STUDIA INFORMATICA 2002
Volume 23 Number 4 (51)

Jean-Michel FOURNEAU
PRiSM, Université de Versailles Saint Quentin en Yvelines, France

DOMINO SYNCHRONIZATION: PRODUCT FORM SOLUTION FOR STOCHASTIC AUTOMATA NETWORKS

Summary. We present a new kind of synchronization which allows Stochastic Automata Networks (SAN) to have a product form steady-state distribution. Unlike previous models on SAN with product form solutions, our model allows synchronization between three automata. The synchronization is not the usual "Rendez-Vous" but an ordered list of transitions. Each transition may fail. When a transition fails, the synchronization ends but all the transitions already executed are kept. This class of SAN is a generalization of Gelenbe's networks with triggered customer movement. Finally, our result suggests an approximation based on product form for SAN whose synchronization are ordered lists of transitions of arbitrary size.

SYNCHRONIZACJA DOMINO: FORMA ILOCZYNOWA DLA SIECI AUTOMATÓW STOCHASTYCZNYCH

Streszczenie. W artykule przedstawiony jest nowy rodzaj synchronizacji, który pozwala przedstawić w formie iloczynowej rozwiązanie modelu Sieci Automatów Stochastycznych. W przeciwieństwie do wcześniejszych modeli w postaci Sieci Automatów Stochastycznych o rozwiązaniu produktowym, ta propozycja umożliwia synchronizację trzech automatów. Ten rodzaj synchronizacji nie reprezentuje zwykłego typu "Rendez-Vous". Jest on bowiem reprezentowany przez uporządkowaną listę tranzycji, z których każda może być aktywowana. Po zakończeniu reaktywacji tranzycje są zachowywane. Ta klasa Sieci Automatów Stochastycznych stanowi uogólnienie sieci Gelenbego z odpalanym ruchem klientóów. Uzyskane wyniki prowadzą do aproksymacji opartej na Sieciach Automatów Stochastycznych o rozwiązaniu iloczynowym, w których synchronizacje są wprowadzana na zasadzie listy tranzycji o określonym rozmiarze.

1. Introduction

Since they have been introduced by B. Plateau [14] to evaluate the performance of distributed algorithms, Stochastic Automata Networks (SAN for short) have been associated to new research on numerical solvers. The key idea is to take into account the tensor decomposition of the transition matrix of a SAN to improve the storage of the model and the complexity of the vector-matrix product [6]. The first algorithm proposed was a numerical resolution of steady-state distribution of the Markov chain associated to a SAN [15] using the power method. Since then, several numerical methods have been investigated ([5], [19], [8]). And SAN have been used to obtain loss rates in ATM networks [7] , blocking probabilities in multistage interconnection networks [1] or for the performance evaluation of a bandwidth allocation mechanism in Wireless ATM [20].

As a SAN is a modular decomposition into automata which are connected by synchronized transitions, SAN are closely related to Stochastic Process Algebra. Therefore, new results on SAN may be easily translated into other models based on composition such as process algebra, for instance PEPA ([12]). The tensor decomposition of the generator has been generalized for Stochastic Petri Nets (see for instance [4]) and other modular specification methods as well [13].

Recently, some analytical results for SAN have been presented. First, B. Plateau et al. [16] have considered SAN without synchronization. They proved that a product form steady-state distribution exists as soon as some local balance conditions are satisfied. Even without synchronization, the transitions of the automata are still dependent because of functional rates. Plateau's result is closely related to Boucherie's result on Markov chains in competition [3] and Robertazzi's theorems on Petri nets [17]. Similarly, using the same type of argument (i.e. group local balance), Sereno has proved a sufficient condition to obtain a product form solution for a PEPA model [18]. Different results have been obtained by Harrison using reversibility theory for PEPA models [11].

In [2], we have considered SAN with a special case of synchronization denoted as limited synchronization. In a limited synchronization, only two automata are active. We also restrict ourself to SAN without functional rates. We proved a sufficient condition to have a product form steady-state distribution: existence of a solution for a fixed-point system between the instantaneous arrival rate and the steady-state distributions of the automata in isolation. Some typical queueing networks such as Jackson's networks or Gelenbe's networks of positive and negative customers [9] have been shown to be examples of this type of SAN. For both networks, the fixed-point system is equivalent to the well-known flow equation. Our proof was based on global balance equation. Indeed, there is no local balance (in the usual sense) for Gelenbe's networks. However to generalize our former result to a more general class of SAN, we have to change the description of synchronization.

The assumption on synchronization used to define the SAN methodology was the "Rendez-Vous". Here, we consider a completely different kind of synchronization: the Domino synchronization that we will introduce more formally in the next section. Briefly, a Domino synchronization is an ordered list of tuples (automaton number, list of transitions inside this automaton). The synchronization takes place according to the order of the list. The synchronization may completely succeed or be only partial if some conditions are not satisfied.

The rest of the paper is organized as follows: in section II, we describe Domino synchronization. In section III, we state the main theorem of the paper, the proof of which is postponed into an appendix. Section IV is devoted to examples. Finally, we give some conclusions and some perspectives to extend our results to more general synchronizations.

