(s)} \boldsymbol{P}_2(s, s) - 1 \Big) \\ &= \left(r_1 + r_2, \tfrac{r_1}{r_1+r_2} \boldsymbol{P}_1 + \tfrac{r_2}{r_1+r_2} \boldsymbol{P}_2 \right) (s, s) \end{aligned}$$

∎

Theorem 13. *For all well-formed PEPA models C, the Markov chain given by the generator matrix $\mathcal{Q}(C)$ and the Markov chain induced by the semantics of PEPA have consistent rates of transition between states.*

Proof: Since, in a PEPA model, the transitions of each action type are independent from one another, we can consider them separately. We therefore need to prove that for each action type, the transition rates induced by the Kronecker form are identical to those induced by the PEPA semantics, as given in [8].

We proceed by induction on the structure of the system equation. In the base case, for a sequential component C_i, $\mathcal{Q}_a(C_i) = (r_{i,a}, \boldsymbol{P}_{i,a})$ corresponds, by Definition 12, precisely to those activities of type a that C_i can perform. In other words, for $s, s' \in S_i$, $r_{i,a}(s)$ is the apparent rate of action type a in state s, and $\boldsymbol{P}_{i,a}(s, s')$ is the relative probability of moving to state s', if we perform an a activity in state s.

For the inductive case, we make the hypothesis that there is a transition $C_1 \xrightarrow{(a,r)} C_2$ induced by the PEPA semantics of a component C — where C may be a composition of sequential components, and $C_1, C_2 \in \mathrm{ds}(C)$ — if and only if $\mathcal{Q}_a(C)(C_1, C_2) = r_a(C_1)\boldsymbol{P}_a(C_1, C_2) = r$. We can ignore the case when $C_1 = C_2$ as self loops cancel out in the generator matrix.

Assume that there is an additional component C' such that $C'_1 \xrightarrow{(a,r')} C'_2$ iff $\mathcal{Q}_a(C')(C'_1, C'_2) = r'$, as above. We will prove that for all sets of action types L, $C \underset{L}{\bowtie} C'$ induces a transition $C_1 \underset{L}{\bowtie} C'_1 \xrightarrow{(a,R)} C_2 \underset{L}{\bowtie} C'_2$ iff $\mathcal{Q}_a(C \underset{L}{\bowtie} C')(C_1 \underset{L}{\bowtie} C'_1, C_2 \underset{L}{\bowtie} C'_2) = R$.

Consider the case $a \in L$. Then:

$$\begin{aligned} & \mathcal{Q}_a(C \underset{L}{\bowtie} C')(C_1 \underset{L}{\bowtie} C'_1, C_2 \underset{L}{\bowtie} C'_2) \\ =\; & (\mathcal{Q}_a(C) \circledast \mathcal{Q}_a(C'))(C_1 \underset{L}{\bowtie} C'_1, C_2 \underset{L}{\bowtie} C'_2) \\ =\; & ((r_a, \boldsymbol{P}_a) \circledast (r'_a, \boldsymbol{P}'_a))(C_1 \underset{L}{\bowtie} C'_1, C_2 \underset{L}{\bowtie} C'_2) \\ =\; & (\min\{r_a, r'_a\}, \boldsymbol{P}_a \otimes \boldsymbol{P}'_a)(C_1 \underset{L}{\bowtie} C'_1, C_2 \underset{L}{\bowtie} C'_2) \\ =\; & \min\{r_a(C_1), r'_a(C'_1)\}(\boldsymbol{P}_a(C_1, C_2) \times \boldsymbol{P}'_a(C'_1, C'_2)) \\ =\; & \min\{r_a(C_1), r'_a(C'_1)\} \tfrac{r}{r_a(C_1)} \tfrac{r'}{r'_a(C'_1)} \end{aligned}$$

where the final step follows from the induction hypothesis. This is equal by definition to the PEPA semantics of cooperation for action types $a \in L$, hence gives the rate R of the transition $C_1 \underset{L}{\bowtie} C'_1 \xrightarrow{(a,R)} C_2 \underset{L}{\bowtie} C'_2$ induced by the PEPA semantics.

