Separable equilibrium state probabilities via time reversal in Markovian process algebra

P.G. Harrison and T.T. Lee

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

Abstract

The Reversed Compound Agent Theorem (RCAT) is a compositional result that uses Markovian process algebra (MPA) to derive the reversed process of certain interactions between two continuous time Markov chains at equilibrium. From this reversed process, together with the given, forward process, the joint state probabilities can be expressed as a product-form, although no general algorithm has previously been given. This paper first generalizes RCAT to multiple (more than two) cooperating agents, which removes the need for multiple applications and inductive proofs in cooperations of an arbitrary number of processes. A new result shows a simple stochastic equivalence between cooperating, synchronised processes and corresponding parallel, asynchronous processes. This greatly simplifies the proof of the new, multi-agent theorem, which includes a statement of the desired product-form solution itself as a product of given state-probabilities in the parallel components. The reversed process and product-form thus derived rely on a solution to certain rate equations and it is shown, for the first time, that a unique solution exists under mild conditions—certainly for queueing networks and G-networks.

1 Introduction

Stochastic process algebra (SPA) is an extension of classical process algebra with time delays and probabilities, aimed at providing performance descriptions of concurrent systems. The inherent compositional structure separates the model of a system into successively more fundamental components and, through the interactions among the components, performance characteristics of complex systems can be assessed. This is one aid to the development of

Email addresses: pgh@doc.ic.ac.uk (P.G. Harrison), ttl01@doc.ic.ac.uk (T.T. Lee).

efficient computer and communication systems in that it can provide crucial performance analysis during the design phase. In the last decade or so, a number of SPA modelling formalisms have been developed, such as Timed Processes and Performance Evaluation (TIPP) [11], the first process algebra used for performance modelling, and Performance Evaluation Process Algebra (PEPA) [12], which are both Markovian. A generalised, more expressive language, Extended Markovian Process Algebra (EMPA) [2] incorporated immediate actions, chosen probabilistically through *weights*. PEPA is the simplest language, having the fewest combinators, and we use a subset of it for the present investigation, using only the *prefix*, *cooperation* and (implicitly) *choice* combinators.

The quest for so-called *product-form solutions* for the equilibrium state probabilities in stochastic networks has been a major research area in performance modelling for over 30 years, e.g. [1,13,14]. As the name implies, such a solution is expressed as a product of terms, each of which relates to only one of a collection of interacting component processes. Most attention has been given to queueing networks and their variants such as G-networks [5], but there have also been other significant examples. However, these have typically been derived in a rather ad-hoc way: guessing that such a solution exists, then verifying that the Kolmogorov equations of the defining Markov chain are satisfied and appealing to uniqueness.

The Reversed Compound Agent Theorem (RCAT) is a compositional result that uses Markovian process algebra (MPA) to derive the reversed process of certain cooperations between two continuous time Markov chains at equilibrium. From a reversed process, together with the given, forward process, the joint state probabilities follow as a product of ratios of rates in these two processes, yielding a product-form when one exists. RCAT thereby provides an alternative methodology, with syntactically checkable conditions, which unifies many product-forms, far beyond those for queueing networks. At the time, the original study of G-networks was considered a major departure from previous product-form analyses since the property of 'local balance' [1] did not hold and the traffic equations were non-linear. In contrast, the RCAT-based approach goes through unchanged—the only difference is that there are now cooperations between two types of departure transitions at different queues, as well as between departure transitions and arrival transitions, as in conventional queueing networks [9]. Prior to the advent of G-networks, many believed that partial balance was a necessary condition for a product-form, but G-networks are no different in the RCAT approach: negative customers satisfy the same conditions (with respect to different action types) as do positive ones.

This paper first generalizes RCAT to multiple (more than two) cooperating agents, which removes the need for multiple applications and inductive proofs in cooperations of an arbitrary number of processes; the generalized result is called MARCAT, for 'multi-agent RCAT'. A new result shows a simple stochastic equivalence between synchronised and corresponding parallel, asynchronous transitions in cooperating processes. This greatly simplifies the proof of the generalized (and original) RCAT and makes a product-form easy to extract as a product of given state-probabilities in the parallel components. In this way, the reversed process itself can be by-passed; it is only needed as some suitable dual process, which can be calculated directly and from which the equilibrium probabilities follow. The reversed process and product-form thus derived rely on a solution to certain rate equations—traffic equations in the case of a queueing network. To date, the existence of such a solution had not been proved but assumed as a condition of RCAT. Here, however, we show a unique solution exists under certain conditions which are quite mild and easy to check. Many existing product-form results, including all the variants of G-networks, each with its own customised proof of the existence of a solution to its non-linear traffic equations, are thereby subsumed.

The main contribution of the present paper is:

- generalisation (MARCAT) of RCAT to multiple cooperating agents;
- simpler proof method giving direct access to product-forms;
- proof of existence of a unique solution to the rate equations;

In the next section the essential background material on reversed, stationary Markov processes and MPA is reviewed; the basic definition of the MPA PEPA is given in the Appendix. This section also includes the new result referred to above relating certain parallel and synchronising processes. The main result extending RCAT to multiple agents is given in section 3, together with the associated existence result, and the ensuing simplified methodology is illustrated by a detailed consideration of an M-node queueing network in section 4. The paper concludes in section 5.

2 Foundations

The formalism chosen to present the results of this paper is a Markovian process algebra based on the well-known PEPA language, which is reviewed in the Appendix for self containedness. PEPA deals with *agents*, which are syntactic entities denoting *processes* and *states* in an underlying continuous time Markov chain (CTMC) that constitutes the semantic model. We will refer to both agents and processes in this paper, referring respectively to PEPA syntax, based on the actions of agents, and CTMC semantics, describing transitions between states in a process. As explained in the appendix, our algebra is a subset of σ-PEPA, i.e. parameterised PEPA extended to countably infinite state spaces. Its semantics is inherited from PEPA's.

As stated in the introduction, the goal is to find separable solutions for the equilibrium probabilities in a stationary Markov process by using its reversed process. A reversed process is actually just the process you get by looking 'backwards in time'. It matters not where you start looking from since the process is assumed stationary. However, we are not concerned with the semantics of reversed processes *per se*, we just want a process with the right properties for getting separable solutions efficiently. The primary such property is that the reversed Markov process of a stationary Markov process $\{X_t\}$ with state-space Ω, generator matrix Q and stationary probabilities $\boldsymbol{\pi}$ has generator matrix Q' defined by

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

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

This result is standard, see for example [14], and immediately yields a product-form solution for $\boldsymbol{\pi}$. This is because, in an irreducible Markov process, we may choose a reference state 0 arbitrarily, find a sequence of connected states, in either the forward or reversed process, $0, \ldots, j$ (i.e. with either $q_{i,i+1} > 0$ or $q'_{i,i+1} > 0$ for $0 \leq i \leq j-1$) for any state j and calculate

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

The fact that Q' is the generator matrix of what is actually the reversed process is somewhat irrelevant for our purposes. All we need is a straightforward way to calculate Q'—whatever it may represent. For a pair of synchronised stationary Markov processes, the original RCAT and its generalisation [10] provides this. It states that, under certain conditions, the reversed agent of a cooperation $P \underset{L}{\bowtie} Q$ between two agents P and Q is a cooperation between the reversed agents of P and Q, after some re-parameterisation [7].

2.1 State transition paths

Once a reversed process is known, a solution for a stationary Markov process's equilibrium probabilities follows as a product of ratios of forward and reversed rates when an appropriate path has been found from a chosen reference state to the state in question. The problem remains how to determine a suitable reference state and appropriate paths to other states. Of course, in many cases the choice is obvious, for example birth-death processes like queues, where there is a well-defined ordering of states with transitions between adjacent states.

In the case of a cooperation between two agents, representing two Markov processes, we assume inductively that a reference state, 0 say, and paths to the other states are known in each component process considered separately. In the case of a network of two queues, the joint reference state would typically be $(0, 0)$ but we cannot be sure about the paths. In the special case that the cooperation set is empty, so that the two components evolve independently, the paths are obvious—to go to state (i, j), just follow the path from 0 to i in the first component process with the second component in state 0, and then follow the path from 0 to j in the second component with the first in state i. In other words, follow a *rectilinear path* in one dimension at a time. In this case, the joint equilibrium probability for state (i, j) will very simply be the product of the equilibrium probabilities of the states i, j in each component process. However, this approach becomes more complicated when the transitions encountered on the chosen paths participate in the synchronisation—it may well be that there are no paths with solely non-synchronising transitions, or even no such rectilinear path at all, as in a two-node cyclic queueing network for example.

