The Theory of Interactive Generalized Semi-Markov Processes

Mario Bravetti, Roberto Gorrieri

Università di Bologna, Dipartimento di Scienze dell'Informazione
Mura Anteo Zamboni 7, 40127 Bologna, Italy
E-mail: {bravetti, gorrieri}@cs.unibo.it

Abstract

In this paper we introduce the calculus of Interactive Generalized Semi-Markov Processes (*IGSMPs*), a stochastic process algebra which can express probabilistic timed delays with general distributions and synchronizable actions with zero duration, and where choices may be probabilistic, non-deterministic and prioritized. *IGSMP* is equipped with a structural operational semantics which generates semantic models in the form of Generalized Semi-Markov Processes (*GSMPs*), i.e. probabilistic systems with generally distributed time, extended with action transitions representing interaction among system components. This is obtained by expressing the concurrent execution of delays through a variant of ST semantics which is based on dynamic names. The fact that names for delays are generated dynamically by the semantics makes it possible to define a notion of observational congruence for *IGSMP* (that abstracts from internal actions with zero duration) simply as a combination of standard observational congruence and probabilistic bisimulation. We also present a complete axiomatization for observational congruence over *IGSMP*. Finally, we show how to derive a *GSMP* from a given *IGSMP* specification in order to evaluate the system performance and we present a case study.

Key words: stochastic process algebras, generalized semi-Markov processes, probabilistic bisimulation, observational congruence

1 Introduction

Stochastically timed process algebras (see e.g. [20,12,1,6,10,16,28,5,11,17,22]) are formal specification languages which describe concurrent systems both from the viewpoint of interaction and from the viewpoint of performance. They extend the expressiveness of classical process algebras by introducing a notion of *time* in the form of *delays* with probabilistic duration. The advantages of integrating the description of interaction with the description of performance are

Preprint submitted to Elsevier Preprint *26 June 2000*

several. First of all we can specify and analyze systems for combined behavioral and performance properties, e.g. via a notion of *integrated equivalence*, that relates terms with the same behavioral and performance characteristics. In doing this we can take advantage of the feature of compositionality offered by process algebras, which describe a concurrent system in term of the behavior of its composing processes. Secondly, we can analyze behavioral and performance properties of a system, separately on two projected semantic models (a standard transition system labeled with actions and a stochastic process with some kind of *Markov property*), which are automatically derived from the initial integrated specification. This has the advantage that such models are guaranteed to be consistent, since they are formally derived from the same initial integrated specification.

A lot of work has been previously done in the field of *Markovian process algebras* (see e.g. [20,6,17] and the references therein). They are stochastically timed process algebras, where the probabilistic distribution of a delay is assumed to be *exponential*. This causes the passage of time to be "memoryless" and has the consequence that the system behavior can be described (via interleaving operational semantics) by expressing the execution of a time delay as an atomic transition without explicitly representing durations for delays. Moreover the limitation to exponential distributions allows a straightforward transformation of the semantic model of a system into a *Continuous Time Markov Chain (CTMC)*. The limitation imposed over durations is very strong from a modeling viewpoint because, e.g., not even deterministic (fixed) durations can be expressed. The capability of expressing *general probabilistic distributions* would give the possibility of producing much more realistic specifications of systems. Even system activities which have an uncertain duration could be represented probabilistically by more adequate distributions than exponential ones (e.g. Gaussian distributions or experimentally determined distributions).

Some previous efforts have been done in order to try to extend the expressiveness of Markovian process algebras to probabilistic time with general distribution [12,1,10,16,28,5,11,22]. The main point in doing this is to understand how to define semantic models and semantic reasoning, e.g. the definition of an adequate notion of bisimulation based equivalence. In probability theory systems capable of executing parallel activities with generally distributed durations are represented by *Generalized Semi-Markov Processes (GSMPs)* [24]. Previously [5] we have studied how to develop an adequate operational semantics for a process algebra with general distributions which generates semantic models in the form of *GSMPs*. In [5] we have shown that the problem of representing time delays in semantic models is basically the same as describing the behavior of a system via ST semantics [13,4,9,7]. According to ST semantics, the evolution of a delay is represented in semantic models, similarly as in *GSMPs*, as a combination of the two events of delay start and delay termination, where the termination of a delay is uniquely related to its start by, e.g., identifying each delay with a unique name. As we will see this approach is very natural for expressing time delays, especially when a duration is expressed through general probability distributions. Moreover the use of ST semantics leads to a notion of choice among delays which is based on *preselection policy*. A choice among alternative delays is resolved by first performing a probabilistic choice among the possible delays, and then executing the selected delay. Therefore the choice of a delay is naturally represented in semantic models as a probabilistic choice among transitions representing delay starts. As we also show in [5] this method of solving choices, compared to the *race policy* used in Markovian process algebras, is very adequate and simple when dealing with generally distributed durations. On the other hand this adheres to the fact that, while in *CTMC* probabilistic choices are implicitly expressed through a "race" of exponential distributions, in *GSMPs* they are explicitly expressed via a probabilistic selection mechanism. From the semantic model of a system, derived in this way, it is easy to derive a performance model in the form of a *GSMP*. A *GSMP* can then be analyzed through well established mathematical or simulative techniques in order to obtain performance measures of the system (see e.g. [14]).

In this paper we consider a variant of the algebra of [5] that allows us to define a notion of observational congruence which abstracts from internal computations which are not visible from an external observer (τ actions). This is desirable because it may lead to a tremendous state space reduction of semantic models. Technically this is obtained by restricting the possible durations of synchronizable actions to zero durations only. More precisely, following an approach which is quite usual in real-time process algebras (see e.g. [27]) and which has been imported in the stochastic process algebra community in [19,17], we distinguish between actions f representing a delay whose duration is given by the probability distribution f (itself) and standard actions of CCS/CSP [25,21] (including internal τ actions) with zero duration. In analogy to [17] we call the resulting algebra: calculus of *Interactive Generalized Semi-Markov Processes*. The name reflects the separated orthogonal treatment of delays and standard actions.

