A Distributed Semantics for EMPA Based on Stochastic Contextual Nets

Marco Bernardo and Nadia Busi[1] and Roberto Gorrieri

Università di Bologna, Dipartimento di Scienze dell'Informazione, Piazza di Porta S. Donato 5, 40127 Bologna, Italy
E-mail: {bernardo, gorrieri}@cs.unibo.it
[1] *Università di Siena, Dipartimento di Matematica, Via del Capitano 15, 53100 Siena, Italy*
E-mail: busi@cs.unibo.it

Extended Markovian Process Algebra (EMPA) is a stochastic process algebra equipped with an interleaving semantics, a Markovian semantics and a net semantics. The main drawback of its net semantics is that it usually associates huge nets with EMPA terms. Here we propose a new net semantics, based on contextual nets, in order to obtain more compact net representations for EMPA terms.

1. INTRODUCTION

Extended Markovian Process Algebra (EMPA for short) (Bernardo *et al.*, 1995b) is a process algebra whose actions are durational. The durations are expressed by means of exponentially distributed random variables, as proposed in other stochastic process algebras like MTIPP (Götz *et al.*, 1993), PEPA (Hillston, 1994), and MPA (Buchholz, 1994). Also immediate actions are used in EMPA, thus yielding further expressive power.

Each EMPA term E is equipped with three semantics: the labelled transition system $\mathcal{I}[\![E]\!]$, which is the interleaving model of E, the Markov chain $\mathcal{M}[\![E]\!]$, which is the stochastic model of E, and the stochastic Petri net $\mathcal{L}oc\mathcal{ON}[\![E]\!]$, which is the distributed model of E. These three semantics are at the base of the integrated approach for modelling and analysing concurrent systems described in (Bernardo *et al.*, 1994a).

The net semantics $\mathcal{L}oc\mathcal{ON}[\![E]\!]$ has been defined by resorting to a suitable extension of the structured operational semantics approach for nets (Degano *et al.*, 1988; Olderog, 1991), which consists of associating with each term E a place/transition net such that: places correspond to the sequential subterms of E and its derivatives, transitions are defined by induction on the syntactical structure of the sets of sequential subterms, and markings correspond roughly to E and its derivatives. Since EMPA is a stochastic process algebra, its terms have been translated into stochastic Petri nets (Ajmone Marsan *et al.*, 1984). In (Bernardo *et al.*, 1995a) we proved that this is a good net semantics because it satisfies both the *functional and performance retrievability principles* and the *concurrency principle*: the first and second principles guarantee, for each term, the consistency of its net semantics with respect to its interleaving semantics both from a functional point of view and from a performance evaluation point of view, and the third one assures that for each term all the intended concurrency is captured by its net semantics.

The main drawback of such a net semantics is that it usually associates huge nets with EMPA terms. This is a drawback shared by all the net semantics exploiting the syntactical structure of the terms (notably, the parallel composition operator) to define the set of places (Degano *et al.*, 1988). These are sometimes called *location-oriented* net semantics. However, there is another approach, called *label-oriented*, which partly ignores the syntactical structure of terms and is based mainly on information about transition labels (Goltz, 1988; Gorrieri & Montanari, 1995). While the former approach has the merit of having been successfully applied to several process algebras, the latter has the merit of producing smaller net representations.

The purpose of this paper is to define a new net semantics for EMPA in the label-oriented approach. Following the proposal of (Busi & Gorrieri, 1995a; Busi & Gorrieri, 1995b), we can define a label-oriented net semantics by exploiting inhibitor arcs as follows:

- For each action type there is a place in the net which, when containing at least one token, inhibits the transitions labelled with that action type.
- Each term of the form $E_1 + E_2$ is interpreted as the parallel composition of kE_1 and $\bar{k}E_2$, where each place representing a subterm of E_1 (E_2) is decorated with a conflict name k ($\bar{k}$). The net will contain a place for any conflict name which, when holding at least one token, inhibits all the transitions starting from places decorated with the complementary conflict name.

In order to model the functional abstraction operator and the functional relabelling operator of EMPA, we need also contextual arcs, helpful in correctly implementing the action type binding and the related mechanism for generating fresh action types. Thus, the semantic model we use is the stochastic version of the class of contextual nets (Montanari & Rossi, 1995).

Finally, we show that most inhibitor arcs (all, if we restrict to guarded sum) and all the contextual arcs can be safely removed, obtaining an optimised, compact net semantics. The advantages of this new net semantics with respect to the location-oriented one are:

- Nets are no more 1-safe, as places can hold many tokens.
- The alternative composition has a linear representation: if we need p_1 places for representing E_1 and p_2 places for E_2, then we need $O(p_1{+}p_2)$ places for $E_1{+}E_2$. In the location-oriented approach, it would be $O(p_1 \cdot p_2)$.

This paper is organised as follows. In Section 2 some notions concerning EMPA are recalled. In Section 3 the class of stochastic contextual nets is introduced. In Section 4 we define an operational label-oriented net semantics for EMPA. The adequacy of this semantics is assessed in Section 5 by verifying that it satisfies the functional and the performance retrievability principles. In Section 6 we present the optimised version of the net semantics, which is compared in Section 7 with the location-oriented one. Finally, some concluding remarks are reported in Section 8.

2. EMPA: A BRIEF OVERVIEW

In this section we introduce the syntax and the semantics for EMPA. The reader is invited to consult (Bernardo *et al.*, 1995b) * for more details.

2.1. Syntax

Let $Act = AType \times ARate$ be a set of *actions*, where $AType$ is a set of *action types* and $ARate = \{0\} \cup \mathbb{R}_+ \cup Inf$ is a set of *action rates*, where $Inf = \{\infty_{l,w} \mid l \in \mathbb{N}_+ \wedge w \in \mathbb{R}_+\}$. Depending on the type, actions are divided into: *external or observable* actions and one *internal* action τ, which is invisible to an external observer. Depending on the rate, actions are divided into:

- *Passive actions*, i.e. actions with undefined duration, expressed by the execution rate zero. The duration is fixed only upon a synchronisation with an active action of the same type.
- *Active actions*, whose execution rate is nonzero, in turn divided into:
  - *Timed actions*, whose execution rate is finite. The duration is given by an exponentially distributed random variable with action rate as parameter.
  - *Immediate actions*, whose execution rate is infinite (hence with zero duration). They are given a priority level l and a weight w.

*This technical report is available via `ftp` from the area `ftp.cs.unibo.it:/pub/TR/UBLCS` in compressed PostScript format, or via `www` with the URL `http://www.cs.unibo.it`.

Let Con be a set of *constants*, and let $\Phi = \{\varphi : AType \longrightarrow AType \mid \varphi(\tau) = \tau \wedge \varphi(AType - \{\tau\}) \subseteq AType - \{\tau\}\}$ be a set of *relabelling functions*.

Definition 2.1. *The set $\mathcal{L}$ of EMPA processes is defined as the set of terms generated by the following syntax:*

$$E ::= \underline{0} \mid <a, \tilde{\lambda}>.E \mid E/L \mid E \backslash H \mid E[\varphi] \mid E{+}E \mid E \|_S E \mid A$$

where $<a, \tilde{\lambda}> \in Act$, $L, S \subseteq AType - \{\tau\}$, $H \subseteq AType$, $A \in Con$. □

The *null term* "$\underline{0}$" is a nullary operator representing a term which can execute no action.
The *prefix operator* "$<a, \tilde{\lambda}>.\_$" represents the sequential composition of an action and a term.
The *functional abstraction operator* "$\_/L$" expresses the abstraction from the type of actions whenever it is in L, i.e. the action type is turned to τ (see the hiding operator of TCSP (Hoare, 1985)).
The *temporal restriction operator* "$\_\backslash H$" prevents the execution of passive actions whose type is in H.
The *functional relabelling operator* "$\_[\varphi]$" changes the type of the actions according to φ.
The *alternative composition operator* "$\_{+}\_$" expresses a choice between two terms. Unlike classical process algebras, such a choice is not nondeterministic but probabilistic, as we shall see in Section 2.2.
The *parallel composition operator* "$\_\|_S\_$" expresses the parallel execution of two terms according to the TCSP (Hoare, 1985) synchronisation discipline on action types. Hence, term $E_1 \|_S E_2$, where S is called the *synchronisation set*, can execute asynchronously actions whose type does not appear in S from E_1 *or* E_2, and synchronously actions whose type appears in S from E_1 *and* E_2. The actual executability of synchronisations may depend on the rate of the involved actions: action $<a, \tilde{\lambda}>$ can be synchronised with action $<a, \tilde{\mu}>$ if and only if $\min(\tilde{\lambda}, \tilde{\mu}) = 0$. Since we require that in a synchronisation at most one active action is involved, the synchronisation discipline turns out to be based on the client-server paradigm.
Finally, each constant A is used as a shorthand for a term E through its *defining equation* $A \stackrel{\Delta}{=} E$. Constants are used for recursive definitions, which are well-defined for *guardedly closed* terms. In the following we consider only the set $\mathcal{G}$ of guardedly closed terms in $\mathcal{L}$.

2.2. Race policy

Since several active actions may be simultaneously executable, and since in an interleaving model only one action at a time can be done, it is necessary to choose a policy determining which of them is to be executed first. In this framework we adopt the *race policy*, which chooses the active action having the least duration. As a consequence, immediate actions take precedence over timed actions. Furthermore, since each immediate action is equipped with a priority level and a weight, when several immediate actions are simultaneously executable only those having the highest priority level are actually executable: the choice among them is probabilistically made by giving each of them an execution probability proportional to its weight. Passive actions are not affected by the race policy.

Before continuing, we recall the following property of the exponentially distributed random variables.

Proposition 2.1. *If* $\{Y_i \mid 1 \leq i \leq n\}$ *is a set of* $n \geq 2$ *exponentially distributed random variables with rates* $\lambda_1, \lambda_2, \ldots, \lambda_n$*, respectively, and a random variable* Y *is defined by* $Y = \min\{Y_i \mid 1 \leq i \leq n\}$*, then* Y *is exponentially distributed with rate* $\lambda = \sum_{i=1}^{n} \lambda_i$*.* □

As a consequence of the previous remarks and the proposition above, we can assess the *execution probability* of each active action in a given term:

- If n timed actions of the form
$$<a_1, \lambda_1>, <a_2, \lambda_2>, \ldots, <a_n, \lambda_n>$$
are executable and no immediate actions are executable, then the execution probability of $<a_k, \lambda_k>$ is given by
$$\lambda_k / \sum_{i=1}^{n} \lambda_i$$
- If n immediate actions of the form
$$<a_1, \infty_{l,w_1}>, <a_2, \infty_{l,w_2}>, \ldots, <a_n, \infty_{l,w_n}>$$
are executable and no immediate actions with higher priority are executable, then the execution probability of $<a_k, \infty_{l,w_k}>$ is given by
$$w_k / \sum_{i=1}^{n} w_i$$

Furthermore, the execution probability of a passive action is fixed only upon its synchronisation with an active action of the same type. Whenever n passive actions can be separately synchronised with the same active action, we assume that each of the synchronisations is given the same execution probability: such a probability is equal to the execution probability of the active action divided by n.

We conclude this brief presentation about the race policy by introducing a problem common to all the stochastic process algebras. In the case of a classical process algebra, $a.E + a.E$ can only perform a thus becoming E, so it is equivalent to $a.E$. In the case of a stochastic process algebra such as EMPA, given $<a, \lambda>.E + <a, \lambda>.E$ we must remember that there are two executable actions because the race policy has been adopted and therefore the exit rate from this term is not λ but 2λ (by virtue of Proposition 2.1). Since in EMPA this is achieved by resorting to the formalism of multisets, we recall below some relevant definitions about them.