In general, a component's action that is involved in the cooperation with another component is *split* into two sub-actions, one of which synchronises (with a (sub-)action in the other component) the other proceeding independently (and concurrently). For example, a service completion at a queue can cause either an external departure or the transfer of a customer to the other queue. This notion of splitting is made formal in the following:

Definition 1 *An action $\alpha = (a, \lambda)$ in agent P is said to be* split *into sub-actions $\alpha_1 = (a_1, \lambda_1)$ and $\alpha_2 = (a_2, \lambda_2)$ if P is semantically equivalent (w.r.t. the underlying Markov chains) to $P\{\alpha.E \leftarrow \alpha_1.E + \alpha_2.E \mid E \sqsubset P\}$. Conversely, the action α is called the* complete *action of the sub-actions α_1, α_2.*

Notice that α_1 and α_2 become parallel actions in P, with the same combined semantics as a single action, also parallel, with rate $\lambda_1+\lambda_2$. Hence the definition requires $\lambda_1 + \lambda_2 = \lambda$.

In an application of RCAT, each sub-action (even if passive, i.e. having unspecified rate) has a well defined rate and we need to decide how to apportion the rate of their reversed *complete* action between the reversed sub-actions. The reversed rate of the complete action is given solely by the single component in which it occurs. In general a cooperating action can be split into more than two sub-actions, corresponding to multiple synchronisations—i.e. one sub-action for each potential synchronisation partner, with possibly a concurrent non-synchronising sub-action. Definition 1 is generalised inductively in the obvious way to $n \geq 2$ sub-actions.

We define the reversed rate of the complete action to be distributed amongst the reversed sub-actions in proportion to their forward transition rates. This definition is not arbitrary but the one that ensures that the process so defined is the unique reversed process, a property that was not observed in [7], incidentally. We state it as a proposition.

Proposition 1 *The reversed sub-actions of multiple sub-actions* (a_i, λ_i) *for* $1 \leq i \leq n$ *in an agent* P *are respectively*

$$\left(\overline{a_i}, (\lambda_i/\lambda)\overline{\lambda}\right)$$

where $\lambda = \lambda_1 + \ldots + \lambda_n$ *and* $\overline{\lambda}$ *is the reversed rate*[1] *of the complete action* (a, λ) *denoting the corresponding transition with rate* λ *in the underlying Markov process of* P*.*

Proof

We use *Kolmogorov's (generalised) criteria* [7], which states that X and Y are mutually reversed processes if and only if (a) the sum of the outgoing rates from every state (reciprocal of the mean state holding time) is the same in both X and Y and (b) the ratio of the products of the transition rates around every cycle in X and of those in the corresponding reversed cycle in Y is unity.

The first of Kolmogorov's criteria holds trivially since it considers only total rates out of each state. Regarding the second, the definition of the rate of a reversed sub-action ensures that the ratio of a sub-action's forward and reversed rates is the same as the ratio of its complete action's forward and reversed rates. Hence the second condition must also be satisfied under the hypothesis that the reversed process $\overline{P}$ of the process P with unsplit actions (only the aggregate, complete actions) was known. ♠

Returning now to the choice of paths in a cooperation, if every cooperation involves only a sub-action in each component, with another sub-action of each respective complete action not participating, there will always be a rectilinear path to every state from the reference state. Then a separable solution can be found exactly as in the case of parallel independent agents described above, with the reparameterisation of the components given by RCAT. Note that parallel agents are a special case with a null reparameterisation.

[1] Reversed entities, e.g. reversed rates, are denoted by an overbar—see the Appendix

We can guarantee that paths do exist which are identical to paths in the isolated component processes by augmenting active synchronising actions with *residual actions* or *ϵ-actions*. These are parallel to (concurrent with) the synchronising actions but do not participate in the cooperation.

Definition 2 *Suppose (a, λ) is an action in some agent P. The agent $P^{a+\epsilon}$ is obtained by splitting the action (a, λ) into sub-actions with rates $(1-\epsilon)\lambda$ and $\epsilon\lambda$ and with respective types a and a^ϵ. That is*

$$P^{a+\epsilon} = P\{(a, \lambda).E \leftarrow (a, (1-\epsilon)\lambda).E + (a^\epsilon, \epsilon\lambda).E \mid E \sqsubset P\}$$

for some real number $\epsilon, 0 < \epsilon < 1$, where the action type a^ϵ does not occur in P. The residual action *$(a^\epsilon, \epsilon\lambda)$ is called an* ϵ-action.

The agent $P^{a+\epsilon}$ denotes the Markov process with the same generator matrix as that of the Markov process underlying P, but with every element denoted by the action type a being interpreted as a sum of the quantities $(1-\epsilon)\lambda$ (for the original action with type a) and $\epsilon\lambda$ (for the ϵ-action). That is, the Markov process underlying $P^{a+\epsilon}$ has the same transitions as for P except that the rate of a is reduced by a factor of $1-\epsilon$ and there are additional transitions of rate $\epsilon\lambda$ parallel to (with the same source and destination states) all those denoted by action type a. Clearly, $\lim_{\epsilon\to 0} P^{a+\epsilon} = P$. We cannot assume anything about ergodicity and its preservation in this limit, but this is not an issue here since all processes are assumed stationary. Ergodicity conditions require a separate analysis. Similarly, even when ergodicity is preserved, the equilibrium probabilities, $\boldsymbol{\pi}^\epsilon$ say, may not be continuous, i.e. it may not be the case that $\lim_{\epsilon\to 0} \boldsymbol{\pi}^\epsilon = \boldsymbol{\pi}$. We will see that this is true, however, for cooperations satisfying RCAT and its extensions.

Notice that a reversed residual action $\overline{(a^\epsilon, \epsilon\lambda)} = (\overline{a^\epsilon}, \epsilon\overline{\lambda})$ by proposition 1, i.e. its rate is the product of ϵ and the reversed rate of the complete (unsplit) action a.

It is not meaningful to split an action with unspecified rate.[2] Therefore, in a cooperation of agents with ϵ-actions, we must split an action α into residual and cooperating parts *before* making its cooperating part passive (only in the component in which α is passive, of course). For brevity, we denote an agent P, in which an action type a is made passive, by $P(a, \top) \triangleq P\{\alpha_i \leftarrow (a, \top) \mid \alpha_i \in \mathcal{I}(a)\}$, where $\mathcal{I}(a)$ is the set of all instances of an action with

[2] Unless weights are specified for each sub-action, as in PEPA [12]. This is essentially the same as pre-assigning rates and subsequently making actions passive, as below in Lemma 1.

type a in P, i.e. of the form $\alpha_i = (a, \lambda_i)$, where the rate λ_i may differ at different instances $\alpha_i \in \mathcal{I}(a)$. This notation is extended in the obvious way to a set of action types $a \in S$, say, which each become passive in the agent $P(S, \top) \equiv P\{(a, \top) \mid a \in S\} \overset{\Delta}{=} P\{(a, \lambda_i) \leftarrow (a, \top) \mid (a, \lambda_i) \in \mathcal{I}(a), a \in S\}$.

We can now write $P^{a+\epsilon}(a, \top)$ to define a modified agent P with passive action type a split to introduce a parallel residual action with rate $\epsilon\lambda$, which does not synchronise, where λ is the rate of a in P (possibly different at each instance).

We have the following simple but important property for certain cooperations with residual actions.

Lemma 1 *Consider agents R, S with no passive actions and let action type a in R have rate λ_a, action type a in S have rate μ_a. Let r_a and $\overline{r_a}$ be the rates of a and $\overline{a}$ at a particular instance of a in the cooperations*

$$R^{a+\epsilon} \underset{L}{\bowtie} S^{a+\epsilon}(a, \top) \quad \textit{and} \quad \overline{R^{a+\epsilon}}(\overline{a}, \top) \underset{\overline{L}}{\bowtie} \overline{S^{a+\epsilon}}$$

respectively, where $a \in L$. Then

$$\frac{r_a}{\overline{r_a}} = \frac{\lambda_a \mu_a}{\overline{\lambda_a \mu_a}} \quad \textit{if and only if} \quad \mu_a = \overline{\lambda_a}.$$

Proof
By definition of the cooperation combinator and the splitting of the action with type a, $r_a = (1 - \epsilon)\lambda_a$ and, by proposition 1, $\overline{r_a} = (1 - \epsilon)\overline{\mu_a}$ and so the result follows. ♠

This lemma means that paths including a cooperating action are equivalent to paths that do not, in the sense of equilibrium state probabilities as follows.

Remarks

(1) As written in the lemma, there could be synchronisations between action types $b \in L$ which are active in both components R and S. Their semantics is just that of PEPA, but in practice we would not be applying RCAT to these comnponents since its conditions would not be satisfied;
(2) The splitting of the action type a does not lead to the four possible synchronisations that would be obtained in PEPA. This is because when we split an action to introduce a residual action we rename the residual action's type. Hence there remains only one instance of type a for each of its original instances and so there is only the one synchronisation possibility corresponding to each original synchronisation.