Following the ideas of [5], we define the operational semantics of a delay f in *IGSMP* through ST semantics. Hence a delay is represented in semantic models as a combination of the event of start of the delay f^+ and the event of termination of the delay f^-. Moreover we assign names (consisting of indexes i) to delays so that the execution of a delay is represented by the two events f_i^+ and f_i^- and no confusion arises (in the connection between delay starts and delay terminations) when multiple delays with the same distribution f are concurrently executed.

In this paper we employ the new technique for expressing ST semantics that we have introduced in [7] which is based on dynamic names. As opposed to the technique employed in [5], which is based on static names, the technique we use here allows us to establish equivalence of systems via the standard notion of bisimulation (so that existing results and tools can be exploited), nevertheless preserving the possibility of obtaining finite ST semantic models even in the case of recursive systems. By exploiting the fact that this technique is also compositional, we define ST semantics through Structural Operational Semantics (SOS) and we produce a complete axiomatization for ST bisimulation over finite state processes.

As in [5], we resolve choices among several delays by means of preselection policy. We associate with each delay a weight w: in a choice a delay is selected with probability proportional to its weight. For instance $<f,w>.\underline{0}+<g,w'>.\underline{0}$ represents a system which performs a delay of distribution f with probability $w/(w+w')$ and a delay of distribution g with probability $w'/(w+w')$. Choices are expressed in semantic models by associating weights to transitions f_i^+ representing the start of a delay.

The semantics of standard actions a (including internal actions τ) in *IGSMP* is just the standard interleaving semantics. This reflects the fact that these actions have zero duration and can be considered as being executed atomically. As in [19,17] the choice among standard actions is just non-deterministic. We can express external choices (e.g. $a + b$) which are based on the behavior of other system components, but also non-deterministic internal choices (e.g. $\tau + \tau$) which cannot be resolved through interaction. This can be seen as an expressive feature, since it allows for an underspecification of the system performance, but has the drawback that it makes sometimes impossible to derive a purely probabilistic model of the system (see Sect. 4). We assume the so-called *maximal progress* [27]: τ actions have priority over delays, thus expressing that the system cannot wait if it has something internal to do, i.e. $\tau.P + <f,w>.Q = \tau.P$.

We present a formal procedure for transforming the semantic model obtained from a suitable specification of a system (see Sect. 4) into a *GSMP*. Such a procedure just turns each delay of the system into a different element of the *GSMP* and system weighted choices into probabilistic choices of a *GSMP*.

Finally, as an example of modular *IGSMP* specification, we consider Queueing Systems $G/G/1/q$, i.e. queuing systems with one server and a FIFO queue with q-1 seats, where interarrival time and service time are generally distributed. Moreover we show how to derive the performance model of such queuing systems (a *GSMP*) by applying the formal procedure above.

Summing up, the contribution of this work is a weak semantics for a language expressing generally distributed durations, probabilistic choices (preselection policy), non-determinism and priority. The use of the technique of [7] for expressing ST bisimulation allows us to define observational congruence for *IGSMP* simply as a combination of the standard notion of observational congruence and probabilistic bisimulation [23] and to produce a complete axiomatization for this equivalence. Moreover we show how to automatically derive *GSMPs* from *IGSMP* specifications and we present the example of Queueing Systems $G/G/1/q$.

The paper is structured as follows. In Sect. 2 we present the calculus of Interactive *GSMPs* and its operational semantics. In Sect. 3 we present the notion of observational congruence and its complete axiomatization. In Sect. 4 we present the formal procedure for deriving a *GSMP* from a complete system specification. In Sect. 5 we present the example of Queueing Systems $G/G/1/q$. Finally, in Sect. 6 we report some concluding remarks including comparison with related work and directions for future research.

2 The Calculus of Interactive GSMPs

2.1 Syntax of Terms and Informal Semantics of Operators

The calculus of interactive *GSMPs* is an extension of a standard process algebra with operators of CCS/CSP [25,21], which allows us to express priority, probabilistic choices and probabilistic delays with arbitrary distributions. This is done by including into the calculus, in addition to standard actions, a special kind of actions representing delays. Delays are represented as $<f, w>$ and are characterized by a *weight* w and a *duration distribution* f. The weight w determines the probability of choosing the delay in a choice among several delays. The set of weights is $\mathbb{R}^+$, ranged over by $w, w', \ldots$. The duration distribution f denotes the probability distribution function of the delay duration. The set of duration probability distribution functions is PDF^+, i.e. the set of probability distribution functions f such that $f(x) = 0$ for $x < 0$, ranged over by f, g, h. The possibility of expressing priority derives from the interrelation of delays and standard τ actions. In particular we make the maximal progress assumption: the system cannot wait if it has something internal to do. Therefore we assume that, in a choice, τ actions have priority over delays, i.e. $\tau.P + <f, w>.Q$ behaves as $\tau.P$.

Let *Act* be the set of action types containing a distinguished type τ representing an internal computation. *Act* is ranged over by $a, b, c, \ldots$. Let $TAct = \{<f, w> \mid f \in PDF^+ \wedge w \in \mathbb{R}^+\}$ be the set of delays. [1] Let *Var* be a set of

[1] In the following we consider f to be a shorthand for $<f, 1>$ when this is clear

process variables ranged over by X, Y, Z. Let $ARFun = \{\varphi : Act \longrightarrow Act \mid \varphi(\tau) = \tau \wedge \varphi(Act - \{\tau\}) \subseteq Act - \{\tau\}\}$ be a set of *action relabeling functions*, ranged over by φ.