Consider the case $a \notin L$. Then:

$$
\begin{aligned}
&\mathcal{Q}_a(C \underset{L}{\bowtie} C')(C_1 \underset{L}{\bowtie} C_1', C_2 \underset{L}{\bowtie} C_2') \\
&= (\mathcal{Q}_a(C) \odot \mathcal{Q}_a(C'))(C_1 \underset{L}{\bowtie} C_1', C_2 \underset{L}{\bowtie} C_2') \\
&= ((r_a, \boldsymbol{P}_a) \odot (r_a', \boldsymbol{P}_a'))(C_1 \underset{L}{\bowtie} C_1', C_2 \underset{L}{\bowtie} C_2') \\
&= ((r_a, \boldsymbol{P}_a) \circledast (r_\top, \boldsymbol{I}) + (r_\top, \boldsymbol{I}) \circledast (r_a', \boldsymbol{P}_a')) \\
&\qquad (C_1 \underset{L}{\bowtie} C_1', C_2 \underset{L}{\bowtie} C_2') \\
&= \min\{r_a, \top\}(C_1, C_1')\boldsymbol{P}_a(C_1, C_2)\boldsymbol{I}(C_1', C_2') + \\
&\qquad \min\{\top, r_a'\}(C_1, C_1')\boldsymbol{I}(C_1, C_2)\boldsymbol{P}_a'(C_1', C_2') \\
&= r_a(C_1)\boldsymbol{P}_a(C_1, C_2)\boldsymbol{I}(C_1', C_2') + \\
&\qquad r_a'(C_1')\boldsymbol{I}(C_1, C_2)\boldsymbol{P}_a'(C_1', C_2') \\
&= \begin{cases} r_a(C_1)\boldsymbol{P}_a(C_1, C_2) & \text{if } C_2' = C_1' \\ r_a'(C_1')\boldsymbol{P}_a'(C_1', C_2') & \text{if } C_2 = C_1 \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} r & \text{if } C_2' = C_1' \\ r' & \text{if } C_2 = C_1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

where the final step follows from the induction hypothesis. This corresponds to the PEPA semantics of cooperation for action types $a \notin L$, where the activities of the two components take place independently. In other words, $C_1 \underset{L}{\bowtie} C_1' \xrightarrow{(a,r)} C_2 \underset{L}{\bowtie} C_1'$ if $C_1 \xrightarrow{(a,r)} C_2$ in component C, and $C_1 \underset{L}{\bowtie} C_1' \xrightarrow{(a,r')} C_1 \underset{L}{\bowtie} C_2'$ if $C_1' \xrightarrow{(a,r')} C_2'$ in C'. ∎

Theorem 17. *If $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I}) \leq_{\mathrm{rst}} \boldsymbol{Q}_a' = r_a'(\boldsymbol{P}_a' - \boldsymbol{I}')$ and for all $s \in S$, $r_a(s) \leq r_a'(s)$, then $\boldsymbol{Q}_a \leq_{\mathrm{st}} \boldsymbol{Q}_a'$.*

Proof: For any λ that is not exceeded in magnitude by any diagonal element of $\boldsymbol{Q}_a$ or $\boldsymbol{Q}_a'$, we need to show that $\frac{\boldsymbol{Q}_a}{\lambda} + \boldsymbol{I} \leq_{\mathrm{st}} \frac{\boldsymbol{Q}_a'}{\lambda} + \boldsymbol{I}$, by the definition of the stochastic ordering on CTMCs. This corresponds to showing that:

$$
\begin{aligned}
&\frac{r_a \boldsymbol{P}_a}{\lambda} + \left(1 - \frac{r_a}{\lambda}\right)\boldsymbol{I} \\
\leq_{\mathrm{st}}\ &\frac{r_a' \boldsymbol{P}_a'}{\lambda} + \left(1 - \frac{r_a'}{\lambda}\right)\boldsymbol{I}
\end{aligned}
$$

remembering that r_a and r_a' are apparent rate functions, which can be written as vectors. By the definition of the strong stochastic ordering, this requires that, for each row s and for all states s':

$$
\begin{aligned}
&\frac{r_a(s)}{\lambda}\sum_{t \succ s'} \boldsymbol{P}_a(s,t) + \left(1 - \frac{r_a(s)}{\lambda}\right)\mathbf{1}_{s' \prec s} \\
\leq\ &\frac{r_a'(s)}{\lambda}\sum_{t \succ s'} \boldsymbol{P}_a'(s,t) + \left(1 - \frac{r_a'(s)}{\lambda}\right)\mathbf{1}_{s' \prec s}
\end{aligned}
$$