2. Domino Synchronization

An automaton consists of states and transitions which represent the effects of events. These events are classified into two types: local events or synchronizing events. A local event affects a single automaton and is modeled by some local transitions. On the opposite, a synchronizing event modifies the state of more than one automaton (but loops are considered as valid transitions). Transitions rates may be fixed or functions of the states of the whole set of automata.

In this paper, we consider that the transition rates are fixed. The SAN methodology allows functional rates to couple the automata. However it is possible to replace functional rates by synchronization with loops. Each value of the function is replaced by a synchronization with loops and a fixed transition rate (i.e. the value of the function). Functions have been added in the SAN methodology to make more compact the representation using less synchronizations.

So we restrict ourself to continuous-time SAN without functions. The state space of the system is the cartesian product of the states of the automata which are combined in the network. The effective state space is in general only a subset of this product. Because of synchronizations, an automaton by itself is not Markovian. To obtain a multidimensional

Markov chain for the whole network, we assume exponentially distributed transition durations.

The synchronization formerly used for SAN are defined as "Rendez-Vous". This simply says that a synchronized transition is possible, if and only if, all automata are ready for this synchronized transition. We have to consider a completely different type of synchronization: the domino of three automata. The name comes from a group of domino tiles which falls one after the other. Of course, if one tile does not fall, the domino effect stops but the tiles already fallen stay down.

Let r be a synchronization number or label. The Domino synchronization consists of an ordered list of three automata called the master $msr(r)$, the slave $sl(r)$ and the relay $rl(r)$. The synchronization is performed according to the list order. The master of synchronization r is the initiator of the synchronization. It performs its transition. The slave may obey or not to the request of the master. If it does not follow the master, it makes a loop and the synchronization stops without any interaction with the third automaton (i.e. the relay). But the transition of the master is kept. If the slave obeys, it performs a real transition (i.e. not a loop) and the third automaton (i.e. the relay) now has to make a transition. This transition is either a loop (the relay refuses to follow) or a real transition (the relay obeys). In both cases, the master and the slave perform their transitions. The relay and the slave follow the master according to their local state and the list of transitions marked by label r. Note that this definition of synchronization implies that the master is never blocked by the slave or the relay (it is not a rendez-vous). This implies that every state of the automata $sl(r)$ and $rl(r)$ is the origin of at least one synchronized transition marked by synchronization label r.

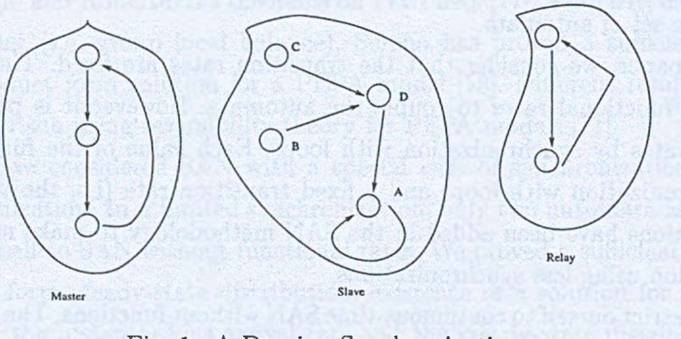

Fig. 1. A Domino Synchronization
Rys. 1. Synchronizacja Domino

Finally, let us remark that the name "relay" for the third automaton is justified by the following idea. Assume that automaton A_l is the relay for synchronization $r1$. It may also be the master for synchronization $r2$. Assume that the effect of synchronization $r1$ on A_l is that synchronization $r2$ becomes firable. Then, if the transition rate for synchronization $r2$ is higher than the other rates, it is more likely that $r2$ will almost immediately follow $r1$. Furthermore, as it may be seen in the next section, increasing the transition rate of some synchronization do not remove the product form solution.

3. Product Form Solution

We now establish a sufficient condition for a SAN with domino synchronization to have steady-state distribution which is obtained as the product of steady-state distribution of isolated automata. The automata are defined by the following matrices which may be either finite or infinite:
N matrices denoted as F_l which contains the transition rates of local transitions for automaton l. The matrices are normalized, i.e.

$$F_l[k,i] \geqslant 0 \;\text{ if }\; i \neq k \;\text{ and }\; \sum_i F_l[k,i] = 0$$

R tuples of three matrices $(D^r, E^r; T^r)$ which represents the synchronizations. In D^r we find the transitions due to synchronization r on the master automaton. It is assumed that the synchronizations always have an effect on the master (i.e. its transition is not a loop). All matrices are normalized, i.e. for all k we have:

$$\begin{array}{lll} D^r[k,i] \geqslant 0 \text{ if } i \neq k & \text{and} & \sum_i D^r[k,i] = 0 \\ E^r[k,i] \geqslant 0 \text{ if } i \neq k & \text{and} & \sum_i E^r[k,i] = 0 \\ T^r[k,i] \geqslant 0 \text{ if } i \neq k & \text{and} & \sum_i T^r[k,i] = 0 \end{array}$$

The effect of synchronization r on the slave (i.e. automaton $sl(r)$) is specified by matrix E^r. The synchronization may have no effect on the slave when it is in state k (i.e $E^r[k,k]$ is zero). It is said that the synchronization r fails during the second step. The synchronized transition takes place on the master but there is no effect on the slave and the synchronization is stopped at this step. Thus, the relay do not synchronize. For instance, in Fig. 1, the slave does not follow the synchronization when it is in state A and it performs a loop.
Otherwise, row k of matrix E^r gives the transition probability out of state k for the slave.