Definition 2.2. *Given a set* S*, a multiset over* S *is a function* $M : S \longrightarrow \mathbb{N}$*, and a finite multiset over* S *is a function* $M : S \longrightarrow \mathbb{N}$ *such that the set* $\mathit{dom}(M) = \{s \in S \mid M(s) \neq 0\}$ *is finite. The value* $M(s)$ *is called the multiplicity of element* s*. We denote with* $\mathcal{M}u(S)$ *the set of all the multisets over* S*, with* $\mathcal{M}u_{fin}(S)$ *the set of all the finite multisets over* S*, and with* $\mathcal{P}_{fin}(S)$ *the set of all the finite sets over* S*.* □

Definition 2.3. *Given a set* S*, let* $M_1, M_2, M \in \mathcal{M}u(S)$ *and* $M' \in \mathcal{M}u(S \times S)$*. Then:*

- $s \in M \iff M(s) > 0$;
- $M_1 \subseteq M_2 \iff \forall s \in S.\, M_1(s) \leq M_2(s)$;
- $M = M_1 \oplus M_2 \iff \forall s \in S.\, M(s) = M_1(s) + M_2(s)$;
- $M = M_1 \ominus M_2 \iff \forall s \in S.\, M(s) = \max(M_1(s) - M_2(s), 0)$. □

We use "$\{\!|$" and "$|\!\}$" as brackets for multisets, and "$\emptyset$" to denote the empty multiset. When describing multisets by means of brackets, we adopt the following two conventions about the multiplicity of their elements.
Let $S_1, \ldots, S_n$ be $n \geq 1$ sets, let $M_1, \ldots, M_n$ be n multisets over $S_1, \ldots, S_n$, respectively, let M' be a multiset over $S_1 \times \ldots \times S_n$, and let P be a predicate over $S_1 \times \ldots \times S_n$. Then $M' = \{\!| (s_1, \ldots, s_n) \in S_1 \times \ldots \times S_n \mid s_1 \in M_1 \wedge \ldots \wedge s_n \in M_n \wedge P(s_1, \ldots, s_n) |\!\}$ if and only if, for each $s = (s_1, \ldots, s_n) \in S_1 \times \ldots \times S_n$, $P(s) \Longrightarrow M'(s) = \prod_{i=1}^{n} M_i(s_i)$ and $\neg P(s) \Longrightarrow M'(s) = 0$.
Let $S_1, \ldots, S_n$ be $n \geq 1$ sets, let M be a multiset over $S_1 \times \ldots \times S_n$, let M' be a multiset over S_i where $1 \leq i \leq n$, and let P be a predicate over S_i. Then $M' = \{\!| s \in S_i \mid (s_1, \ldots, s, \ldots, s_n) \in M \wedge P(s) |\!\}$ if and only if, for each $s \in S_i$, $P(s) \Longrightarrow M'(s) = \sum_{s_1 \in S_1 \wedge \ldots \wedge s_n \in S_n} M(s_1, \ldots, s, \ldots, s_n)$ and $\neg P(s) \Longrightarrow M'(s) = 0$.

Definition 2.4. *Let* $S_1, \ldots, S_n$ *be* $n \geq 1$ *sets. The* i^{th} *multiset projection over the given sets,* $1 \leq i \leq n$*, is the function* $\pi_i : \mathcal{M}u(S_1 \times \ldots \times S_n) \longrightarrow \mathcal{M}u(S_i)$ *defined by* $\pi_i(M) = \{\!| s_i \in S_i \mid (s_1, \ldots, s_i, \ldots, s_n) \in M |\!\}$*.* □

2.3. Operational interleaving semantics

The operational interleaving semantics for EMPA is defined by means of a slight modification of the SOS approach, in which two distinct relations are introduced. This is due to the fact that the actions executable by a given term may have different priority levels: as we already know, by virtue of the race policy, only those having the highest priority are actually executable. If we call *(potential) move* of a given term a pair composed of an action executable by the term and another term obtained by performing the action, the idea is that of inductively computing all the moves of a given term regardless of priorities, and then imposing the race policy by discarding those moves having a lower priority. Computing inductively all the moves of a given term instead of a single move is motivated, in a stochastic framework, by the fact that the actual executability as well as the execution probability of an action depends upon all the actions which are executable at the same time when it is executable. As a consequence, only if we know all the moves of the subterms of a given term, can we correctly determine its transitions and their rates.

The first relation, denoted by "$\longrightarrow$", is the least subset of $\mathcal{G} \times Act \times \mathcal{G}$ satisfying the inference rule reported in the upper part of Table 1. The side condition associated with the inference rule enforces the race policy by selecting among the moves of a term those having the highest priority level, and then merges together all the moves in which the same action type as well as the same term occur.
The first operation is carried out through function $Race : \mathcal{M}u_{fin}(Act \times \mathcal{G}) \longrightarrow \mathcal{M}u_{fin}(Act \times \mathcal{G})$ defined by

$$\begin{aligned} Race(M) = \{\!| (<a, \tilde{\lambda}>, E) \in M \mid & PL(<a, \tilde{\lambda}>) = -1 \vee \\ & \forall (<b, \tilde{\mu}>, E') \in M.\, PL(<a, \tilde{\lambda}>) \geq PL(<b, \tilde{\mu}>) |\!\} \end{aligned}$$

where function $PL : Act \longrightarrow \mathbf{Z}$ is such that:

- $PL(<a, 0>) = -1$;
- $PL(<a, \lambda>) = 0$;
- $PL(<a, \infty_{l,w}>) = l$.

The second operation is carried out through function $Melt : \mathcal{M}u_{fin}(Act \times \mathcal{G}) \longrightarrow \mathcal{P}_{fin}(Act \times \mathcal{G})$ defined by

$$\begin{aligned} Melt(M) = \{(<a, \tilde{\lambda}>, E) \mid\ & (<a, \tilde{\mu}>, E) \in M \wedge \\ & \tilde{\lambda} = Min\{\!| \tilde{\gamma} \mid (<a, \tilde{\gamma}>, E) \in M \wedge \\ & \qquad PL(<a, \tilde{\gamma}>) = PL(<a, \tilde{\mu}>) |\!\}\} \end{aligned}$$

where function $Min : (ARate \times ARate) \longrightarrow\!\!\!\!\circ\ ARate$ is such that:

- $0 \, Min \, 0 = 0$;
- $\lambda \, Min \, \lambda' = \lambda + \lambda'$;
- $\infty_{l,w} \, Min \, \infty_{l,w'} = \infty_{l,w+w'}$,

and, with abuse of notation, it is extended to multisets of action rates having the same priority by assuming both it acts as the identity function when applied to singletons, and it is commutative and associative when applied to multisets with more than one element.
Observe that function $Race$ never rules out passive moves, and that it cannot be applied while inductively computing moves in order for some admissible transitions not to be eliminated. Furthermore, note that operation Min is a restriction (to the exponential case) of the operation computing the minimum of a set of random variables.

Example 2.1. Consider term

$$E \equiv (<a, \lambda>.\underline{0} + <b, \infty_{1,1}>.\underline{0}) \|_{\{a,b\}} <a, 0>.\underline{0}$$

This term has only one transition labelled with a, λ. It would be ruled out if $Race$ were inductively applied since $PL(<b, \infty_{1,1}>) > PL(<a, \lambda>)$. □

Example 2.2. Consider term

$$E \equiv <a, \lambda>.\underline{0} + <a, \lambda>.\underline{0}$$

It has only one transition labelled with $a, 2\lambda$. □

The second relation, denoted by "$\vdash$", is defined by structural induction as the least subset of $\mathcal{G} \times \mathcal{M}u_{fin}(Act \times \mathcal{G})$ satisfying the axiom and the inference rules reported in the lower part of Table 1. This relation computes the multiset of all the moves of each term, regardless of action priorities. The way in which such a multiset is worked out should result clear from the informal explanation of the semantics of operators given in Section 2.1. The only caution concerns the need to normalise the rates of moves resulting from the synchronisation of the same active action with several passive actions which are either independent of each other (i.e. composed in parallel with a synchronisation set not containing the action type at hand), or mutually exclusive (i.e. composed in alternative). In order to clarify the problem, consider the following example.

Example 2.3. Let us focus our attention on term

$$E \equiv <a, \lambda>.\underline{0} \|_{\{a\}} (<a, 0>.\underline{0} \|_{\emptyset} <a, 0>.\underline{0})$$

which is constructed from the parallel composition of a term that can execute a timed action of type a with a term deriving from the parallel composition of 2 independent terms each of which can execute a passive action of type a. If we compute the rate of a transition resulting from a synchronisation simply as the maximum of the rates of the two interacting actions, for E we get the transitions

$$E \xrightarrow{a,\lambda} \underline{0} \|_{\{a\}} (\underline{0} \|_{\emptyset} <a, 0>.\underline{0})$$
$$E \xrightarrow{a,\lambda} \underline{0} \|_{\{a\}} (<a, 0>.\underline{0} \|_{\emptyset} \underline{0})$$

The problem is that the rate computed for each of these transitions is λ, whereas their correct rate is $\lambda/2$ because in E there exists only one active action whose rate is λ (hence the exit rate from E is λ), and 2 passive actions giving rise to 2 different synchronisations having the same execution probability (due to the assumption made in Section 2.2. about passive actions). If each transition had λ as rate (and this is what happens above), then the exit rate from E would be $2 \cdot \lambda$ (by virtue of Proposition 2.1). The same phenomenon happens if in E the occurrence of "$\|_{\emptyset}$" is replaced by an occurrence of "+". Moreover, the same argument is valid if we have $\infty_{l,w}$ instead of λ; in this case, the weight w must be normalised. □

The operation above is carried out through function $Norm : (AType \times ARate \times ARate \times \mathcal{M}u_{fin}(Act \times \mathcal{G}) \times \mathcal{M}u_{fin}(Act \times \mathcal{G})) \longrightarrow\!\!\!\!\circ\ ARate$ defined as follows:

- $Norm(a, 0, 0, M_1, M_2) = 0$;
- $Norm(a, \lambda, 0, M_1, M_2) = \lambda / (\pi_1(M_2))(<a, 0>)$;
- $Norm(a, 0, \lambda, M_1, M_2) = \lambda / (\pi_1(M_1))(<a, 0>)$;
- $Norm(a, \infty_{l,w}, 0, M_1, M_2) = \infty_{l, w / (\pi_1(M_2))(<a,0>)}$;
- $Norm(a, 0, \infty_{l,w}, M_1, M_2) = \infty_{l, w / (\pi_1(M_1))(<a,0>)}$.

It is worth noting that $Norm(a, \tilde{\lambda}, \tilde{\mu}, M_1, M_2)$ is defined if and only if $\min(\tilde{\lambda}, \tilde{\mu}) = 0$, which is the condition we have required in Section 2.1. in order for a synchronisation to be permitted.

$$\frac{E \vdash M}{E \xrightarrow{a,\tilde{\lambda}} E'} \qquad [(<a, \tilde{\lambda}>, E') \in \mathit{Melt}(\mathit{Race}(M))]$$

$$<a, \tilde{\lambda}>.E \vdash \{\!|\, (<a, \tilde{\lambda}>, E) \,|\!\}$$

$$\frac{E \vdash M}{\begin{array}{l} E/L \vdash \{\!|\, (<a, \tilde{\lambda}>, E'/L) \mid (<a, \tilde{\lambda}>, E') \in M \wedge a \notin L \,|\!\} \oplus \\ \qquad\quad \{\!|\, (<\tau, \tilde{\lambda}>, E'/L) \mid (<a, \tilde{\lambda}>, E') \in M \wedge a \in L \,|\!\} \end{array}}$$

$$\frac{E \vdash M}{E \backslash H \vdash \{\!|\, (<a, \tilde{\lambda}>, E' \backslash H) \mid (<a, \tilde{\lambda}>, E') \in M \wedge \neg(a \in H \wedge \tilde{\lambda} = 0) \,|\!\}}$$

$$\frac{E \vdash M}{E[\varphi] \vdash \{\!|\, (<\varphi(a), \tilde{\lambda}>, E'[\varphi]) \mid (<a, \tilde{\lambda}>, E') \in M \,|\!\}}$$

$$\frac{E_1 \vdash M_1 \qquad E_2 \vdash M_2}{E_1 + E_2 \vdash M_1 \oplus M_2}$$

$$\frac{E_1 \vdash M_1 \qquad E_2 \vdash M_2}{\begin{array}{l} E_1 \|_S E_2 \vdash \{\!|\, (<a, \tilde{\lambda}>, E_1' \|_S E_2) \mid a \notin S \wedge (<a, \tilde{\lambda}>, E_1') \in M_1 \,|\!\} \oplus \\ \qquad \{\!|\, (<a, \tilde{\lambda}>, E_1 \|_S E_2') \mid a \notin S \wedge (<a, \tilde{\lambda}>, E_2') \in M_2 \,|\!\} \oplus \\ \qquad \{\!|\, (<a, \tilde{\gamma}>, E_1' \|_S E_2') \mid a \in S \wedge (<a, \tilde{\lambda}>, E_1') \in M_1 \wedge (<a, \tilde{\mu}>, E_2') \in M_2 \wedge \tilde{\gamma} = \mathit{Norm}(a, \tilde{\lambda}, \tilde{\mu}, M_1, M_2) \,|\!\} \end{array}}$$

$$\frac{E \vdash M}{A \vdash M} \qquad [A \stackrel{\Delta}{=} E]$$

TABLE 1. Inductive rules for EMPA interleaving semantics

EXAMPLE 2.4. Consider term

$$E \equiv E_1 \|_{\{a\}} (E_2 \|_{\emptyset} E_3)$$

where

$$E_1 \equiv <a, \lambda>.\underline{0}$$
$$E_2 \equiv <a, 0>.\underline{0} + <a, 0>.\underline{0}$$
$$E_3 \equiv <a, 0>.\underline{0}$$

Then E_1 has one move $(<a, \lambda>, \underline{0})$, E_2 has one move $(<a, 0>, \underline{0})$ with multiplicity 2, and E_3 has one move $(<a, 0>, \underline{0})$. As a consequence, $E_2 \|_{\emptyset} E_3$ has both move $(<a, 0>, \underline{0} \|_{\emptyset} E_3)$ with multiplicity 2 and move $(<a, 0>, E_2 \|_{\emptyset} \underline{0})$. Therefore, when computing the moves for E, function Norm produces both $(<a, \lambda/3>, \underline{0} \|_{\{a\}} (\underline{0} \|_{\emptyset} E_3))$ with multiplicity 2 and $(<a, \lambda/3>, \underline{0} \|_{\{a\}} (E_2 \|_{\emptyset} \underline{0}))$, and subsequently function Melt produces both $(<a, 2\lambda/3>, \underline{0} \|_{\{a\}} (\underline{0} \|_{\emptyset} E_3))$ and $(<a, \lambda/3>, \underline{0} \|_{\{a\}} (E_2 \|_{\emptyset} \underline{0}))$. □

We are now in a position of introducing the labelled transition system (LTS) representing the semantics of a given term.