where $\mathbf{1}_{s' \prec s}$ is the indicator function, evaluating to one if the condition $s' \prec s$ holds, and to zero otherwise. If we are above the diagonal element (i.e. the indicator term evaluates to zero), then the relation holds since $r_a(s) \leq r_a'(s)$ and $\boldsymbol{P}_a \leq_{\mathrm{st}} \boldsymbol{P}_a'$. Otherwise, we have, for $s' \prec s$:

$$
\begin{aligned}
&\frac{r_a(s)}{\lambda}\sum_{t \succ s'} \boldsymbol{P}_a(s,t) - \frac{r_a(s)}{\lambda} \\
\leq\ &\frac{r_a'(s)}{\lambda}\sum_{t \succ s'} \boldsymbol{P}_a'(s,t) - \frac{r_a'(s)}{\lambda}
\end{aligned}
$$

which, on re-arranging, gives:

$$
\frac{r_a'(s)}{r_a(s)} \leq \frac{1 - \sum_{t \succ s'} \boldsymbol{P}_a(s,t)}{1 - \sum_{t \succ s'} \boldsymbol{P}_a'(s,t)}
$$

But since we know that the left-hand side is less than or equal to the minimum of all possible ratios on the right-hand side, this holds for all s'. ∎

Theorem 18. *If $\boldsymbol{Q}_a = r_a(\boldsymbol{P}_a - \boldsymbol{I})$ is rate-wise monotone, and for all $s \prec s' \in S$, $r_a(s) \leq r_a(s')$, then $\boldsymbol{Q}_a$ is monotone.*

Proof: For any λ that is not exceeded in magnitude by any diagonal element of $\boldsymbol{Q}_a$, we need to show that $\frac{\boldsymbol{Q}_a}{\lambda} + \boldsymbol{I}$ is monotone, by the definition of monotonicity for CTMCs. This corresponds to showing that:

$$\frac{r_a \boldsymbol{P}_a}{\lambda} + \left(1 - \frac{r_a}{\lambda}\right) \boldsymbol{I}$$

is monotone, where we write the apparent rate function r_a as a vector. By the definition of monotonicity, we require for all states s and s', such that $s \prec s'$ (two rows that we compare), and for all states s'' (elements along the row):

$$\begin{aligned} &\frac{r_a(s)}{\lambda} \sum_{t \succ s''} \boldsymbol{P}_a(s,t) + \left(1 - \frac{r_a(s)}{\lambda}\right) \mathbf{1}_{s'' \prec s} \\ \leq\ &\frac{r_a(s')}{\lambda} \sum_{t \succ s''} \boldsymbol{P}_a(s',t) + \left(1 - \frac{r_a(s')}{\lambda}\right) \mathbf{1}_{s'' \prec s'} \end{aligned}$$

If we are above the diagonal element in both rows (i.e. the indicator terms $\mathbf{1}_{s'' \prec s}$ and $\mathbf{1}_{s'' \prec s'}$ both evaluate to zero), then the relation holds since $r_a(s) \leq r_a(s')$ and $\boldsymbol{P}_a$ is monotone. Otherwise, we have to consider two cases; when $s \prec s'' \prec s'$, and when $s'' \prec s$.

When $s \prec s'' \prec s'$, we have:

$$\begin{aligned} &\frac{r_a(s)}{\lambda} \sum_{t \succ s''} \boldsymbol{P}_a(s,t) \\ \leq\ &\frac{r_a(s')}{\lambda} \sum_{t \succ s''} \boldsymbol{P}_a(s',t) + 1 - \frac{r_a(s')}{\lambda} \end{aligned}$$

which holds as before, since $1 - \frac{r_a(s')}{\lambda} > 0$.

