Operational GSPN Semantics of MPA

Marco Bernardo  Lorenzo Donatiello  Roberto Gorrieri

Technical Report UBLCS-94-12
May 1994

Laboratory for Computer Science
University of Bologna
Piazza di Porta S. Donato, 5
40127 Bologna (Italy)
The University of Bologna Laboratory for Computer Science Research Technical Reports are available via anonymous FTP from the area 
ftp.cs.unibo.it:/pub/TR/UBLCS in compressed PostScript format. Abstracts are available from the same host in the directory /pub/TR/UBLCS/ABSTRACTS in plain text format. All local authors can be reached via e-mail at the address last-name@cs.unibo.it.

**UBLCS Technical Report Series**

93-7 Distributed Programming with Logic Tuple Spaces, P. Ciancarini, April 1993.
93-8 Coordinating Rule-Based Software Processes with ESP, P. Ciancarini, April 1993.
93-14 Guard Compilation in Logic Shared Dataspace Languages, M. Gaspari, June 1993.
94-6 Exploring the Coordination Space with LO, S. Castellani, P. Ciancarini, April 1994.
94-7 Comparative Semantics of LO, S. Castellani, P. Ciancarini, April 1994.
Operational GSPN Semantics of MPA

Marco Bernardo 1       Lorenzo Donatiello 1       Roberto Gorrieri 1

Technical Report UBLCS-94-12
May 1994

Abstract

In this technical report we define an operational net semantics for the stochastic process algebra MPA based on generalized stochastic Petri nets.
As a consequence of the definition of such a semantics, we present an interesting modeling technique for concurrent systems which is based on MPA.

1. Dipartimento di Matematica, Università di Bologna, Piazza di Porta S. Donato 5, 40127 Bologna, Italy.
1 Introduction

Process algebras constitute one of the most important tools for modeling and analyzing concurrent systems, i.e. systems which exhibit parallelism and synchronization between their components. In the 90’s a remarkable research effort has been made to tackle and solve an important problem of the process algebras: the impossibility of expressing the temporal aspect of the behavior of concurrent systems.

One of the main solutions to this problem consists of allowing also the action durations to be expressed in a probabilistic way by means of random variables. This solution has led to the birth of stochastic process algebras like MTIPP, proposed by N. Götz-U. Herzog-M. Rettelbach [7][8], PEPA, proposed by J. Hillston [9][6], and MPA, proposed by M. Bernardo-L. Donatiello-R. Gorrieri [2][3].

In this work we shall consider the stochastic process algebra MPA. Each MPA term has two semantic interpretations: $I[E]$, which is an automaton representing the operational interleaving semantics of $E$ hence the functional aspect of the behavior of the concurrent system modeled by $E$, and $M[E]$, which is an automaton representing the markovian semantics of $E$ hence the temporal aspect of the behavior of the concurrent system modeled by $E$.

If we restrict ourselves to the operational interleaving semantics of MPA terms, we see that such a semantics is not a real concurrent semantics; in fact, it is not possible to distinguish by means of sequential automata between parallelism and causal dependence because sequential automata simulate a parallel execution by means of alternative executions obtained by interleaving the actions forming the parallel execution itself. From this point of view, a more satisfying semantics can be obtained if we consider Petri nets [21] instead of automata because Petri nets are extensions of automata in which the notion of state is described in a distributed way.

The purpose of this work is to define a net semantics for MPA. By resorting to the structured operational semantics approach it is possible to define an operational net semantics for a process algebra. The idea, due to P. Degano-R. De Nicola-U. Montanari and then refined by the same authors [4] and by E. Olderog [20], consists of associating with each term of the process algebra a place/transition net such that:

- Net places correspond to the sequential subterms of the term at hand and of the terms obtainable from it by action execution.
- Net transitions are defined by induction on the syntactical structure of the sets of sequential subterms following the structured operational semantics rules of the process algebra.
- Net markings “correspond” to the above terms.

Here we want to extend such an idea to MPA; since MPA is a stochastic process algebra, its terms will be translated into stochastic Petri nets.

This technical report is organized as follows. In Section 2 some notions concerning place/transition nets are recalled. In Section 3 stochastic Petri nets are introduced and briefly explained. In Section 4 generalized stochastic Petri nets are introduced and briefly explained. In Section 5 an operational net semantics based on generalized stochastic Petri nets is defined for MPA; the goodness of this semantics is then assessed by verifying whether it satisfies the retrievability principle and the concurrency principle. In Section 6 some concluding remarks are reported.  

---

2. In this technical report we shall exhibit neither the definition of the syntax, the operational interleaving
In this section we recall some notions concerning place/transition nets; they are taken from
the textbook of W. Reisig [22], from a paper of T. Murata [18] and from the textbook of
E.-R. Olderog [20].

Place/transition nets constitute a particular class of Petri nets which are useful for mod-
eling systems where an organization is necessary for taking into account modifications of the
number and the distribution of objects or resources in order to keep such modifications within
specified bounds.

Definition 2.1 A place/transition net (PTN) is a tuple
\[(S, T, I, O, M_0)\]
such that:
- \(S\) is a set whose elements are called places;
- \(T\) is a set whose elements are called transitions;
- \(S \cap T = \emptyset\);
- \(I : T \rightarrow (S \rightarrow \mathbb{N})\) is called input function;
- \(O : T \rightarrow (S \rightarrow \mathbb{N})\) is called output function;
- \(M_0 : S \rightarrow \mathbb{N}\) is called initial marking.

Let \(N = (S, T, I, O, M_0)\) be a PTN; its graphic representation is as follows:
- Places are represented as circles.
- Transitions are represented as boxes.
- The input function is represented by means of arcs between pairs composed of a place
  and a transition. The arc \((s, t) \in S \times T\) is labeled by \((I(t))(s)\) and it is not drawn if
  \((I(t))(s) = 0\).
- The output function is represented by means of arcs between pairs composed of a
  transition and a place. The arc \((t, s) \in T \times S\) is labeled by \((O(t))(s)\) and it is not drawn
  if \((O(t))(s) = 0\).
- The initial marking is represented by putting \(M_0(s)\) tokens into each place \(s \in S\).

Notice that \(M_0\) and \(I(t)\) and \(O(t)\) are multisets, hence they can be regarded as vectors and
vectorial operations can be applied to them.

Definition 2.2 Let \(N = (S, T, I, O, M_0)\) be a PTN:
- \(M\) is a marking of \(N\) iff
  \[M : S \rightarrow \mathbb{N}\]
- \(t \in T\) is enabled by marking \(M\) iff
  \[\forall s \in S. M(s) \geq (I(t))(s);\]
we denote with \(E(M)\) the set of transitions enabled by marking \(M\) and we write
  \[M \mid t \rightarrow M'\]

semantics and the markovian semantics of MPA terms, nor notions concerning automata, nor properties of
exponentially distributed random variables we shall use in the following, nor the meaning of the shorthand
";"; before proceeding, the reader is then referred to [3].
to indicate that firing $t \in E(M)$ produces the new marking

$$M' = M - I(t) + O(t).$$

We assume that the firing of a transition is an atomic operation, i.e. we assume that upon a transition firing tokens are withdrawn from transition input places and deposited into transition output places in a single indivisible action.