DEFINITION 2.5. *The operational interleaving semantics of a term $E \in \mathcal{G}$ is the LTS*

$$\mathcal{I}[\![E]\!] = (\uparrow E, Act, \longrightarrow_E, E)$$

where:

- *$\uparrow E$ is the least subset of $\mathcal{G}$ such that:*
  - *$E \in \uparrow E$;*
  - *if $E_1 \in \uparrow E$ and $E_1 \xrightarrow{a,\tilde{\lambda}} E_2$, then $E_2 \in \uparrow E$;*
- *$\longrightarrow_E$ is the restriction of $\longrightarrow$ to $\uparrow E$.* □

Moreover, given a term $E \in \mathcal{G}$, we denote by $\mathcal{F}[\![E]\!]$ its *functional semantics*, i.e. the LTS obtained from $\mathcal{I}[\![E]\!]$ by dropping action rates.

Based on the operational interleaving semantics, we can formalise the property of temporal closure.

DEFINITION 2.6. *A term $E \in \mathcal{G}$ is said to be temporally closed if and only if $\mathcal{I}[\![E]\!]$ is isomorphic to $\mathcal{I}[\![E \backslash ATy\!pe]\!]$.* □

A temporally closed term cannot execute passive actions, so it models a system whose behaviour is completely specified from the temporal viewpoint. We denote by $\mathcal{T}$ the set of terms in $\mathcal{L}$ that are temporally closed. We also denote by $\mathcal{E}$ the set of terms in $\mathcal{L}$ that are guardedly and temporally closed, i.e. $\mathcal{E} = \mathcal{G} \cap \mathcal{T}$.

2.4. Markovian semantics

The *Markovian semantics* of a term $E \in \mathcal{E}$ is the homogeneous continuous time Markov chain (HCTMC) (Kemeny & Snell, 1977) denoted by $\mathcal{M}[\![E]\!]$, obtained by applying to $\mathcal{I}[\![E]\!]$ an algorithm organised in two phases. The first phase eliminates all the immediate transitions occurring in $\mathcal{I}[\![E]\!]$, thus producing a HCTMC. The second one detects and merges states which are equivalent according to the notion of lumping.

3. STOCHASTIC CONTEXTUAL NETS

In this paper we shall be concerned with the class of the *generalized stochastic Petri nets (GSPNs)* (Ajmone Marsan *et al.*, 1984), which are essentially place/transition nets (Reisig, 1985) equipped with inhibitor arcs whose transitions are either timed (i.e. their durations are expressed by means of exponentially distributed random variables) or immediate (i.e. their durations are zero, so they take precedence over timed ones). Furthermore, GSPN transitions are divided into priority levels and have weights which can depend on the current marking M_{curr}. The race policy is adopted whenever several transitions are simultaneously executable.

Since GSPNs do not admit passive transitions (not involved in the priority mechanism) as well as contextual arcs (Montanari & Rossi, 1995), we propose a new class of nets by introducing the required two extensions.

DEFINITION 3.1. *A contextual generalized stochastic Petri net (CGSPN) is a tuple*

$$(P, U, T, M_0, L, W)$$

such that:

- P *is a set whose elements are called places;*
- U *is a set whose elements are called labels;*
- $T \subseteq \mathcal{M}u_{fin}(P) \times \mathcal{P}_{fin}(P) \times \mathcal{P}_{fin}(P) \times U \times \mathcal{M}u_{fin}(P)$ *whose elements are called transitions;*
- $M_0 \in \mathcal{M}u_{fin}(P)$ *is called the initial marking;*
- $L : T \longrightarrow \mathbb{Z}$ *, called priority function, is such that:*
  - $L(t) = -1$ *if* t *is passive;*
  - $L(t) = 0$ *if* t *is timed;*
  - $L(t) \in \mathbb{N}_+$ *is the priority level of* t *if* t *is immediate;*
- $W : (T \times \mathcal{M}u_{fin}(P)) \longrightarrow \mathbb{R}$*, called weight function, is such that:*
  - $W(t, M_{curr}) = 0$ *if* $L(t) = -1$*;*
  - $W(t, M_{curr}) \in \mathbb{R}_+$ *is the rate of the exponential distribution associated with* t *if* $L(t) = 0$*;*
  - $W(t, M_{curr}) \in \mathbb{R}_+$ *is the weight of* t *if* $L(t) \in \mathbb{N}_+$. □

Given a CGSPN, its places are drawn as circles and its transitions are drawn as either boxes (if timed), or bars (if immediate), or black boxes (if passive), with the appropriate labels. If the current marking of the net is M_{curr}, we draw $M_{curr}(p)$ black dots (called *tokens*) in every place p. Each transition t can be written as

$$({}^{\bullet}t, {}^{\circ}t, \hat{t}) \xrightarrow{u_t} t^{\bullet}$$

where ${}^{\bullet}t$ is the *preset* of t (tokens to be consumed), ${}^{\circ}t$ is the *inhibitor set* of t (tokens to be tested for absence), $\hat{t}$ is the *contextual set* of t (tokens to be tested for presence), u_t is the *label* of t, and $t^{\bullet}$ is the *postset* of t (tokens to be produced). Places and transitions are linked as follows: given a transition t, there is an arrow-headed arc from each place in ${}^{\bullet}t$ to t, a circle-headed arc from each place in ${}^{\circ}t$ to t, an arc from each place in $\hat{t}$ to t, and an arrow-headed arc from t to each place in $t^{\bullet}$.

DEFINITION 3.2. *Let* $N = (P, U, T, M_0, L, W)$ *be a CGSPN.*

- *A marking of* N *is an element of* $\mathcal{M}u_{fin}(P)$.
- *A transition* t *is enabled at marking* M *if* ${}^{\bullet}t \subseteq M$, $dom(M) \cap {}^{\circ}t = \emptyset$ *and* $\hat{t} \subseteq dom(M)$. $E(M)$ *is the set of transitions enabled at marking* M.
- *Transition* t *enabled at* M *can fire whenever either* $L(t) = -1$ *or* $L(t)$ *is the highest priority level among the transitions in* $E(M)$*; the firing of* t *produces marking* $M' = (M \ominus {}^{\bullet}t) \oplus t^{\bullet}$*, written* $M\,[u_t\rangle\, M'$.
- *The reachability set* $R(M)$ *of a marking* M *is the least subset of* $\mathcal{M}u_{fin}(P)$ *such that:*
  - $M \in R(M)$*;*
  - *if* $M_1 \in R(M)$ *and* $M_1\,[u_t\rangle\, M_2$*, then* $M_2 \in R(M)$.
- *The reachability graph of* N *is the LTS* $\mathcal{RG}[\![N]\!] = (R(M_0), U, [\rangle, M_0)$. □

If $U = Act$, from $\mathcal{RG}[\![N]\!]$ one can extract the *functional semantics* of N as the LTS $\mathcal{F}[\![N]\!]$ obtained by keeping only the action type information. And – provided that $\mathcal{RG}[\![N]\!]$ has no passive transitions – also the HCTMC $\mathcal{M}[\![N]\!]$ underlying N by applying the algorithm sketched in Section 2.4.

4. LABEL-ORIENTED NET SEMANTICS

The label-oriented net semantics for EMPA translates its guardedly closed terms into CGSPNs. In this section we start with the definition of places and the function which maps terms onto places, then we proceed with the axiom schema generating the transitions. Finally, we define the net associated with each term.

4.1. Places

The first step consists of introducing a set of places onto which terms will be mapped. Such a set

$$P_{EMPA} = \boldsymbol{CSeq} \cup (\boldsymbol{Conf} \cup \boldsymbol{I}_{AType}) \cup \boldsymbol{Ren}$$

is composed of three classes of places:

- sequential terms decorated with a conflict set;
- conflicts and action inhibitors;
- renaming functions.

The first class of places is based on the following definition.

Definition 4.1. *A term $E \in \mathcal{G}$ is said to be sequential if and only if:*

- $E \equiv \underline{0}$, *or*
- $E \equiv <a, \tilde{\lambda}>.E'$, *or*
- $E \equiv E_1 + E_2$ *with* E_1 *and* E_2 *sequential, or*
- $E \equiv A$ *with* $A \stackrel{\Delta}{=} E'$ *and* E' *sequential.* □

The first class of places

$$\boldsymbol{CSeq} = \{KE \mid K \subseteq \boldsymbol{Conf} \wedge E \equiv \sum_{h=1}^{n} <a_h, \tilde{\lambda}_h>.E_h \in \mathcal{G}\}$$

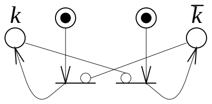
is made out of sequential terms E, each juxtaposed to a conflict set K. When used in the syntax of places, we assume that the sum is associative and commutative, whence derives the summation notation, and it admits $\underline{0}$ as neutral element (the sum over an empty set of sequential terms is taken to be $\underline{0}$). Places in $\boldsymbol{CSeq}$ will have neither inhibitor arcs nor contextual arcs exiting from them.

The second class of places is composed of conflicts and action inhibitors.

Given a set of symbols χ, we define the *set of conflicts*

$$\boldsymbol{Conf} = \chi \cup \bar{\chi}$$

where $\forall k \in \chi.\, \bar{\bar{k}} = k$, and we call *conflict set* any subset K of $\boldsymbol{Conf}$; if K is a conflict set, then the *complement* of K is $\bar{K} = \{\bar{k} \mid k \in K\}$. We introduce a place for each conflict. Place k will be used to prevent the execution of any transition from any place $KE \in \boldsymbol{CSeq}$ such that $\bar{k} \in K$. This allows us to model terms such as $E_1 + E_2$, which is interpreted as the parallel composition of $\{k\}E_1$ and $\{\bar{k}\}E_2$; the associated net is of the following form:

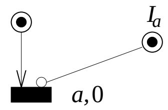


where the transition which fires first prevents the other from occurring. Thus, we are able to give a fully distributed account of the alternative composition operator.

After defining the *set of action inhibitors*

$$\boldsymbol{I}_{AType} = \{I_a \mid a \in \boldsymbol{AType}\}$$

we introduce a place for every action inhibitor. Place I_a will be used to prevent the execution of any transition labelled with a which is forbidden due to a functional abstraction, a temporal restriction, a functional relabelling or a synchronisation on a. For example, the net we wish to associate with $(<a, 0>.\underline{0})\backslash\{a\}$ has the following form:



Places in $\boldsymbol{Conf} \cup \boldsymbol{I}_{AType}$ will have only inhibitor arcs exiting from them.

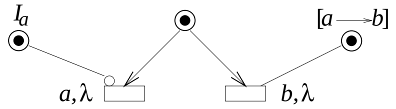
The third class of places

$$\begin{aligned} \boldsymbol{Ren} = \; & \{[a_1 \mapsto a_2] \mid a_1, a_2 \in \boldsymbol{AType} \text{ distinct}\} \cup \\ & \{[a_1 \mapsto_0 a_2] \mid a_1, a_2 \in \boldsymbol{AType} \text{ distinct}\} \cup \\ & \{[a_1, a_2 \mapsto a] \mid a_1, a_2, a \in \boldsymbol{AType} \text{ distinct}\} \end{aligned}$$

is made out of *renaming functions* from $\boldsymbol{Act}$ to $\boldsymbol{Act}$ of the following three types:

- Place $[a_1 \mapsto a_2]$ corresponds to partial function ρ such that
$$\rho(\{<a_1, \tilde{\lambda}>\} \cup V) = \{<a_2, \tilde{\lambda}>\} \cup V$$
- Place $[a_1 \mapsto_0 a_2]$ corresponds to partial function ρ such that: if $\tilde{\lambda} \neq 0$, then
$$\rho(\{<a_1, \tilde{\lambda}>\} \cup V) = \{<a_2, \tilde{\lambda}>\} \cup V$$
- Place $[a_1, a_2 \mapsto a]$ corresponds to partial function ρ such that: if $\min(\tilde{\lambda}, \tilde{\mu}) = 0$, then
$$\rho(\{<a_1, \tilde{\lambda}>, <a_2, \tilde{\mu}>\} \cup V) = \{<a, \max(\tilde{\lambda}, \tilde{\mu})>\} \cup V$$

Places in $\boldsymbol{Ren}$ of the first type are necessary to implement the functional abstraction operator as well as the functional relabelling operator. For example, the net we wish to associate with $(<a, \lambda>.\underline{0})[b/a]$ has the following form:



Moreover, since information about occurrences of functional abstraction, temporal restriction, functional relabelling, and parallel composition operators is not retained when associating places in $\boldsymbol{CSeq}$ with terms (as we shall see after introducing the decomposition function), we always need fresh action types in order to rename binders composed of the occurrences of these operators. The correct link between binders and the related places in $\boldsymbol{CSeq}$ is established by means of places in $\boldsymbol{Ren}$, which are of the first type in the case of functional abstraction or functional relabelling operators, of the second type in the case of temporal restriction operators, and of the third type in the case of parallel composition operators. To achieve this, we must avoid action type clashes, so we assume there is a mechanism for generating new action types whenever necessary. The example below illustrates the problem.