Finally, when $s'' \prec s$, we have:

$$\begin{aligned} &\frac{r_a(s)}{\lambda} \sum_{t \succ s''} \boldsymbol{P}_a(s,t) - \frac{r_a(s)}{\lambda} \\ \leq\ &\frac{r_a(s')}{\lambda} \sum_{t \succ s''} \boldsymbol{P}_a(s',t) - \frac{r_a(s')}{\lambda} \end{aligned}$$

which, on re-arranging, gives:

$$\frac{r_a(s')}{r_a(s)} \leq \frac{1 - \sum_{t \succ s''} \boldsymbol{P}_a(s,t)}{1 - \sum_{t \succ s''} \boldsymbol{P}_a(s',t)}$$

But since we know that the left-hand side is less than or equal to the minimum of all possible ratios on the right-hand side, this holds for all s''. ∎

Theorem 24 (Montonicity). *Let two components, C_1 and C_2, occur in contexts $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, such that $C_1 \in \mathcal{C}_2$ and $C_2 \in \mathcal{C}_1$. Let $\mathcal{C}_1$ be internally bounded by B^1_{int}, and $\mathcal{C}_2$ by B^2_{int}, for action type a.*

If the matrices $\boldsymbol{Q}_{1,a} = r_{1,a}(\boldsymbol{P}_{1,a} - \boldsymbol{I})$ of C_1 and $\boldsymbol{Q}_{2,a} = r_{2,a}(\boldsymbol{P}_{2,a} - \boldsymbol{I})$ of C_2 are context-bounded rate-wise monotone by B^1_{int} and B^2_{int} respectively, then $(r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a})$ is context-bounded rate-wise monotone by the internal bound B^3_{int} of the context $\mathcal{C}_3$ of $C_1 \underset{L}{\bowtie} C_2$, for all action sets L.

To prove this theorem, we will first establish the following two lemmas. Lemma 27 shows that the Kronecker product preserves monotonicity, and Lemma 28 shows that the minimum of two monotone functions is also monotone. We omit the subscript a for brevity, hence we write $\boldsymbol{P}_i$ in place of $\boldsymbol{P}_{i,a}$, and r_i in place of $r_{i,a}$ for $i \in \{1,2\}$:

Lemma 27. *Let $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ be monotone stochastic matrices describing the PEPA components C_1 and C_2 respectively, which have state spaces $(S_1, \prec_1)$ and $(S_2, \prec_2)$. Then $\boldsymbol{P}_1 \otimes \boldsymbol{P}_2$ is also monotone under the lifted orderings $\prec^L_1$ and $\prec^L_2$ on $S_1 \times S_2$.*

Proof: Consider states $(s_1, s_2) \prec^L_1 (s'_1, s'_2)$, recalling that this implies that $s_1 \prec_1 s'_1$ and $\neg\exists s''_2.\ s''_2 \prec_2 s_2$. We need to show that the following inequality holds, for all $s \in \mathrm{ds}(C_1)$ and $t \in \mathrm{ds}(C_2)$:

$$\begin{aligned} &\sum_{(s',t') \succ^L_1 (s,t)} \boldsymbol{P}_1(s_1,s')\boldsymbol{P}_2(s_2,t') \\ \leq &\sum_{(s',t') \succ^L_1 (s,t)} \boldsymbol{P}_1(s'_1,s')\boldsymbol{P}_2(s'_2,t') \end{aligned}$$

But in order for there to be any states $(s', t') \succ (s, t)$, t must be the smallest state in $(S_2, \prec_2)$. Hence this is equivalent to:

$$\begin{aligned} &\sum_{s' \succ_1 s} \sum_{t' \in S_2} \boldsymbol{P}_1(s_1,s')\boldsymbol{P}_2(s_2,t') \\ \leq &\sum_{s' \succ_1 s} \sum_{t' \in S_2} \boldsymbol{P}_1(s'_1,s')\boldsymbol{P}_2(s'_2,t') \end{aligned}$$

which we rewrite to give:

$$\sum_{s' \succ_1 s} \boldsymbol{P}_1(s_1,s') \leq \sum_{s' \succ_1 s} \boldsymbol{P}_1(s'_1,s')$$

This holds since $\boldsymbol{P}_1$ is monotone under $(S_1, \prec_1)$. The proof of monotonicity under $(S_1 \times S_2, \prec^L_2)$ follows similarly. ∎

Lemma 28. *Let r_1 and r_2 be monotone functions. Then $r_3(s_1, s_2) = \min\{r_1(s_1), r_2(s_2)\}$ is also monotone, under the orderings $(S_1 \times S_2, \prec^L_1)$ and $(S_1 \times S_2, \prec^L_2)$.*