- The reachability graph of $N$ is the automaton

$$RG[N] = (R(M_0), T, [], M_0)$$

where $R(M_0)$, called the reachability set of $N$, is the least subset of $\mathbb{N}^S$ such that

$$M_0 \in R(M_0)$$

$$M \in R(M_0) \quad M[t] M' \quad M' \in R(M_0)$$

PTNs can be characterized by means of properties which can depend on the initial marking or not; among this properties we quote boundedness and liveness, which are very important for analyzing the partial correctness and the total correctness, respectively, of the concurrent systems modeled by means of PTNs.

**Definition 2.3** Let $N = (S, T, I, O, M_0)$ be a PTN:

- $N$ is ordinary iff

$$\forall t \in T. \forall s \in S. (I(t))(s) \leq 1 \wedge (O(t))(s) \leq 1.$$

- $N$ is $k$-bounded, $k \in \mathbb{N}$ iff

$$\forall M \in R(M_0). \forall s \in S. M(s) \leq k.$$

- $N$ is safe iff $N$ is $k$-bounded with $k = 1$.

- $N$ is live iff

$$\forall t \in T. \forall M \in R(M_0). \exists M' \in R(M). t \in E(M').$$

We conclude this section with the two following definitions.

**Definition 2.4** The PTN $N_1 = (S_1, T_1, I_1, O_1, M_{01})$ is isomorphic to the PTN $N_2 = (S_2, T_2, I_2, O_2, M_{02})$ iff there exists a bijection

$$\beta : S_1 \cup T_1 \rightarrow S_2 \cup T_2$$

such that:

- $\beta(S_1) = S_2$ and $\beta(T_1) = T_2$;

- $\forall s \in S_1, \forall t \in T_1. (I_1(t))(s) = (I_2(\beta(t)))(\beta(s)) \wedge (O_1(t))(s) = (O_2(\beta(t)))(\beta(s))$;

- $\forall s \in S_1. M_{01}(s) = M_{02}(\beta(s))$. 

UBLCS-94-12
Definition 2.5 The ordinary and safe PTN $N_1 = (S_1, T_1, I_1, O_1, M_{01})$ is strongly bisimilar to the ordinary and safe PTN $N_2 = (S_2, T_2, I_2, O_2, M_{02})$ iff there exists a relation

$$B \subseteq S_1 \times S_2$$

such that:

- $M_{01} \not\sim M_{02}$;
- $\forall M_1, M'_1 \in R(M_{01}). \forall M_2 \in R(M_{02}). \forall t_1 \in T_1. (M_1 \not\sim M_1' \wedge M_1[t_1] M'_1 \implies \exists M'_2 \in R(M_{02}). \exists t_2 \in T_2. M_2[t_2] M'_2)$$;
- $\forall M_2, M'_2 \in R(M_{02}). \forall M_1 \in R(M_{01}). \forall t_2 \in T_2. (M_1 \not\sim M_2 \wedge M_2[t_2] M'_2 \implies \exists M'_1 \in R(M_{01}). \exists t_1 \in T_1. M_2[t_2] M'_1) \wedge t_1 \not\sim t_2$.

where:

- $M_1 \not\sim M_2$ means that $B \cap (M_1 \times M_2)$ is a bijection;
- $t_1 \not\sim t_2$ means that $t_1 = t_2 \wedge \{s \in S_1 \mid (I_1(t_1))(s) = 1\} \not\sim \{s \in S_2 \mid (I_2(t_2))(s) = 1\} \wedge \{s \in S_1 \mid (O_1(t_1))(s) = 1\} \not\sim \{s \in S_2 \mid (O_2(t_2))(s) = 1\}.$

3 Stochastic Petri nets

Stochastic Petri nets have been proposed by researchers active in the field of the applied stochastic modeling to develop a tool which integrates formal description, correctness verification and performance evaluation. As far as performance evaluation is concerned, the purpose was that of obtaining an equivalence between models based on stochastic Petri nets and models based on homogeneous continuous time Markov chains (HCTMCs); this is the reason for which stochastic Petri nets are essentially PTNs whose transitions have durations associated with them expressed by means of exponentially distributed random variables with parameters possibly dependent of the current marking (the duration of a transition is intended to be the time that has to elapse before the transition fires, the transition firing remaining an atomic operation).

An overview of stochastic Petri nets is reported in a paper of M. A. Marsan [10]; in this section we shall be concerned with a particular class of stochastic Petri nets (denoted with SPN) proposed by M. K. Molloy [16][17] and S. Natkin [19][5].

Definition 3.1 A stochastic Petri net (SPN) is a tuple

$$(S, T, I, O, M_0, \Lambda)$$

such that:

- $(S, T, I, O, M_0)$ is a PTN;
- $\Lambda : T \rightarrow (\mathbb{N} \rightarrow \mathbb{R}^+)$ where $\Lambda(t)$ is called the parameter of $t$. 

Remark 3.2 In the definition of SPN the parameters of the exponential distributions associated with transitions can depend on the current marking. This dependence permits to obtain very concise models in which a single transition represents a subsystem whose aggregated delay depends on the population inside the system, but on the contrary it makes necessary
to check if each marking-dependent parameter is well defined (i.e., positive for each marking which enables the corresponding transition).

From the graphic point of view, the only difference between SPNs and PTNs is that SPN transitions are labeled by the parameter of the corresponding exponential distribution; such a parameter is either a number or a function depending on whether it depends on the current marking or not. We also write

$$M \mid t_i (A(t))(M) \rangle M'$$

to indicate that firing transition $t \in E(M)$ produces the new marking $M'$.

Since a marking can enable several transitions, the firing rule for SPNs must include an execution policy which determines the enabled transition to be fired. The execution policy usually adopted is the race policy, which states that the transition to be fired is the transition which has sampled from its distribution the minimal duration among the transitions enabled by the current marking. There are three variants of the race policy:

- The first variant, called race with resampling, states that upon each marking change all the transitions enabled by the new marking resample a new duration. Supposing that with each transition is associated an age variable which takes into account the work done by the activity corresponding to the transition, race with resampling states that upon each marking change all the age variables are reset to zero (and that the age variables associated with the enabled transitions are then incremented at a constant rate until one of them equals the duration sampled by the corresponding transition).

Due to a well known property of exponentially distributed random variables, we have that:

- The mean sojourn time in a marking $M$ is given by

$$\frac{1}{\sum_{t_i \in E(M)} (A(t_i))(M)}.$$

- The firing probability of transition $t_j \in E(M)$ is given by

$$\frac{(A(t_j))(M)}{\sum_{t_i \in E(M)} (A(t_i))(M)}.$$

- The transition rate from marking $M$ to marking $M'$ is given by

$$\sum_{t_i \in E(M) \land M \not\vdash (A(t_i))(M) \mid M'} (A(t_i))(M).$$

- The second variant, called race with enabling memory, states that upon each marking change only the transitions enabled by the new marking but not by the old marking resample a new duration. In terms of age variables, race with enabling memory states that all the age variables are reset to zero except for those associated with transitions which remain enabled after the marking change, so that the age variable associated with a transition takes into account the work done by the corresponding activity since the last time that the transition has been enabled provided that it has not been disabled after that moment.
The third variant, called \textit{race with age memory}, states that upon each marking change only the transitions enabled by the new marking and never enabled after their last firing resample a new duration. In terms of age variables, race with age memory states that upon each marking change only the age variable of the transition whose firing caused the marking change is reset to zero, so that the age variable associated with a transition takes into account the work done by the corresponding activity since the first time that the transition has been enabled after its last firing regardless of whether it has been disabled or not after that moment.