Lemma 2 *In the notation of the previous lemma, let r_R^ϵ, r_S^ϵ be the rates of the residual action type a^ϵ in R, S respectively and let $\overline{r_R^\epsilon}$, $\overline{r_S^\epsilon}$ be the respective reversed rates of $\overline{a^\epsilon}$. Then, at a particular instance of a,*

$$\frac{r_a}{\overline{r_a}} = \frac{r_R^\epsilon\, r_S^\epsilon}{\overline{r_R^\epsilon}\,\overline{r_S^\epsilon}} \quad \textit{if and only if} \quad \mu_a = \overline{\lambda_a}.$$

Proof
$r_R^\epsilon = \epsilon\lambda_a$ and $\overline{r_R^\epsilon} = \epsilon\overline{\lambda_a}$ by proposition 1. Similarly, $r_S^\epsilon = \epsilon\mu_a$ and $\overline{r_S^\epsilon} = \epsilon\overline{\mu_a}$ so that the ϵ factors cancel in the ratio and the result follows by lemma 1. ♠

Suppose, then, that a cooperating action type a denotes a transition between states (i, j) and (i', j') in $R \underset{L}{\bowtie} S$; i.e. it also denotes transitions $i \to i'$ in R and $j \to j'$ in S. Then, the paths $(i, j) \to (i', j) \to (i', j')$, $(i, j) \to (i, j') \to (i', j')$ (via residual transitions) and $(i, j) \to (i', j')$ (via the synchronised transition) are equivalent in the sense that the products of the ratios of the forward and reversed rates of each transition in each path are equal. This is a necessary condition for $\overline{R^{a+\epsilon}}(\overline{a}, \top) \underset{L}{\bowtie} \overline{S^{a+\epsilon}}$ to be the reversed process of $R^{a+\epsilon} \underset{L}{\bowtie} S^{a+\epsilon}(a, \top)$, a property that will be used to prove the multiple agent RCAT in the next section. Moreover, it will also be used to find simple separable solutions directly for the equilibrium state probabilities of cooperations satisfying that theorem.

3 Multiple agent cooperations

In the PEPA cooperation $P \underset{L}{\bowtie} Q$, the subset of action types in a cooperation set L which are *passive* (i.e. have unspecified rate $\top$) with respect to an agent P is denoted by $\mathcal{P}_P$ and the corresponding subset of *active* action types is denoted by $\mathcal{A}_P$; similarly for agent Q. For the purposes of RCAT, it is also assumed that, in $P \underset{L}{\bowtie} Q$:

- any active action in P has a corresponding passive action in Q, and vice versa;
- every action with type in L must be either passive at every instance in P or active at every instance in P (and similarly for Q).

Therefore, $\mathcal{P}_P = \mathcal{A}_Q$, $\mathcal{A}_P = \mathcal{P}_Q$ and $\mathcal{P}_P \cap \mathcal{A}_P = \mathcal{P}_Q \cap \mathcal{A}_Q = \emptyset$. This defines the subset of PEPA cooperations that we are considering in this paper. Moreover, in RCAT and MARCAT we consider only irreducible state transition graphs in the underlying Markov chains of cooperations. Hence all states are reachable from each other and questions of redundant or blocking agents and action types do not arise. In any analysis of a Markov chain at equilibrium, irreducibility is usually a prerequisite that is established separately. This is often relatively straightforward, as in single-class queueing networks, which require only that the task-routing matrix (of dimension equal to the number of queues, not states) be irreducible, i.e. that every queue is reachable from every other by any task. However, in multi-class networks, this may not apply to every class and so sub-chains have to be identified prior to applying a product-form solution for the equilibrium state probabilities. In the worst case, state space exploration must be undertaken, but this is much cheaper than direct solution of the underlying Markov chain. Indeed, it can be combined with computation of the normalising constant of a closed network, which is also not in the scope of the present paper.

In an extension of PEPA, consider now a multiple-agent, pairwise cooperation $\underset{L}{\overset{n}{\underset{k=1}{\bowtie}}} P_k$ $(n \geq 2)$, where $L = \bigcup_{k=1}^{n} L_k$ and $L_k = \mathcal{P}_k \cup \mathcal{A}_k$ is the set of synchronising action types that occur in agent P_k (abbreviating $\mathcal{P}_{P_k}$ by $\mathcal{P}_k$ and $\mathcal{A}_{P_k}$ by $\mathcal{A}_k$). Every action in each of the n agents cooperates with (at most) one other and so $\mathcal{P}_k \subset \bigcup_{\substack{j=1\\ j\neq k}}^{n} \mathcal{A}_j$ and $\mathcal{A}_k \subset \bigcup_{\substack{j=1\\ j\neq k}}^{n} \mathcal{P}_j$. We provide the semantics of multi-agent cooperation by defining it in terms of PEPA's cooperation combinator:

$$\underset{L}{\overset{n}{\underset{k=1}{\bowtie}}} P_k = (\ldots((P_1 \underset{M_2}{\bowtie} P_2) \underset{M_3}{\bowtie} P_3) \underset{M_4}{\bowtie} \ldots \underset{M_{n-1}}{\bowtie} P_{n-1}) \underset{M_n}{\bowtie} P_n$$

where $M_k = L_k \cap \left(\bigcup_{j=1}^{k-1} L_j \right)$. Note the subtle change in the bowtie symbol used for multi-agent cooperations.

3.1 Notation

We now define the following notation, generalising that of [10]:

$\mathcal{P}_k^{i\rightarrow}$ denotes the set of action types in L_k that are *passive* in P_k and correspond to transitions *out of* state i in the Markov process of P_k;
$\mathcal{P}_k^{i\leftarrow}$ denotes the set of action types in L_k that are *passive* in P_k and correspond to transitions *into* state i in the Markov process of P_k;
$\mathcal{A}_k^{i\rightarrow}$ denotes the set of action types in L_k that are *active* in P_k and correspond to transitions *out of* state i in the Markov process of P_k;
$\mathcal{A}_k^{i\leftarrow}$ denotes the set of action types in L_k that are *active* in P_k and correspond to transitions *into* state i in the Markov process of P_k;
$\mathcal{P}^{\underline{i}\rightarrow}$ denotes the set of action types in $L = \bigcup_{k=1}^{n} L_k$ that are *passive* and correspond to transitions *out of* state $\underline{i} = (i_1, i_2, \ldots, i_n)$ in the Markov process of $\underset{L}{\overset{n}{\underset{k=1}{\bowtie}}} P_k$;

$\mathcal{P}^{\underline{i}\leftarrow}$ denotes the set of action types in L that are *passive* and correspond to transitions *into* state $\underline{i}$ in the Markov process of $\underset{k=1}{\overset{n}{\underset{L}{\bowtie}}} P_k$;

$\mathcal{A}^{\underline{i}\rightarrow}$ denotes the set of action types in L that are *active* and correspond to transitions *out of* state $\underline{i}$ in the Markov process of $\underset{k=1}{\overset{n}{\underset{L}{\bowtie}}} P_k$;

$\mathcal{A}^{\underline{i}\leftarrow}$ denotes the set of action types in L that are *active* and correspond to transitions *into* state $\underline{i}$ in the Markov process of $\underset{k=1}{\overset{n}{\underset{L}{\bowtie}}} P_k$;

$\alpha_a^{\underline{i}}$ denotes the instantaneous transition rate *out of* state $\underline{i}$ in the Markov process of $\underset{k=1}{\overset{n}{\underset{L}{\bowtie}}} P_k$ corresponding to *active* action type $a \in L$;

$\top_a$ denotes the unspecified rate associated with the action type a in the action $(a, \top_a)$;

$\boldsymbol{x}$ denotes the vector $(x_{a_1}, \ldots, x_{a_m})$ of positive real variables x_{a_i} when $L = \{a_1, \ldots, a_m\}$;

$\overline{\beta_a^{\underline{i}}}(\boldsymbol{x})$ denotes the instantaneous transition rate *out of* state $\underline{i}$ in the reversed Markov process of $\underset{k=1}{\overset{n}{\underset{L}{\bowtie}}} P_k\{\top_a \leftarrow x_a \mid a \in L\}$ corresponding to *passive* action type $a \in L$; note that a is *incoming* to state $\underline{i}$ in the forwards process. We also write $\overline{\beta_{k;a}^{i_k}}(\boldsymbol{x}) \equiv \overline{\beta_a^{\underline{i}}}(\boldsymbol{x})$ where P_k is the component in which a is passive (incoming to state i_k).

3.2 Reversed compound multi-agent theorem

We are now in a position to state and prove an extended RCAT, generalised to multiple agents, which also leads (in section 3.5) to a direct product-form solution for the equilibrium state probabilities of the underlying Markov process.