Proof: Consider states $(s_1, s_2) \prec^L_1 (s'_1, s'_2)$. By definition, $s_1 \prec_1 s'_1$ and either $s_2 \prec_1 s'_2$ or $s_2 = s'_2$. There are two cases to consider:

Case 1: $\min\{r_1(s'_1), r_2(s'_2)\} = r_1(s'_1)$. Then:

$$\begin{aligned} \min\{r_1(s_1), r_2(s_2)\} &\leq r_1(s_1) \\ &\leq r_1(s'_1) \\ &\leq \min\{r_1(s'_1), r_2(s'_2)\} \end{aligned}$$

Case 2: $\min\{r_1(s_1'), r_2(s_2')\} = r_2(s_2')$. Then:

$$\begin{aligned}\min\{r_1(s_1), r_2(s_2)\} &\le r_2(s_2)\\ &\le r_2(s_2')\\ &\le \min\{r_1(s_1'), r_2(s_2')\}\end{aligned}$$

Hence r_3 is monotone with respect to $(S_1 \times S_2, \prec_1^L)$. The proof of monotonicity under $(S_1 \times S_2, \prec_2^L)$ follows similarly. ■

Proof: **[Theorem 24]** Let $(S_1, \prec_1)$ and $(S_2, \prec_2)$ be the state spaces of components C_1 and C_2 respectively. We will show that the generator matrix $\min\{r_1, r_2\}(\boldsymbol{P}_1 \otimes \boldsymbol{P}_2 - \boldsymbol{I})$ is monotone with respect to $(S_1 \times S_2, \prec_1^L)$.

We know from Lemma 27 that the matrix $\boldsymbol{P}_1 \otimes \boldsymbol{P}_2$ is monotone, and from Lemma 28 that the apparent rate function $\min\{r_1, r_2\}$ is monotone increasing. Hence, for all states $(s_1, s_2) \prec_1^L (s_1', s_2')$, we need to show that:

$$\max\left\{B_{int}^3, \frac{\min\{r_1(s_1'), r_2(s_2')\}}{\min\{r_1(s_1), r_2(s_2)\}}\right\} \le \min_{(t_1,t_2)\prec_1^L(s_1,s_2)} \left\{\frac{1 - \sum_{(t_1',t_2')\succ_1^L(t_1,t_2)} \boldsymbol{P}_1(s_1,t_1')\boldsymbol{P}_2(s_2,t_2')}{1 - \sum_{(t_1',t_2')\succ_1^L(t_1,t_2)} \boldsymbol{P}_1(s_1',t_1')\boldsymbol{P}_2(s_2',t_2')}\right\}$$

where B_{int}^3 is the internal bound of the context $\mathcal{C}''$ of $C \bowtie_L C'$:

Let (t_1, t_2) be the state under which the ratio on the right hand side is at a minimum. Since $t_1 \prec_1 s_1$ by definition of $\prec_1^L$, we know that the following relation holds:

$$\max\left\{B_{int}^1, \frac{r_1(s_1')}{r_1(s_1)}\right\} \le \frac{1 - \sum_{t_1'\succ_1 t_1} \boldsymbol{P}_1(s_1,t_1')}{1 - \sum_{t_1'\succ_1 t_1} \boldsymbol{P}_1(s_1',t_1')}$$

Furthermore, since by definition $B_{int}^1 \ge \frac{r_2(s_2')}{r_2(s_2)}$, $B_{int}^2 \ge \frac{r_1(s_1')}{r_1(s_1)}$, and $B_{int}^3 \le \min\{B_{int}^1, B_{int}^2\}$, we can infer that:

$$\max\left\{B_{int}^3, \frac{r_1(s_1')}{r_1(s_1)}, \frac{r_2(s_2')}{r_2(s_2)}\right\} \le \max\left\{B_{int}^1, \frac{r_1(s_1')}{r_1(s_1)}\right\}$$

To complete the proof, we need to make use of the following observation:

Observation 29. *For all positive* $a, b, c, d \in \mathbb{R}$*:*

$$\max\left\{\frac{a}{b}, \frac{c}{d}\right\} \ge \frac{\min\{a, c\}}{\min\{b, d\}}$$

since $\frac{a}{b} \ge \frac{\min a,c}{b}$ *and* $\frac{c}{d} \ge \frac{\min a,c}{d}$.