Example 4.1. Consider the following terms:

- $E_/ \equiv (<a, 0>.\underline{0})/\{a\} + <a, 0>.\underline{0}$;
- $E_{[]} \equiv (<a, 0>.\underline{0})[b/a] + <a, 0>.\underline{0}$;
- $E_\backslash \equiv (<a, 0>.\underline{0})\backslash\{a\} + <a, 0>.\underline{0}$;

- $E_{\|} \equiv (<a, 0>.\underline{0} \|_{\{a\}} <a, 0>.\underline{0}) + <a, 0>.\underline{0}$.

If we want to give a label-oriented net semantics to each of them, then the binder occurring in each of them must affect only its scope by completely disregarding the right-hand side operand of the alternative composition operator. To achieve this, we replace the type of the appropriate actions inside the scope of the binder thus obtaining the following terms equivalent to the initial ones:

- $E'_{/} \equiv (<a_{new}, 0>.\underline{0})/\{a\} + <a, 0>.\underline{0}$;
- $E'_{[]} \equiv (<a_{new}, 0>.\underline{0})[b/a] + <a, 0>.\underline{0}$;
- $E'_{\backslash} \equiv (<a_{new}, 0>.\underline{0})\backslash\{a\} + <a, 0>.\underline{0}$;
- $E'_{\|} \equiv (<a_{new,l}, 0>.\underline{0} \|_{\{a\}} <a_{new,r}, 0>.\underline{0}) + <a, 0>.\underline{0}$,

provided that the following inhibitors and renamings are introduced:

- $I_{a_{new}}$ and $[a_{new} \mapsto \tau]$;
- $I_{a_{new}}$ and $[a_{new} \mapsto b]$;
- $I_{a_{new}}$ and $[a_{new} \mapsto_0 a]$;
- $I_{a_{new,l}}$, $I_{a_{new,r}}$ and $[a_{new,l}, a_{new,r} \mapsto a]$. □

Places in Ren will have only contextual arcs exiting from them.

After defining the set of places, we map terms onto them through the *decomposition function*

$$dec : \mathcal{G} \longrightarrow \mathcal{M}u_{fin}(P_{EMPA})$$

which is defined by induction on the syntactical structure of terms as follows: [†]

- $dec(\underline{0}) = \emptyset$;
- $dec(<a, \tilde{\lambda}>.E) = \{\| \emptyset <a, \tilde{\lambda}>.E \|\}$;
- $dec(E/L) = dec(E\langle L := L_{new}\rangle) \oplus \{\| I_{a_{new}} \mid a_{new} \in L_{new} \|\} \oplus \{\| [a_{new} \mapsto \tau] \mid a_{new} \in L_{new} \|\}$;
- $dec(E\backslash H) = dec(E\langle H := H_{new}\rangle) \oplus \{\| I_{a_{new}} \mid a_{new} \in H_{new} \|\} \oplus \{\| [a_{new} \mapsto_0 a] \mid a_{new} \in H_{new} \|\}$;
- $dec(E[\varphi]) = dec(E\langle ch(\varphi) := ch(\varphi)_{new}\rangle) \oplus \{\| I_{a_{new}} \mid a_{new} \in ch(\varphi)_{new} \|\} \oplus \{\| [a_{new} \mapsto \varphi(a)] \mid a_{new} \in ch(\varphi)_{new} \|\}$;
- $dec(E_1 + E_2) = \{k_{new}\}(dec(E_1)) \oplus \{\bar{k}_{new}\}(dec(E_2))$ if $E_1 + E_2$ is not sequential, $dec(E_1 + E_2) = \{\| \emptyset(E_1 + E_2) \|\}$ if $E_1 + E_2$ is sequential;
- $dec(E_1 \|_S E_2) = dec(E_1\langle S := S_{new,l}\rangle) \oplus dec(E_2\langle S := S_{new,r}\rangle) \oplus \{\| I_{a_{new,x}} \mid a_{new,x} \in S_{new,x}, x \in \{l, r\} \|\} \oplus \{\| [a_{new,l}, a_{new,r} \mapsto a] \mid a_{new,x} \in S_{new,x}, x \in \{l, r\} \|\}$;
- $dec(A) = dec(E)$ if $A \stackrel{\Delta}{=} E$.

[†] Given $D \subseteq AType$, with $E\langle D := D_y\rangle$ we denote the term obtained from E by replacing occurrences of elements in D with occurrences of corresponding elements in $D_y = \{a_y \mid a \in D\}$. We also denote by $ch(\varphi)$ the set $\{a \in AType \mid \varphi(a) \neq a\}$. Finally, if $K \subseteq Conf$ and $P \in \mathcal{M}u_{fin}(P_{EMPA})$, then $K(P) = \{\| K(p) \mid p \in \bar{P} \wedge (p = K'E \in CSeq \Longrightarrow K(p) = (K \cup K')E) \wedge (p \in Conf \cup I_{AType} \cup Ren \Longrightarrow K(p) = p) \|\}$.

To help the reader in understanding the decomposition function, for each clause we have written sets of places of different classes on different lines. It is worth noting that there are two clauses for the alternative composition operator: the first one implements the scheme shown when discussing places in $Conf$, the second one is an optimisation of the previous one exploiting the sequentiality of terms in order to reduce the size of nets. Finally, observe that dec is not injective.

EXAMPLE 4.2. Consider term

$$<a, \tilde{\lambda}>.E + <b, \tilde{\mu}>.F$$

Creating (by means of the second clause) only one place $\emptyset(<a, \tilde{\lambda}>.E + <b, \tilde{\mu}>.F)$ with two transitions and two arcs is more convenient than creating (through the first clause) four places $\{k\}<a, \tilde{\lambda}>.E$, $\{\bar{k}\}<b, \tilde{\mu}>.F$, k and $\bar{k}$ with two transitions and six arcs. □

EXAMPLE 4.3. The following pairs of terms give rise to the same decomposition:

$$\begin{array}{rcl} \underline{0} & \text{and} & \underline{0} \|_{\emptyset} \underline{0} \\ E \|_{\emptyset} F & \text{and} & F \|_{\emptyset} E \\ <a, \tilde{\lambda}>.E + <b, \tilde{\mu}>.F & \text{and} & <b, \tilde{\mu}>.F + <a, \tilde{\lambda}>.E \\ <a, \tilde{\lambda}>.E + \underline{0} & \text{and} & <a, \tilde{\lambda}>.E \end{array}$$

Additionally, note that term

$$<a, \tilde{\lambda}>.\underline{0} \|_{\emptyset} <a, \tilde{\lambda}>.\underline{0}$$

is mapped to only one place given by $\emptyset <a, \tilde{\lambda}>.\underline{0}$ with multiplicity 2. This is typical of the label-oriented approach, whereas in the location-oriented one the same term would be mapped to two distinct places as the information about the presence of the parallel composition operator would be encoded inside the two places themselves. □

4.2. Transitions

The second step in the definition of the net semantics consists of connecting places by means of transitions. The set of transitions T_{EMPA} is defined as the least subset of $\mathcal{M}u_{fin}(P_{EMPA}) \times \mathcal{P}_{fin}(P_{EMPA}) \times \mathcal{P}_{fin}(P_{EMPA}) \times (AType \times ARate^{\mathcal{M}u_{fin}(P_{EMPA})}) \times \mathcal{M}u_{fin}(P_{EMPA})$ generated by the axiom schema reported in Table 2, which is subject to the following side conditions: [‡]

1. $n, n_i \in \mathbb{N}_+$ and $m \in \mathbb{N}$;
2. each place in the preset has multiplicity 1, and $a_i \neq a_{i'}$ as well as $K_i \cap \bar{K}_{i'} = \emptyset$ whenever $i \neq i'$;
3. $PL(<a_i, \tilde{\lambda}_{i,h}>) = PL(<a_i, \tilde{\lambda}_{i,h'}>)$, $dec(E_{i,h}) = dec(E_{i,h'})$, and $G_i \equiv \sum_d <b_{i,d}, \tilde{\mu}_{i,d}>.F_{i,d}$ such that $b_{i,d} \neq a_i$ or $PL(<b_{i,d}, \tilde{\mu}_{i,d}>) \neq PL(<a_i, \tilde{\lambda}_{i,1}>)$ or $dec(F_{i,d}) \neq dec(E_{i,1})$;
4. $\{<a, \tilde{\lambda}>\} = (\bigcirc_{j=1}^{m} \rho_j)(\bigcup_{i=1}^{n} \{<a_i, Min\{\| \tilde{\lambda}_{i,h} \mid 1 \leq h \leq n_i\}> \|\})$ where $\bigcirc_{j=1}^{m}$ is the m-ary composition of functions;

[‡] Let $i, i' = 1, \ldots, n$ and $h, h' = 1, \ldots, n_i$ and $j = 1, \ldots, m$.

$$(\bigoplus_{i=1}^{n} \{\!| K_i \sum_{h=1}^{n_i} <a_i, \tilde{\lambda}_{i,h}>.E_{i,h} + G_i |\!\}, \bigcup_{i=1}^{n} \bar{K}_i \cup \{I_a\}, \bigcup_{j=1}^{m} \{\rho_j\}) \xrightarrow{a, norm(<a,\tilde{\lambda}>, f_1, f_2, f_3, E)} \bigoplus_{i=1}^{n} dec(E_{i,1}) \oplus \bigoplus_{i=1}^{n} K_i$$

TABLE 2. Axiom schema for EMPA label-oriented net semantics

5. $f_1 = M_{curr}(K_i \sum_{h=1}^{n_i} <a_i, \tilde{\lambda}_{i,h}>.E_{i,h} + G_i)$ if there exists i such that $\tilde{\lambda}_{i,1} \in \mathbb{R}_+ \cup Inf$, $f_1 = 1$ otherwise;
6. $f_2 = 1$ if there is no i such that $\tilde{\lambda}_{i,1} = 0$, $f_2 = \prod \{\!| n_i \cdot M_{curr}(K_i \sum_{h=1}^{n_i} <a_i, \tilde{\lambda}_{i,h}>.E_{i,h} + G_i) \mid 1 \leq i \leq n \wedge \tilde{\lambda}_{i,1} = 0 |\!\}$ otherwise;
7. $f_3 = 1$ if $m = 0$ or $M_{curr}(\rho_j) > 0$ for every j, $f_3 = 0$ otherwise;
8. $E \equiv E_{i,1}$ if there exists i such that $\tilde{\lambda}_{i,1} \in \mathbb{R}_+ \cup Inf$, $E \equiv \perp$ otherwise;
9. $norm : (Act \times \mathbb{N}^{\mathcal{M}u_{fin}(P_{EMPA})} \times \mathbb{N}^{\mathcal{M}u_{fin}(P_{EMPA})} \times \mathbb{N}^{\mathcal{M}u_{fin}(P_{EMPA})} \times (\mathcal{G} \cup \perp)) \longrightarrow ARate^{\mathcal{M}u_{fin}(P_{EMPA})}$ such that:

   - $norm(<a, 0>, f_1, f_2, f_3, E) = 0$;
   - $norm(<a, \lambda>, f_1, f_2, f_3, E) = \lambda \cdot f_1 \cdot f_2 / s$ where $s = \sum \{\!| f_2' \cdot f_3' \mid (M_1', K' \cup I_a, C', <a, norm(<a, \lambda>, f_1, f_2', f_3', E')>, M_2') \in T_{EMPA} \wedge dec(E') = dec(E) |\!\}$;
   - $norm(<a, \infty_{l,w}>, f_1, f_2, f_3, E) = \infty_{l,w \cdot f_1 \cdot f_2 / s}$ where $s = \sum \{\!| f_2' \cdot f_3' \mid (M_1', K' \cup I_a, C', <a, norm(<a, \infty_{l,w}>, f_1, f_2', f_3', E')>, M_2') \in T_{EMPA} \wedge dec(E') = dec(E) |\!\}$.

Let us explain the axiom schema.
The preset of the transition is a finite nonempty set of places in $CSeq$, which contains more than one place if and only if the transition is due to a synchronisation.
The inhibitor set of the transition is a finite nonempty set of places in $Conf \cup I_{AType}$. There is a place for every conflict name which is complementary to a conflict name appearing in the conflict set of some place in the preset. There is also a place for the action type labelling the transition. These places prevent the transition from firing (i.e. they contain tokens) if either a transition with a place in its preset having a complementary conflict name has been previously fired, or the action type labelling the transition is subject to functional abstraction, temporal restriction, functional relabelling or synchronisation.
The contextual set of the transition is a finite (possibly empty) set of places in Ren. These places allow the transition to be fired (i.e. they contain tokens) if there is a composition of renamings that, when applied to the set of actions appearing in the preset, produces the action labelling the transition.
The postset of the transition is the multiset composed of both the places representing the decomposition of terms appearing in the preset, and the places associated with the conflict names appearing in the preset. As a consequence, if the transition fires then all the transitions with a place in their preset having a complementary conflict name are inhibited.
The axiom can be applied if the side conditions listed above are met.
Condition 2 states that all the places in the preset are distinct and have different action types as well as disjoint conflict sets. This avoids the generation of synchronisation transitions whose presets contain places associated with terms which cannot be synchronised.