Theorem 1 *(MARCAT)*
Suppose that the cooperation $\underset{k=1}{\overset{n}{\underset{L}{\bowtie}}} P_k$ of agents P_k, denoting stationary Markov processes, has a derivation graph[3] *with an irreducible subgraph G. Given that every instance of a reversed action, type $\overline{a}$, of an* active *action type $a \in \mathcal{A}_k$ has the same rate $\overline{p_a}$ in $\overline{P_k}$ $(1 \le k \le n)$, the reversed agent $\overline{\underset{k=1}{\overset{n}{\underset{L}{\bowtie}}} P_k}$, with derivation graph containing the reversed subgraph $\overline{G}$, is*

$$\underset{k=1}{\overset{n}{\underset{\overline{L}}{\bowtie}}} \overline{R_k}\{(\overline{a}, \top) \mid a \in \mathcal{A}_k\}$$

where

$$R_k = P_k\{\top_a \leftarrow x_a \mid a \in \mathcal{P}_k\} \qquad k = 1, \ldots, n$$

[3] As defined in PEPA's semantics, see the Appendix.

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

$$\{\top_a = \overline{p_a} \mid a \in \mathcal{A}_k, 1 \leq k \leq n\} \tag{1}$$

and $\overline{p_a}$ *is the symbolic rate of action type* $\overline{a}$ *in* $\overline{P_k}$*, provided that*

$$\sum_{a\in\mathcal{P}^{\underline{i}\rightarrow}} x_a - \sum_{a\in\mathcal{A}^{\underline{i}\leftarrow}} x_a = \sum_{a\in\mathcal{P}^{\underline{i}\leftarrow}\backslash\mathcal{A}^{\underline{i}\leftarrow}} \overline{\beta_a^{\underline{i}}(\boldsymbol{x})} - \sum_{a\in\mathcal{A}^{\underline{i}\rightarrow}\backslash\mathcal{P}^{\underline{i}\rightarrow}} \alpha_a^{\underline{i}} \tag{2}$$

Proof
The existence of solutions $\{x_b\}$ to the rate equations 1 is established by theorem 2 in section 3.4.

Let the instantaneous transition rate in the Markov chain of P_k (respectively R_k) out of state i corresponding to action type a be $p_{k;ia}$ (respectively $r_{k;ia}$) and let $p_{k;i} = \sum_{a\in\mathcal{O}_{k;i}} p_{k;ia}$ (similarly for r_{ki}), where $\mathcal{O}_{k;i}$ is the set of all outgoing action types in (the derivative of P_k corresponding to) state i. In other words, $p_{k;i}$ is the total outgoing rate from state i in the Markov chain of P_k.

In $\underset{L}{\overset{n}{\underset{k=1}{\bowtie}}} P_k$, the total rate out of any state $\underline{i} = (i_1, i_2, \ldots, i_n) \in G$ is

$$\sum_{k=1}^{n} p_{k;i_k}\{\top \leftarrow 0\} - \sum_{k=1}^{n} \sum_{a\in\mathcal{A}_k^{i_k\rightarrow}\backslash\mathcal{P}^{\underline{i}\rightarrow}} p_{k;i_k a}$$

where $\{\top \leftarrow 0\}$ denotes setting every occurrence of an unspecified rate corresponding to action types in L_k to zero, which is an abbreviation for $\{\top_a \leftarrow 0 \,|\, a \in L_k\}$. Note that $\mathcal{A}_k^{i_k\rightarrow}\backslash\mathcal{P}^{\underline{i}\rightarrow}$ is the set of active actions in P_k that do not have passive actions to synchronize with in state $\underline{i}$; these disabled active actions do not contribute to the total rate out of state $\underline{i}$. Since $\mathcal{A}_1^{i_1\rightarrow}, \mathcal{A}_2^{i_2\rightarrow}, \ldots,$ and $\mathcal{A}_n^{i_n\rightarrow}$ are disjoint, the total rate out of state $\underline{i}$ in $\underset{L}{\overset{n}{\underset{k=1}{\bowtie}}} P_k$ can be simplified to:

$$\sum_{k=1}^{n} p_{k;i_k}\{\top \leftarrow 0\} - \sum_{a\in\mathcal{A}^{\underline{i}\rightarrow}\backslash\mathcal{P}^{\underline{i}\rightarrow}} \alpha_a^{\underline{i}} \tag{3}$$

Now, consider the total outgoing rates in the reversed agent $\overline{\underset{L}{\overset{n}{\underset{k=1}{\bowtie}}} P_k}$.

$$r_{ki_k} = p_{k;i_k}\{\top_a \leftarrow x_a \mid a \in \mathcal{P}_k\} \qquad k = 1, \ldots, n$$

are the total rates out of state i_k in R_k with $k = 1, \ldots, n$. Thus,

$$r_{ki_k} = p_{k;i_k}\{\top \leftarrow 0\} + \sum_{a\in\mathcal{P}_k^{i_k\rightarrow}} x_a \qquad k = 1, \ldots, n$$

In the reversed cooperation, the rates of reversed components' outgoing actions which are made passive must be subtracted from the sum of all the outgoing rates in the agents $\overline{R_k}$. Such an action $(\overline{a}, \overline{\lambda_a})$ is the reverse of an incoming active action (a, λ_a) and so its rate is $\overline{\lambda_a} = x_a$. Hence, by the first of Kolmogorov's criteria [7], the total rate out of state $\underline{i}$ in

$$\underset{\underset{L}{k=1}}{\overset{n}{\bowtie}} \overline{R_k}\{(\overline{a}, \top) \mid a \in \mathcal{A}_k\}$$

is (remembering to also subtract out the disabled, reversed, outgoing, active actions' rates)

$$\sum_{k=1}^{n} \left(p_{k;i_k}\{\top \leftarrow 0\} + \sum_{a \in \mathcal{P}_k^{i_k \rightarrow}} x_a - \sum_{a \in \mathcal{A}_k^{i_k \leftarrow}} x_a - \sum_{a \in \mathcal{P}_k^{i_k \leftarrow} \setminus \mathcal{A}^{\underline{i} \leftarrow}} \overline{p_{k;i_k a}} \right)$$

which can be simplified to:

$$\sum_{k=1}^{n} p_{k;i_k}\{\top \leftarrow 0\} + \sum_{a \in \mathcal{P}^{\underline{i} \rightarrow}} x_a - \sum_{a \in \mathcal{A}^{\underline{i} \leftarrow}} x_a - \sum_{a \in \mathcal{P}^{\underline{i} \leftarrow} \setminus \mathcal{A}^{\underline{i} \leftarrow}} \overline{\beta_a^{\underline{i}}}(\boldsymbol{x})$$

Equating this with expression 3, to satisfy the first of Kolmogorov's criteria in the cooperation, gives the last condition of the theorem.

To complete the proof by Kolmogorov's criteria, we need to verify that the product of transition rates in any forward cycle in $\underset{\underset{L}{k=1}}{\overset{n}{\bowtie}} P_k$ is equal to that in its reversed cycle in $\underset{\underset{L}{k=1}}{\overset{n}{\bowtie}} \overline{R_k}\{(\overline{a}, \top) \mid a \in \mathcal{A}_k\}$. But $\underset{\underset{L}{k=1}}{\overset{n}{\bowtie}} P_k \equiv \underset{\underset{L}{k=1}}{\overset{n}{\bowtie}} R_k\{(a, \top) \mid a \in \mathcal{P}_k\}$. By hypothesis, all synchronisations are pairwise and the conditions of lemmas 1 and 2 are satisfied when we introduce ϵ-transitions on all synchronising actions $a \in L$; in the notation of these lemmas, $\mu_a = \overline{\lambda_a} = x_a$. Consequently, the products of the ratios of forward rates to reversed rates in any cycle are invariant when we replace any synchronising action by its two residual actions. Hence we only need to consider the rectilinear cycles. But, since the transition rates in agent P_k are independent of the states of all agents P_j for $j \neq k, 1 \leq k \leq n$, any rectilinear cycle is simply a union of cycles within single component processes, and these satisfy the second of Kolmogorov's criteria by hypothesis. ♠

3.3 Conditions of MARCAT and RCAT

In its most general form, equation 2 of theorem 1 is a complex condition to check—prohibitively so in large (and especially infinite) state-spaces. This is because it must hold in every joint state of the cooperation. However, in important special cases checking is straightforward or trivial—for example when the passive actions are 'invisible', leading from a state to itself, whereupon the term $\overline{\beta_a^{\underline{i}}(\boldsymbol{x})} = x_a$ [10]. More generally, it is typical for equation 2 to hold for all values of the rates $\boldsymbol{x}$; we will require this property when we consider the existence of solutions to the rate equations in section 3.4.

