A Kronecker Representation for PEPA net Models

Jane Hillston* and Leïla Kloul*†

1 Introduction

PEPA nets [2] extend the PEPA stochastic process algebra [3] by allowing PEPA components to be used as the tokens of a coloured stochastic Petri net. Communication between tokens at different places is not allowed; for tokens to communicate they must first meet at one of the places of the net. While they are together in the same place they may communicate but when one or other of them moves to another place the communication between them must cease until another firing of the net brings the tokens together again in the same place.

In addition to the PEPA components which are used as tokens, a PEPA net contains *static components*. Each static component is resident in a place of the net and is unable to move. Thus each place is a PEPA context[1]. Static components may communicate with tokens while they are present or they can communicate with other static components in the same place. Communication between static components at different places is not allowed.

PEPA nets provide a rich expressive high-level language for continuous time Markov chains (CTMCs). As with most state-based system description techniques it faces the problem of state space explosion. Indeed this problem is exacerbated by the characteristics of the large distributed nature of mobile computation systems.

In this paper we present a Kronecker representation for PEPA nets. It is well known that there is a close relationship between Kronecker sum and the composition of independent components. However, the model elements which we compose here are far from independent and intimately affect each other's evolution. There have been previous characterisations of both classes of SPN [1] and PEPA [4] in terms of Kronecker algebra, and we take inspiration from these. Nevertheless our problem is more complex as there are

*LFCS, The University of Edinburgh, Edinburgh EH9 3JZ, UK. Email: {`jeh`, `leila`}`@lfcs.ed.ac.uk`

†On leave from PRISM, Université de Versailles, 45, Avenue des Etats-Unis 78000 Versailles, France.

[1] A *context* is a process algebra expression with "holes" or cells into which components may be inserted

two considerations which must be considered before each state change: the state of each component and the location of each token component. As we will see these key pieces of information are represented explicitly in our Kronecker expression. In this paper we restrict the class of PEPA nets which we consider to have only one token of each type.

2 The Global Generator Matrix

We denote by $\mathcal{P}(T_k)$ the set of all possible places or locations of token T_k, i.e. the set of places which contain a cell of type T_k. Thus if P_i belongs to $\mathcal{P}(T_k)$, then P_i is a place where token T_k can move to after one or several firing transitions, because it includes a cell of the appropriate type.

For each token type k, $k = 1, \ldots, K$, in the PEPA net, we associate a matrix $E_{k,\alpha}$ of size $|\mathcal{P}(T_k)| \times |\mathcal{P}(T_k)|$. We refer to $E_{k,\alpha}$ as the *location matrix* of token k for activity α. This matrix allows us to know in which place the token is when performing activity α. Moreover, if α is a firing activity, this matrix indicates to which place the token is moving to after performing α. In particular, we denote by E_j the identity matrix associated with a token of type T_j specifying that the token may be at any place of the net.

In the following we use *net components* to refer to PEPA components within the net: these can be either static components which stay resident within a place, or dynamic components (tokens) which circulate between the places of the net. Moreover, considering two matrices A and B, we define the binary operator $\bigodot$ as:

$$A \bigodot B = \begin{cases} A \bigoplus B & \text{if } A \neq 0 \\ 0 & \textit{otherwise} \end{cases}$$

2.1 The Kronecker Representation

Consider a PEPA net model where components may have individual and/or cooperating activities. Using Kronecker algebra, the corresponding generator matrix Q can be represented as stated in the following equation.

$$Q = \left(\bigoplus_{j=1}^{N} W_j \right) \bigodot \left(\bigoplus_{j=1}^{K} E_j \right) + \sum_{\alpha \in \mathcal{Z} \cup \mathcal{F}} r_\alpha \left(\bigotimes_{j=1}^{N} R_{j,\alpha} - \bigotimes_{j=1}^{N} \overline{R}_{j,\alpha} \right) \otimes \left(\bigotimes_{j=1}^{K} E_{j,\alpha} \right) \quad (2.1)$$

where

- K is the total number of token types,
- N is the total number of PEPA net components, including static components and tokens,

- $\mathcal{Z}$ is the set of cooperating action types which occur within places (between tokens, between static components, or between tokens and static components),

- $\mathcal{F}$ is the set of firing action types which label transitions of the Petri net,

- W_i is the transition matrix of net component C_i relating solely to its individual activities,

- $E_{j,\alpha}$ is the location matrix of the token of type T_j for action type α.