And the synchronization tries now to trigger a transition of the automaton $rl(r)$. Due to the normalization, we have:

$$E^r[k,k] = -1 \quad or \quad E^r[k,k] = 0$$

Similarly, the synchronization may have an effect on the relay (a real transition and a probability in matrix T^r) or it may fail (it is represented by a loop in T^r). The matrix is suitably normalized:

$$T^r[k,k] = -1 \quad or \quad T^r[k,k] = 0$$

Remark 1

- *We note $\odot$ the product vector-matrix.*
- *To keep the notation as clear as possible, we use in the following the indices i, j, k and m for states, l for an automaton, r for a synchronization.*
- *Finally, we denote by $((k_1, k_2, \cdots, k_n)\Diamond(list\ (automaton, state)))$ the state where all automata are in the state defined by $(k_1, k_2, \cdots, k_n)$, except the ones in the list. So, $((k_1, k_2, \cdots, k_n)\Diamond((l,i)))$ represents the state where for all m, automaton m is in state k_m except automaton l which is in state i.*

Theorem 1 *If there exists a solution $(g_l, \Gamma_r, \Omega_r)_{l,r}$ to the fixed point system*

$$\left\{\begin{array}{l} g_l \odot \left[F_l + \sum_{r=1}^{R}\left(D^r 1_{msr(r)=l} + \Gamma_r E^r 1_{sl(r)=l} + \Gamma_r \Omega_r T^r 1_{rl(r)=l}\right)\right] = 0 \\ \Gamma_r\ g_l = g_l \odot \overline{D^r} \quad if \quad msr(r) = l \\ \Omega_r\ g_l = g_l \odot \overline{E^r} \quad if \quad sl(r) = l \end{array}\right. \tag{1}$$

where

- $\overline{D^r} = D^r - diag(D^r)$
- $\overline{E^r} = E^r - diag(E^r)$
- $\Gamma_r, \Omega_r \in \mathcal{R}^+$
- *g_l is a distribution of probability on the state space X_l*

Then, the steady-state distribution has a product form solution.

$$Pr((X_1, X_2, \cdots, X_n)) = C \prod_{l=1}^{n} g_l(X_l) \tag{2}$$

and C is a normalization constant.

The proof is based on algebraic manipulation of the global balance equation (see below). For the sake of readability, it is postponed into an appendix. Here, we just explain the various terms which appear in this equation.

$$\begin{aligned}
Pr(\vec{k})(\sum_{l=1}^{n}\sum_{i\neq k_l} F_l[k_l,i] + \sum_{r=1}^{R}\sum_{i\neq k_{msr(r)}} D^r[k_{msr(r)},i]) \quad (3)\\
= \sum_{l=1}^{n}\sum_{i\neq k_l} F_l[i,k_l] Pr(\vec{k}\diamond((l,i)))\\
+ \sum_{r=1}^{R}\sum_{i\neq k_{msr(r)}} D^r[i,k_{msr(r)}] \sum_{j\neq k_{sl(r)}} E^r[(j,k_{sl(r)})] \sum_{m\neq k_{rl(r)}} T^r[m,k_{rl(r)}] \times\\
Pr(\vec{k}\diamond((msr(r),i),(sl(r),j),(rl(r),m)))\\
+ \sum_{r=1}^{R}\sum_{i\neq k_{msr(r)}} D^r[i,k_{msr(r)}] \sum_{j\neq k_{sl(r)}} E^r(j,k_{sl(r)}) 1_{T^r[k_{rl(r)},k_{rl(r)}]=0}\\
Pr(\vec{k}\diamond((msr(r),i),(sl(r),j)))\\
+ \sum_{r=1}^{R}\sum_{i\neq k_{msr(r)}} D^r[i,k_{msr(r)}] 1_{E^r[k_{sl(r)},k_{sl(r)}]=0} Pr(\vec{k}\diamond((msr(r),i)))
\end{aligned}$$

- On the left-hand-side, $F_l[k_l,i]$ is the rate for local transition out of state k_l for automaton l and $D^r[k_l,i]$ is the transition rate of a synchronization which jumps out of state k_l,
- on the right-hand-side, the first term describes local transitions into state k_l,
- the second term is associated to a complete synchronization of the three automata,
- in the third term, we consider a synchronization which fails at the third step (i.e. the relay),
- and finally, the last term describes a synchronization which fails at the second step. The slave refuses the transition.