Most importantly, in applications of the original RCAT of [7], extended to multiple cooperations, all passive actions are outgoing from every joint state of the cooperation and all active actions are incoming to every joint state. The condition is then satisfied trivially since the terms on the left hand side cancel and those on the right are both empty sums. In particular, MARCAT applies to all of the standard queueing networks, including G-networks and their extensions.

A sufficient set of n conditions to replace equation 2 is, for each k, $1 \leq k \leq n$:

$$\sum_{a \in L_k \cap \mathcal{P}^{\underline{i}\rightarrow}} x_a - \sum_{a \in L_k \cap \mathcal{A}^{\underline{i}\leftarrow}} x_a = \sum_{a \in L_k \cap (\mathcal{P}^{\underline{i}\leftarrow} \setminus \mathcal{A}^{\underline{i}\leftarrow})} \overline{\beta_a^{\underline{i}}(\boldsymbol{x})} - \sum_{a \in L_k \cap (\mathcal{A}^{\underline{i}\rightarrow} \setminus \mathcal{P}^{\underline{i}\rightarrow})} \alpha_a^{\underline{i}} \tag{4}$$

for all valid joint states $(i_1, \ldots, i_n)$ in the irreducible chain G of theorem 1. Summing equation 4 over k yields precisely double equation 2 since synchronisations are pairwise: every action type appears in exactly two of the sets L_k $(1 \leq k \leq n)$. In simpler form, this equation can be written:

$$\begin{aligned} \sum_{a \in \mathcal{P}_k^{i_k\rightarrow}} x_a + \sum_{a \in \mathcal{P}_{k^a}^{i_k\rightarrow}} x_a - \sum_{a \in \mathcal{A}_k^{i_k\leftarrow}} x_a - \sum_{a \in \mathcal{A}_{k^a}^{i_k\leftarrow}} x_a = \\ \sum_{a \in \mathcal{P}_k^{i_k\leftarrow} \setminus \mathcal{A}_{k^a}^{i_k\leftarrow}} \overline{\beta_{k;a}^{i_k}(\boldsymbol{x})} + \sum_{a \in \mathcal{P}_{k^a}^{i_k\leftarrow} \setminus \mathcal{A}_k^{i_k\leftarrow}} \overline{\beta_{k;a}^{i_k}(\boldsymbol{x})} \\ - \sum_{a \in \mathcal{A}_k^{i_k\rightarrow} \setminus \mathcal{P}_{k^a}^{i_k\rightarrow}} \alpha_a^{\underline{i}} - \sum_{a \in \mathcal{A}_{k^a}^{i_k\rightarrow} \setminus \mathcal{P}_k^{i_k\rightarrow}} \alpha_a^{\underline{i}} \end{aligned} \tag{5}$$

where P_{k^a} is the component agent that synchronises with the action a in P_k. Thus, even the worst case requires only checking actions componentwise—the number of checks is of the order of the product of the numbers of local states in each component process, not combinatorial in these numbers. Moreover, the states of a component process will usually be parameterised in the process algebraic specification, e.g. a queue of positive length corresponds to the PEPA agent P_{n+1} and the empty queue to P_0, giving just two parameterised states. The actual number of checks required is the product of the numbers of *parameterised* states.

It remains to establish that the rate equations for the variables x_a do indeed have a solution which is unique. This is proved in theorem 2 below. First we define some more notation , for $1 \leq k \leq n$:

- $X_S = \sum\limits_{a \in S} x_a$ for $S \in 2^L$;
- $\{\top \leftarrow \boldsymbol{y}\}$ is an abbreviation for $\{\top_b \leftarrow y_b \mid b \in L\}$
- $\gamma_{k;i}$ is the total outgoing rate from state i in P_k contributed by non-passive actions, i.e. a constant, independent of rates assigned to passive actions;
- when it exists, $\pi_k(j)$ is the (unnormalised) marginal equilibrium probability of state j in process P_k, a function of the unspecified rates $\top_a$ (regarded symbolically as variable names) for $a \in \mathcal{P}_k$;
- $c : s_c(k) \to d_c(k)$ is an action with type c in component P_k that denotes a transition from a source state $s_c(k)$ to a destination state $d_c(k)$, the argument (k) being omitted where there is no ambiguity;
- an active action type $a : i' \to i$ which is part of a split action in some component is selected with constant probability $f_a \leq 1$, where $f_a = 1$ when the active action is not split;
- an active action type $o \in \mathcal{A}_k$ in component P_k is *strong* if

$$\liminf_{\boldsymbol{y} > \mathbf{0}} \left(\overline{p_o}\{\top \leftarrow \boldsymbol{y}\} / \overline{p_a}\{\top \leftarrow \boldsymbol{y}\}\right) > 0$$

  for all $a \in \mathcal{A}_k^{h\leftarrow}$ where h is a destination state of o ;
- a strong active action o which is part of a split action is said to be *strongly split* ;
- $p_{k;i'a^+} = p_{k;i'a}/f_a$ is the rate of the whole action that splits into $a : i' \to i$ and (possibly) other sub-action(s);
- $\overline{p_{a^+}} = \overline{p_{k;i'a^+}} = \overline{p_{k;i'a}}/f_a = \overline{p_a}/f_a$, which is well defined by proposition 1 and the hypothesis in theorem 1 that reversed rates of active actions are constant;
- $g_{ki}(\boldsymbol{y}) = \dfrac{\sum\limits_{a \in \mathcal{A}_k^{i\leftarrow}} \overline{p_a\{\top \leftarrow \boldsymbol{y}\}}}{\sum\limits_{a \in \mathcal{A}_k^{i\leftarrow}} \overline{p_{a^+}\{\top \leftarrow \boldsymbol{y}\}}} \leq 1$. Thus, $g_{ki} < u$, for some constant $u < 1$, when $\exists o \in \mathcal{A}_k^{i\leftarrow}$ which is strongly split;
- $\overline{p_{k;i\leftarrow}}(\boldsymbol{y}) = \sum_{a:i' \to i} \overline{p_{k;i'a}\{\top \leftarrow \boldsymbol{y}\}}$ denotes the total reversed rate into state i in P_k (out of state i in $\overline{P_k}$), with the given renaming.

An active action a that is part of a split action (by a slight abuse of terminology we also say a is a *split active action*) has a positive proportion of the total rate of the split action, f_a. In general this cannot be said of the reversed rate, but it is true for strongly split active actions. In general it may not be easy to prove an action is strong since its reversed rate may depend on the variables x_b. However, in two situations that are commonplace the proof is trivial:

- when the active incoming actions $a : i' \to i$ in state i of component process P_k all have the same source state i', then the reversed rate of an active action $o : i' \to i$ with (forward) rate λ_o is the positive fraction $\lambda_o / \sum_{a \in \mathcal{A}_k^{i\leftarrow}} \lambda_a$ of the total reversed rate of all actions $a \in \mathcal{A}_k^{i\leftarrow}$ by proposition 1, where λ_a is the forward rate of action a. This is the situation in all G-networks where the active actions are all split departures, except for active resets of the kind introduced in [9];
- an active incoming action $o : i' \to i$ is not strong in a stationary process P_k if and only if $\liminf\limits_{\boldsymbol{y}>\mathbf{0}} \pi_k(i')/\pi_k(i) > 0$, by the fundamental result given in section 2. This limit is usually known by construction and may be quite simple to calculate. In the case of the active resets of [9] the appropriate ratio $\pi_k(i')/\pi_k(i)$ is a positive constant.

The strongness property essentially ensures that no incoming reversed active action vanishes, in the sense that its rate approaches zero asymptotically for some assignments of values y_b to the passive rates. Strongness is a technical condition required in the following theorem, which proves that MARCAT's rate equations have a solution under quite mild conditions. Such a solution is unique when it can be normalised, by the uniqueness of the equilibrium probabilities, when they exist, of Markov chains. As one class of examples, it is easy to verify that most queueing networks, certainly all variants of G-networks and BCMP networks, have solutions since they possess the strongness property.

Theorem 2 *In the cooperation* $\underset{k=1}{\overset{n}{\bowtie}}{}_{L}\, P_k$ *of stationary Markov processes* P_k*, defined in theorem 1, the equations for* x_a*,* $a \in \mathcal{A}_k, 1 \le k \le n$*,*

$$x_a = \overline{p_a}\{\top_b \leftarrow x_b \mid b \in L\}$$

where $\overline{p_a}$ *is the reversed rate of* a*, have a unique solution if for each* k*,* $1 \le k \le n$ *:*

$$\sum_{a \in L_k \cap \mathcal{P}^{\underline{i}\rightarrow}} x_a - \sum_{a \in L_k \cap \mathcal{A}^{\underline{i}\leftarrow}} x_a = \sum_{a \in L_k \cap (\mathcal{P}^{\underline{i}\leftarrow} \setminus \mathcal{A}^{\underline{i}\leftarrow})} \overline{\beta_a^{\underline{i}}(\boldsymbol{y})} - \sum_{a \in L_k \cap (\mathcal{A}^{\underline{i}\rightarrow} \setminus \mathcal{P}^{\underline{i}\rightarrow})} \alpha_a^{\underline{i}}$$

for all positive vectors $\boldsymbol{y}$ *and provided that at least one active action* $o \in \mathcal{A}^{\underline{i}\leftarrow}$ *is strongly split for all states* $\underline{i}$*.*

Notice that the first condition automatically satisfies the condition of theorem 1 (at $\boldsymbol{y} = \boldsymbol{x}$) and that the second condition is checkable by the hypothesis that the reversed agents of the components P_k are known.