EXAMPLE 4.4. If we consider term

$$<a, \lambda>.E + <b, 0>.F$$

then its decomposition results in place $\emptyset <a, \lambda>.E + <b, 0>.F$. Without the first constraint of the condition, a synchronisation of this place with itself would be made possible by renaming $[a, b \mapsto c]$. Therefore, the first constraint avoids the synchronisation of a place with itself, and this is consistent with the clause of dec for the parallel composition operator.
Consider now term

$$<a, \lambda>.E \|_{\{a\}} (<a, 0>.F \|_{\emptyset} <a, 0>.G)$$

whose decomposition generates places $\emptyset <a_{new,l}, \lambda>.E$, $\emptyset <a_{new,r}, 0>.F$, and $\emptyset <a_{new,r}, 0>.G$ in the case $F \not\equiv G$. Without constraint $a_i \neq a_{i'}$, a synchronisation of the second place with the third one would be possible. Also this constraint is consistent with the clause of dec for the parallel composition operator.
Finally, consider term

$$(<a, \lambda>.E \|_{\emptyset} \underline{0}) + (<b, 0>.F \|_{\emptyset} \underline{0})$$

whose decomposition generates places $\{k\} < a, \lambda > .E$ and $\{\bar{k}\} <b, 0>.F$. Without constraint $K_i \cap \bar{K}_{i'} = \emptyset$, a synchronisation between these places would be made possible by renaming $[a, b \mapsto c]$. Such a constraint is consistent with the first clause of dec for the alternative composition operator. □

Condition 3 states that the summands highlighted for each place in the preset of the transition are all and only those involving the same action type a_i, actions $< a_i, \tilde{\lambda}_{i,h} >$ having the same priority, and terms $E_{i,h}$ having the same decomposition. By proceeding in this way, we reduce the number of transitions (as it will result clear after explaining the fourth condition). We require terms to have the same decomposition instead of being syntactically identical because function dec is not injective.
Condition 4 states that the action type and the rate

(hereafter called the *basic rate*) labelling the transition are the result of the application of the composition of the renaming functions appearing in the contextual set to the set of actions highlighted for each place in the preset (if the contextual set is empty, then the composition is taken to be the identity function over Act). For each place in the preset, the set of actions is collapsed into a single action with the same type whose rate is computed according to operation Min.

EXAMPLE 4.5. If we consider term

$$<a, \lambda>.E\|_{\{a\}}(<a, 0>.F\|_{\{a\}}<a, 0>.G)$$

then its decomposition comprises places $\emptyset<a_{new,l}, \lambda>.E$, $\emptyset < a'_{new,l}, 0 > .F$, and $\emptyset < a'_{new,r}, 0 > .G$ in the case $F \not\equiv G$, as well as renamings $[a_{new,l}, a_{new,r} \mapsto a]$ and $[a'_{new,l}, a'_{new,r} \mapsto a_{new,r}]$. Applying the second renaming to $\{<a_{new,l}, \lambda>,<a'_{new,l}, 0>,<a'_{new,r}, 0>\}$ results in $\{<a_{new,l}, \lambda>,<a_{new,r}, 0>\}$, and then we obtain $\{<a, \lambda>\}$ through the first renaming. □

EXAMPLE 4.6. If we consider term

$$<a, \lambda>.E+<a, \mu>.F$$

then its decomposition is place $\emptyset<a, \lambda>.E+<a, \mu>.F$ and we generate only one transition labelled with $a, \lambda+\mu$ instead of two transitions labelled with a, λ and a, μ, respectively, whenever $dec(E) = dec(F)$. □

Condition 5 defines factor f_1 as the number of tokens in the place (if any) of the preset which contributes to the transition with an active action. We must take into account this factor because the rate of the transition is proportional to it.

EXAMPLE 4.7. If we consider term

$$<a, \lambda>.E\|_{\emptyset}<a, \lambda>.E$$

then its decomposition gives rise to place $\emptyset<a, \lambda>.E$ with multiplicity 2, i.e. marked with two tokens. This place has one outgoing transition whose rate must be 2λ at the first firing and λ at the second firing, in order for the reachability graph of the net semantics of the term at hand to be consistent with its interleaving semantics. The needed piece of information is recorded in factor f_1. □

Condition 6 defines factor f_2 as the product, over the set of places (if any) of the preset which contribute to the transition with a passive action, of the number of highlighted summands times the number of tokens. We must take into account this factor because the rate of the transition is proportional to it.

EXAMPLE 4.8. If we consider term

$$<a, \lambda>.E\|_{\{a\}}((<a, 0>.F\|_{\emptyset}<a, 0>.F\|_{\emptyset}<a, 0>.F)+ (<a, 0>.G+<a, 0>.G))$$

then its decomposition produces three places in $CSeq$ given by $p_1 = \emptyset<a_{new,l}, \lambda>.E$ marked with one token, $p_2 = \{k\}<a_{new,r}, 0>.F$ marked with three tokens, and $p_3 = \{k\}<a_{new,r}, 0>.G+<a_{new,r}, 0>.G$ marked with one token. Place p_1 has two outgoing transitions $t_{1,2}$ and $t_{1,3}$, labelled with a, resulting from the synchronisation with p_2 and p_3, respectively. Transition $t_{1,2}$ has basic rate λ, factor $f_1 = M_{curr}(p_1)$, and factor $f_2 = 1 \cdot M_{curr}(p_2)$; when fired, its rate is $\lambda \cdot 1 \cdot (1 \cdot 3) = 3 \cdot \lambda$. Transition $t_{1,3}$ has basic rate λ, factor $f_1 = M_{curr}(p_1)$, and factor $f_2 = 2 \cdot M_{curr}(p_3)$; when fired, its rate is $\lambda \cdot 1 \cdot (2 \cdot 1) = 2 \cdot \lambda$. However, to be consistent with the interleaving semantics of the term at hand, the right rate for the firing of $t_{1,2}$ is $3\lambda/5$ and right rate for the firing of $t_{1,3}$ is $2\lambda/5$. The normalizing factor $5 = 3 + 2$ will be computed by $norm$. □

Condition 7 defines factor f_3 to be 1 if all the places in the contextual set are marked, 0 otherwise. We need this factor to correctly carry out the normalisation.

EXAMPLE 4.9. Reconsider the term of Example 4.8. Due to renamings $[a \mapsto b]$ and $[b \mapsto a]$, we can envisage other two transitions $t'_{1,2}$ and $t'_{1,3}$ obtained from $t_{1,2}$ and $t_{1,3}$, respectively, by applying these two renamings as well. As a consequence, the normalizing factor for the four transitions becomes $3 + 2 + 3 + 2 = 10$. To obtain again the right normalizing factor, we multiply each of its summands by the related factor f_3. In the case studied, we have $3 \cdot 1 + 2 \cdot 1 + 3 \cdot 0 + 2 \cdot 0 = 5$. □

Condition 8 defines E to be the term appearing in the place (if any) of the preset which contributes to the transition with an active action. We need this term in order to correctly carry out the normalisation.

EXAMPLE 4.10. If we consider term

$$(<a, \lambda>.F+<a, \lambda>.G)\|_{\{a\}}<a, 0>.\underline{0}$$

where $dec(F) \neq dec(G)$, then its decomposition consists of places $\emptyset<a_{new,l}, \lambda>.F+<a_{new,l}, \lambda>.G$ and $\emptyset<a_{new,r}, 0>.\underline{0}$. There are two transitions both labelled with a, λ whose postsets are $dec(F)$ and $dec(G)$, respectively. The rates of these transitions must be separately normalised since they result from the participation of two different active actions. This is achieved by labelling the first transition with F and the second transition with G. □

Condition 9 defines the normalising function. It computes the rate of each active transition t by dividing the product of the related basic rate and factors f_1 and f_2 by the normalising factor s. To determine s, all the transitions labelled as t (up to the second and the third factors) are to be singled out, because these transitions are all and only those having in their presets the same place contributing with the same active action. The normalising factor, given by the sum of the products of the second and the third factors of the transitions above, comprises infinitely many marking-dependent summands: when a marking is fixed, only finitely many summands will be nonzero since the domain of a marking is finite.

4.3. Nets associated with EMPA terms

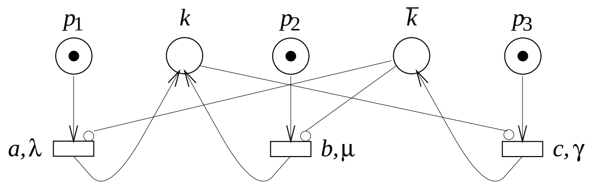
We are now able to define the label-oriented net semantics for EMPA.

Definition 4.2. *The label-oriented net semantics of a term $E \in \mathcal{G}$ is the CGSPN*
$\mathcal{L}ab\mathcal{ON}[\![E]\!] = (P_{EMPA}, U, T_{EMPA}, M_0, L, W)$
where:

- $U = AType \times ARate^{\mathcal{M}u_{fin}(P_{EMPA})}$;
- $M_0 = dec(E)$;
- $L : T_{EMPA} \longrightarrow \mathbb{Z}$ *where:*
  - $L(M_1, K \cup I_a, C, <a, norm(<a, 0>, f_1, f_2, f_3, E)>, M_2) = -1$;
  - $L(M_1, K \cup I_a, C, <a, norm(<a, \lambda>, f_1, f_2, f_3, E)>, M_2) = 0$;
  - $L(M_1, K \cup I_a, C, <a, norm(<a, \infty_{l,w}>, f_1, f_2, f_3, E)>, M_2) = l$;
- $W : (T_{EMPA} \times \mathcal{M}u_{fin}(P_{EMPA})) \longrightarrow \mathbb{R}$ *where:*
  - $W((M_1, K \cup I_a, C, <a, norm(<a, 0>, f_1, f_2, f_3, E)>, M_2), M_{curr}) = 0$;
  - $W((M_1, K \cup I_a, C, <a, norm(<a, \lambda>, f_1, f_2, f_3, E)>, M_2), M_{curr}) = (norm(<a, \lambda>, f_1, f_2, f_3, E))(M_{curr})$;
  - $W((M_1, K \cup I_a, C, <a, norm(<a, \infty_{l,w}>, f_1, f_2, f_3, E)>, M_2), M_{curr}) = w'(M_{curr})$ *whenever* $\infty_{l,w'} = norm(<a, \infty_{l,w}>, f_1, f_2, f_3, E)$. □

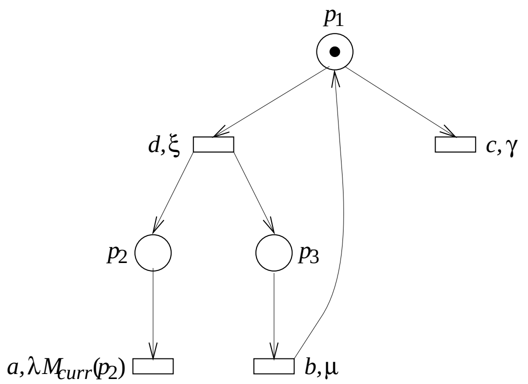
We report below some examples showing that the CGSPNs associated with EMPA terms are not necessarily 1-safe and give (except for sequential terms) a linear representation of alternative compositions. In these examples we draw only reachable places, and the factors involved in the normalisation will be written accordingly.