Definition 2.1 We define the language *IGSMP* as the set of terms generated by the following syntax

$$P ::= \underline{0} \mid X \mid <f, w>.P \mid a.P \mid P + P \mid P/L \mid P[\varphi] \mid P \parallel_S P \mid recX.P$$

where $L, S \subseteq Act - \{\tau\}$. An *IGSMP* process is a closed term of *IGSMP*. We denote by $IGSMP_g$ the set of strongly guarded terms of *IGSMP*. [2] ■

"$\underline{0}$" denotes a process that cannot move. The operators "." and "+" are the CCS prefix and choice. The choice among delays is carried out through the preselection policy by giving each of them a probability proportional to its weight. Note that alternative delays are not executed concurrently, first one of them is chosen probabilistically and then the selected delay is executed. Moreover τ actions have priority over delays in a choice. "$/L$" is the hiding operator which turns into τ the actions in L, "$[\varphi]$" is the relabeling operator which relabels visible actions according to φ. "$\parallel_S$" is the CSP parallel operator, where synchronization over actions in S is required. Finally "*recX*" denotes recursion in the usual way.

2.2 Operational Semantics

As we will formally see in Sect. 4.3, a Generalized Semi-Markov Process (*GSMP*) represents the behavior of a system by employing a set of *elements*, which are similar to the clocks of a timed automata [3]. Each element has an associated duration distribution (element lifetime) and its execution in a *GSMP* is characterized by the two events of start (when the element is born) and termination (when the element dies).

Since *IGSMPs* extend *GSMPs* with the capability of interacting via standard actions, the semantic model of an *IGSMP* process is a labeled transition system, where a transition represents a basic event: the execution of a standard action, the start of a delay or the termination of a delay.

Similarly as in *GSMPs*, the execution of a delay is represented in semantic models by the two events of delay start and delay termination and enough information is provided (delays are given unique *element names*) in order to ensure that each event of delay termination is uniquely related to the corresponding event of delay start. This corresponds to representing the execution

---

from the context.

[2] We consider the delay $<f, w>$ as being a guard in the definition of strong guardedness.

of delays with ST semantics [13,4,9,7]. As we also show in [5] this semantics is just what we need for representing durational actions, if the duration is expressed with general probability distributions. On the other hand, by employing a realization of ST semantics based on the identification of delays with names, the effect of applying ST semantics to an *IGSMP* process is to generate the names for elements of the underlying *GSMP*.

While in [5] ST semantics is expressed by assigning *static names* to delays according to their syntactical position in the system, here we employ a new technique for generating ST semantic models, that we have introduced in [7], which is based on *dynamic names*, i.e. names computed dynamically while the system evolves. The advantage of this technique is that it allows us to establish ST bisimulation of systems via the standard notion of observational congruence [25] and to preserve the finiteness of ST semantic models even in the presence of recursion. On the contrary a technique for establishing ST bisimulation of two processes based on static names must employ a more complex definition of bisimulation which associates the names of the delays of one process with the names of the corresponding delays used by the other one [4,5].

In *IGSMP* the technique of [7] is employed for giving semantics to delays. The "type" of a delay is simply its duration distribution and what we observe of a system is its ability of performing delays of certain types $f \in PDF^+$. The problem of preserving the relationship between starts and terminations of delays arises, like in the ST semantics of standard process algebras, when several delays of the same type f are being executed in parallel. When a delay f terminates (event f^-) we need some information for establishing which event of delay start (f^+) it refers to. The technique introduced in [7] is based on the idea of dynamically assigning, during the evolution of the system, a new name to each delay that starts execution, on the basis of the names assigned to the delays already started. Names consist of indexes that distinguish delays with the same duration distribution. In particular the event of a delay start $<f, w>$ is represented in semantic models by a transition labeled by $<f_i^+, w>$ where i is the minimum index not already used by the other delays with distribution f that have started but not yet terminated. This rule for computing indexes guarantees that names are reused and that finite models can be obtained also in the presence of recursion. The termination of the delay is simply represented by a transition labeled by f_i^-, where the "identifier" i uniquely determines which delay f is terminating. Since the method to compute the index for a starting delay is fixed, it turns out that delays of processes that perform the same execution traces of delays get the same names. As a consequence, contrary to [4,5], ST bisimilarity can simply be checked by applying standard bisimilarity to the semantic models of processes.

Moreover the technique introduced in [7] allows us to dynamically assign names to delays, according to the rule formerly described, via SOS semantics (hence in a *compositional way*) through the idea of *levelwise renaming*. In order to obtain structural compositionality it is necessary to determine, e.g. in the case of the parallel composition operator, the computations of $P \,\|\, Q$ from the computations of P and Q. This is done by parameterizing in state terms each parallel operator with a mapping M. For every delay f started by $P \,\|_{S,M}\, Q$, M records the association between the name f_i, generated according to the fixed rule above for identifying f at the level of $P \,\|_{S,M}\, Q$, and the name f_j (which in general is different from f_i), generated according to the same rule for identifying the same delay f inside P (or Q). In this way when, afterwards, such a delay f terminates in P (or Q) the name f_j can be re-mapped to the correct name f_i at the level of $P \,\|_{S,M}\, Q$, by exploiting the information included in M. In M the delay f of $P \,\|_{S,M}\, Q$ which gets index i is uniquely identified by expressing the unique name j it gets in P or in Q and the "location" of the process that executes it: *left* if P, *right* if Q. Such an association is represented inside M by the triple (f, i, loc_j) with $f \in PDF^+$, indices $i, j \in \mathbb{N}^+$ and location $loc \in Loc = \{l, r\}$, where "l" stands for left and "r" for right. In the following we use $f : (i, loc_j)$ to stand for $(f, i, loc_j) \in M$.

The weight w associated with the start of a delay determines, as already explained, the probability that the delay is chosen in spite of other delays and, therefore, the probability that the delay starts.

As it is natural in the context of a stochastic process algebra, we assume delay starts to be *urgent*. Therefore we have that, if a system state can perform a delay start, it does not let time pass, so possible delays in execution cannot terminate in that state. This causes another form of priority in our language: the priority of delay starts over delay terminations, i.e. $<f_i^+, w>.P + g_j^-.Q = <f_i^+, w>.P$.