These three variants of the race policy are equivalent in the context of the SPNs due to the memoryless property of the exponential distributions associated with SPN transitions; however, race with age memory is to be preferred, except for some particular cases, because it is:

- More general, as with arbitrary distributions the remaining time for the firing of a transition has distribution other than that of the total firing time of the same transition (hence other than that from which the sampling would be made).
- More useful, as it permits to actually model (by means of transitions) activities which have a duration, which can be started and temporarily suspended during the dynamics of the model and that, when completed, cause a state change described by the corresponding marking change.

Notice that, the time being expressed through continuous distributions, the probability that two transitions are simultaneously fired is zero. As a consequence, a model based on SPNs evolves by firing a transition at a time.

Let $N = \langle S, T, I, O, M_0, A \rangle$ be a SPN:

- The reachability set of $N$ is the same as that of the underlying PTN because the probability density functions associated with the transitions have unlimited support (in other words, all the transitions enabled by a marking have a nonzero firing probability in that marking because they sample their durations in $\mathbb{R}^+$), hence $N$ and the underlying PTN have the same structural properties. To be more precise, this is true provided that the marking-dependent parameters of the exponential distributions associated with the transitions are well defined, because if this is not the case then a transition enabled by a marking could have a zero firing probability in that marking.
- The reachability graph of $N$ is defined as the automaton
  \begin{equation*}
  \mathcal{RG}[N] = (R(M_0), T \times \mathbb{R}^+, \emptyset, M_0)
  \end{equation*}
  and it is isomorphic to the reachability graph of the PTN underlying $N$.
- The HCTMC associated with $N$ is defined as the automaton
  \begin{equation*}
  \mathcal{M}[N] = (R(M_0), \mathbb{R}^+, \rightarrow, P)
  \end{equation*}
  where:
  - $\rightarrow$ is the least subset of $R(M_0) \times \mathbb{R}^+ \times R(M_0)$ such that
    \begin{align*}
    M_1 \left[t_1, (A(t_1))(M_1)\right] \quad &\rightarrow \quad M_2 \quad \cdots \quad M_1 \left[t_n, (A(t_n))(M_1)\right] \quad \rightarrow \quad M_2 \\
    \sum_{i=1}^{n} (A(t_i))(M_1) &\rightarrow M_1 \rightarrow M_2
    \end{align*}
4 Generalized stochastic Petri nets

The main weakness of models based on SPNs lies in the high complexity of their analysis, which is due to both the number of reachable markings and to the presence of activities faster or slower than those important for the performance evaluation. Since from the point of view of the correctness of the models such activities cannot be neglected and since there can exist in the models transitions corresponding to logical aspects of the modeled systems with which no reasonable timing can be associated, generalized stochastic Petri nets have been proposed by M. A. Marsan-G. Balbo-G. Conte [15] and then revisited by the same authors and others [13][11][12][14]; they constitute a generalization of the SPNs because they permit to specify also transitions whose duration is zero, inhibitor arcs and priority levels. Generalized stochastic Petri nets then admit the simultaneous presence of timed transitions and immediate transitions; since immediate transitions have duration zero, they take precedence over timed transitions. In the next section we shall use generalized stochastic Petri nets for defining the operational net semantics of MPA terms.

Definition 4.1 A generalized stochastic Petri net (GSPN) is a tuple

\[(S, T, I, O, H, M_0, L, W)\]

such that:

- \((S, T, I, O, M_0)\) is a PTN;
- \(H : T \rightarrow (S \rightarrow \mathbb{N})\) is called inhibition function;
- \(L : T \rightarrow \mathbb{N}\) is called priority function and is such that:
  - \(L(t) = 0\) if \(t\) is timed;
  - \(L(t) \in \mathbb{N}^+\) if \(t\) is immediate;
- \(W : T \rightarrow (\mathbb{N}^\ast \rightarrow \mathbb{R}^+)\) such that:
  - \(W(t)\) is called the parameter of the exponential distribution associated with \(t\) if \(L(t) = 0\);
  - \(W(t)\) is called the weight of \(t\) if \(L(t) \in \mathbb{N}^+\).

Remark 4.2 The definition of GSPN given in [13] is more restrictive than that given above because it requires that the subnets composed of immediate transitions are without structural confusion and that the marking-dependent weights do not cause stochastic confusion, where confusion stands for a situation in which the resolution of a conflict among transitions depends on the order in which other nonconflicting transitions have been previously fired.
(in case of confusion the effect is that a given marking is reached with probabilities which vary depending on the firing order). Here we ignore such requirements; however, there are some sufficient conditions which guarantee the absence of confusion and that need a structural analysis for their verification, i.e. an analysis fully independent of the current marking hence automatizable (such a structural analysis, as well as the structural analysis necessary for assigning weights to immediate transitions, is already present in some software tools for constructing and solving models based on GSPNs and is described in [13]).

From the graphic point of view the differences between GSPNs and SPNs are the following:
- The priority function is partly represented by using boxes for timed transitions and bars for immediate transitions.
- The inhibition function is represented by means of arcs lying between pairs composed of a place and a transition which are drawn as dashed lines beginning at the place and terminating at the transition with a circle. The arc \((s, t) \in S \times T\) is labeled by \((H(t))(s)\) and is not drawn if \((H(t))(s) = 0\).

**Definition 4.3** Let \(N = (S, T, I, O, H, M_0, I, W)\) be a GSPN:
- \(t \in T\) is **enabled** by marking \(M\) iff
  \[
  \forall s \in S. M(s) \geq (I(t))(s) \land ((H(t))(s) > 0 \implies M(s) < (H(t))(s));
  \]
  we write
  \[
  M \left[ t, (W(t))(M) \right] M'
  \]
  to indicate that firing \(t \in E(M)\) produces the new marking \(M'\).
- A marking \(M\) is **tangible** iff
  \[
  \forall t \in E(M). L(t) = 0.
  \]
- A marking \(M\) is **vanishing** iff
  \[
  \exists t \in E(M). L(t) \in \mathbb{N}^+.
  \]
- We denote with \(E_l(M)\) the set of transitions enabled by marking \(M\) whose priority level is \(l\).

The firing rule for the GSPNs is the following:
- If the current marking \(M\) is tangible then one of the variants of the race policy defined for the SPNs is used.
- If the current marking \(M\) is vanishing then:
  - The set \(E_{l_{\max}}(M)\) is computed, where \(l_{\max} = \max_{E_l(M) \neq \emptyset} l\).
  - The transition to be fired is chosen in \(E_{l_{\max}}(M)\) after associating with each \(t_j \in E_{l_{\max}}(M)\) a firing probability given by
    \[
    \frac{(W(t_j))(M)}{\sum_{t_i \in E_{l_{\max}}(M)} (W(t_i))(M)}.
    \]
Let $N = (S, T, I, O, H, M_0, L, W)$ be a GSPN:  
- The reachability set of $N$ does not coincide in general with that of the underlying PTN due to the presence of priority levels and inhibitor arcs which usually causes a reduction in the number of reachable markings. 
- The reachability graph of $N$ is defined as the automaton 

$$\mathcal{RG}[N] = (R(M_0), T \times \mathbb{R}^+, \longrightarrow, M_0)$$

where:
- $R(M_0)$ is the least subset of $\mathbb{N}^S$ such that

$$M_0 \in R(M_0)$$

\[
\frac{M_1 \in R(M_0) \quad M_1 \mid (W(t))(M_1) \quad M_2 \mid -(M_1 \mid (t', (W(t'))(M_1)) \quad M_2')}{M_2 \in R(M_0)}
\text{ if } L(t) < L(t')
\]

- $\longrightarrow$ is the least subset of $R(M_0) \times (T \times \mathbb{R}^+) \times R(M_0)$ such that

\[
\frac{M_1 \mid (W(t))(M_1) \quad M_2 \mid -(M_1 \mid (t', (W(t'))(M_1)) \quad M_2')}{M_1 \mid (t', (W(t'))(M_1)) \quad M_2}
\text{ if } L(t) < L(t')
\]

- The HCTMC associated with $N$ is defined through an automaton deriving from a contraction of $\mathcal{RG}[N]$ which causes the elimination of the states associated with the vanishing reachable markings (for a more detailed description, the reader is referred to [15]).