Example 4.11. Consider term
$E \equiv (<a, \lambda>.\underline{0} \|_{\emptyset} <b, \mu>.\underline{0}) + <c, \gamma>.\underline{0}$
Then $\mathcal{L}ab\mathcal{ON}[\![E]\!]$ is the following net:



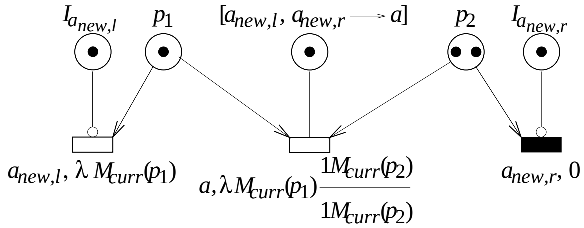
where $p_1 = \{k\} < a, \lambda > .\underline{0}$, $p_2 = \{k\} < b, \mu > .\underline{0}$, and $p_3 = \{\bar{k}\} <c, \gamma>.\underline{0}$. Since E is not sequential, the second clause of dec for the alternative composition operator cannot be applied. □

Example 4.12. Take the recursive sequential term
$A \stackrel{\Delta}{=} <d, \xi>.(<a, \lambda>.\underline{0} \|_{\emptyset} <b, \mu>.A) + <c, \gamma>.\underline{0}$
Then $\mathcal{L}ab\mathcal{ON}[\![A]\!]$ is the following net:

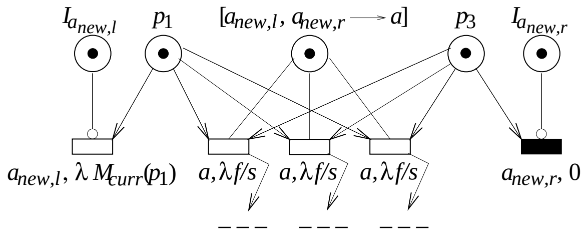


where $p_1 = \emptyset A$, $p_2 = \emptyset <a, \lambda>.\underline{0}$, and $p_3 = \emptyset <b, \mu>.A$. Since in this case A is sequential, the second clause of dec for the alternative composition operator has been applied. If the first clause had been used instead, we would have produced a net with infinitely many reachable places (as well as infinitely many live transitions), because conflicts would have come into play: each application of dec to A would have generated new conflict names, thus preventing places from being reused. □

Example 4.13. Consider term
$E \equiv <a, \lambda>.\underline{0} \|_{\{a\}} (<a, 0>.\underline{0} \|_{\emptyset} <a, 0>.\underline{0})$
Then $\mathcal{L}ab\mathcal{ON}[\![E]\!]$ is the following net:



where $p_1 = \emptyset <a_{new,l}, \lambda>.\underline{0}$ and $p_2 = \emptyset <a_{new,r}, 0>.\underline{0}$. Place p_2 contains more than one token, hence the net is not 1-safe. There is only one enabled transition; it can fire only once with rate λ.
If we consider term
$E' \equiv (<a, \lambda>.\underline{0} \|_{\emptyset} <a, \lambda>.\underline{0}) \|_{\{a\}} (<a, 0>.\underline{0} \|_{\emptyset} <a, 0>.\underline{0})$
then we get the same net as above, but with two tokens in place p_1. Again there is only one enabled transition; it can fire the first time with rate 2λ and the second time with rate λ.
Finally, consider term
$E'' \equiv <a, \lambda>.\underline{0} \|_{\{a\}} (<a, 0>.F_1 + <a, 0>.F_2 + <a, 0>.F_3)$
where $dec(F_1) \neq dec(F_2)$, $dec(F_1) \neq dec(F_3)$, $dec(F_2) \neq dec(F_3)$. Then $\mathcal{L}ab\mathcal{ON}[\![E'']\!]$ is the following net:



where $p_3 = \emptyset <a_{new,r}, 0>.F_1 + <a_{new,r}, 0>.F_2 + <a_{new,r}, 0>.F_3$ and $f = M_{curr}(p_1) \cdot 1 \cdot M_{curr}(p_3)$ and $s = 1 \cdot$

$M_{curr}(p_3) + 1 \cdot M_{curr}(p_3) + 1 \cdot M_{curr}(p_3)$. There are three mutually exclusive enabled transitions: each of them can fire only once with rate $\lambda/3$. □

5. RETRIEVABILITY PRINCIPLES

In this section we want to prove that the label-oriented net semantics satisfies the two following consistency principles:

- *Functional retrievability principle*: the functional semantics of each term is retrievable from its net semantics. This can be formalised by requiring that $\mathcal{F}[\![E]\!]$ is bisimilar to $\mathcal{F}[\![\mathcal{L}ab\mathcal{ON}[\![E]\!]]\!]$ for each term $E \in \mathcal{G}$.
- *Performance retrievability principle*: the Markovian semantics of each term is retrievable from its net semantics. This can be formalised by requiring that $\mathcal{M}[\![E]\!]$ is bisimilar to $\mathcal{M}[\![\mathcal{L}ab\mathcal{ON}[\![E]\!]]\!]$ for each term $E \in \mathcal{E}$.

To this aim, we do not separately consider the functional behaviour and the performance behaviour. We resort instead to a proper relation accounting for both aspects, called *strong extended Markovian bisimulation* (Bernardo *et al.*, 1995b), which is a generalisation of an idea in (Hillston, 1994; Hermanns & Rettelbach, 1994) to our setting with immediate actions.

DEFINITION 5.1. *Let $A_1 = (S_1, U, \longrightarrow_1, s_{01})$ and $A_2 = (S_2, U, \longrightarrow_2, s_{02})$ be two LTSs such that $U = Act$. We say that A_1 is strongly extended-Markovian bisimilar (strongly EMB) to A_2 if and only if there exists a relation $\mathcal{B} \subseteq S_1 \times S_2$ such that:*

- $s_{01}\,\mathcal{B}\,s_{02}$;
- *for every $s_1 \in S_1$ and $s_2 \in S_2$, if $s_1\,\mathcal{B}\,s_2$ then for each $<a, \tilde{\lambda}> \in Act$*
  - *for every $s_1' \in S_1$ such that*
  $$s_1 \xrightarrow{a,\tilde{\lambda}}_1 s_1'$$
  there exist $s_2' \in S_2$ and $\tilde{\mu} \in ARate$ such that
  $$s_2 \xrightarrow{a,\tilde{\mu}}_2 s_2'$$
  and
  $$s_1'\,\mathcal{B}\,s_2'$$
  and
  $$PL(<a, \tilde{\lambda}>) = PL(<a, \tilde{\mu}>)$$
  and
  $$\begin{aligned} &Min\{\!|\, \tilde{\gamma} \mid s_1 \xrightarrow{a,\tilde{\gamma}}_1 s_1'' \wedge \\ &\qquad PL(<a, \tilde{\gamma}>) = PL(<a, \tilde{\lambda}>) \,|\!\} = \\ &Min\{\!|\, \tilde{\gamma} \mid s_2 \xrightarrow{a,\tilde{\gamma}}_2 s_2'' \wedge \\ &\qquad PL(<a, \tilde{\gamma}>) = PL(<a, \tilde{\mu}>) \,|\!\} \end{aligned}$$
  - *for every $s_2' \in S_2$ such that*
  $$s_2 \xrightarrow{a,\tilde{\lambda}}_2 s_2'$$
  there exist $s_1' \in S_1$ and $\tilde{\mu} \in ARate$ such that
  $$s_1 \xrightarrow{a,\tilde{\mu}}_1 s_1'$$
  and
  $$s_1'\,\mathcal{B}\,s_2'$$
  and
  $$PL(<a, \tilde{\lambda}>) = PL(<a, \tilde{\mu}>)$$
  and
  $$\begin{aligned} &Min\{\!|\, \tilde{\gamma} \mid s_1 \xrightarrow{a,\tilde{\gamma}}_1 s_1'' \wedge \\ &\qquad PL(<a, \tilde{\gamma}>) = PL(<a, \tilde{\mu}>) \,|\!\} = \\ &Min\{\!|\, \tilde{\gamma} \mid s_2 \xrightarrow{a,\tilde{\gamma}}_2 s_2'' \wedge \\ &\qquad PL(<a, \tilde{\gamma}>) = PL(<a, \tilde{\lambda}>) \,|\!\} \end{aligned}$$

□

We write $s_1 \sim_{EM} s_2$ to indicate that the two states are strongly EMB.

PROPOSITION 5.1. *Let $E \in \mathcal{G}$. For each $<a, \tilde{\lambda}> \in Act$ and $E' \in \mathcal{G}$ such that*
$$E \xrightarrow{a,\tilde{\lambda}} E'$$
there exist $\tilde{\mu} \in ARate$ and $M' \in \mathcal{M}u_{fin}(P_{EMPA})$ such that
$$dec(E) \xrightarrow{a,\tilde{\mu}} M'$$
and
$$E' \sim_{EM} M'$$
and
$$PL(<a, \tilde{\lambda}>) = PL(<a, \tilde{\mu}>)$$
and
$$\begin{aligned} &Min\{\!|\, \tilde{\gamma} \mid E \xrightarrow{a,\tilde{\gamma}} E' \wedge \\ &\qquad PL(<a, \tilde{\gamma}>) = PL(<a, \tilde{\lambda}>) \,|\!\} = \\ &Min\{\!|\, \tilde{\gamma} \mid dec(E) \xrightarrow{a,\tilde{\gamma}} M'' \wedge \\ &\qquad PL(<a, \tilde{\gamma}>) = PL(<a, \tilde{\mu}>) \,|\!\} \end{aligned}$$

Proof. Assume that the race policy is adopted neither in EMPA nor in CGSPNs (note that if two states are strongly EMB, they remain so after applying the race policy). Since function $Melt$ is irrelevant from the point of view of the strong EMB, we do not consider it. We demonstrate the result by induction on the proof of transition $E \xrightarrow{a,\tilde{\lambda}} E'$.

- If the rule applied to obtain the transition is that for the prefix operator, then $E \equiv <a, \tilde{\lambda}>.E'$ and $dec(E) = \{\!|\, \emptyset E \,|\!\}$. Therefore $dec(E) \xrightarrow{a,\tilde{\lambda}} dec(E')$ and the result follows. Note that, for the net transition, $f_1 = f_2 = f_3 = s = 1$.
- If the last rule applied to obtain the transition is that for the functional abstraction operator, then $E \equiv F/L$ and $E' \equiv F'/L$ where $F \xrightarrow{b,\tilde{\lambda}} F'$. We consider only the case $b \in L$ and $a = \tau$. By the induction hypothesis, we obtain that $F \sim_{EM} dec(F)$ hence $F\langle L := L_{new}\rangle \sim_{EM} dec(F\langle L := L_{new}\rangle)$. Since $dec(F/L) = dec(F\langle L := L_{new}\rangle) \oplus \{\!|\, I_{a_{new}} \mid a_{new} \in L_{new} \,|\!\} \oplus \{\!|\, [a_{new} \mapsto \tau] \mid a_{new} \in L_{new} \,|\!\}$, the result follows. Observe that we have applied the fact that, given markings M_1 and M_2 where action type c occurs in both of them while action type c_{new} occurs in none of them, if $M_1 \sim_{EM} M_2$ then $M_1\langle c := c_{new}\rangle \oplus I_{c_{new}} \oplus [c_{new} \mapsto c] \sim_{EM} M_2\langle c := c_{new}\rangle \oplus I_{c_{new}} \oplus [c_{new} \mapsto c]$.

- The other unary operators can be treated similarly.
- If the last rule applied to obtain the transition is that for the alternative composition operator, then $E \equiv E_1 + E_2$. We consider only the case $E' \equiv E'_1$.
If E is sequential, then it is easy to check that $\mathit{dec}(E_1 + E_2) \xrightarrow{a,\tilde{\mu}} \mathit{dec}(E'_1)$ and the result immediately follows.
If E is not sequential, then the result stems from the fact that, given markings M_1, M_2 and M_3 where the conflicts and the fresh action types occurring in M_3 are distinct from the ones occurring in M_1 and M_2, and given a new conflict k, we have $M_1 \oplus \{k\} \oplus \{k\}(M_2) \oplus \{\bar{k}\}(M_3) \sim_{EM} M_1 \oplus M_2$.
- If the last rule applied to obtain the transition is that for the parallel composition operator, then $E \equiv E_1 \|_S E_2$. We consider only the case of a synchronisation on $a \in S$, and we assume that $E' \equiv E'_1 \|_S E'_2$ where $E_1 \xrightarrow{a,\lambda} E'_1$ and $E_2 \xrightarrow{a,0} E'_2$. By the induction hypothesis, we have that $E_1 \sim_{EM} \mathit{dec}(E_1)$ and $E_2 \sim_{EM} \mathit{dec}(E_2)$, thus $E_1\langle S := S_{new,l}\rangle \sim_{EM} \mathit{dec}(E_1\langle S := S_{new,l}\rangle)$ and $E_2\langle S := S_{new,r}\rangle \sim_{EM} \mathit{dec}(E_2\langle S := S_{new,r}\rangle)$. Since $\mathit{dec}(E_1\|_S E_2) = \mathit{dec}(E_1\langle S := S_{new,l}\rangle) \oplus \mathit{dec}(E_2\langle S := S_{new,r}\rangle) \oplus \{\!|\, I_{a_{new,x}} \mid a_{new,x} \in S_{new,x}, x \in \{l,r\} \,|\!\} \oplus \{\!|\, [a_{new,l}, a_{new,r} \mapsto a] \mid a_{new,x} \in S_{newx}, x \in \{l,r\} \,|\!\}$, the result follows. Concerning rates, we have that, by definition of Norm, $\tilde{\lambda}$ is given by λ divided by the number of passive moves of E_2 labelled with a. In turn, rate λ is the result of the division of a basic rate ξ (occurring in E_1) by the number of alternative passive moves with which the move labelled with a, ξ can be synchronised. If we apply operation Min to the rates labelling all the alternative transitions deriving from a synchronisation involving the move above, then Norm guarantees us that the result is ξ. From the point of view of the net, if we apply operation Min to the rates $\xi \cdot f_1 \cdot f_2 \cdot f_3/s$ labelling all the firable alternative transitions deriving from a synchronisation involving the place in which ξ occurs, the result is exactly $f_1 \cdot \xi$ where f_1 is due to the unsafeness of the net.