Summing up, we have two forms of priority in our semantic models: the priority of τ actions over delays (starts or terminations) and the priority of delay starts over delay terminations. As opposed to [17], where a similar notion of priority is captured in the definition of equivalence among systems, we prefer to express priority by cutting transitions which cannot be performed directly in semantic models (a solution also hinted in [18]). This allows us to have smaller semantic models and to define the notion of equivalence more simply, without having to discard any transitions when establishing bisimulation.

In order to define the operational semantics for the processes of *IGSMP*, we need a richer syntax to represent states. Let $TAct^+ = \{<f_i^+, w> \mid f \in PDF^+ \wedge i \in \mathbb{N}^+ \wedge w \in \mathbb{R}^+\}$ be the set of delay starts, where $<f_i^+, w>$

$$a.P \xrightarrow{a} P$$

$$\frac{P \xrightarrow{a} P'}{P+Q \xrightarrow{a} P'} \qquad \frac{Q \xrightarrow{a} Q'}{P+Q \xrightarrow{a} Q'}$$

$$\frac{P \xrightarrow{a} P'}{P/L \xrightarrow{a} P'/L}\ a \notin L \qquad \frac{P \xrightarrow{a} P'}{P/L \xrightarrow{\tau} P'/L}\ a \in L$$

$$\frac{P \xrightarrow{a} P'}{P[\varphi] \xrightarrow{\varphi(a)} P'[\varphi]}$$

$$\frac{P \xrightarrow{a} P'}{P \parallel_{S,M} Q \xrightarrow{a} P' \parallel_{S,M} Q}\ a \notin S \qquad \frac{Q \xrightarrow{a} Q'}{P \parallel_{S,M} Q \xrightarrow{a} P \parallel_{S,M} Q'}\ a \notin S$$

$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q'}{P \parallel_{S,M} Q \xrightarrow{a} P' \parallel_{S,M} Q'}\ a \in S$$

$$\frac{P\{recX.P/X\} \xrightarrow{a} P'}{recX.P \xrightarrow{a} P'}$$

Fig. 1. Standard Rules

represents the beginning of the delay $<f, w>$ identified by i. [3] Besides let $TAct^- = \{f_i^- \mid f \in PDF^+ \wedge i \in \mathbb{N}^+\}$ be the set of delay terminations, where f_i^-, represents the termination of the delay with duration distribution f identified by i. η ranges over $Act \cup TAct \cup TAct^+ \cup TAct^-$. We denote an *index association*, whose elements are associations (i, loc_j), with $iassoc$ which ranges over the set $IAssoc$ of partial bijections from $\mathbb{N}^+$ to $Loc \times \mathbb{N}^+$. Finally a mapping M is a relation from PDF^+ to $\mathbb{N}^+ \times (Loc \times \mathbb{N}^+)$ such that $\forall f \in PDF^+ . M_f \in IAssoc$ [4], i.e. M is a set including an independent index association for each different duration distribution.

The set $IGSMP_s$ of state terms of $IGSMP$ is generated by the following syntax:

$$P ::= \underline{0} \mid X \mid \eta.P \mid P + P \mid P/L \mid P[\varphi] \mid P \parallel_{S,M} P \mid recX.P$$

We denote by $IGSMP_{sg}$ the set of strongly guarded terms of $IGSMP_s$. We consider the operators "$\parallel_S$" occurring in a $IGSMP$ term P as being "$\parallel_{S,\emptyset}$" when P is regarded as a state.

The semantics of state terms produces a transition system labeled over $Act \cup$

[3] In the following we consider f_i^+ to be a shorthand for $<f_i^+, 1>$ when this is clear from the context.

[4] Given a relation M from A to B, we denote with M_a the set $\{b \in B \mid (a, b) \in M\}$.

$$\frac{(<f_i^+, w>, P') \in \mathit{melt}(\mathit{SM}(P)) \wedge \nexists Q : P \xrightarrow{\tau} Q}{P \xrightarrow{<f_i^+, w>} P'}$$

$$\mathit{SM}(\underline{0}) = \mathit{SM}(a.P) = \mathit{SM}(f_i^-.P) = \emptyset \qquad \mathit{SM}(<f, w>.P) = \{\!|\, (<f_1^+, w>, f_1^-.P) \,|\!\}$$

$$\mathit{SM}(P/L) = \mathit{SM}(P[\varphi]) = \mathit{SM}(P) \qquad \mathit{SM}(P + Q) = \mathit{SM}(P) \oplus \mathit{SM}(Q)$$

$$\begin{aligned} \mathit{SM}(P \,\|_{S,M}\, Q) = {} & \\ & \{\!|\, (<f^+_{n(M_f)}, w>, P' \,\|_{S, M \cup \{f:(n(M_f), l_i)\}}\, Q) \mid (<f_i^+, w>, P') \in \mathit{SM}(P) \,|\!\} \oplus \\ & \{\!|\, (<f^+_{n(M_f)}, w>, P \,\|_{S, M \cup \{f:(n(M_f), r_i)\}}\, Q') \mid (<f_i^+, w>, Q') \in \mathit{SM}(Q) \,|\!\} \end{aligned}$$

$$\mathit{SM}(\mathit{rec}X.P) = \mathit{SM}(P\{\mathit{rec}X.P/X\})$$

$$\begin{aligned} \mathit{melt}(\mathit{SM}) = \{ & (<f_i^+, w>, P) \mid \exists w' : (<f_i^+, w'>, P) \in \mathit{SM} \wedge \\ & w = \textstyle\sum \{\!|\, w'' \mid (<f_i^+, w''>, P) \in \mathit{SM} \,|\!\} \} \end{aligned}$$