Proof

Let the $|L|-$vector $\boldsymbol{x} = (\ldots x_a \ldots)$ for all $a \in L$. Then $\boldsymbol{x}$, if it exists, is the solution of the fixed point problem

$$\boldsymbol{x} = \mathbf{F}(\boldsymbol{x})$$

where the vector-valued function $\mathbf{F}$ is defined by $F_a(\boldsymbol{x}) = \overline{p_a}\{\top \leftarrow \boldsymbol{x}\} = \pi_k(j)\{\top \leftarrow \boldsymbol{x}\}p_{k;ja}/\pi_k(i)\{\top \leftarrow \boldsymbol{x}\}$ and P_k is the (unique) component of the cooperation in which $a : j \to i$ is active. By hypothesis, the (unnormalized) marginal probabilities $\pi_k(\cdot)\{\top \leftarrow \boldsymbol{x}\}$ exist as continuous, positive-valued functions of $\boldsymbol{x}$, being a solution of linear equations. Hence $\mathbf{F}$ is continuous and so the fixed point exists provided it is a mapping between compact spaces, by Brouwer's theorem.

This may be proved by showing that, in the iteration $\boldsymbol{x}^{m+1} = \mathbf{F}(\boldsymbol{x}^m)$, for all $a \in L, m \geq 0, x_a^m$ is positive and bounded. Now consider the incoming active actions in component P_1 at joint state $\underline{i}$. For brevity, we use * to denote the renaming $\{\top \leftarrow \boldsymbol{x}^m\}$, so that, for example, $p_{k;i}^* \equiv p_{k;i}\{\top \leftarrow \boldsymbol{x}^m\}$. By the definition in theorem 1,

$$\begin{aligned}
X^{m+1}_{\mathcal{A}_1^{i\leftarrow}} &= \sum_{a \in \mathcal{A}_1^{i\leftarrow}} \overline{p_a}^{\,*} \\
&\leq g_{1i}(\boldsymbol{x}^m)\left(\overline{p_{1;i\leftarrow}}^{\,*} - \sum_{a \in \mathcal{P}_1^{i\leftarrow}} \overline{p_a}^{\,*}\right) \\
&= g_{1i}(\boldsymbol{x}^m)\left(p_{1;i}^* - \sum_{a \in \mathcal{P}_1^{i\leftarrow}} \overline{p_a}^{\,*}\right) \qquad (6) \\
&= g_{1i}(\boldsymbol{x}^m)\left(X^m_{\mathcal{P}_1^{i\rightarrow}} + \gamma_{1;i} - \sum_{a \in \mathcal{P}_1^{i\leftarrow}} \overline{p_a}^{\,*}\right)
\end{aligned}$$

where $\gamma_{k;i} \geq 0$ is the total rate out of state i in component process P_k due to non-passive actions, a constant.

We now proceed by induction and suppose initially that $n = 2$ and, without loss of generality, that P_1 has a strongly split, active, incoming action in all states. Summing over the components P_1, P_2 at joint state $\underline{i}$ and using the first condition of the proposition, we have

$$X^{m+1}_{\mathcal{A}^{i\leftarrow}} \leq u_{\underline{i}}\left(X^m_{\mathcal{A}^{i\leftarrow}} + \Gamma_{\underline{i}}\right)$$

where $u_{\underline{i}} = \max(g_{1i_1}(\boldsymbol{x}^m), g_{2i_2}(\boldsymbol{x}^m)) < 1$ by the strong splitting (second condition) and $\Gamma_{\underline{i}} = \sum_{k=1}^{2} \gamma_{k;i_k} - \sum_{a \in \mathcal{A}^{\underline{i}\rightarrow} \setminus \mathcal{P}^{\underline{i}\rightarrow}} \alpha_a^{\underline{i}} \geq 0$. Since this inequality holds for all states $\underline{i}$ and all active actions must be incoming to some state, we have, by summing over appropriate states if necessary,

$$X_{\mathcal{A}_1}^{m+1} \leq U\left(X_{\mathcal{A}_1}^m + \Gamma\right) \tag{7}$$

for non-negative constants Γ and $U < 1$. This sequence is increasing and it is routine to show that it is bounded, the upper bound being at least $\Gamma/(1-U)$, which proves the first part of the proposition for $n = 2$. Now consider the two-component cooperation $P_1 \underset{L_1}{\bowtie} \left(\underset{\substack{k=2 \\ L}}{\overset{n}{\bowtie}} P_k \right)$ and assume inductively that $X_{L\backslash\mathcal{A}_1}^m$ is bounded. Proceeding as before (in the base case of the induction) we observe that the first condition of the theorem again holds and obtain

$$X_{\mathcal{A}_1}^{m+1} \leq U'\left(X_{\mathcal{A}_1}^m + \Gamma\right) \tag{8}$$

where $U' < 1$. Hence $X_{\mathcal{A}_1}^m$ is also bounded and so is $X_{\mathcal{A}_1}^m + X_{L\backslash\mathcal{A}_1}^m = X_L^m$ ♠

Remarks

(1) Notice that the proof accomodates transition rates that are locally state dependent, i.e. action rates that can depend on the derivative of the agent in which they are active. This can be seen by the explicit dependence on the source state i of a transition in the proof. In contrast, the reversed rates $\overline{p_a}$ must be constant over the instances of action type a, as required in the conditions of theorem 1.

(2) For the original RCAT, extended to multiple cooperations, the proof is easier since the passive actions in each component are outgoing from all that process's states; similarly for its incoming active actions. Hence we only need the intermediate equation (6), which implies

$$X_{\mathcal{A}_k^{i\leftarrow}}^{m+1} \leq g_{ki}(\boldsymbol{x}^m)\left(X_{\mathcal{P}_k^{i\rightarrow}}^m + \gamma_{k;i}\right)$$

for $k = 1, 2$. Summing over k then yields equation 7 directly.

(3) If $\Gamma = 0$ in equation 7, the only solution is $\boldsymbol{x} = \mathbf{0}$. An example of this is a queueing network with no external arrivals but some departures, which eventually becomes empty.

(4) The conditions of the proposition are sufficient and quite general, but not necessary. In the special case that the second condition is absent, we would have in the above proof that $X_L^{m+1} \leq X_L^m + \Gamma$ where $\Gamma \geq 0$. If $X_L^{m+1} = X_L^m + \Gamma$ and $\Gamma > 0$, the sequence (X_L^m) diverges and there may be no fixed point solution for $\boldsymbol{x}$. An example of this is a queueing network with no external departures but some arrivals—hence no strongly split active actions. Such a closed network has no steady state, always increasing its population. If $\Gamma = 0$, however, the sequence is bounded and so a fixed point exists. One solution is $\boldsymbol{x} = \mathbf{0}$, but others may exist in a homogeneous set of equations that merely scale the ensuing equilibrium

probabilities. A queueing example is the well known closed network with no arrivals or departures which has a steady state with product-form solution [13,6].

3.5 Separable equilibrium probabilities

The introduction of residual actions means that we can directly obtain product-form solutions for the equilibrium probabilities of multi-agent cooperations. We present this result as a corollary to Theorem 1.