The equations in theorem 1 are quite complex, but a simple interpretation may be given to all of them. The first equation defines g_l as the invariant distribution of a continuous-time Markov chain which models the automaton in isolation (i.e. $g_l M_l = 0$), with

$$M_l = F_l + \sum_{r=1}^{R}\left(D^r 1_{msr(r)=l} + \Gamma_r E^r 1_{sl(r)=l} + \Gamma_r \Omega_r T^r 1_{rl(r)=l}\right)$$

Clearly, as F_l, D^r, E^r and T^r are generators and Γ_r and Ω_r are positive, matrix M_l is the generator of a continuous-time Markov chain. Of course, this construction does not prove in general that the chain is ergodic. However, if the chain is finite and if the matrix F_l is irreducible, then the matrix M_l is irreducible and the chain of the automaton in isolation is ergodic.

Furthermore, the four terms of the summation have an intuitive interpretation. The first term corresponds to the local transitions. The last three terms represent the effects of the synchronization on the automata involved. The effect on the master are explicitly represented by the transition matrix D^r while the effect on the slave and the relay are represented by the normalized matrices E^r and T^r multiplied by some rates denoted Ω_r and Γ_r. These rates are defined by the last two equations of the fixed point system. Consider the first one:

$$\Gamma_r \; g_l = g_l \odot \overline{D^r}$$

This equation states that Γ_r is the left-eigenvalue associated to the eigenvector g_l for an operator obtained from matrix D^r by zeroing the diagonal elements. The examples presented in the next section show that this equation is a generalization of queueing networks flow equation. Similarly, Ω_r is defined as the eigenvalue of a modified version of E^r. Note that, like in product form queueing network, the existence of these flows (Γ_r, Ω_r) does not imply that the whole network send a Poisson streams of synchronization on automaton l. Similarly, the product form holds even if the underlying Markov chain is not reversible.

Remark however that the assumptions of the theorem are quite restrictive. We have seen before that, for finite automata with an irreducible local transition matrix F_l, vector g_l exists and is unique because chain M_l is ergodic. But the two last equations establish that g_l is also an eigenvector of operators $\overline{D^r}$ and $\overline{E^r}$. Furthermore we also have a fixed point on the eigenvalues Ω_r and Γ_r. Clearly, this part of the system may have no solution. For instance, as Ω_r and Γ_r are positive, matrices $\overline{D^r}$ and $\overline{E^r}$ do not have rows full of zero. Therefore, every state of the master and the slave must be the destination of at least one synchronized transition. We present in the next section some examples where the product form holds.

4. Example

We define some notation for the various matrices used to describe the SAN:

- I: the identity matrix,
- U: the matrix full of 0 except the main upper diagonal which is 1,
- L: the matrix full of 0 except the main lower diagonal which is 1,
- I^0: the identity matrix except the first diagonal element which is 0.

4.1. Gelenbe's networks with customer triggered movement

The concept of Generalized networks (G-networks for short) have been introduced by Gelenbe in [9]. These networks contain customers and signals. In the first papers on this topic, signals were also denoted as negative customers. Signals are not queued in the network. They are sent into a queue, and disappear instantaneously. But before they disappear they may act upon some customers present in the queue. As customers may, at the completion of their service, become signals and be routed into another queue, G-networks exhibit some synchronized transitions which are not modeled by Jackson networks. Usually the signal implies the deletion of at least one customer. These networks have a steady-state product form solution under usual Markovian assumptions. Then the effects have been extended to include the synchronization between three queues : a signal originated from queue i and which arrives into queue j triggers a customer movement into queue k, if queue j is not empty. Gelenbe has proved that these networks still have a product form solution under the same assumptions [10]. For the sake of simplicity, we assume that there is no arrival of signals from the outside. We also restrict ourselves to networks where at the completion of their services, the customers become signals or leave the network.

We consider an infinite state space. Each automaton models the number of positive customers in a queue. The signal are not represented in the states as they vanish instantaneously. The local transitions are the external arrivals (rate λ_l) and the departures to the outside (rate μ_l multiplied by probability d_l). The synchronization describes the departure of a customer on the master (the end of service with rate μ_l and probability

$(1-d_l)$), the departure of a customer on the slave (a customer movement, if there is any), the arrival of a customer on the relay (always accepted).

$$
\begin{array}{l}
F_l = \lambda_l(U - I) + \mu_l d_l(L - I^0) \\
D^r = \mu_l(1 - d_l)(L - I^0) \\
E^r = (L - I^0) \\
T^r = (U - I)
\end{array}
\qquad (4)
$$

After substitution in the system considered in theorem 1, it must be clear that matrix M_l is tridiagonal with constant diagonals. Thus, g_l has a geometric distribution with rate ρ_l:

$$\rho_l = \frac{\lambda_l + \sum_{r=1}^{R} \Omega_r \Gamma_r 1_{rl(r)=l}}{\mu_l + \sum_{r=1}^{R} \Gamma_r 1_{sl(r)=l}}$$

Of course, one must check that for all l, ρ_l is smaller than 1. Because of its geometric distribution, g_l is an eigenvector of operators $\overline{D^r}$ and $\overline{E^r}$. Finally, we obtain:

$$\Omega_r = \rho_{sl(r)} \quad \textit{and} \quad \Gamma_r = \rho_{msr(r)}\mu_{msr(r)}(1 - d_{msr(r)})$$

which is roughly the generalized flow equation which has been found in [10]. This provide a new proof of Gelenbe's theorem.