Fig. 2. Rules for Start Moves

$$\frac{P \overset{f_i^-}{\rightarrowtail} P' \wedge \nexists Q : P \xrightarrow{\tau} Q \wedge \nexists f, w, i, Q : P \xrightarrow{<f_i^+, w>} Q}{P \xrightarrow{f_i^-} P'}$$

$$f_i^-.P \overset{f_i^-}{\rightarrowtail} P$$

$$\frac{P \overset{f_i^-}{\rightarrowtail} P'}{P + Q \overset{f_i^-}{\rightarrowtail} P'} \qquad \frac{Q \overset{f_i^-}{\rightarrowtail} Q'}{P + Q \overset{f_i^-}{\rightarrowtail} Q'}$$

$$\frac{P \overset{f_i^-}{\rightarrowtail} P'}{P/L \overset{f_i^-}{\rightarrowtail} P'/L} \qquad \frac{P \overset{f_i^-}{\rightarrowtail} P'}{P[\varphi] \overset{f_i^-}{\rightarrowtail} P'[\varphi]}$$

$$\frac{P \overset{f_i^-}{\rightarrowtail} P' \;\wedge\; f:(j, l_i) \in M}{P \,\|_{S,M}\, Q \overset{f_j^-}{\rightarrowtail} P' \,\|_{S, M - \{f:(j, l_i)\}}\, Q} \qquad \frac{Q \overset{f_i^-}{\rightarrowtail} Q' \;\wedge\; f:(j, r_i) \in M}{P \,\|_{S,M}\, Q \overset{f_j^-}{\rightarrowtail} P \,\|_{S, M - \{f:(j, r_i)\}}\, Q'}$$

$$\frac{P\{\mathit{rec}X.P/X\} \overset{f_i^-}{\rightarrowtail} P'}{\mathit{rec}X.P \overset{f_i^-}{\rightarrowtail} P'}$$

Fig. 3. Rules for Termination Moves

$TAct^+ \cup TAct^-$, ranged over by $\gamma, \gamma', \ldots$. The transition relation is defined by the standard operational rules of Fig. 1 and by the two operational rules in the first part of Fig. 2 and Fig. 3.

The rule of Fig. 2 defines the transitions representing the start of a delay, by taking into account the priority of "τ" actions over delays and by employing the function SM. $SM(P)$ evaluates the multiset of start moves leaving state P, represented as pairs $(<f_i^+, w>, P')$, where terms P' are the derivatives of the moves. We use multisets so that we take into account several occurrences of the same weight w. $SM(P)$ is defined by structural induction as the least element of $\mathcal{M}u_{fin}(TAct^+ \times IGSMP_{sg})$ [5] satisfying the rules in the second part of Fig. 2. The meaning of the rule for $P \parallel_{S,M} Q$ is the following. When P performs f_i^+ then a new index $n(M_f)$ is determined for identifying the delay f at the level of "$\parallel_{S,M}$" and the new association $f : (n(M_f), l_i)$ is added to M. The function $n : IAssoc \longrightarrow \mathbb{N}^+$ computes the new index to be used for identifying the delay f that is starting execution by choosing the minimum index not used by the other delays f already in execution: $n(iassoc) = min\{k \mid k \notin dom(iassoc)\}$. A symmetric mechanism takes place for a move f_i^+ of Q. The function $melt : \mathcal{M}u_{fin}(TAct^+ \times IGSMP_{sg}) \longrightarrow \mathcal{P}_{fin}(TAct^+ \times IGSMP_{sg})$, defined in the third part of Fig. 2, merges the start moves with the same label and the same derivative state by summing their weights.

The rule of Fig. 3 defines the transitions representing the termination of a delay, by taking into account the priority of "τ" actions over delays and the priority of delay starts over delay terminations, and by employing the auxiliary transition $\rightarrowtail$. The transition relation $\rightarrowtail$, labeled over $TAct^-$, is defined in the second part of Fig. 3. The meaning of the operational rules for "$P \parallel_{S,M} Q$" is the following. When P performs f_i^- the delay f with index j associated to l_i in M terminates at the level of the parallel operator. A symmetric mechanism takes place for a move f_i^- of Q.

Note that even if the two rules in the first part of Fig. 2 and Fig. 3 include negative premises, the operational semantics is nevertheless correct [15]. This because negative premises are not in the rules which induce on the term structure, but only in "top-level" rules. Moreover the definition of delay start transitions of Fig. 2 is based on the definition of standard action transitions of Fig. 1 only, and the definition of delay termination transitions of Fig. 3 is based on the definition of standard action transitions of Fig. 1 and of delay start transitions of Fig. 2.

We are now in a position to define the *integrated* (representing both interaction and performance) semantic model of a process.

[5] We denote by $\mathcal{M}u_{fin}(S)$ the set of finite multiset over S, we use $\{\!|$ and $|\!\}$ as multiset parentheses, and we use $\oplus$ to denote multiset union.

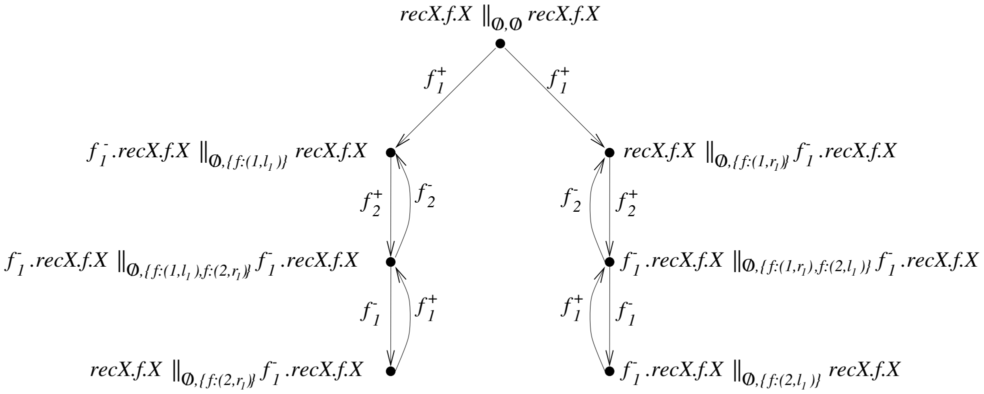


Fig. 4. Example of recursive system

Definition 2.2 The *integrated semantic model* $\mathcal{I}[\![P]\!]$ of $P \in IGSMP_g$ is the labeled transition system (LTS) defined by:

$$\mathcal{I}[\![P]\!] = (S_P, L, \longrightarrow_P, P)$$

where:

- S_P is the least subset of $IGSMP_{sg}$ such that:
  - $\cdot$ $P \in S_P$
  - $\cdot$ if $s \in S_P$ and $s \xrightarrow{\gamma} s'$, then $s' \in S_P$
- $L = Act \cup TAct^+ \cup TAct^-$ is the set of labels.
- $\longrightarrow_P$ is the restriction of $\longrightarrow$ to $S_P \times L \times S_P$. ■

Example 2.3 In Fig. 4 we depict the integrated semantic model of $recX.f.X \,\|_\emptyset\, recX.f.X$. ■

In the following theorem, where we consider "P/L", "$P[\varphi]$", and "$P \,\|_S\, P$" to be *static* operators [25], we show that finite semantic models are obtained for a wide class of recursive systems.

Theorem 2.4 Let P be a $IGSMP_g$ process such that for each subterm $recX.Q$ of P, X does not occur free in Q in the context of a static operator. Then P is a finite state process.

Proof The proof of this theorem derives from the fact that the number of states of the semantics of P which differ only for the contents of mappings M parameterizing parallel operators, are always finite, because the maximum index a delay may assume is bounded by the maximum number of processes that may run in parallel in a state. ■

Note that the class of processes considered in this corollary includes strictly the class of *nets of automata*, i.e. terms where no static operator occurs in the in the scope of any recursion.

3 Observational Congruence for IGSMP

The notion of observational congruence for *IGSMP* is defined, similarly as in [19,17], as a combination of the classical notion of observational congruence [25] and the notion of probabilistic bisimulation of [23].

In our context we express cumulative probabilities by aggregating weights.

Definition 3.1 The function $w : \mathit{IGSMP}_{sg} \times \mathit{PDF}^+ \times \mathcal{P}(\mathit{IGSMP}_{sg}) \longrightarrow \mathbb{R}^+$, which computes the aggregated weight that a state $P \in \mathit{IGSMP}_{sg}$ reaches a set of states $C \in \mathcal{P}(\mathit{IGSMP}_{sg})$ by starting a delay with duration distribution $f \in \mathit{PDF}^+$ is defined as:

$$w(P, f, C) = \sum \{\!| \, w \mid \exists i \in \mathbb{N}^+, Q \in C : P \xrightarrow{<f_i^+, w>} Q \, |\!\} \text{ [6]} \qquad \blacksquare$$

We are now in a position to define the notion of strong bisimilarity for terms of IGSMP_{sg}. Let $\mathit{NPAct} = \mathit{Act} \cup \mathit{TAct}^-$, the set of non-probabilistic actions, be ranged over by α.

Definition 3.2 An equivalence relation β over closed terms of IGSMP_{sg} is a *strong bisimulation* iff $P \; \beta \; Q$ implies

- for every $\alpha \in \mathit{NPAct}$,
  $P \xrightarrow{\alpha} P'$ implies $Q \xrightarrow{\alpha} Q'$ for some Q' with $P' \; \beta \; Q'$,
- for every $f \in \mathit{PDF}^+$ and equivalence class C of β,
  $w(P, f, C) = w(Q, f, C)$

Two closed terms P, Q of IGSMP_{sg} are strongly bisimilar, written $P \sim Q$, iff (P, Q) is included in some strong bisimulation. $\blacksquare$

We consider $\sim$ as being defined also on the open terms of IGSMP_{sg} by extending strong bisimilarity with the standard approach of [25].

The definition of weak bisimilarity is an adaptation of that presented in [19,17] to our context.

Let $\stackrel{\alpha}{\Longrightarrow}$ denote $(\xrightarrow{\tau})^* \xrightarrow{\alpha} (\xrightarrow{\tau})^*$, i.e. a sequence of transitions including a single α transition and any number of τ transitions. Moreover we define $\stackrel{\hat{\alpha}}{\Longrightarrow} \; = \; \stackrel{\alpha}{\Longrightarrow}$ if $\alpha \neq \tau$ and $\stackrel{\hat{\tau}}{\Longrightarrow} \; = (\xrightarrow{\tau})^*$, i.e. a possibly empty sequence of τ

[6] The summation of an empty multiset is assumed to yield 0. Since the method for computing the new index of a delay f that starts in a state P is fixed, we have that several transitions f^+ leaving P have all the same index i.

transitions. Moreover we let C^τ denote the set of processes that may silently evolve into an element of C, i.e. $C^\tau = \{P \mid \exists Q \in C : P \stackrel{\hat{\tau}}{\Longrightarrow} Q\}$.

Definition 3.3 An equivalence relation β over closed terms of $IGSMP_{sg}$ is a *weak bisimulation* iff $P \; \beta \; Q$ implies

- for every $\alpha \in NPAct$,
  $$P \stackrel{\hat{\alpha}}{\Longrightarrow} P' \text{ implies } Q \stackrel{\hat{\alpha}}{\Longrightarrow} Q' \text{ for some } Q' \text{ with } P' \; \beta \; Q',$$
- $P \stackrel{\hat{\tau}}{\Longrightarrow} P'$ implies $Q \stackrel{\hat{\tau}}{\Longrightarrow} Q'$ for some Q' such that, for every $f \in PDF^+$ and equivalence class C of β,
  $$w(P', f, C^\tau) = w(Q', f, C^\tau)$$

Two closed terms P, Q of $IGSMP_{sg}$ are weakly bisimilar, written $P \approx Q$, iff (P, Q) is included in some weak bisimulation. ■

Differently from [19,17] we do not need to express conditions about the stability of bisimilar processes because we consider only strongly guarded processes. As a consequence there is no process of $IGSMP_{sg}$ that is forced in a τ loop and we do not have to recognize this situation. A justification for the fact that we do not consider processes with weakly guarded recursion is the following one. A process that is forced in a τ loop can be seen as a Zeno process, i.e. a processes which performs infinite computations without going beyond a certain point in time. Discarding weakly guarded processes allows us to avoid the technical complications deriving from the treatment of Zenoness (see [8]) and on the other hand seems not to be so restrictive.