5 Operational GSPN semantics of MPA terms

The net semantics we shall develop for MPA involves translating MPA terms into GSPNs hence it will be said to be the operational GSPN semantics of MPA terms. In this section we shall define such a semantics and we shall wonder whether it is a good net semantics; to this end we shall follow the proposal of P. Degano-R. De Nicola-U. Montanari, adopted also by E. Olderog in [20], which consists of evaluating the goodness of a net semantics for a process algebra based on the following two principles:

- The **retrievability principle**, which states that the operational interleaving semantics of any term of the process algebra should be retrievable from the net semantics of the term itself. Such a principle is usually formalized by requiring that the automaton representing the operational interleaving semantics of any term of the process algebra is strongly bisimilar to the automaton representing the reachability graph of the net semantics of the term itself.

- The **concurrency principle**, which states that the net semantics of any term of the process algebra should represent the intended concurrency of the term itself. Such a principle is usually formalized by requiring that the net semantics is compositional w.r.t. the standard operators on nets.

This section is organized as follows. In Section 5.1 the operational GSPN semantics of MPA terms is defined. In Section 5.2 the semantics is checked for retrievability. In Section 5.3 the semantics is checked for concurrency. In Section 5.4 some examples are shown in which the operational GSPN semantics of some MPA terms is computed.
5.1 Definition
The first step in the definition of the operational GSPN semantics of MPA terms consists of establishing a correspondence between net places and sequential subterms so as to induce, together with the definition of net transitions, a correspondence between net markings and MPA terms. We thus define the set $\mathcal{P}_{GSPN}$ of places as the set of terms $P$ generated by the following production:

$$P ::= \emptyset | \vartriangleleft a, \lambda \triangleright.E | P/L | P\backslash H | P[\varphi] | P+P | P\parallel s \text{id} | i d \parallel s P | A$$

where $a \in Act, \lambda \in Rate, E \in \mathcal{G}, L \subseteq Com, H \subseteq Act, \varphi \in \Phi, S \subseteq Com, A \in \mathcal{G} \cap Con$, and we define the decomposition function

$$\text{dec}_{GSPN} : \mathcal{G} \rightarrow 2^{\mathcal{P}_{GSPN}}$$

by induction on the syntactical structure of the terms in $\mathcal{G}$ in the following way:

- $\text{dec}_{GSPN}(\emptyset) = \{\emptyset\}$;
- $\text{dec}_{GSPN}(\vartriangleleft a, \lambda \triangleright.E) = \{\vartriangleleft a, \lambda \triangleright.E\};$
- $\text{dec}_{GSPN}(E/L) = \text{dec}_{GSPN}(E) \parallel \{P/L | P \in \text{dec}_{GSPN}(E)\};$
- $\text{dec}_{GSPN}(E\backslash H) = \text{dec}_{GSPN}(E) \parallel \{P\backslash H | P \in \text{dec}_{GSPN}(E)\};$
- $\text{dec}_{GSPN}(E[\varphi]) = \text{dec}_{GSPN}(E) [\varphi] = \{P[\varphi] | P \in \text{dec}_{GSPN}(E)\};$
- $\text{dec}_{GSPN}(E_1\parallel E_2) = \text{dec}_{GSPN}(E_1) \parallel \text{dec}_{GSPN}(E_2) = \{P \parallel \{\text{id} \parallel s P | P \in \text{dec}_{GSPN}(E_1)\} \cup \{\text{id} \parallel s P | P \in \text{dec}_{GSPN}(E_2)\}\};$
- $\text{dec}_{GSPN}(A) = \text{dec}_{GSPN}(E) \text{ if } A \triangleleft E,$

where $Q \in 2^{\mathcal{P}_{GSPN}}$ is said to be complete if and only if $\exists E \in \mathcal{G}, \text{dec}_{GSPN}(E) = Q$. We would like to point out that, in order to syntactically express the decomposition into sequential subterms, the binary operator “$\parallel$” has been replaced by the two unary operators “$\parallel \text{id}$” and “$\parallel i d$”.

The decomposition function is well defined because we consider only guardedly closed terms. Besides, by virtue of its last clause, it would be more exact naming it decomposition/expansion function; the above clause and the clause concerning the alternative composition operator have been introduced as refinement of the original proposal of P. Degano-R. De Nicola-U. Montanari in order to correctly translate terms in which both the parallel composition operator and either constants or the alternative composition operator are present.

The second step in the definition of the operational GSPN semantics of MPA terms consists of introducing an appropriate relation on $2^{\mathcal{P}_{GSPN}}$ by means of which net transitions will be constructed. We thus define the relation $\longrightarrow_{GSPN}$ as the least subset of $2^{\mathcal{P}_{GSPN}} \times (Act \times Rate \times Aux) \times 2^{\mathcal{P}_{GSPN}}$ such that

$$\vartriangleleft a, 0 \triangleright.E \xrightarrow{a_0^\lambda}_{GSPN} \text{dec}_{GSPN}(E) \quad \vartriangleleft a, \lambda \triangleright.E \xrightarrow{a_\lambda^t}_{GSPN} \text{dec}_{GSPN}(E) \quad \vartriangleleft a, \infty_\lambda \triangleright.E \xrightarrow{a_\infty_\lambda^i_\beta^j}_{GSPN} \text{dec}_{GSPN}(E)$$

3. We shall use $P, P', P''$, ... as metavariables for $\mathcal{P}_{GSPN}$ and $Q, Q', Q''$, ... as metavariables for $2^{\mathcal{P}_{GSPN}}$.
4. We shall use $R, R', R''$, ... as metavariables for complete elements of $2^{\mathcal{P}_{GSPN}}$. 
the same nonzero firing probability. We thus define the transitions whose firing rule states that in each marking the enabled passive transitions have first of all we introduce the notion of steps. Since GSPNs do not admit passive transitions whereas MPA admits passive actions, of associating with each MPA term an appropriate GSPN by exploiting the previous two decomposition into sequential subterms is given by For example, if we consider the term been discarded yet due to an action previously executed by a sequential subterm of it is necessary to realize that only one part of the sequential subterms (i.e. of the alternative composition operator need to be explained. In particular, it is necessary to realize that only one part of the sequential subterms (i.e. \( Q_2 \)) needs to have an alternative and that such an alternative (i.e. \( R \)) must be a complete set of sequential subterms; the completeness of \( R \) guarantees that none of the sequential subterms in \( R \) has been previously active hence it guarantees that \( Q_2 \) (which is the alternative of \( R \)) has not been discarded yet due to an action previously executed by a sequential subterm of \( R \). For example, if we consider the term \(<a, \lambda > \emptyset || \emptyset < b, \mu > \emptyset + <c, \gamma > \emptyset \) we have that its decomposition into sequential subterms is given by \( \langle a, \lambda > \emptyset || \emptyset id > \rangle + \langle c, \gamma > \emptyset \} \cup \{id || \emptyset b, \mu > \emptyset \} + \{c, \gamma > \emptyset \} \); by executing the action \(<a, \lambda > \) we obtain \( \emptyset || \emptyset id > \} \cup \{id || \emptyset b, \mu > \emptyset \} + \{c, \gamma > \emptyset \} \) where the action \(<c, \gamma > \) has already been discarded (in fact its alternative is not complete).