Corollary 1 *Assuming the cooperation set L is finite, the cooperation of theorem 1 has product-form solution $\pi(\underline{i}) \propto \prod_{k=1}^{n} \pi_k(i_k)$ for the equilibrium probability of state $\underline{i} = (i_1, \ldots, i_n)$, where $\pi_k(i_k)$ is proportional to the equilibrium probability of state i_k in the process denoted by R_k.*

Proof
Let the cooperation of theorem 1 be $C = \underset{L}{\overset{n}{\underset{k=1}{\bowtie}}} P_k$. In the cooperation, C^ϵ say, with ϵ-transitions on every synchronising action, one path from some chosen reference state $\mathbf{0}$ to an arbitrary state $(i_1, \ldots, i_n)$ takes $i_1, \ldots, i_n$ steps in each dimension in succession; i.e. proceeds from $\mathbf{0}$ to $(i_1, 0, \ldots, 0)$ to $(i_1, i_2, 0, \ldots, 0)$ to $(i_1, i_2, i_3, 0, \ldots, 0)$ ... to $(i_1, \ldots, i_n)$. The segment in dimension k is precisely the path from state 0 to i_k in the process denoted by R_k, for which the reversed process and product-form is known by hypothesis $(1 \leq k \leq n)$. Hence this segment contributes the factor $\pi_k^\epsilon(i_k)$, a product of the ratios of forward to reversed rates on each of the i_k steps. In the application of MARCAT to C^ϵ, let the solution for the rates $\top_a$ be x_a^ϵ and the resulting product-form be $\pi^\epsilon(\underline{i})$. Then, since L is finite, $\lim_{\epsilon\to 0} x_a^\epsilon = x_a$ and hence $\lim_{\epsilon\to 0} \pi^\epsilon = \pi$. ♠

4 Illustrative example: Jackson's theorem

Consider an M-node Jackson network, with respective external arrival rates $\lambda_1, \ldots, \lambda_M$, service rates $\mu_1, \ldots, \mu_M$, and routing probability p_{ij} from node i to node j $(1 \leq i \neq j \leq M)$, where $p_{ii} = 0$. Tasks leave the network from node i with probability $p_{i0} = 1 - \sum_{j=1}^{M} p_{ij}$. We do not consider departures from a node back to itself as this is considered part of the definition of the component process for that node. Such departures can be included easily with more complex components.

This network can be described by the PEPA expression $\underset{L}{\overset{M}{\underset{k=1}{\bowtie}}} P_{k,0}$ (starting with an empty network), where, for $1 \leq k \leq M$:

$$\begin{array}{ll} P_{k,n} = (e_k, \lambda_k).P_{k,n+1} & n \geq 0 \\ P_{k,n} = (a_{jk}, \top_{jk}).P_{k,n+1} & n \geq 0, 1 \leq j \neq k \leq M \\ P_{k,n} = (d_k, p_{k0}\mu_k).P_{k,n-1} & n > 0 \\ P_{k,n} = (a_{kj}, p_{kj}\mu_k).P_{k,n-1} & n > 0, 1 \leq j \neq k \leq M \end{array}$$

with $L_k = \{a_{kj} \mid j \neq k\} \cup \{a_{jk} \mid j \neq k\}$. In the sequel, we use the abbreviations $\top_{ij}$ for $\top_{a_{ij}}$ and x_{ij} for $x_{a_{ij}}$, $1 \leq i \neq j \leq M$. It would have been just as easy in principle to define a much more complex G-network with triggers, resets and batches, as considered in two-node networks in [9] for example, but the PEPA definition would have been much longer, the notation more dense and the clarity somewhat obscured. The correct traffic equations would emerge via the rate equations in exactly the same way, however.

In this example, $\mathcal{P}^{\underline{i}\rightarrow} = \mathcal{A}^{\underline{i}\leftarrow} = \bigcup_{k=1}^{M} L_k$ which satisfies the last condition of theorem 1,

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

trivially (this is, of course, an application of the multi-agent version of the original, two-component RCAT [7]).

Every instance of the reversed action of an active action, type $a_{kj} \in \mathcal{A}_k$ say $(1 \leq k \neq j \leq M)$, has rate which is a constant fraction p_{kj} of the net arrival rate at the M/M/1 queue represented by component k. This follows from proposition 1 and since every active action represents a departure from some queue. Hence, considering the symbols $\top_{jk}$ as variables denoting real rates in preparation for an application of MARCAT,

$$\overline{p_{a_{kj}}} = p_{kj}(\lambda_k + \sum_{j \neq k} \top_{jk})$$

at all instances of a_{kj} since λ_k is state-independent. We can therefore apply MARCAT and obtain the following equations in the variables $\top_{jk}$ corresponding to agent $P_{1,0}$, where we set $\top_{ii} = 0$ for uniformity of the exposition, $1 \leq i \leq M$:

$$\begin{aligned} \top_{21} &= p_{21}(\lambda_2 + \top_{12} + \top_{22} + \ldots + \top_{M2}) \\ &\vdots \\ \top_{M1} &= p_{M1}(\lambda_M + \top_{1M} + \top_{2M} + \ldots + \top_{MM}) \end{aligned}$$

and in general, $\top_{ij} = p_{ij}\left(\lambda_i + \sum_{k=1}^{M} \top_{ki}\right)$ for agent $P_{j,0}$, $j = 1, \ldots, M$. These equations have a solution, say $\top = \boldsymbol{x}$, by theorem 2 or by a more conventional, direct analysis of the linear equations.

Let $v_i = \lambda_i + \sum_{k=1}^{M} x_{ki}$ for $1 \leq i \leq M$ so that:

$$x_{ij} = v_i p_{ij} \qquad\qquad 1 \leq i, j \leq M$$

These are precisely the traffic equations for the internal flows, where x_{ij} is the internal traffic rate from node i to node j. In fact, summing over i we obtain

$$v_j - \lambda_j = \sum_{i=1}^{M} v_i p_{ij}$$

which are the usual traffic or 'visitation rate' equations for the network, v_i being the average number of visits made to node i in unit time at equilibrium in an open network—and proportional to this quantity in a closed network.

4.1 Reversed process

By applying MARCAT, we can now easily obtain the reversed PEPA agent of $\underset{L}{\overset{M}{\underset{k=1}{\bowtie}}} P_{k,0}$, which is $\underset{\overline{L}}{\overset{M}{\underset{k=1}{\bowtie}}} X_{k,0}$, where, for $1 \leq j, k \leq M$:

$$\begin{aligned}
X_{k,n} &= (\overline{e_k}, \frac{\lambda_k}{v_k}\mu_k).X_{k,n-1} && n > 0 \\
X_{k,n} &= (\overline{a_{jk}}, \frac{x_{jk}}{v_k}\mu_k).X_{k,n-1} && n > 0 \\
X_{k,n} &= (\overline{d_k}, (1 - \sum_{j \neq k} p_{kj})v_k).X_{k,n+1} && n \geq 0 \\
X_{k,n} &= (\overline{a_{kj}}, \top).X_{k,n+1} && n \geq 0
\end{aligned}$$

with $\overline{L_k} = \{\overline{a_{kj}} \mid j \neq k\} \cup \{\overline{a_{jk}} \mid j \neq k\}$.

The rates for the reversed actions are easily calculated by proposition 1. For example, consider the reversed external arrivals at node 1, which have type $\overline{e_1}$. The total departure rate of node 1 is μ_1 and the proportion of e_1 in the forward process is $\frac{\lambda_1}{v_1}$. By proposition 1, the rate for the reversed action $\overline{e_1}$ is $\frac{\lambda_1}{v_1}\mu_1$.

Compared to the original theorem of [7], MARCAT significantly reduces the effort in finding the reversed agent of a cooperation with more than two components. Moreover, it also yields the product-form solution for the equilibrium state probabilities, which is what is usually sought after, directly—without knowledge of the reversed process at all, as we will now see.

4.2 Product-form

In the application of MARCAT in the previous sub-section, for $1 \leq k \leq M$, the agents R_k of theorem 1 are immediately seen to be $M/M/1$ queues with total arrival rate $\lambda_k + \sum_{j=1}^{M} x_{jk} = v_k$ and total service rate $\sum_{j=0}^{M} \mu_k p_{kj} = \mu_k$. Hence an unnormalised equilibrium probability for state i_k in the process denoted by R_k is $\left(v_k/\mu_k\right)^{i_k}$. Jackson's theorem then follows—for either an open or a closed network—directly from corollary 1, i.e. the equilibrium probability for state $\underline{i}$ in the network is proportional to

$$\prod_{k=1}^{M} \left(\frac{v_k}{\mu_k}\right)^{i_k}$$

Notice how the MARCAT methodology is far more concise than that of the original RCAT [7] in four ways:

- the enabling conditions all relate to single queues and are uniform so all can be checked at once;
- there is no need to seek suitable paths in the forward and reversed processes—the product-form comes directly from the corollary to theorem 1;
- there is no need for a lengthy, inductive proof for networks with more than two nodes;
- in more complex networks with non-linear rate equations, it is easy to prove the existence of a solution using proposition 2, rather than a customised analysis.

5 Conclusion

The Multiple Agents Reversed Compound Agent Theorem (MARCAT) greatly simplifies the use of its predecessor, RCAT, for cooperations of an arbitrary number of agents. In terms of automation, its greatest advantage is that inductive proofs of reversed processes, and hence product-forms, are unnecessary. Our example illustrates how MARCAT deals with a cooperation of any number of agents that synchronise pairwise. Clearly the same applies to all the variants of product-form G-networks, as investigated extensively in the context of RCAT in [9] recently. Furthermore, much simpler proofs of RCAT and MARCAT were presented, based on the notion of residual, ϵ-actions. These essentially allow every synchronising action to be ignored, being substituted by a pair of independent actions, one in each of the cooperating components. Perhaps more importantly, residual actions provide a direct route to finding product-forms, which explains the syntactic similarities between the results obtained for open and closed networks, initially by Jackson [13].