□

PROPOSITION 5.2. *Let $E \in \mathcal{G}$. For each $<a, \tilde{\lambda}> \in \mathit{Act}$ and $M' \in \mathcal{M}u_{fin}(P_{EMPA})$ such that*

$$\mathit{dec}(E) \xrightarrow{a,\tilde{\lambda}} M'$$

there exist $\tilde{\mu} \in \mathit{ARate}$ and $E' \in \mathcal{G}$ such that

$$E \xrightarrow{a,\tilde{\mu}} E'$$

and

$$E' \sim_{EM} M'$$

and

$$PL(<a, \tilde{\lambda}>) = PL(<a, \tilde{\mu}>)$$

and

$$\begin{aligned} \mathit{Min}\{\!|\, \tilde{\gamma} \mid E \xrightarrow{a,\tilde{\gamma}} E' \wedge \\ PL(<a,\tilde{\gamma}>) = PL(<a,\tilde{\mu}>) \,|\!\} = \\ \mathit{Min}\{\!|\, \tilde{\gamma} \mid \mathit{dec}(E) \xrightarrow{a,\tilde{\gamma}} M'' \wedge \\ PL(<a,\tilde{\gamma}>) = PL(<a,\tilde{\lambda}>) \,|\!\} \end{aligned}$$

Proof. By induction on the definition of dec.

□

The following theorem is a straightforward consequence of the propositions above.

THEOREM 5.3. *Let $E \in \mathcal{G}$. Then $\mathcal{I}[\![E]\!]$ is strongly EMB to $\mathcal{RG}[\![\mathcal{L}ab\mathcal{ON}[\![E]\!]]\!]$.* □

As shown in (Bernardo *et al.*, 1995b), the following results can be obtained by the theorem above.

COROLLARY 5.4. *Let $E \in \mathcal{G}$. Then $\mathcal{F}[\![E]\!]$ is bisimilar to $\mathcal{F}[\![\mathcal{L}ab\mathcal{ON}[\![E]\!]]\!]$.* □

COROLLARY 5.5. *Let $E \in \mathcal{E}$. Then $\mathcal{M}[\![E]\!]$ is bisimilar to $\mathcal{M}[\![\mathcal{L}ab\mathcal{ON}[\![E]\!]]\!]$.* □

6. NET OPTIMISATION

The purpose of this paper is to introduce a net semantics for EMPA which generates nets which are more compact than those obtained following the location-oriented approach. At first sight, this goal has not been achieved. In fact, so far, we have constructed nets with infinitely many places and infinitely many transitions.

To overcome this, we have firstly to consider only reachable places (as done implicitly in all the examples of Section 4); such a set can be inductively defined by starting with the set of places in the initial marking. Secondly, looking carefully at the label-oriented net semantics and in particular its decomposition function, we can realize that a lot of places, together with the related transitions, can be ruled out.

A first relevant instance is given by the inhibitor places associated with action inhibitors, which can all be removed. The reason is that, as soon as a token is deposited in an inhibitor place, all the transitions having this place in their inhibitor set cannot fire any more; in fact, tokens are never removed from inhibitor places. Likewise, all the contextual places associated with renaming functions can be ruled out as well. Here, the reason is that, as soon as a token is deposited in a contextual place, all the transitions having this place in their contextual set can always fire; again, tokens are never removed from contextual places.

The thoughts above lead us to the following definition, where the semantic model adopted is the class of *passive generalized stochastic Petri nets (PGSPNs)*, which are GSPNs enriched with passive transitions not involved in the priority mechanism.

DEFINITION 6.1. *The optimised label-oriented net semantics of a term $E \in \mathcal{G}$ is the PGSPN*

$$\mathcal{OL}ab\mathcal{ON}[\![E]\!] = (P, U, T, M_0, L, W)$$

where:

- $P = P_0 - (I_{AType} \cup Ren)$ *where* P_0 *is the least subset of* P_{EMPA} *such that:*
  - $dom(dec(E)) \subseteq P_0$;
  - *if* $dom(M_1 \oplus C) \subseteq P_0$ *and* $(M_1, K \cup I_a, C, <a, norm(<a, \tilde{\lambda}>, f_1, f_2, f_3, E)>, M_2) \in T_{EMPA}$, *then* $dom(M_2 \oplus K) \subseteq P_0$;
- $U = AType \times ARate^{\mathcal{M}u_{fin}(P)}$;
- $T = \{(M_1,\ K,\ <a,\ norm(<a,\ \tilde{\lambda}>, f_1,\ f_2,\ 1,\ E)>, M_2 - (I_{AType} \cup Ren)) \in \mathcal{M}u_{fin}(P) \times \mathcal{P}_{fin}(P) \times U \times \mathcal{M}u_{fin}(P) \mid (M_1, K \cup I_a, C, <a, norm(<a, \tilde{\lambda}>, f_1, f_2, f_3, E)>, M_2) \in T_{EMPA} \wedge I_a \notin P_0 \wedge C \in P_0\}$;
- $M_0 = dec(E) - (I_{AType} \cup Ren)$;
- $L : T \longrightarrow \mathbb{Z}$ *and* $W : (T \times \mathcal{M}u_{fin}(P)) \longrightarrow \mathbb{R}$ *are defined as for* $\mathcal{L}ab\mathcal{ON}[\![E]\!]$. □

The following theorem justifies that the present optimisation is conservative, in the sense that it does not change the behaviour. Then, a corollary follows, showing that for specific classes of EMPA terms we obtain specific classes of nets.

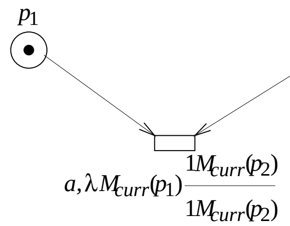
Theorem 6.1. *Let* $E \in \mathcal{G}$. $\mathcal{RG}[\![\mathcal{OL}ab\mathcal{ON}[\![E]\!]]\!]$ *is isomorphic to* $\mathcal{RG}[\![\mathcal{L}ab\mathcal{ON}[\![E]\!]]\!]$. □

Corollary 6.2. *(i)* $\mathcal{OL}ab\mathcal{ON}[\![E]\!]$ *is a GSPN if and only if* $E \in \mathcal{E}$.
(ii) If in $E \in \mathcal{G}$ *the alternative composition operator is applied to sequential terms only, then* $\mathcal{OL}ab\mathcal{ON}[\![E]\!]$ *has no inhibitor arcs.* □

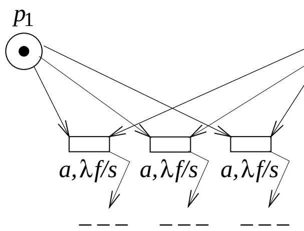
Now we reconsider the nets shown in the examples of Section 4 in order to see what improvements we gain with the optimisation defined above.

Example 6.1. The nets associated with terms considered in Examples 4.11 and 4.12 were already depicted in their optimised version: we drew only reachable places for the sake of simplicity. We want to remember that they actually had infinitely many places as well as infinitely many transitions. □

Example 6.2. Let us turn our attention to Example 4.13. Then $\mathcal{OL}ab\mathcal{ON}[\![E]\!]$ is the following net:



and $\mathcal{OL}ab\mathcal{ON}[\![E'']\!]$ is the following net:



□

7. COMPARING NET SEMANTICS

Now we want to compare the label-oriented net semantics and the location-oriented net semantics for EMPA with respect to the complexity of net representations and the complexity of their definitions. Finally, we consider an interesting example.

7.1. Complexity in net representation

The label-oriented net semantics is remarkably advantageous since it produces significantly smaller nets for almost all the EMPA terms. The three main reasons are the following.

Unsafeness: The decomposition function of the label-oriented approach assigns the same place to syntactically identical terms which are composed in parallel whenever the synchronisation set does not contain action types occurring in such terms (since the mechanism for generating fresh action types does not come into play). This is not the case for the location-oriented approach because each place keeps track of the syntactical structure of the term, and notably the presence of parallel composition operators: dec is injective in this approach. As a consequence, in the former case nets are not necessarily 1-safe, whereas in the latter case they are so. It is instructive, in this respect, to consider term A of Example 4.12: $\mathcal{OL}ab\mathcal{ON}[\![A]\!]$ is a *finite* unsafe net, while $\mathcal{L}oc\mathcal{ON}[\![A]\!]$ is an *infinite* 1-safe net.

The non injectivity of dec in the label-oriented approach induces identifications on EMPA terms; e.g., the terms which are congruent modulo associativity, commutativity and "$\underline{0}$ as neutral element" for $\|_\emptyset$ are mapped to the very *same* net. In the location-oriented approach, instead, each term is assigned a different net; however, if two terms are congruent modulo the axioms above, then they give rise to *isomorphic* nets.

The price to pay for obtaining smaller nets by dropping safeness is the presence of marking-dependent rates, which do not come into play in the location-oriented approach.

Linearly distributed choice: The decomposition function of the label-oriented net semantics handles the alternative composition operator applied to nonsequential terms as if it were a parallel composition operator without synchronisation set by preserving the mutual exclusion through conflicts. It is easy to see that, if we need p_1 places for representing E_1 and p_2 places for representing E_2, then $E_1 + E_2$ is represented with $p_1 + p_2 + 2$ places. On the contrary, the location-oriented approach handles the alternative composition operator by means of Cartesian products: $E_1 + E_2$ is represented with $p_1 \cdot p_2$ places.

It is instructive to see that the location-oriented approach is better than the label-oriented one only when $p_1 = 1 \wedge p_2 > 1$ and when $p_1 = 2 \wedge 2 \leq p_2 \leq 3$ (and symmetrically). The two approaches give exactly the same construction for $p_1 = 1 \wedge p_2 = 1$ (thanks to the second clause of $\boldsymbol{dec}$ for the alternative composition operator) and coincide, albeit with a different number of arcs, for $p_1 = 2 \wedge p_2 = 4$. One of these few cases ($p_1 = 2$ and $p_2 = 1$) is reported in Example 4.11.

Of course, the price to pay for producing smaller nets is the presence of inhibitor places associated with conflicts, while the location-oriented approach needs no inhibitor arcs.

Null terms optimisation: The label-oriented net semantics does not produce useless places for terms such as $\underline{0}$, $\underline{0}\|_S\underline{0}$ and so on.

Further optimisations related to useless places could remove places not appearing in the preset of any transition, i.e. places associated with terms equivalent to $\underline{0}$ which become evident by virtue of the elimination of dead transitions resulting in the optimised version of the label-oriented net semantics.

7.2. Complexity in definition

Not only the complexity of net representations is different in the two approaches, but also the complexity of description of the two semantics is different. In fact, the label-oriented net semantics has only one axiom schema for generating all the transitions, whilst the location-oriented net semantics produces transitions by as many axioms and inference rules as the interleaving semantics for EMPA has. However, there is an extra complexity due to the computation of transition rate parameters which are easily computed in the location-oriented approach because independent of the current marking. On the contrary, here we introduced a lot of technical details to compute the correct, marking-dependent, transition rate.

Both semantics satisfy the consistency retrievability principles. However, in the case of the location-oriented approach there is a *strong* retrievability result, as for each term its interleaving semantics is isomorphic to the reachability graph of its net semantics. For the label-oriented net semantics, we have only a *weak* retrievability result, as for each term its interleaving semantics is strongly EMB to the reachability graph of its net semantics, and the isomorphism does not hold in general. E.g., consider term E of Example 4.13: its interleaving semantics has two transitions each labelled with $a, \lambda/2$, while the reachability graph of its label-oriented net semantics has only one transition labelled with a, λ.

7.3. An example

We conclude this section by presenting an example of a queueing system (QS) (Kleinrock, 1975), showing the net representations resulting from the application of the two approaches; as expected, the label-oriented approach is more convenient than the location-oriented one.

The QS we want to examine is $M/M/2/3/4$ with arrival rate λ and service rate μ. Such a queueing system is supplied with two independent servers and a queue with capacity one which provide service to four independent customers according to the FIFO discipline. The interarrival time of each customer is exponentially distributed with rate λ, and the service time of each server is exponentially distributed with rate μ.