The third step in the definition of the operational GSPN semantics of MPA terms consists of associating with each MPA term an appropriate GSPN by exploiting the previous two steps. Since GSPNs do not admit passive transitions whereas MPA admits passive actions, first of all we introduce the notion of modified GSPN (MGSPN) as GSPN supplied with passive transitions whose firing rule states that in each marking the enabled passive transitions have the same nonzero firing probability. We thus define the operational MGSPN semantics of \( E \in \mathcal{G} \) as

\[
\mathcal{MGSPN}[E] = (Q, T, I, O, \emptyset, M_0, L, W)
\]

where:

UBLCS-94-12
5 Operational GSPN semantics of MPA terms

• $Q$ is the least subset of $\mathcal{P}_{GSPN}$ such that

\[
\text{dec}_{GSPN}(E) \subseteq Q
\]

\[
Q_1 \subseteq Q \xrightarrow{a,\lambda, w}_{GSPN} Q_2
\]

\[
Q_2 \subseteq Q
\]

• $T = \{(Q_1, a, \lambda, w, Q_2) \in 2^{\mathcal{Q}} \times \text{Act} \times \text{Rate} \times \text{Aux} \times 2^{\mathcal{Q}} | Q_1 \xrightarrow{a,\lambda, w}_{GSPN} Q_2\}$;

• $I : T \rightarrow (Q \rightarrow \mathbb{N})$ such that

\[
\forall P \in Q_1. (I(Q_1, a, \lambda, w, Q_2))(P) = 1
\]

\[
\forall P \in Q - Q_1. (I(Q_1, a, \lambda, w, Q_2))(P) = 0
\]

• $O : T \rightarrow (Q \rightarrow \mathbb{N})$ such that

\[
\forall P \in Q_2. (O(Q_1, a, \lambda, w, Q_2))(P) = 1
\]

\[
\forall P \in Q - Q_2. (O(Q_1, a, \lambda, w, Q_2))(P) = 0
\]

• $M_0 : Q \rightarrow \mathbb{N}$ such that

\[
\forall P \in \text{dec}_{GSPN}(E), M_0(P) = 1
\]

\[
\forall P \in Q - \text{dec}_{GSPN}(E), M_0(P) = 0
\]

• $L : T \rightarrow (Q \rightarrow \mathbb{N})$ such that

\[
L(Q_1, a, \lambda, w, Q_2) = 0 \text{ if } \lambda \in \mathbb{R}^+
\]

\[
L(Q_1, a, \lambda, w, Q_2) = l \text{ if } \lambda = \infty_{l,p}
\]

• $W : T \rightarrow (\mathbb{N}^Q \rightarrow \mathbb{R}^+)$ such that

\[
(W(Q_1, a, \lambda, w, Q_2))(M) = \lambda \text{ if } \lambda \in \mathbb{R}^+
\]

\[
(W(Q_1, a, \lambda, w, Q_2))(M) = p \text{ if } \lambda = \infty_{l,p}
\]

It is interesting to identify a class of terms in $\mathcal{G}$ such that for each term $E$ in this class it turns out that $\mathcal{M}_{GSPN}[E]$ is a GSPN; as we can expect, the above class is given by $\mathcal{E}$ and this will be proved in Section 5.2.

Now we show some properties of the operational MGSPN semantics.

**Theorem 5.1** Let $E \in \mathcal{G}$. It turns out that:

(i) $\mathcal{M}_{GSPN}[E]$ is ordinary and without inhibitor arcs and its transitions have parameters independent of the current marking.

(ii) $\mathcal{M}_{GSPN}[E]$ is safe.

(iii) $\mathcal{M}_{GSPN}[E]$ is finite if each subterm of $E$ of the form $E'/L, E'\backslash H, E'[\varphi], E_1||S E_2$ is without constants.

**Proof**

(i) It immediately follows from the definition of $\mathcal{M}_{GSPN}[E]$.

(ii) The scheme of the proof is analogous to that developed in [20].

(iii) The scheme of the proof is analogous to that developed in [20].

UBLCS-94-12 13
5.2 Retrievalability

In this section we shall prove that the operational MGSPN semantics of MPA terms satisfies the retrievability principle.

Theorem 5.2 For each $E \in \mathcal{G}$ it turns out that $\mathcal{R}\mathcal{G}[\mathcal{MGSNP}[E]]$ is strongly bisimilar to $\mathcal{T}[E]$.

Proof

The scheme of the proof is analogous to that developed e.g. in [20]. In such a proof transitions with different priority levels associated with them are not considered, but it suffices noting that the priority mechanism for MPA actions coincides with the priority mechanism for GSPN (hence MGSPN) transitions.

In fact, if an action occurring in a term does not result in a transition in the automaton representing the operational interleaving semantics of the term itself, then it does not result in a fireable transition in the MGSPN representing the operational MGSPN semantics of the term. Here we only give a sketch of the proof of this fact. There are only two cases in which the priority mechanism of MPA comes into play:

- Consider the term $E_1 + E_2$ and assume that $E_1$ can execute only the action $<a_1, \lambda_1>$, that $E_2$ can execute only the action $<a_2, \lambda_2>$ and that the first action takes precedence over the second one. It follows that $\mathcal{T}[E_1 + E_2]$ has only a transition labeled by $a_1, \lambda_1$ whereas $\mathcal{MGSNP}[E_1 + E_2]$ has two mutually exclusive transitions labeled by $a_1, \lambda_1$ and $a_2, \lambda_2$ where the first one takes precedence over the second one. Due to the mutual exclusion, the first transition prevents the second one from being fired; therefore, it is clear that $\mathcal{R}\mathcal{G}[\mathcal{MGSNP}[E]]$ is strongly bisimilar to $\mathcal{T}[E_1 + E_2]$.

- Consider the term $E_1 || S E_2$ and assume that $E_1$ can execute only the action $<a_1, \lambda_1>$, $a_1 \notin S$, that $E_2$ can execute only the action $<a_2, \lambda_2>$, $a_2 \notin S$, and that the first action takes precedence over the second one. It follows that $\mathcal{T}[E_1 || S E_2]$ has two transitions labeled by $a_1, \lambda_1$ and $a_2, \lambda_2$ where the second one can take place only after the first one, while $\mathcal{MGSNP}[E_1 || S E_2]$ has two independent transitions labeled by $a_1, \lambda_1$ and $a_2, \lambda_2$ where the first one takes precedence over the second one hence the second one can be fired only after the first one. Therefore, it is clear that $\mathcal{R}\mathcal{G}[\mathcal{MGSNP}[E]]$ is strongly bisimilar to $\mathcal{T}[E_1 + E_2]$.