Similarly as in [19,17] it is possible to reformulate weak bisimilarity in the following way, which is simpler but less intuitive.

Lemma 3.4 An equivalence relation β over closed terms of $IGSMP_{sg}$ is a *weak bisimulation* iff $P \; \beta \; Q$ implies

- for every $\alpha \in NPAct$,
  $$P \stackrel{\alpha}{\longrightarrow} P' \text{ implies } Q \stackrel{\hat{\alpha}}{\Longrightarrow} Q' \text{ for some } Q' \text{ with } P' \; \beta \; Q',$$
- $Q \stackrel{\hat{\tau}}{\Longrightarrow} Q'$ for some Q' such that, for every $f \in PDF^+$ and equivalence class C of β,
  $$w(P, f, C) = w(Q', f, C)$$ ■

The proof that this reformulation is correct derives from that given in [17], simply by substituting rates of exponential distributions with weights.

The definition of observational congruence, where again we discard the requirement about stability, is the following one.

Definition 3.5 Two closed terms P, Q of $IGSMP_{sg}$ are observational congruent, written $P \simeq Q$, iff:

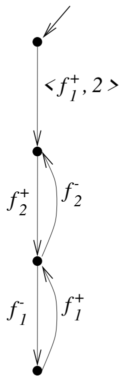


Fig. 5. Minimal semantic model

- for every $\alpha \in NPAct$,
  $$P \xrightarrow{\alpha} P' \text{ implies } Q \overset{\alpha}{\Longrightarrow} Q' \text{ for some } Q' \text{ with } P' \approx Q',$$
- for every $\alpha \in NPAct$,
  $$Q \xrightarrow{\alpha} Q' \text{ implies } P \overset{\alpha}{\Longrightarrow} P' \text{ for some } P' \text{ with } P' \approx Q',$$
- for every $f \in PDF^+$ and equivalence class C of β,
  $$w(P, f, C) = w(Q, f, C)$$ ■

Again we consider $\simeq$ as being defined also on the open terms of $IGSMP_{sg}$ by extending observational congruence with the standard approach of [25].

Theorem 3.6 $\simeq$ is a congruence w.r.t. all the operators of $IGSMP$, including recursion.

Proof The proof of this theorem follows the lines of the similar proof in [25] that is adapted to our setting. The only relevant case is that of parallel composition operator. It suffices to show that $\{(P_1 \,\|_{S,M}\, Q, P_2 \,\|_{S,M}\, Q) \mid P_1 \simeq P_2\}$ is a (weak) bisimulation. ■

Example 3.7 In Fig. 5 we depict the minimum semantic model for the recursive system of Fig. 4, which is obtained by merging bisimilar states. The weight 2 of the initial transition derives from the aggregation of the weights of the two initial transitions in the model of Fig. 4. However since in the initial state there is no alternative to such a transition, its weight is not relevant for the actual behavior (in isolation) of the system. ■

3.1 Axiomatization

In this section we present an axiom system which is complete for $\simeq$ on finite state $IGSMP_{sg}$ terms.

The axiom system $\mathcal{A}_{IGSMP}$ for $\simeq$ on $IGSMP_{sg}$ terms is formed by the axioms presented in Fig. 6. In this figure "$\lfloor\!\lfloor$" and "$|$" denote, respectively, the left merge and synchronization merge operators (see e.g. [2]). Moreover θ ranges over $TAct^+ \cup TAct^-$. We recall from Sect. 2.2 that $\gamma, \gamma', \ldots$ range over $Act \cup TAct^+ \cup TAct^-$.

The axioms ($Pri1$) and ($Pri2$) express the two kinds of priorities of *IGSMP*, respectively, priority of τ actions over (semi-)delays and priority of delay starts over delay terminations. The axiom (Par) is the standard one except that when the position of processes P and Q is exchanged we must invert left and right inside M. The inverse $\overline{M}$ of a mapping M is defined by $\overline{M} = \{f:(i, r_j) \mid f:(i, l_j) \in M\} \cup \{f:(i, l_j) \mid f:(i, r_j) \in M\}$. Axioms ($LM4$) and ($LM5$) just reflect the operational rules of the parallel operator for a delay move of the left-hand process. The axioms ($Rec1-3$) handle strongly guarded recursion in the standard way [26].

If we consider the obvious operational rules for "$\lfloor\!\lfloor_{S,M}$" and "$|_{S,M}$" that derive from those we presented for the parallel operator [7] then the axioms of $\mathcal{A}_{IGSMP}$ are sound.

A sequential state is defined to be one which includes "$\underline{0}$", "X" and operators ".", "+", "*recX*" only; leading to the following theorem.

Theorem 3.8 If an $IGSMP_{sg}$ process P is finite state, then $\exists P' : \mathcal{A}_{IGSMP} \vdash P = P'$ with P' sequential state.