4.2. Extension

It is worthy to remark that matrices T^r and F_l only appear in one equation. Therefore, it is possible to extend Gelenbe's result in several directions keeping the geometric distribution for g_l and the matrices $\overline{D^r}$ and $\overline{E^r}$ unchanged. Indeed, the last two equations of the fixed point system are still verified for the eigenvector. And this gives two relations between the eigenvalues Γ_r and Ω_r and the rate of the geometric distribution of g_l.

Theorem 2 *Assume that* $\overline{D^r} = \alpha L$ *and* $E^r = L$*, then for every matrices* F_l *and* T^r *which imply a geometric distribution for* g_l *with rate* ρ_l*, the SAN has a product form distribution if the flow equation in* ρ_l *has a solution whose components are smaller than* 1.

For instance, if T^r or F_l combines queue flushing, deletion of a batch of customers and arrival of one customer (with natural representation of these effects on the automata), the distribution g_l is geometric and we have a product form as soon as the ρ_l exist and are smaller than 1.

Another easy extension comes from the representation of customer migration in a Jackson network. Remember that we do not have consider these transitions in the simplified model of Gelenbe's network. We now show how to add them into the model. A customer arrival cannot be represented by matrix E^r because of the eigenvector and eigenvalue constraint (i.e. in this case, the first column of $\overline{E^r}$ is null). Therefore it must be represented by matrix T^r. To represent a customer movement, we add a slave which always follows the master and allows the transfer of customers between the master and the relay. The slave must have at least two states. For instance, $E^r = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$ is a good solution to allow a solution for the eigenvector constraint. This construction allows also some movement with constraints represented by E^r.

Finally, we may represent domino synchronization of larger size. Indeed, the relay for synchronization $r1$ may be the master for a new synchronization $r2$. Assume that the state space of this automaton is the set of integers. Assume that the only transitions on this automaton are the synchronization $r1$ and $r2$. Let lr be the automaton number. If $T^{r1} = (U - I)$ and $D^{r2} = \alpha(L - I^0)$. Then, g_{lr} has a geometric distribution with rate $\frac{\Omega_{r1}\Gamma_{r1}}{\alpha}$ and the eigenvector relation is satisfied. Thus, we have two domino synchronization of three automata which are connected by the relay. The domino effect apply now on five automata with a loop for the relay and some time spent to fire the relay. This is depicted in Fig. 2 where synchonisation 1 is depicted by plain arcs while synchronization 2 is modelled by dotted ones. Automaton 3 is the relay for synchronization 1 and the master for synchronization 2.

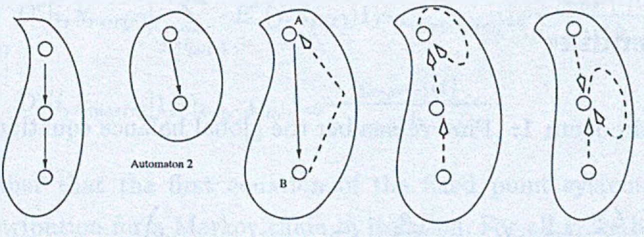

Fig. 2. Two linked Domino Synchronizations
Rys. 2. Dwie połączone Synchronizacje Domino

Furthermore, one must remark that this result still holds if we increase the value of α. Indeed, if $\frac{\Omega_{r1}\Gamma_{r1}}{\alpha} < 1$ for a particular value of α, the relation is still true for a larger value. But the larger the rate, the more likely $r2$ will follow $r1$ very quickly. This gives an approximation to analyze networks where the domino synchronization involve more than three automata. One may also expect from this limit behavior or from the algebraic proof of the theorem that this product form for domino is not limited to synchronization between three automata.

5. Conclusions

Domino synchronization limited to three modules allow SAN and more generally SPA to have a product form steady-state distribution. This result is based on the synchronization description and the algebraic analysis of the global balance. Our result holds even if the underlying Markov chain is not reversible. Similarly, local balances do not hold (at least for Gelenbe's networks which are included in our model). However, Domino synchronization are far less powerful for specification than the usual "Rendez-Vous". For instance, they do not allow the blocking of the master by the slave. We expect that it will be possible to make a perturbation of the automata with events of small probability to transform a "rendez-vous" into a domino. And the product form result will provide an approximation of the solution. More theoretically, the Domino synchronization with product form is much more general than the three automata case we have presented here. It remains to generalize to arbitrary size Domino synchronization and to explain why they allow product form. Finally, one must consider functional transitions.