Corollary 5.3 It turns out that $\mathcal{MGSNP}[E]$ is a GSPN iff $E \in \mathcal{E}$. In such a case $\mathcal{MGSNP}[E]$ is denoted with $\mathcal{GSPN}[E]$ and is said to be the operational GSPN semantics of $E$.

Proof

($\Rightarrow$) Suppose that $\mathcal{MGSNP}[E]$ is a GSPN, i.e. suppose that $\mathcal{MGSNP}[E]$ has not passive transitions. Then $\mathcal{R}\mathcal{G}[\mathcal{MGSNP}[E]]$ has not passive transitions hence, by virtue of Theorem 5.2, also $\mathcal{T}[E]$ has not passive transitions so $E \in \mathcal{E}$.

($\Leftarrow$) Suppose that $E \in \mathcal{E}$, i.e. suppose that $\mathcal{T}[E]$ has not passive transitions; we prove that $\mathcal{MGSNP}[E]$ is a GSPN by proceeding by induction on the syntactical structure of $E$:

- If $E \equiv 0$ then $\mathcal{MGSNP}[E]$ is obviously a GSPN.
- Let $E \equiv <a, \lambda>.E'$ and let the thesis hold for $E'$. From $E \in \mathcal{E}$ it follows that $\lambda \neq 0$ and that $E' \in \mathcal{E}$ and therefore, by the induction hypothesis, we have that $\mathcal{MGSNP}[E']$ is
a GSPN hence $MGS\!N[E]$ is a GSPN too.

- Let $E \equiv E'/L$ and let the thesis hold for $E'$. From $E \in \mathcal{E}$ it follows that $E' \in \mathcal{E}$ and therefore, by the induction hypothesis, we have that $MGS\!N[E']$ is a GSPN hence $MGS\!N'[E]$ is a GSPN too.
- Let $E \equiv E'\setminus H$ and let the thesis hold for $E'$; there are two cases:
  - If $E' \in \mathcal{E}$ then, by the induction hypothesis, we have that $MGS\!N[E'] = MGS\!N[E]$ is a GSPN.
  - If $E' \notin \mathcal{E}$ then $E'$ can execute some passive actions which, due to the fact that $E \in \mathcal{E}$, have type present in $H$. Since the rule of $G\!N$ for the temporal restriction operator is analogous to the transition rule for such an operator, the passive transitions in $MGS\!N'[E]$ cannot be present in $MGS\!N[E]$ hence $MGS\!N[E]$ is a GSPN.
- Let $E \equiv E'[g]$ and let the thesis hold for $E'$. From $E \in \mathcal{E}$ it follows that $E' \in \mathcal{E}$ and therefore, by the induction hypothesis, we have that $MGS\!N[E']$ is a GSPN hence $MGS\!N'[E]$ is a GSPN too.
- Let $E \equiv E_1\parallel E_2$ and let the thesis hold for $E_1$ and for $E_2$. From $E \in \mathcal{E}$ it follows that $E_1 \in \mathcal{E} \land E_2 \in \mathcal{E}$ and therefore, by the induction hypothesis, we have that $MGS\!N[E_1]$ and $MGS\!N[E_2]$ are two GSPNs hence $MGS\!N'[E]$ is a GSPN too.
- Let $E \equiv E_1\setminus g E_2$ and let the thesis hold for $E_1$ and for $E_2$; there are two cases:
  - If $E_1 \in \mathcal{E} \land E_2 \notin \mathcal{E}$ then, by the induction hypothesis, we have that $MGS\!N[E_1]$ and $MGS\!N[E_2]$ are two GSPNs hence $MGS\!N'[E]$ is a GSPN too.
  - If $E_1 \notin \mathcal{E} \land E_2 \notin \mathcal{E}$ then $E_1$ or $E_2$ can execute some passive actions which, due to the fact that $E \in \mathcal{E}$, have type present in $S$ and either do not synchronize at all or synchronize with active actions of the same type present in the other subterm. Since the rules of $G\!N$ for the parallel composition operator are analogous to the transition rules for such an operator, the passive transitions present in $MGS\!N[E_1]$ or in $MGS\!N[E_2]$ cannot be present in $MGS\!N'[E]$ hence $MGS\!N'[E]$ is a GSPN.
- Let $E \equiv A, A \equiv E'$. Since $A \in \mathcal{E}$, hence in particular $A \in G$, it is possible to obtain from $E'$ a term $E''$ other than a constant by substituting constants by means of their corresponding defining equations (if $E' \notin Con$ then $E'' \equiv E'$). After getting $E''$, since $E'' \in \mathcal{E} - Con$ we can apply to it one of the previous cases and then exploit the fact that $MGS\!N[E] = MGS\!N'[E] = MGS\!N'[E''].$

5.3 Concurrency

In this section we shall prove that the operational MGSPN semantics of MPA terms satisfies the concurrency principle. To this end, we define first of all the following operators on the ordinary and safe MGSPNs without inhibitor arcs:

- $0_{MGS\!P\!N}$ = $\{P\}, 0, 0, 0, \emptyset, \emptyset, \{(P, 1)\}, 0, 0$;
- $<a, \lambda>_{MGS\!P\!N}(Q, T, I, O, \emptyset, M_0, L, W) = (Q', T', I', O', \emptyset, M'_0, I', W')$ where:
  - $Q' = Q \cup \{P'\}, P' \notin Q$;
  - $T' = T \cup \{t'\}, t' = \{\{P'\}, a, \lambda, w, \{P \in Q \mid M_0(P) = 1\}\}$ where $w = p$ if $\lambda = 0$, $w = t$ if $\lambda \in I^+\!, w = i$ if $\lambda \in I^{nf};$
  - $I' = \{(Q_1, a, \lambda, w, Q_2), (P, 1)\} \cup \{(Q_1, a, \lambda, w, Q_2), (P, 0)\} \cup \{(Q_1, a, \lambda, w, Q_2) \in T' \land P \in Q' - Q_1\};$
- \( O' = \{(Q_1, a, \lambda, w, Q_2), (P, 1) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q_2 \} \cup \{(Q_1, a, \lambda, w, Q_2), (P, 0) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q' \setminus Q_2 \};
- \( M'_0 = \{(P', 1) \} \cup \{(P, 0) | P \in Q \};
- \( L' = \{(Q_1, a, \lambda, w, Q_2), I \} \in T' \times N \mid (\lambda \in \mathbb{R}^+ \land l = 0) \lor \lambda = \infty_{l,p} \};
- \( W' = \{(Q_1, a, \lambda, w, Q_2), (M, p) \} \in T' \times (\mathbb{N}_0 \times \mathbb{R}^+) \mid (\lambda \in \mathbb{R}^+ \land p = \lambda) \lor \lambda = \infty_{l,p} \};
- \( (Q, T, I, O, \emptyset, M_0, L, W) \rangle_{MGSPN} L = (Q, T', I', O', \emptyset, M_0, L', W') \) where:
  - \( T' = \{(Q_1, a, \lambda, w, Q_2) \in T | a \notin L \} \cup \{(Q_1, \tau, \lambda, w, Q_2) | \exists a \in L. (Q_1, a, \lambda, w, Q_2) \in T \};
  - \( I' = \{(Q_1, a, \lambda, w, Q_2), (P, 1) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q_1 \} \cup \{(Q_1, a, \lambda, w, Q_2), (P, 0) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q - Q_1 \};
  - \( O' = \{(Q_1, a, \lambda, w, Q_2), (P, 1) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q_2 \} \cup \{(Q_1, a, \lambda, w, Q_2), (P, 0) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q - Q_2 \};
  - \( L'' = \{(Q_1, a, \lambda, w, Q_2), I \} \in T' \times N \mid (\lambda \in \mathbb{R}^+ \land l = 0) \lor \lambda = \infty_{l,p} \};
  - \( W'' = \{(Q_1, a, \lambda, w, Q_2), (M, p) \} \in T' \times (\mathbb{N}_0 \times \mathbb{R}^+) \mid (\lambda \in \mathbb{R}^+ \land p = \lambda) \lor \lambda = \infty_{l,p} \};
- \( (Q, T, I, O, \emptyset, M_0, L, W)[\varphi] \rangle_{MGSPN} H = (Q, T', I', O', \emptyset, M_0, L', W') \) where:
  - \( T' = \{(Q_1, a, \lambda, w, Q_2) \in T | \neg(a \in H \land \lambda = 0) \};
  - \( I' = \{(Q_1, a, \lambda, w, Q_2), (P, 1) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q_1 \} \cup \{(Q_1, a, \lambda, w, Q_2), (P, 0) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q - Q_1 \};
  - \( O' = \{(Q_1, a, \lambda, w, Q_2), (P, 1) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q_2 \} \cup \{(Q_1, a, \lambda, w, Q_2), (P, 0) \} | (Q_1, a, \lambda, w, Q_2) \in T' \wedge P \in Q - Q_2 \};
  - \( L' = \{(Q_1, a, \lambda, w, Q_2), I \} \in T' \times N \mid (\lambda \in \mathbb{R}^+ \land l = 0) \lor \lambda = \infty_{l,p} \};
  - \( W' = \{(Q_1, a, \lambda, w, Q_2), (M, p) \} \in T' \times (\mathbb{N}_0 \times \mathbb{R}^+) \mid (\lambda \in \mathbb{R}^+ \land p = \lambda) \lor \lambda = \infty_{l,p} \};
- \( (Q', T_1, I_1, O_1, \emptyset, M_0, L, W_1) +_{MGSPN} (Q'', T_2, I_2, O_2, \emptyset, M_0, L, W_2) = (Q, T, I, O, \emptyset, M_0, L, W) \) where:
  - \( Q = Q' \cup Q'' \cup \{(P_1, P_2) \in Q' \times Q'' | M_{01}(P_1) = 1 \land M_{02}(P_2) = 1 \} \setminus Q'' \); 
  - \( T = \{(Q_1, \{P \in Q'' | M_{02}(P) = 1\} \cup Q'_1, a, \lambda, w, Q_2) | Q_1 \in \{P \in Q' | M_{01}(P) = 1\} \land Q_1 \cap Q'_1 = \emptyset \land (Q_1 \cup Q'_1, a, \lambda, w, Q_2) \in T_1 \} \cup \{(Q_1, \{P \in Q' | M_{01}(P) = 1\} \cup Q'_1, a, \lambda, w, Q_2) \in T_1 \} \cup \{(Q_1, \{P \in Q'_1 | M_{02}(P) = 1\} \land Q_1 \cap Q'_1 = \emptyset \land (Q_1 \cup Q'_1, a, \lambda, w, Q_2) \in T_2 \};
  - \( I = \{(Q_1, a, \lambda, w, Q_2), (P, 1) \} | (Q_1, a, \lambda, w, Q_2) \in T \wedge P \in Q_1 \} \cup \{(Q_1, a, \lambda, w, Q_2), (P, 0) \} | (Q_1, a, \lambda, w, Q_2) \in T \wedge P \in Q - Q_1 \};
  - \( O = \{(Q_1, a, \lambda, w, Q_2), (P, 1) \} | (Q_1, a, \lambda, w, Q_2) \in T \wedge P \in Q_2 \} \cup \{(Q_1, a, \lambda, w, Q_2), (P, 0) \} | (Q_1, a, \lambda, w, Q_2) \in T \wedge P \in Q - Q_2 \};
  - \( M_0 = \{(P_1, P_2), (P_1, P_2) \in Q' \times Q'' \wedge M_{01}(P_1) = 1 \land M_{02}(P_2) = 1 \} \cup \{(P, 0) | P \in Q' \cup Q'' \};
  - \( L = \{(Q_1, a, \lambda, w, Q_2), I \} \in T \times N \mid (\lambda \in \mathbb{R}^+ \land l = 0) \lor \lambda = \infty_{l,p} \};
  - \( W = \{(Q_1, a, \lambda, w, Q_2), (M, p) \} \in T \times (\mathbb{N}_0 \times \mathbb{R}^+) \mid (\lambda \in \mathbb{R}^+ \land p = \lambda) \lor \lambda = \infty_{l,p} \);
In this section we compute the operational GSPN semantics of some MPA terms.

**Theorem 5.4** Having denoted with \( sb \) the strong bisimulation relation, it turns out that:

- \( \forall E \in G. \forall a \in Act. \forall \lambda \in Rate. MGSPN[N][a, \lambda >. E] sb <a, \lambda >.MGSPN.MGSPN'[E]. \)
- \( \forall E \in G. \forall L \subseteq Com. MGSPN[N][E / L] sb MGSPN[N][E] / MGSPN L. \)
- \( \forall E \in G. \forall H \subseteq Act. MGSPN[N][E \setminus H] sb MGSPN[N][E] \setminus MGSPN H. \)
- \( \forall E \in G. \forall \varphi \in \Phi. MGSPN[N][E][\varphi] sb MGSPN[N][E][\varphi] / MGSPN. \)
- \( \forall E_1, E_2 \in G. MGSPN[N][E_1 \uplus E_2] sb MGSPN[N][E_1] + MGSPN MGSPN[N][E_2]. \)
- \( \forall E_1, E_2 \in G. VS \subseteq Com. MGSPN[N][E_1 \uplus E_2] sb MGSPN[N][E_1] || MGSPN[N][E_2]. \)
- \( \forall A \in G \cap Con. A \cong E \implies MGSPN[N][A] = MGSPN[N][E]. \)

**Proof**
The scheme of the proof is analogous to that developed in [20].

---

**5.4 Examples**

In this section we compute the operational GSPN semantics of some MPA terms.

**Example 5.5** Let \( E \equiv <a, \lambda >.0 || <b, \mu >.0 \); then \( GSPN[N][E] \) is the following net:

\[
\begin{array}{c}
\text{id} || 0 <b, \mu > 0 \\
\end{array}
\]

**Example 5.6** Let \( E \equiv <a, \lambda >. <b, \mu >.0 || <b, \mu >. <a, \lambda >.0 \); then \( GSPN[N][E] \) is the following net:

\[
\begin{array}{c}
\text{id} || 0
\end{array}
\]
Example 5.7 Let $E \equiv \langle a, \lambda >.0+<a, \lambda >.0$; then $\mathcal{GSPN}[E]$ is the following net:

![Diagram](attachment:image.png)

Example 5.8 Let $E \equiv (\langle a, \lambda >.b, \mu >.0{\{a\}} \langle a, 0 >.c, \mu >.0{\{a\}});$ then $\mathcal{GSPN}[E]$ is the following net:

![Diagram](attachment:image.png)

Example 5.9 Let $E \equiv (\langle a, 0 >.0{\{a\}}\langle a, 0 >.0{\{a\}})_{\{a\}} \langle a, 0 >.0;\{a\}}_{\{a\}}\langle a, \infty >.0;\{a\}}_{\{a\}}$; then $\mathcal{GSPN}[E]$ is the following net:
Example 5.10 Let $E \equiv A, A \seteq \langle a, \lambda \rangle \cdot \langle b, \mu \rangle \cdot A \| \varnothing, \gamma \rangle \Omega$; then $GSPN[E]$ is the following net:

Example 5.11 Let $E \equiv \langle a, \lambda \rangle \cdot \varnothing \| \varnothing \langle b, \mu \rangle \cdot \Omega \langle c, \gamma \rangle \| \Omega$; then $GSPN[E]$ is the following net:
Example 5.12 A queueing system $M/M/1$ with arrival rate $\lambda$ and service rate $\mu$ is a model representing a service center composed of an infinite queue and a server which provide a given service following the FCFS discipline to the customers arrived at the service center, where the customer interarrival time is exponentially distributed with parameter $\lambda$ and the customer service time is exponentially distributed with parameter $\mu$.