- E_j is the identity matrix associated with token of type T_j.

- $R_{i,\alpha}$ is the probability transition matrix of component C_i due to activity of α,

- $\overline{R}_{i,\alpha}$ is the normalisation matrix associated with the shared activity α.

Unlike the local transition matrices W_j, matrices $R_{j,\alpha}$ are not generators. Therefore, matrices $\overline{R}_{j,\alpha}$ are introduced in equation (2.1) as diagonal corrector matrices of $R_{j,\alpha}$.

2.2 Example

Consider the PEPA net of Figure 1. This net is composed of three places $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ where two tokens $Agent_{006}$ and $Agent_{007}$ can move. The mobility of the first token is restricted to places $\mathbf{P}_1$ and $\mathbf{P}_2$, whereas the mobility of the second is restricted to places $\mathbf{P}_2$ and $\mathbf{P}_3$.

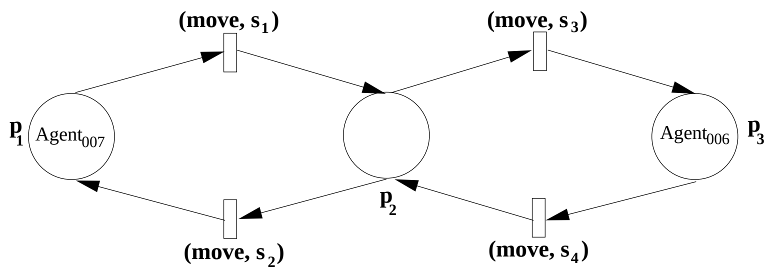

Figure 1: A PEPA net example

The static components Inside each place there is a static component with which the current component token in this place must cooperate to achieve an activity. These components are given in the following in the numbering order of the places where they are.

$$\begin{array}{lcl} Remote & \stackrel{def}{=} & (download, \top).Remote \\ Connection & \stackrel{def}{=} & (save, \top).Connection + (select, \top).Connection \\ MI5 & \stackrel{def}{=} & (process, \top).MI5 \end{array}$$

The components tokens

$$\begin{array}{lcl} Agent_{007} & \stackrel{def}{=} & (download, r_1).Agent'_{007} + (save, r_2).Agent'_{007} \\ Agent'_{007} & \stackrel{def}{=} & (\mathbf{move}, \top).Agent_{007} \end{array}$$

$$\begin{array}{lcl} Agent_{006} & \stackrel{def}{=} & (select, r_3).Agent'_{006} + (process, r_4).Agent'_{006} \\ Agent'_{006} & \stackrel{def}{=} & (\mathbf{move}, \top).Agent_{006} \end{array}$$

The places

$$\begin{array}{lclcl} \mathbf{P_1} & \stackrel{def}{=} & Remote & \underset{\{download\}}{\bowtie} & Agent_{007}[Agent_{007}] \\ \mathbf{P_2} & \stackrel{def}{=} & Connection & \underset{\{save, select\}}{\bowtie} & (Agent_{007}[\_] \mid\mid Agent_{006}[\_]) \\ \mathbf{P_3} & \stackrel{def}{=} & MI5 & \underset{\{process\}}{\bowtie} & Agent_{006}[Agent_{006}] \end{array}$$