6. Appendix

Proof of theorem 1: First remember the global balance equation:

$$Pr(\vec{k})\left(\sum_{l=1}^{n}\sum_{i\neq k_l} F_l[k_l,i] + \sum_{r=1}^{R}\sum_{i\neq k_{msr(r)}} D^r[k_{msr(r)},i]\right) \tag{5}$$

$$= \sum_{l=1}^{n}\sum_{i\neq k_l} F_l[i,k_l] Pr(\vec{k}\diamond((l,i)))$$

$$+ \sum_{r=1}^{R}\sum_{i\neq k_{msr(r)}} D^r[i,k_{msr(r)}] \sum_{j\neq k_{sl(r)}} E^r[(j,k_{sl(r)})] \sum_{m\neq k_{rl(r)}} T^r[m,k_{rl(r)}] \times$$

$$Pr(\vec{k}\diamond((msr(r),i),(sl(r),j),(ter(r),m)))$$

$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] \sum_{j \neq k_{sl(r)}} E^r(j, k_{sl(r)}) 1_{T^r[k_{rl(r)}, k_{rl(r)}]=0} Pr(\vec{k} \diamond ((msr(r), i), (sl(r), j)))$$

$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] 1_{E^r[k_{sl(r)}, k_{sl(r)}]=0} Pr(\vec{k} \diamond ((msr(r), i)))$$

Step 1: Remember that F_l and D^r are generators. Thus,

$$\sum_{l=1}^{n} \sum_{i \neq k_l} F_l[k_l, i] + \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[k_{msr(r)}, i] = - \sum_{l=1}^{n} F_l[k_l, k_l] - \sum_{r=1}^{R} D^r[k_{msr(r)}, k_{msr(r)}]$$

After substitution in global balance equation, we simplify the left-hand-side (l.h.s. for short).

Step 2: Then, we divide both sides of equation (5) by $Pr(\vec{k})$ and we assume a product form solution to simplify the ratios of probability.

$$- \sum_{l=1}^{n} F_l[k_l, k_l] - \sum_{r=1}^{R} D^r[k_{msr(r)}, k_{msr(r)}] \tag{6}$$

$$= \sum_{l=1}^{n} \sum_{i \neq k_l} F_l[i, k_l] \frac{g_l(i)}{g_l(k_l)}$$

$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] \sum_{j \neq k_{sl(r)}} E^r[(j, k_{sl(r)})] \sum_{m \neq k_{rl(r)}} T^r[m, k_{rl(r)}] \times$$

$$\frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} \frac{g_{sl(r)}(j)}{g_{sl(r)}(k_{sl(r)})} \frac{g_{rl(r)}(m)}{g_{rl(r)}(k_{rl(r)})}$$

$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] \sum_{j \neq k_{sl(r)}} E^r(j, k_{sl(r)}) 1_{T^r[k_{rl(r)}, k_{rl(r)}]=0} \frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} \frac{g_{sl(r)}(j)}{g_{sl(r)}(k_{sl(r)})}$$

$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] 1_{E^r[k_{sl(r)}, k_{sl(r)}]=0} \frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})}$$

Step 3: Remember that the first equation of the fixed point system defines g_l as a steady-state distribution for a Markov chain in isolation. For all k_l we have:

$$\sum_i g_l(i) \left[F_l[i, k_l] + \sum_{r=1}^{R} \left(D^r[i, k_l] 1_{msr(r)=l} + \Gamma_r E^r[i, k_l] 1_{sl(r)=l} + \Gamma_r \Omega_r T^r[i, k_l] 1_{ter(r)=l} \right) \right] = 0 \tag{7}$$

Step 4: After some algebraic manipulation of equation (7), we obtain a new relation for the l.h.s. of equation (6):

$$-F_l[k_l, k_l] - \sum_{r=1}^{R} D^r[k_l, k_l] 1_{msr(r)=l} \tag{8}$$
$$= \sum_{i \neq k_l} F_l[i, k_l] \frac{g_l(i)}{g_l(k_l)} + \sum_{r=1}^{R} \sum_{i \neq k_l} D^r[k_l, k_l] 1_{msr(r)=l} \frac{g_l(i)}{g_l(k_l)}$$
$$+ \sum_{r=1}^{R} \sum_{i} \Gamma_r E^r[i, k_l] 1_{sl(r)=l} \frac{g_l(i)}{g_l(k_l)} + \sum_{r=1}^{R} \sum_{i} \Gamma_r \Omega_r T^r[i, k_l] 1_{ter(r)=l} \frac{g_l(i)}{g_l(k_l)}$$

Step 5: Combining equation (8) and equation (6) we get after some cancellation of terms:

$$\sum_{r=1}^{R} \sum_{i \neq k_l} D^r[k_l, k_l] 1_{msr(r)=l} \frac{g_l(i)}{g_l(k_l)} + \sum_{r=1}^{R} \sum_{i} \frac{g_l(i)}{g_l(k_l)} \Gamma_r \left(E^r[i, k_l] 1_{sl(r)=l} + \Omega_r T^r[i, k_l] 1_{ter(r)=l} \right)$$
$$= \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] \sum_{j \neq k_{sl(r)}} E^r[(j, k_{sl(r)})] \sum_{m \neq k_{rl(r)}} T^r[m, k_{rl(r)}] \times$$
$$\frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} \frac{g_{sl(r)}(j)}{g_{sl(r)}(k_{sl(r)})} \frac{g_{rl(r)}(m)}{g_{rl(r)}(k_{rl(r)})}$$
$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] \sum_{j \neq k_{sl(r)}} E^r(j, k_{sl(r)}) 1_{T^r[k_{rl(r)}, k_{rl(r)}]=0} \frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} \frac{g_{sl(r)}(j)}{g_{sl(r)}(k_{sl(r)})}$$
$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] 1_{E^r[k_{sl(r)}, k_{sl(r)}]=0} \frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} \tag{9}$$