Using this observation, and the fact that t_2 must be the smallest state in $(S_2, \prec_2)$ by the definition of $\prec_1^L$:

$$\begin{aligned}
&\max\left\{B_{int}^3, \frac{\min\{r_1(s_1'), r_2(s_2')\}}{\min\{r_1(s_1), r_2(s_2)\}}\right\}\\
&\le \max\left\{B_{int}^3, \frac{r_1(s_1')}{r_1(s_1)}, \frac{r_2(s_2')}{r_2(s_2)}\right\}\\
&\le \max\left\{B_{int}^1, \frac{r_1(s_1')}{r_1(s_1)}\right\}\\
&\le \frac{1 - \sum_{t_1'\succ_1 t_1} \boldsymbol{P}_1(s_1,t_1')}{1 - \sum_{t_1'\succ_1 t_1} \boldsymbol{P}_1(s_1',t_1')}\\
&= \frac{1 - \sum_{t_1'\succ_1 t_1} \boldsymbol{P}_1(s_1,t_1') \sum_{t_2'\in S_2} \boldsymbol{P}_2(s_2,t_2')}{1 - \sum_{t_1'\succ_1 t_1} \boldsymbol{P}_1(s_1',t_1') \sum_{t_2'\in S_2} \boldsymbol{P}_2(s_2',t_2')}\\
&\le \frac{1 - \sum_{(t_1',t_2')\succ_1^L(t_1,t_2)} \boldsymbol{P}_1(s_1,t_1')\boldsymbol{P}_2(s_2,t_2')}{1 - \sum_{(t_1',t_2')\succ_1^L(t_1,t_2)} \boldsymbol{P}_1(s_1',t_1')\boldsymbol{P}_2(s_2',t_2')}\\
&= \min_{(t_1,t_2)\prec(s_1,s_2)} \left\{\frac{1 - \sum_{(t_1',t_2')\succ_1^L(t_1,t_2)} \boldsymbol{P}_1(s_1,t_1')\boldsymbol{P}_2(s_2,t_2')}{1 - \sum_{(t_1',t_2')\succ_1^L(t_1,t_2)} \boldsymbol{P}_1(s_1',t_1')\boldsymbol{P}_2(s_2',t_2')}\right\}
\end{aligned}$$

The proof of monotonicity under $(S_1 \times S_2, \prec_2^L)$ follows similarly.

Thus context-bounded rate-wise monotonicity is preserved by $\circledast$. ■

Theorem 25 (Lumpability). *Let C_1 and C_2 be PEPA models with generator matrices $\boldsymbol{Q}_1 = \sum_a \boldsymbol{Q}_{1,a}$ and $\boldsymbol{Q}_2 = \sum_a \boldsymbol{Q}_{2,a}$, where $\boldsymbol{Q}_{1,a} = r_{1,a}(\boldsymbol{P}_{1,a} - \boldsymbol{I})$ and $\boldsymbol{Q}_{2,a} = r_{2,a}(\boldsymbol{P}'_{2,a} - \boldsymbol{I})$. Then for all action types a, if the terms $\boldsymbol{Q}_{1,a}$ in $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_{2,a}$ in $\boldsymbol{Q}_2$ are ordinarily lumpable according to the partitions $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively, then the term $\boldsymbol{Q}_a = (r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a})$ in $\boldsymbol{Q} = \sum_a \boldsymbol{Q}_a$ is ordinarily lumpable according to $\mathcal{L}_1 \times \mathcal{L}_2$.*

Proof: Observe that $\boldsymbol{Q}$ is the generator matrix of $C_1 \underset{L}{\bowtie} C_2$, for some cooperation set L. For each action type a, $\boldsymbol{Q}_{1,a}$ is strongly equivalent to the lumped matrix $\mathcal{L}_1(\boldsymbol{Q}_{1,a})$, in that any a activity in $\boldsymbol{Q}_{1,a}$ can be matched by an a activity in $\mathcal{L}_1(\boldsymbol{Q}_{1,a})$. Similarly, $\boldsymbol{Q}_{2,a}$ and $\mathcal{L}_2(\boldsymbol{Q}_{2,a})$ are strongly equivalent.