In this example, we have only synchronising and firing action types. Using equation 2.1, the generator expression becomes:

$$Q = \sum_{\alpha \in \mathcal{Z} \cup \mathcal{F}} r_\alpha \left(\bigotimes_{j=1}^{N} R_{j,\alpha} - \bigotimes_{j=1}^{N} \overline{R}_{j,\alpha} \right) \otimes \left(\bigotimes_{j=1}^{K} E_{j,\alpha} \right)$$

where the cooperation set is $\mathcal{Z} = \{download, save, select, process\}$, $K = 2$ and $N = 5$. As there are four instances of firing activity **move**, we need to distinguish between them. Thus $\mathcal{F} = \{\mathbf{move_{12}}, \mathbf{move_{21}}, \mathbf{move_{23}}, \mathbf{move_{32}}\}$ where $\mathbf{move_{ij}}$ denotes the firing action type from place i to place j.

The matrices related to action types in set $\mathcal{Z}$ are computed as follows:

- $\alpha =$ **download**

$$\begin{array}{rcl} R_{Agent_{007},\alpha} & = & \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}; \qquad R_{Agent_{006},\alpha} = Id_{2\times 2} \\ E_{Agent_{007},\alpha} & = & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}; \qquad E_{Agent_{006},\alpha} = Id_{2\times 2} \\ R_{Remote,\alpha} & = & R_{MI5,\alpha} = R_{Connection,\alpha} = Id_{1\times 1} \end{array}$$

- $\alpha =$ **save**

$$\begin{array}{rcl} R_{Agent_{007},\alpha} & = & \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}; \qquad R_{Agent_{006},\alpha} = Id_{2\times 2} \\ E_{Agent_{007},\alpha} & = & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}; \qquad E_{Agent_{006},\alpha} = Id_{2\times 2} \\ R_{Remote,\alpha} & = & R_{MI5,\alpha} = R_{Connection,\alpha} = Id_{1\times 1} \end{array}$$

- $\alpha =$ **select**

$$\begin{aligned} R_{Agent_{007},\alpha} &= Id_{2\times 2}; \qquad R_{Agent_{006},\alpha} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ E_{Agent_{007},\alpha} &= Id_{2\times 2}; \qquad E_{Agent_{006},\alpha} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ R_{Remote,\alpha} &= R_{MI5,\alpha} = R_{Connection,\alpha} = Id_{1\times 1} \end{aligned}$$

- $\alpha =$ **process**

$$\begin{aligned} R_{Agent_{007},\alpha} &= Id_{2\times 2}; \qquad R_{Agent_{006},\alpha} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ E_{Agent_{007},\alpha} &= Id_{2\times 2}; \qquad E_{Agent_{006},\alpha} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ R_{Remote,\alpha} &= R_{MI5,\alpha} = R_{Connection,\alpha} = Id_{1\times 1} \end{aligned}$$

The matrices related to action types in set $\mathcal{F}$ are computed as follows:

- $\alpha =$ $\mathbf{move_{12}}$

$$\begin{aligned} R_{Agent_{007},\alpha} &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}; \qquad R_{Agent_{006},\alpha} = Id_{2\times 2} \\ E_{Agent_{007},\alpha} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}; \qquad E_{Agent_{006},\alpha} = Id_{2\times 2} \\ R_{Remote,\alpha} &= R_{MI5,\alpha} = R_{Connection,\alpha} = Id_{1\times 1} \end{aligned}$$

- $\alpha =$ $\mathbf{move_{21}}$

$$\begin{aligned} R_{Agent_{007},\alpha} &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}; \qquad R_{Agent_{006},\alpha} = Id_{2\times 2} \\ E_{Agent_{007},\alpha} &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}; \qquad E_{Agent_{006},\alpha} = Id_{2\times 2} \\ R_{Remote,\alpha} &= R_{MI5,\alpha} = R_{Connection,\alpha} = Id_{1\times 1} \end{aligned}$$