Step 6: Rearrange the summation on i in the second part if the l.h.s.:

$$\sum_{r=1}^{R} \sum_{i \neq k_l} D^r[k_l, k_l] 1_{msr(r)=l} \frac{g_l(i)}{g_l(k_l)} + \sum_{r=1}^{R} \sum_{i \neq k_l} \frac{g_l(i)}{g_l(k_l)} \Gamma_r \left(E^r[i, k_l] 1_{sl(r)=l} + \Omega_r T^r[i, k_l] 1_{ter(r)=l} \right)$$
$$+ \sum_{r=1}^{R} \Gamma_r \left(E^r[k_l, k_l] 1_{sl(r)=l} + \Omega_r T^r[k_l, k_l] 1_{ter(r)=l} \right)$$
$$= \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] \sum_{j \neq k_{sl(r)}} E^r[(j, k_{sl(r)})] \sum_{m \neq k_{rl(r)}} T^r[m, k_{rl(r)}] \times$$
$$\frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} \frac{g_{sl(r)}(j)}{g_{sl(r)}(k_{sl(r)})} \frac{g_{rl(r)}(m)}{g_{rl(r)}(k_{rl(r)})}$$
$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] \sum_{j \neq k_{sl(r)}} E^r(j, k_{sl(r)}) 1_{T^r[k_{rl(r)}, k_{rl(r)}]=0} \frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} \frac{g_{sl(r)}(j)}{g_{sl(r)}(k_{sl(r)})}$$
$$+ \sum_{r=1}^{R} \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] 1_{E^r[k_{sl(r)}, k_{sl(r)}]=0} \frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} \tag{10}$$

Step 7: Remember the second part of the fixed point system:

$$\Gamma_r \; g_l = g_l \odot (D^r - diag(D^r)) \quad if \quad msr(r) = l$$

After some algebraic manipulation,

$$\Gamma_r = \sum_{i \neq k_{msr(r)}} D^r[i, k_{msr(r)}] \frac{g_{msr(r)}(i)}{g_{msr(r)}(k_{msr(r)})} = \sum_{i \neq k_l} D^r[k_l, k_l] 1_{msr(r)=l} \frac{g_l(i)}{g_l(k_l)} \tag{11}$$

Similarly we have, according to the assumption of the theorem:

$$\Omega_r \; g_l = g_l \odot \; (E^r - diag(E^r)) \quad if \quad sl(r) = l$$

Thus,

$$\Omega_r = \sum_{j \neq k_{sl(r)}} E^r[(j, k_{sl(r)})] \frac{g_{sl(r)}(j)}{g_{sl(r)}(k_{sl(r)})} \tag{12}$$

Step 8: Substitute relations (11) and (12) into equation (10):

$$\begin{aligned}
&\sum_{r=1}^{R} \Gamma_r + \sum_{r=1}^{R} \Gamma_r \Omega_r + \sum_{r=1}^{R} \Gamma_r E^r[k_{sl(r)}, k_{sl(r)}] \\
&+ \sum_{r=1}^{R} \Gamma_r \Omega_r T^r[k_{rl(r)}, k_{rl(r)}] + \sum_{r=1}^{R} \sum_{i \neq k_l} \Gamma_r \Omega_r \frac{g_l(i)}{g_l(k_l)} T^r[i, k_l] 1_{ter(r)=l} \\
&= \sum_{r=1}^{R} \Gamma_r \Omega_r \sum_{m \neq k_{rl(r)}} T^r[m, k_{rl(r)}] \frac{g_{rl(r)}(m)}{g_{rl(r)}(k_{rl(r)})} \\
&+ \; \sum_{r=1}^{R} \Gamma_r \Omega_r 1_{T^r[k_{rl(r)}, k_{rl(r)}]=0} \\
&+ \; \sum_{r=1}^{R} \Gamma_r 1_{E^r[k_{sl(r)}, k_{sl(r)}]=0}
\end{aligned} \tag{13}$$

Step 9: Remember that E^r and T^r are normalized such that their diagonal elements are 0 or -1. Thus, for all i we have:

$$1_{E^r[i,i]=0} - E^r[i, i] = 1 \quad and \quad 1_{T^r[i,i]=0} - T^r[i, i] = 1$$

Step 10: Thus after some cancellation of terms in equation (13), we get

$$\begin{aligned}
&\sum_{r=1}^{R} \sum_{i \neq k_l} \Gamma_r \Omega_r \frac{g_l(i)}{g_l(k_l)} T^r[i, k_l] 1_{ter(r)=l} \\
&= \; \sum_{r=1}^{R} \Gamma_r \Omega_r \sum_{m \neq k_{rl(r)}} T^r[m, k_{rl(r)}] \frac{g_{rl(r)}(m)}{g_{rl(r)}(k_{rl(r)})}
\end{aligned} \tag{14}$$

which is trivially true. This concludes the proof.