It has been proven in [8] that the PEPA cooperation combinator preserves strong equivalence, and that strong equivalence implies ordinary lumpability. It follows that $\boldsymbol{Q}_a = (r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a})$ is strongly equivalent to:

$$\begin{aligned} &\mathcal{L}_1(r_{1,a}, \boldsymbol{P}_{1,a}) \circledast \mathcal{L}_2(r_{2,a}, \boldsymbol{P}_{2,a}) \\ =\ &(\mathcal{L}_1 \times \mathcal{L}_2)((r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a})) \end{aligned}$$

Hence $\boldsymbol{Q}_a$ is ordinarily lumpable according to the partition $\mathcal{L}_1 \times \mathcal{L}_2$.

Since for all action types $\boldsymbol{Q}_a$ is ordinarily lumpable, it follows that $\boldsymbol{Q} = \sum_a \boldsymbol{Q}_a$ is ordinarily lumpable according to $\mathcal{L}_1 \times \mathcal{L}_2$. Therefore, the operator $\circledast$ preserves ordinary lumpability over all action types. ∎

Theorem 26 (Stochastic Order). *Consider the components C_i and C'_i, with generator matrices $\boldsymbol{Q}_{i,a} = r_{i,a}(\boldsymbol{P}_{i,a} - \boldsymbol{I})$ and $\boldsymbol{Q}'_{i,a} = r'_{i,a}(\boldsymbol{P}'_{i,a} - \boldsymbol{I})$, for $i \in \{1,2\}$ and action type a. Let $\boldsymbol{Q}_{i,a} \leq_{\mathrm{rst}}^{B^i_{comp}} \boldsymbol{Q}'_{i,a}$, with contexts $\mathcal{C}_i \leq_{\mathrm{st}} \mathcal{C}'_i$, where B^i_{comp} is the comparative bound of $\mathcal{C}_i$ and $\mathcal{C}'_i$. If B^3_{comp} is the comparative bound of the contexts $\mathcal{C}_1 \cap \mathcal{C}_2$ and $\mathcal{C}'_1 \cap \mathcal{C}'_2$, we have $(r_{1,a}, \boldsymbol{P}_{1,a}) \circledast (r_{2,a}, \boldsymbol{P}_{2,a}) \leq_{\mathrm{rst}}^{B^3_{comp}} (r'_{1,a}, \boldsymbol{P}'_{1,a}) \circledast (r'_{2,a}, \boldsymbol{P}'_{2,a})$.*

Proof: For brevity, we will omit the subscript a, and hence write $\boldsymbol{P}_i$ in place of $\boldsymbol{P}_{i,a}$, and r_i in place of $r_{i,a}$ for $i \in \{1,2\}$. We omit the proofs that $\boldsymbol{P}_1 \otimes \boldsymbol{P}_2 \leq_{\mathrm{st}} \boldsymbol{P}'_1 \otimes \boldsymbol{P}'_2$ and that $\min\{r_1, r_2\}(s_1, s_2) \leq \min\{r'_1, r'_2\}(s_1, s_2)$ for all $(s_1, s_2) \in S_1 \times S_2$, since they are very similar to Lemma 27 and Lemma 28 from the proof of Theorem 24 (Montonicity).

Let $(S_1, \prec_1)$ and $(S_2, \prec_2)$ be the state spaces of components C_1, C'_1 and C_2, C'_2 respectively. We will show that the generator matrix $\min\{r'_1, r'_2\}(\boldsymbol{P}'_1 \otimes \boldsymbol{P}'_2 - \boldsymbol{I})$ is a context-bounded rate-wise upper bound of $\min\{r_1, r_2\}(\boldsymbol{P}_1 \otimes \boldsymbol{P}_2 - \boldsymbol{I})$, with respect to the ordering $(S_1 \times S_2, \prec_1^L)$.