- $\alpha =$ $\mathbf{move_{23}}$

$$\begin{aligned} R_{Agent_{007},\alpha} &= Id_{2\times 2}; \qquad R_{Agent_{006},\alpha} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \\ E_{Agent_{007},\alpha} &= Id_{2\times 2}; \qquad E_{Agent_{006},\alpha} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ R_{Remote,\alpha} &= R_{MI5,\alpha} = R_{Connection,\alpha} = Id_{1\times 1} \end{aligned}$$

- $\alpha =$ $\mathbf{move_{32}}$

$$\begin{aligned} R_{Agent_{007},\alpha} &= Id_{2\times 2}; \qquad R_{Agent_{006},\alpha} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \\ E_{Agent_{007},\alpha} &= Id_{2\times 2}; \qquad E_{Agent_{006},\alpha} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \\ R_{Remote,\alpha} &= R_{MI5,\alpha} = R_{Connection,\alpha} = Id_{1\times 1} \end{aligned}$$

The corresponding Markov chain is depicted in Figure 2 where the initial state is state 0 and the λ_i, $i = 0, \ldots, 15$ are defined as follows:

$$\begin{array}{lllllllll}
\lambda_0 & = & -(r_1+r_3); & \lambda_1 & = & -(r_1+r_4); & \lambda_2 & = & -(r_2+r_3) \\
\lambda_3 & = & -(r_2+r_4); & \lambda_4 & = & -(r_1+s_3); & \lambda_5 & = & -(r_1+s_4) \\
\lambda_6 & = & -(r_2+s_3); & \lambda_7 & = & -(r_2+s_4); & \lambda_8 & = & -(r_3+s_1) \\
\lambda_9 & = & -(r_4+s_1); & \lambda_{10} & = & -(r_3+s_2); & \lambda_{11} & = & -(r_4+s_2) \\
\lambda_{12} & = & -(s_1+s_3); & \lambda_{13} & = & -(s_1+s_4); & \lambda_{14} & = & -(s_2+s_3) \\
\lambda_{15} & = & -(s_2+s_4) & & & & & &
\end{array}$$

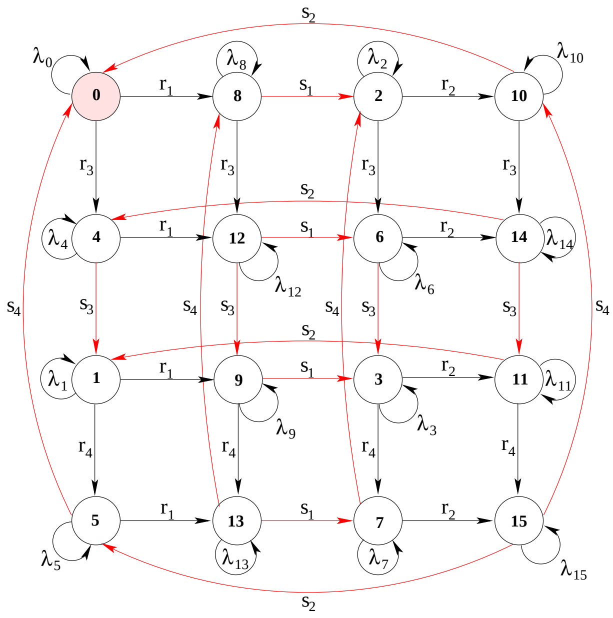

Figure 2: The Markov Chain

References

[1] S. Donatelli. Superposed stochastic automata: a class of stochastic petri nets with parallel solution and distributed state space. *Performance Evaluation*, (18):21–36, 1993.

[2] S. Gilmore, J. Hillston, L. Kloul, and M. Ribaudo. PEPA nets: A structured performance modelling formalism. *Performance Evaluation*, 2003. accepted for publication.

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

[4] J. Hillston and L. Kloul. An efficient kronecker representation for pepa models. *Lecture Notes in Computer Science, Proceedings of the Joint International Workshop, PAPM-PROBMIV 2001, Aachen*, (2165):120–135, 2001.