REFERENCES

1. Badrah A., Fourneau J. M., Quessette F.: Performance Evaluation of Multistage Interconnection Networks with blocking. Proc. of the 11th European Simulation MultiConf., Istambul, 1997.
2. Boujdaine F., Fourneau J. M., Mikou N.: Product Form Solution for Stochastic Automata Networks with synchronization. 5th Process Algebra and Performance Modeling Workshop, 1997, Twente, Netherlands.
3. Boucherie R.: A Characterization of independence for competing Markov chains with applications to stochastic Petri nets. Inria Report RR-1880, Sophia Antipolis, France.
4. Donnatelli S.: Superposed stochastic automata : a class of stochastic Petri nets with parallel solution and distributed state space. Performance Evaluation, V18, 1993, pp. 21-36.
5. Fernandes P., Plateau B., Stewart W. J.: Efficient Descriptor-Vector Multiplications in Stochastic Automata Networks. JACM, 1998, pp. 381-414.
6. Fernandes P., Plateau B., Stewart W. J.: Optimizing tensor product computations in SAN. RAIRO, V 32, N3, pp. 325-351, 1998.
7. Fourneau J. M., Kloul L., Quessette F., Pekergin N., Vèque V.: Modeling Buffer Admission Mechanisms using Stochastic Automata Networks. Annales des Télécoms, Vol. 49, N 5-6, June 1994, pp. 337-349.
8. Fourneau J. M., Quessette F.: Graphs and Stochastic Automata Networks. Proc. of the 2nd International Workshop on the Numerical Solution of Markov Chains, Raleigh, USA, 1995.
9. Gelenbe E.: Product form queueing networks with negative and positive customers. Journal of Applied Probability, V 28, pp. 656–663, 1991.
10. Gelenbe E.: G-networks with triggered customer movement. Journal of Applied Probability, V 30, pp. 742–748, 1993.
11. Harrison G.: Reversed processes in Markovian process algebra. ESM 2001, pp 783-790, Prague, 2001.
12. Hillston J.: A compositional approach to Performance Modeling, PHD Thesis. University of Edinburgh, 1994.
13. Kloul L., Hillston J.: An efficient Kronecker representation for PEPA models. PAPM 2001, Germany, 2001.

14. Plateau B. ; On the Stochastic Structure of Parallelism and Synchronization Models for Distributed Algorithms. Proc. ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, Austin, Texas, August 1985.
15. Plateau B., Fourneau J.M., Lee K.H. PEPS: A Package for Solving Complex Markov Models of Parallel Systems. Proceedings of the 4th Int. Conf. on Modeling Techniques and Tools for Computer Performance Evaluation, Majorca, Spain. Sept. 1988.
16. Plateau B., Stewart W. J.: Stochastic Automata Networks: Product Forms and Iterative Solutions. Inria Report 2939.
17. Lazar A., Robertazzi T.: Markovian Petri Net Protocols with Product Form Solution. Performance Evaluation, 1991, V12, pp. 66-77.
18. Sereno M.: Towards a Product Form Solution for Stochastic Process Algebra. 3rd Process Algebra and Performance Modeling Workshop, 1995, also in The Computer Journal, V38, N7, pp. 622-632.
19. Stewart W. J., Atif K., Plateau B.: The numerical solution of Stochastic Automata Networks. European Journal of Operation Research, V86, N3, pp. 503-525,1995.
20. Vèque V., Ben-Othman J.: MRAP : Multiservice Resource Allocation Policy for Wireless ATM Networks. Int. Journal in Computer networks and ISDN systems, V29, pp. 187-200, 1998.

Recenzent: Prof. dr hab. inż. Tadeusz Czachórski

Wpłynęło do Redakcji 30 stycznia 2002 r.

Streszczenie

Łańcuchy Markowa są często wykorzystywane w modelowaniu systemów informatycznych, ponieważ pozwalają na opis synchronizacji pomiędzy wykonywanymi zadaniami. Jedną z metod tworzenia łańcuchów Markowa dla modeli o bardzo dużej liczbie stanów, w szczególności modeli systemów, w których występuje równolegle wykonywanie zadań, jest metoda Automatów Stochastycznych, podająca algorytm tworzenia macierzy przejść dla łańcucha Markowa, odpowiadającego całości badanego systemu (zwanego Siecią Automatów Stochastycznych) na podstawie modeli cząstkowych, zwanych Automatami Stochastycznymi. W artykule przedstawiony jest nowy rodzaj synchronizacji, który pozwala

przedstawić w formie iloczynowej rozwiązanie modelu Sieci Automatów Stochastycznych. W przeciwieństwie do wcześniejszych modeli w postaci Sieci Automatów Stochastycznych o rozwiązaniu produktowym ta propozycja umożliwia synchronizację trzech automatów. Ten rodzaj synchronizacji nie reprezentuje zwykłego typu "Rendez-Vous". Jest on bowiem reprezentowany przez uporządkowaną listę tranzycji, z których każda może być aktywowana. Po zakończeniu reaktywacji tranzycje są zachowywane. Ta klasa Sieci Automatów Stochastycznych stanowi uogólnienie sieci Gelenbego z odpalanym ruchem klientów. Uzyskane wyniki prowadzą do aproksymacji opartej na Sieciach Automatów Stochastycznych o rozwiązaniu iloczynowym, w których synchronizacje są wprowadzane na zasadzie listy tranzycji o określonym rozmiarze.