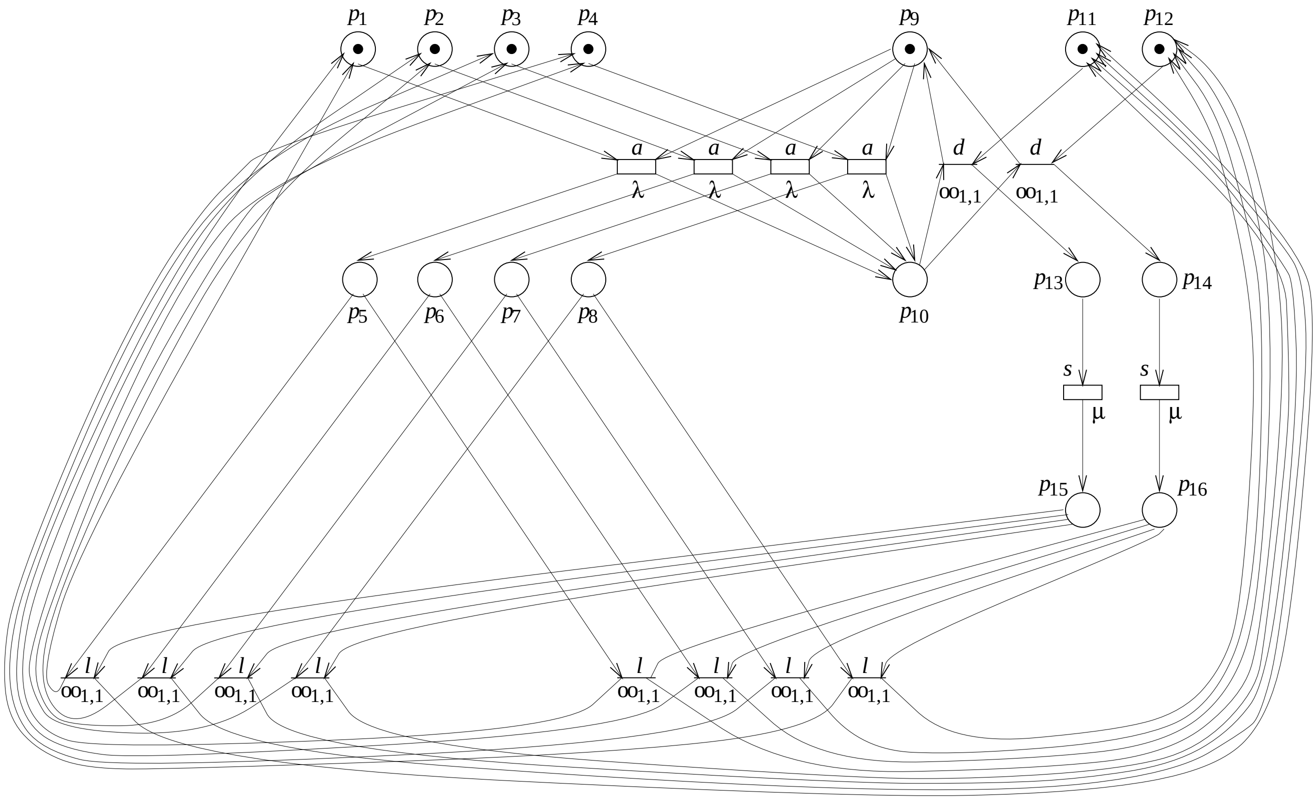
Let us denote by a the action type "a customer arrives at the service centre", d the action type "a customer is delivered by the queue to one of the servers", s the action type "a customer is served by one of the servers", and l the action type "a customer leaves the service centre". Then the QS at hand can be modelled with EMPA as follows:

- $System_{M/M/2/3/4} \triangleq C_4\|_{\{a,l\}}(Q\|_{\{d\}}S_2)$;
- $C_4 \triangleq C\|_\emptyset C\|_\emptyset C\|_\emptyset C$,
  $C \triangleq {<}a, \lambda{>}.{<}l, 0{>}.C$;
- $Q \triangleq {<}a, 0{>}.{<}d, 0{>}.Q$;
- $S_2 \triangleq S\|_\emptyset S$,
  $S \triangleq {<}d, \infty_{1,1}{>}.{<}s, \mu{>}.{<}l, \infty_{1,1}{>}.S$.

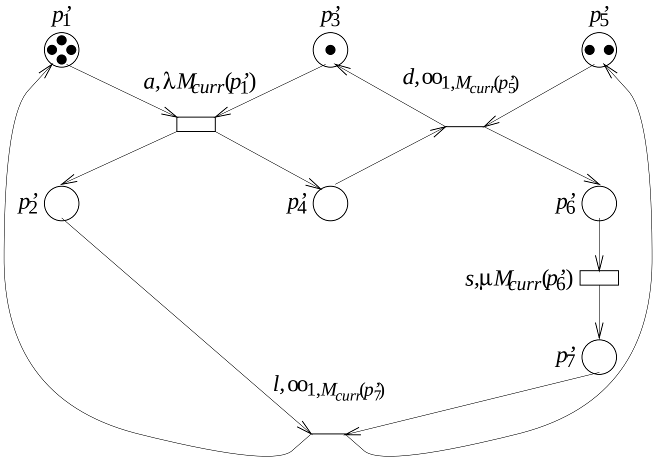
$\mathcal{L}oc\mathcal{ON}[\![System_{M/M/2/3/4}]\!]$ is reported in Figure 1(a), whilst $\mathcal{OL}ab\mathcal{ON}[\![System_{M/M/2/3/4}]\!]$ is reported in Figure 1(b). The names of places have been shortened as follows:

- $p_1 = (((C\|_\emptyset id)\|_\emptyset id)\|_\emptyset id)\|_{\{a,l\}} id$,
  $p_2 = (((id\|_\emptyset C)\|_\emptyset id)\|_\emptyset id)\|_{\{a,l\}} id$,
  $p_3 = ((id\|_\emptyset C)\|_\emptyset id)\|_{\{a,l\}} id$,
  $p_4 = (id\|_\emptyset C)\|_{\{a,l\}} id$,
  $p_5 = ((({<}l, 0{>}.C\|_\emptyset id)\|_\emptyset id)\|_\emptyset id)\|_{\{a,l\}} id$,
  $p_6 = (((id\|_\emptyset {<}l, 0{>}.C)\|_\emptyset id)\|_\emptyset id)\|_{\{a,l\}} id$,
  $p_7 = ((id\|_\emptyset {<}l, 0{>}.C)\|_\emptyset id)\|_{\{a,l\}} id$,
  $p_8 = (id\|_\emptyset {<}l, 0{>}.C)\|_{\{a,l\}} id$;
- $p_9 = id\|_{\{a,l\}}(Q\|_{\{d\}} id)$,
  $p_{10} = id\|_{\{a,l\}}({<}d, 0{>}.Q\|_{\{d\}} id)$;
- $p_{11} = id\|_{\{a,l\}}(id\|_{\{d\}}(S\|_\emptyset id))$,
  $p_{12} = id\|_{\{a,l\}}(id\|_{\{d\}}(id\|_\emptyset S))$,
  $p_{13} = id\|_{\{a,l\}}(id\|_{\{d\}}({<}s, \mu{>}.{<}l, \infty_{1,1}{>}.S\|_\emptyset id))$,
  $p_{14} = id\|_{\{a,l\}}(id\|_{\{d\}}(id\|_\emptyset {<}s, \mu{>}.{<}l, \infty_{1,1}{>}.S))$,
  $p_{15} = id\|_{\{a,l\}}(id\|_{\{d\}}({<}l, \infty_{1,1}{>}.S\|_\emptyset id))$,
  $p_{16} = id\|_{\{a,l\}}(id\|_{\{d\}}(id\|_\emptyset {<}l, \infty_{1,1}{>}.S))$;
- $p'_1 = \emptyset C'$, $p'_2 = \emptyset{<}l, 0{>}.C'$
  with $C' \equiv C\langle\{a, l\} := \{a_{new,l}, l_{new,l}\}\rangle$;
- $p'_3 = \emptyset Q'$, $p'_4 = \emptyset{<}d, 0{>}.Q'$
  with $Q' \equiv Q\langle\{a, d\} := \{a_{new,r}, d_{new,l}\}\rangle$;
- $p'_5 = \emptyset S'$,
  $p'_6 = \emptyset{<}s, \mu{>}.{<}l, \infty_{1,1}{>}.S'$,
  $p'_7 = \emptyset{<}l, \infty_{1,1}{>}.S'$
  with $S' \equiv S\langle\{d, l\} := \{d_{new,r}, l_{new,r}\}\rangle$.

Note that the first net has 16 places, 16 transitions and 60 arcs, whereas the second net has only 7 places, 4 transitions and 14 arcs.



(a) Location-oriented net semantics



(a) Label-oriented net semantics

FIGURE 1. Net semantics of $System_{M/M/2/3/4}$

8. CONCLUDING REMARKS

Besides consistency retrievability principles, a good net semantics should satisfy also some *completeness* principle. In (Olderog, 1991) the *concurrency principle* is proposed, according to which the net semantics should never impose unnatural sequentialisations to actions which are intended to be concurrent in the EMPA specification. In (Bernardo *et al.*, 1995a) we proved that the location-oriented net semantics for EMPA satisfies this principle, by taking a standard (variation of a) denotational net semantics as a reference point and showing that corresponding nets obtained from the operational semantics and the denotational semantics have the same concurrent computations. We leave for future research investigating whether the label-oriented net semantics satisfies the concurrency principle, although we have a strong feeling that this is the case.

Finally, a few words on related works. Apart from the operational location-oriented net semantics for EMPA (Bernardo *et al.*, 1994b; Bernardo *et al.*, 1995a) which has been already compared in great detail, we would like to mention the denotational net semantics for EMPA (Bernardo *et al.*, 1994b; Bernardo *et al.*, 1995a) as well as the work by Ribaudo (Ribaudo, 1995). She gave denotational net semantics for several stochastic process algebras and, in the case of EMPA, developed a semantics which essentially coincides with our denotational one. Since these semantics produce safe nets, the label-oriented net semantics for EMPA allows us to obtain advantages which are similar to those gained with respect to the location-oriented net semantics for EMPA.

ACKNOWLEDGEMENTS

This research has been partially funded by MURST, CNR and ESPRIT BRA project 9102 - Coordination.

REFERENCES

Ajmone Marsan, M., Balbo, G., & Conte, G. 1984. A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *Pages 143–172 of: ACM Trans. on Computer Systems*, vol. 2.

Bernardo, M., Donatiello, L., & Gorrieri, R. 1994a. Integrated Analysis of Concurrent Distributed Systems using Markovian Process Algebra. *Pages 455–457 of:* 7^{th} *Int. Conf. on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE '94)*. Chapman & Hall.

Bernardo, M., Donatiello, L., & Gorrieri, R. 1994b. *Operational GSPN Semantics of MPA*. Tech. rept. UBLCS-94-12. University of Bologna (Italy).

Bernardo, M., Donatiello, L., & Gorrieri, R. 1995a. Giving a Net Semantics to Markovian Process Algebra. *In:* 6^{th} *International Workshop on Petri Nets and Performance Models (PNPM '95)*. IEEE-CS Press.

Bernardo, M., Donatiello, L., & Gorrieri, R. 1995b. *Integrating Performance and Functional Analysis of Concurrent Systems with EMPA*. Tech. rept. UBLCS-95-14. University of Bologna (Italy).

Buchholz, P. 1994. Markovian Process Algebra: Composition and Equivalence. *Pages 11–30 of:* Herzog, U., & Rettelbach, M. (eds), 2^{nd} *Workshop on Process Algebras and Performance Modelling (PAPM '94)*. University of Erlangen (Germany).

Busi, N., & Gorrieri, R. 1995a. Distributed Conflicts in Communicating Systems. *Pages 49–65 of: Object-Based Models and Languages for Concurrent Systems*. LNCS, vol. 924. Springer-Verlag.

Busi, N., & Gorrieri, R. 1995b. A Petri Net Semantics for π-Calculus. *Pages 145–159 of:* 6^{th} *Int. Conf. on Concurrency Theory (CONCUR '95)*. LNCS, vol. 962. Springer-Verlag.

Degano, P., De Nicola, R., & Montanari, U. 1988. A Distributed Operational Semantics for CCS Based on Condition/Event Systems. *Pages 59–91 of: Acta Informatica*, vol. 26.

Goltz, U. 1988. On Representing CCS Programs by Finite Petri Nets. *Pages 339–350 of:* 13^{th} *Int. Symp. on Mathematical Foundations of Computer Science (MFCS '88)*. LNCS, vol. 324. Springer-Verlag.

Gorrieri, R., & Montanari, U. 1995. On the Implementation of Concurrent Calculi in Net Calculi: Two Case Studies. *Pages 195–252 of: Theoretical Computer Science*, vol. 141 (1-2).

Götz, N., Herzog, U., & Rettelbach, M. 1993. Multiprocessor and Distributed System Design: the Integration of Functional Specification and Performance Analysis Using Stochastic Process Algebras. *In:* 16^{th} *Int. Symp. on Computer Performance Modelling, Measurement and Evaluation (PERFORMANCE '93)*. LNCS, vol. 729. Springer-Verlag.

Hermanns, H., & Rettelbach, M. 1994. Syntax, Semantics, Equivalences, and Axioms for MTIPP. *Pages 71–87 of:* Herzog, U., & Rettelbach, M. (eds), 2^{nd} *Workshop on Process Algebras and Performance Modelling (PAPM '94)*. University of Erlangen.

Hillston, J. 1994. *A Compositional Approach to Performance Modelling*. Ph.D. thesis, University of Edinburgh (UK).

Hoare, C. A. R. 1985. *Communicating Sequential Processes*. Prentice Hall.

Kemeny, J. G., & Snell, J. L. 1977. *Finite Markov Chains*. Springer-Verlag.

Kleinrock, L. 1975. *Queueing Systems*. Wiley.

Montanari, U., & Rossi, F. 1995. Contextual Nets. *Pages 545–596 of: Acta Informatica*, vol. 32 (6).

Olderog, E. R. 1991. *Nets, Terms and Formulas*. Cambridge University Press.

Reisig, W. 1985. *Petri Nets: an Introduction*. Springer-Verlag.

Ribaudo, M. 1995. *On the Relationship between Stochastic Process Algebras and Stochastic Petri Nets*. Ph.D. thesis, University of Turin (Italy).