A second major contribution of this paper is the proof of existence of solutions to MARCAT's rate equations. Without this, the derived reversed processes and product-form solutions would be somewhat vaccuous and previously one had to rely on direct analyses of the rate equations in specific applications. Indeed, this is what others such as Gelenbe did to produce rigorous results for product-forms in G-networks [5,4]. With our new general result, however, all of these solutions and their associated existence proofs are particular applications and so are subsumed.

The methodology can be automated and its newly generalised, multi-agent form facilitates the uniform derivation of many diverse separable solutions, as considered just for two-component cooperations in [9,10]. These applications range from multi-class queueing networks, through the numerous variants of G-networks, to networks with mutual exclusion and blocking in critical sections.

Although it applies to multiple cooperations, the new theorem is restricted to actions that can cooperate in only two agents at a time, e.g. representing departures from one queue passing to another. The method of [9], used to model triggers in G-networks, can account for a cascade of transitions in a chain of components, where one transition essentially initiates a sequence of instantaneous secondary transitions in a given order. Complementary to this, ongoing research is investigating the possibility that one active action can cooperate with several passive actions simultaneously. This would not only induce an alternate approach to triggers but could also account for other types of simultaneous movement of customers in queueing networks, as in [4] for example. Other work in progress includes the analogous discrete time version of MARCAT, which introduces the problem of multiple events in a time slot, in contrast to the uniqueness of instantaneous transitions in a continuous time Markov chain.

References

[1] F. Baskett, K. M. Chandy, R. R. Muntz and F. Palacios. Open, Closed and Mixed Networks of Queues with Different classes of Customers. J. ACM, 22(2):248-260, 1975.

[2] M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. Theoretical Computer Science, 202(1–2): 1–54, July 1998.

[3] R.J. Boucherie. A Characterisation of Independence for Competing Markov Chains with Applications to Stochastic Petri Nets. IEEE Transactions on Software Engineering, 20(7): 536–544, July 1994.

[4] X. Chao, M. Miyazawa and M. Pinedo. *Queueing networks: customers, signals and product form solutions.* Wiley, 1999

[5] E. Gelenbe. The first decade of G-networks. European Journal of Operational Research, 126(2): 231-232, October 2000.

[6] W.J. Gordon and G.F. Newell. Closed Queueing Systems with Exponential Servers. Operations Research, 15(2): 254-265, Mar-Apr 1967.

[7] P.G. Harrison. Turning Back Time in Markovian Process Algebra. Theoretical Computer Science, 290(3): 1947-1986, January 2003.

[8] P.G. Harrison. Mechanical Solution of G-networks via Markovian Process Algebra. Proc. International Conference on Stochastic Modelling and the IV International Workshop on Retrial Queues, Cochin, India, December 2002, Notable Publications, 2002.

[9] P.G. Harrison. Compositional reversed Markov processes, with applications to G-networks. Performance Evaluation, 2004.

[10] P.G. Harrison. Reversed processes, product forms and some non-product forms. J. Linear Algebra Applications, 2004.

[11] N. Götz, U. Herzog and M. Rettelbach. Multiprocessor and Distributed System Design: The Integration of Functional Specification and Performance Analysis Using Stochastic Process Algebras. Tutorial proceedings of PERFORMANCE '93, Rome, Springer-Verlag Lecture Notes in Computer Science, Vol. 729, 1993.

[12] J. Hillston. A Compositional Approach to Performance Modelling. PhD thesis, University of Edinburgh, 1994.

[13] J.R. Jackson. Jobshop-like queueing systems. Management Science, 10(1): 131-142, 1963.

[14] F.P. Kelly. *Reversibility and Stochastic Networks.* John Wiley & Sons Ltd, 1979.

Appendix: A PEPA-based MPA

We use a Markovian process algebra language that defines *agents*, which denote continuous time Markov chains. Agents evolve through the execution of *actions*, which have exponentially distributed durations. An action is a pair, the first component of which is its *type* (or name) and the second of which is its *rate*. Thus, agents and actions in an MPA specification correspond to states and transitions respectively in the underlying Markov process. MPA describes systems at a higher level than explicit state-transition diagrams. In particular, the *cooperation* combinator of PEPA defines precisely how agents interact in a concise manner, using generic descriptions of their actions' rates. The precise semantics of the original PEPA language is given in [12], and defines the

Markov process denoted by a PEPA agent. Notice that the term 'agent' is *syntactic*, part of the MPA, whereas 'process' is a semantic entity with a well defined value in the domain of continuous time Markov chains. However, the terms are essentially isomorphic.

In this paper, we use only the *prefix, cooperation, assignment* and *choice* combinators of the MPA PEPA (generalised straightforwardly in the body of the paper to multiple cooperations):

(1) The prefix combinator defines an agent $(a, \lambda).P$ that carries out action (a, λ) of *type* (or 'name') a at *rate* λ and subsequently behaves as agent P;
(2) The agent describing the cooperation of two agents P and Q, which synchronise over actions with types in a specified set L, is written $P \underset{L}{\bowtie} Q$;
(3) A new agent A is defined using the assignment combinator, $A = P$;
(4) Choice is denoted either by the usual combinator symbol $+$ of conventional PEPA or by multiple assignments to a process name, as in [7].

The semantics of these combinators is the same as in PEPA, given by the Markov process defined on the *derivation graph* of an agent [12].

In the cooperations considered in this paper, every action type in L is *active*, i.e. has a specified real valued rate, in exactly one of the agents P, Q and is *passive*, i.e. 'waits', in the other. The rate of the joint action in the cooperation is then that specified for the active action. A passive action is indicated by an *unspecified* rate, denoted $\top$, essentially infinite in the sense that the action will proceed instantly once its synchronising action is ready. Any action with type in L can only proceed simultaneously in both of the cooperating agents. Partly to emphasise this restricted use of cooperation, we use a subtly different bowtie symbol.

The process algebra described above is a subset of PEPA. The alternate syntax for the choice combinator was preferred originally because it makes the synthesis of reversed processes simpler to explain (see [7]) but it also often leads to more concise mathematical arguments. We extend the syntax beyond PEPA to allow:

- *Parameterised agents.* An agent may be named P_i where P is a symbol used to define an agent in plain PEPA and the subscript i is a member of some countable *index set* I, typically the natural numbers. This allows simple processes such as queues to be defined in a concise way—see for example section 4. Since each P_i is a valid PEPA-agent symbol, the extension is still within PEPA if a countably infinite number of agents are allowed. This is not precluded in PEPA.
- *Infinite derivation graphs.* As a result of this, the number of drivatives of

an agent can be (countably) infinite, leading to an underlying Markov process with infinite state-space. Whilst PEPA's semantics is usually associated with a finite state-space Markov chain, this is only for reasons of implementation and numerical solution. There is no semantic reason to restrict PEPA to finite state-spaces.

For clarity, we call 'PEPA with a countably infinite state-space' σ-PEPA when it is necessary or desirable to distinguish it from 'finite state-space PEPA'. Our process algebra is therefore a subset of σ-PEPA and inherits its semantics.

We also make use of the syntactic *relabeling*, $P\{y \leftarrow x\}$, that denotes the process P in which all occurences of the symbol y are changed to x, which may be an expression. Thus, for example, $((a,\lambda).P)\{\lambda \leftarrow \mu\}$ denotes the agent $(a,\mu).P\{\lambda \leftarrow \mu\}$. Any symbol can be relabeled, defining a new process with (possibly) new semantics. When a rate is relabelled, it generates a new symbolic process with a corresponding new symbolic solution for its equilibrium probabilities, when they exist. Of course, if numerical rates are bound to the new symbols, the properties of the ensuing Markov chains may change, for example ergodicity may be lost. But this does not change the respective semantic models, which are allowed to be different.

We say that E is a subexpression of P, written $E \sqsubset P$ if the string of symbols E denotes a valid syntactic expression that occurs in the definition of P. We then extend the definition of relabelling:

$$P\{E[y] \leftarrow X_E \mid \mathcal{C}(E)\}$$

relabels each sub-expression $E[y] \sqsubset P$ that satisfies the condition $\mathcal{C}$ by the expression X_E, which depends on the instance of E satisfying $\mathcal{C}$. Thus, for example, we can change every occurence of a process P_i to Q_i, $i = 1, 2, \ldots$, by writing $P\{P_i \leftarrow Q_i \mid i = 1, 2, \ldots\}$.

Finally, reversed entities (agents, actions, action types, action rates) are denoted with an overbar. Their semantics are defined by the reversed process of the semantic Markov chain denoted by the PEPA agent in which they appear without an overbar.