Such a system can be described in MPA as follows:

- $System_{M/M/1} \triangleq Arrivals\|_{\{a\}}(Queue_0\|_{\{d\}}Server)$;
- $Arrivals \triangleq <a, \lambda>.Arrivals$;
- $Queue_0 \triangleq <a, 0>.Queue_1$, $Queue_i \triangleq <a, 0>.Queue_{i+1}+<d, 0>.Queue_{i-1}$, $i > 0$;
- $Server \triangleq <d, \infty, 1>.S_1$, $S_1 \triangleq <s, \mu>.Server$.

Then $GSPN[ System_{M/M/1} ]$ is the following net:

6 Conclusions

In this technical report we have defined an operational net semantics for MPA based on GSPNs; as far as we know, this is the first time that an operational net semantics is defined for a stochastic process algebra.

This fact is very important because, using MPA, we can exploit a modeling technique for concurrent systems which integrates:

- Different points of view of concurrent systems.
- The qualitative and the quantitative analysis of concurrent systems.

Such a modeling technique can be summarized by the following scheme.
6 Conclusions

which indicates that:

- The first phase consists of specifying the concurrent system as a term of the stochastic process algebra (which is MPA in our case), so as to obtain a first model of the concurrent system which is expressed in an abstract algebraic language (hence, easy to understand and compositional). With this algebraic representation, it is possible to:
  - Perform a functional analysis of the concurrent system by means of equivalence checking (after defining a suitable notion of equivalence, e.g., bisimulation, one can check if two terms are equivalent), preorder checking (after defining a suitable notion of preorder, one can check if a term is more defined than another term), model checking (after choosing an appropriate modal logic, one can determine whether a term satisfies an assertion). Moreover, if the equivalence relation is a congruence, analysis can be performed by equational reasoning defined on its axiomatization. Such analyses can detect qualitative properties of the concurrent system (e.g., deadlock) and can also help in minimizing the state space of the system representation.
  - Evaluate the performance of the concurrent system by resorting to the study of a HCTMC.

- The second phase consists of translating the term of the stochastic process algebra constructed in the previous phase into a stochastic Petri net (which is a GSPN in our case), giving a distributed representation of the same concurrent system. Such a translation makes explicit the parallelism and the causal dependencies among the activities of the concurrent system itself, but the price to pay is that such a model usually has a remarkable graphic complexity. Moreover, it is not easily compositional, hence making hard, in general, the detection and the analysis of its subsystems. With this stochastic Petri net representation, it is possible to:
  - Perform a qualitative analysis of the concurrent system by, e.g., computing net invariants.
  - Evaluate the performance of the concurrent systems by resorting to existing tools for simulation on stochastic Petri nets.

The modeling technique above was firstly proposed by E.-R. Olderog [20]; here it has been extended in order to take into account the quantitative aspect of performance evaluation. It is worth noting that if we regard GSPNs as implementations of concurrent systems then
such a modeling technique automatically guarantees that the implementation obtained for a
given concurrent system is correct, i.e. consistent with the specification of the system itself.

References

“On the Efficient Construction of the Tangible Reachability Graph of Generalized Stochastic
Petri Nets”
in Proceedings of the International Workshop on Petri Nets and Performance Models,
Madison (WI), 1987
IEEE-CS Press, Washington (DC), 1987

“Verso l’Integrazione di Modelli di Concorrenza Stocastici”
Master Thesis
University of Bologna, Bologna (Italy), March 1994

“MPA: a Stochastic Process Algebra”
Technical Report
University of Bologna, Bologna (Italy), 1994

“On the Consistency of Truly Concurrent Operational and Denotational Semantics”
in Proceedings of the Annual Symposium on Logics in Computer Science, Edinburgh
(UK), 1988
IEEE-CS Press, Washington (DC), 1988

“Les Reseaux de Petri Stochastiques”
in Technique et Science Informatiques, vol. 4
1985

“The PEPA Workbench: a Tool to Support a Process Algebra-Based Approach to Performance
Modelling”
Technical Report
University of Edinburgh, Edinburgh (UK), 1993

“TIPP: a Language for Timed Processes and Performance Evaluation”
Technical Report
University of Erlangen-Nürnberg, Erlangen (Germany), 1992

“Multiprocessor and Distributed System Design: the Integration of Functional Specification
and Performance Analysis Using Stochastic Process Algebras”
in Lecture Notes in Computer Science, vol. 729, pp. 121–146
Springer-Verlag, Berlin (Germany), 1993

“PEPA: Performance Enhanced Process Algebra”
Technical Report
University of Edinburgh, Edinburgh (UK), 1993
[10] M. A. Marsan
“Stochastic Petri Nets: an Elementary Introduction”
in Lecture Notes in Computer Science, vol. 424, pp. 1–29
Springer-Verlag, Berlin (Germany), 1990

“On Petri Nets with Stochastic Timing”
in Proceedings of the International Workshop on Timed Petri Nets, Turin (Italy), 1985
IEEE-CS Press, Washington (DC), 1985

“The Effect of Execution Policies on the Semantics and Analysis of Stochastic Petri Nets”
IEEE-CS Press, Washington (DC), 1989

“Generalized Stochastic Petri Nets Revisited: Random Switches and Priorities”
in Proceedings of the International Workshop on Petri Nets and Performance Models,
Madison (WI), 1987
IEEE-CS Press, Washington (DC), 1987

“Generalized Stochastic Petri Nets: a Definition at the Net Level and its Implications”
IEEE-CS Press, Washington (DC), 1993

“A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems”
ACM, 1984

[16] M. K. Molloy
“On the Integration of Delay and Throughput Measures in Distributed Processing Models”
Ph.D. Thesis
UCLA, Los Angeles (CA), 1981

[17] M. K. Molloy
“Performance Analysis Using Stochastic Petri Nets”
IEEE-CS Press, Washington (DC), 1982

[18] T. Murata
“Petri Nets: Properties, Analysis and Applications”
IEEE-CS Press, Washington (DC), 1989

[19] S. Natkin
“Les Reseaux de Petri Stochastiques et leur Application a l’Evaluation des Systemes Informatiques”
Ph.D. Thesis
CNAM, Paris (France), 1980

“Nets, Terms and Formulas. Three Views of Concurrent Processes and their Relationship”
    “Communication with Automata”
    Technical Report
    Rome Air Development Center, Rome (NY), 1966

[22] W. Reisig
    “Petri Nets: an Introduction”
    Springer-Verlag, Berlin (Germany), 1985