We need to show that the following inequality holds, for all states (s_1, s_2):

$$\begin{aligned} &\max\left\{B^3_{comp}, \frac{\min\{r'_1(s_1), r'_2(s_2)\}}{\min\{r_1(s_2), r_2(s_2)\}}\right\} \\ \leq &\min_{(t_1,t_2) \prec_1^L (s_1,s_2)} \left\{\frac{1 - \displaystyle\sum_{(t'_1,t'_2) \succ_1^L (t_1,t_2)} \boldsymbol{P}_1(s_1, t'_1)\boldsymbol{P}_2(s_2, t'_2)}{1 - \displaystyle\sum_{(t'_1,t'_2) \succ_1^L (t_1,t_2)} \boldsymbol{P}'_1(s_1, t'_1)\boldsymbol{P}'_2(s_2, t'_2)}\right\} \end{aligned}$$

Let (t_1, t_2) be the state for which the ratio on the right hand side is at a minimum. Since $t_1 \prec_1 s_1$ by definition of $\prec_1^L$, we know that the following relation holds:

$$\max\left\{B^1_{comp}, \frac{r'_1(s_1)}{r_1(s_1)}\right\} \leq \frac{1 - \displaystyle\sum_{t'_1 \succ_1 t_1} \boldsymbol{P}_1(s_1, t'_1)}{1 - \displaystyle\sum_{t'_1 \succ_1 t_1} \boldsymbol{P}'_1(s_1, t'_1)}$$

Furthermore, since by definition of the comparative bounds, $B^1_{comp} \geq \frac{r'_2(s_2)}{r_2(s_2)}$, and $B^3_{comp} \leq \min\{B^1_{comp}, B^2_{comp}\}$, we can infer that:

$$\max\left\{B^3_{comp}, \frac{r'_1(s_1)}{r_1(s_1)}, \frac{r'_2(s_2)}{r_2(s_2)}\right\} \leq \max\left\{B^1_{comp}, \frac{r'_1(s_1)}{r_1(s_1)}\right\}$$

To complete the proof, we make use of Observation 29 from the proof of Theorem 24 (Montonicity), and the fact that t_2 must be the smallest state in $(S_2, \prec_2)$ by the definition of $\prec_1^L$:

$$\begin{aligned} &\max\left\{B^3_{comp}, \frac{\min\{r'_1(s_1), r'_2(s_2)\}}{\min\{r_1(s_1), r_2(s_2)\}}\right\} \\ \leq\ &\max\left\{B^3_{comp}, \frac{r'_1(s_1)}{r_1(s_1)}, \frac{r'_2(s_2)}{r_2(s_2)}\right\} \\ \leq\ &\max\left\{B^1_{comp}, \frac{r'_1(s_1)}{r_1(s_1)}\right\} \end{aligned}$$

$$\leq \frac{1 - \sum_{t_1' \succ_1 t_1} \boldsymbol{P}_1(s_1, t_1')}{1 - \sum_{t_1' \succ_1 t_1} \boldsymbol{P}_1'(s_1, t_1')}$$

$$= \frac{1 - \sum_{t_1' \succ_1 t_1} \boldsymbol{P}_1(s_1, t_1') \sum_{t_2' \in S_2} \boldsymbol{P}_2(s_2, t_2')}{1 - \sum_{t_1' \succ_1 t_1} \boldsymbol{P}_1'(s_1, t_1') \sum_{t_2' \in S_2} \boldsymbol{P}_2'(s_2, t_2')}$$

$$\leq \frac{1 - \sum_{(t_1', t_2') \succ_1^L (t_1, t_2)} \boldsymbol{P}_1(s_1, t_1') \boldsymbol{P}_2(s_2, t_2')}{1 - \sum_{(t_1', t_2') \succ_1^L (t_1, t_2)} \boldsymbol{P}_1'(s_1, t_1') \boldsymbol{P}_2'(s_2, t_2')}$$

$$= \min_{(t_1, t_2) \prec (s_1, s_2)} \left\{ \frac{1 - \sum_{(t_1', t_2') \succ_1^L (t_1, t_2)} \boldsymbol{P}_1(s_1, t_1') \boldsymbol{P}_2(s_2, t_2')}{1 - \sum_{(t_1', t_2') \succ_1^L (t_1, t_2)} \boldsymbol{P}_1'(s_1, t_1') \boldsymbol{P}_2'(s_2, t_2')} \right\}$$

The proof of stochastic ordering under $(S_1 \times S_2, \prec_2^L)$ follows similarly.

Thus the context-bounded rate-wise stochastic ordering is preserved by $\circledast$. ■