Proof Let $s_1 \ldots s_n$ be the states of the operational semantics of P, $s_n \equiv P$. It can be easily seen that for each $i \in \{1 \ldots n\}$, there exist J_i and $\theta_{i,j}$, $k_{i,j}$ with $j \in J_i$ such that $\mathcal{A}_{IGSMP} \vdash s_i = \sum_{j \in J_i} \theta_{i,j}.s_{k_{i,j}}$ where $\sum_{\emptyset} \equiv \underline{0}$. Then for each i, from 1 to n, we do the following. If i is such that $\exists j \in J_i : k_{i,j} = i$ we have, by applying ($Rec3$), that $s_i = recX.(\sum_{j \in J_i : k_{i,j} \neq i} \theta_{i,j}.s_{k_{i,j}} + \sum_{j \in J_i : k_{i,j} = i} \theta_{i,j}.X)$. Then we replace each subterm s_i occurring in the equations for $s_{i+1} \ldots s_n$ with its equivalent term. When, in the equation for $s_n \equiv P$, we have replaced s_{n-1}, we are done. ■

For sequential states the axioms of $\mathcal{A}_{IGSMP}$ involved are just the standard axioms of [26], and the axioms for priority and probabilistic choice. From Theorem 3.8 and by resorting to arguments similar to those presented in [26] and [17] we derive the completeness of $\mathcal{A}_{IGSMP}$.

Theorem 3.9 $\mathcal{A}_{IGSMP}$ is complete for $\simeq$ over finite state $IGSMP_{sg}$ processes.

Proof The proof of this theorem follows the lines of the proof of [26]. In particular weights are treated as rates of exponential distributions in the proof of [17]. ■

[7] The definition of the operational rule for "$|_{S,M}$" must allow for actions "τ" to be skipped [2], as reflected by axiom ($SM4$).

$(A1)$	$P + Q = Q + P$	$(A2)$	$(P + Q) + R = P + (Q + R)$
$(A3)$	$a.P + a.P = a.P$	$(A4)$	$P + \underline{0} = P$
$(Tau1)$	$\gamma.\tau.P = \gamma.P$	$(Tau2)$	$P + \tau.P = \tau.P$
$(Tau3)$	$a.(P + \tau.Q) + a.Q = a.(P + \tau.Q)$		
$(TAct)$	$<f, w>.P = <f_1^+, w>.f_1^-.P$		
$(Prob)$	$<f_i^+, w>.P + <f_i^+, w'>.P = <f_i^+, w + w'>.P$		
$(Pri1)$	$\tau.P + \theta.Q = \tau.P$	$(Pri2)$	$<f_i^+, w>.P + g_j^-.Q = <f_i^+, w>.P$
$(Hi1)$	$\underline{0}/L = \underline{0}$	$(Hi2)$	$(\gamma.P)/L = \gamma.(P/L) \quad \gamma \notin L$
$(Hi3)$	$(a.P)/L = \tau.(P/L) \quad a \in L$	$(Hi4)$	$(P + Q)/L = P/L + Q/L$
$(Rel1)$	$\underline{0}[\varphi] = \underline{0}$	$(Rel2)$	$(a.P)[\varphi] = \varphi(a).(P[\varphi])$
$(Rel3)$	$(\theta.P)[\varphi] = \theta.(P[\varphi])$	$(Rel4)$	$(P + Q)[\varphi] = P[\varphi] + Q[\varphi]$
(Par)	$P \parallel_{S,M} Q = P \,\lfloor\!\lfloor_{S,M} Q + Q \,\lfloor\!\lfloor_{S,\overline{M}} P + P \mid_{S,M} Q$		
$(LM1)$	$\underline{0} \,\lfloor\!\lfloor_{S,M} P = \underline{0}$		
$(LM2)$	$(a.P) \,\lfloor\!\lfloor_{S,M} Q = \underline{0}$		$a \in S$
$(LM3)$	$(a.P) \,\lfloor\!\lfloor_{S,M} Q = a.(P \parallel_{S,M} Q)$		$a \notin S$
$(LM4)$	$(<f_i^+, w>.P) \,\lfloor\!\lfloor_{S,M} Q = <f_{n(M_f)}^+, w>.(P \parallel_{S,M \cup \{f:(n(M_f), l_i)\}} Q)$		
$(LM5)$	$(f_i^-.P) \,\lfloor\!\lfloor_{S,M} Q = f_j^-.(P \parallel_{S,M-\{f:(j,l_i)\}} Q)$		$f:(j, l_i) \in M$
$(LM6)$	$(P + Q) \,\lfloor\!\lfloor_{S,M} R = P \,\lfloor\!\lfloor_{S,M} R + Q \,\lfloor\!\lfloor_{S,M} R$		
$(SM1)$	$P \mid_{S,M} Q = Q \mid_{S,M} P$		
$(SM2)$	$\underline{0} \mid_{S,M} P = \underline{0}$		
$(SM3)$	$(\gamma.P) \mid_{S,M} (\gamma'.Q) = \underline{0}$		$(\gamma \notin S \ \vee \ \gamma \neq \gamma') \wedge \tau \notin \{\gamma, \gamma'\}$
$(SM4)$	$(\tau.P) \mid_{S,M} Q = P \mid_{S,M} Q$		
$(SM5)$	$(a.P) \mid_{S,M} (a.Q) = a.(P \parallel_{S,M} Q)$		$a \in S$
$(SM6)$	$(P + Q) \mid_{S,M} R = P \mid_{S,M} R + Q \mid_{S,M} R$		
$(Rec1)$	$recX.P = recY.(P\{Y/X\})$ provided that Y is not free in *rec*X.P		
$(Rec2)$	$recX.P = P\{recX.P/X\}$		
$(Rec3)$	$Q = P\{Q/X\} \Rightarrow Q = recX.P$ provided that X is strongly guarded in P		

Fig. 6. Axiomatization for IGSMP

Example 3.10 Let us consider the system $recX.f.X \parallel_\emptyset recX.f.X$ of the previous example. In the following we show how this process can be turned into a sequential process by applying the procedure presented in the proof of Theorem 3.8. In the following we let f_i^+ stand for $<f_i^+, 1>$ and we abreviate $\mathcal{A}_{IGSMP} \vdash P = Q$ with $P = Q$. Moreover we let $P \equiv recX.f.X$ and $P' \equiv f_1^-.recX.f.X$. Initially we note that $P = f.recX.f.X = f_1^+.f_1^-.recX.f.X$, by applying $(Rec2)$ and $(TAct)$. We start the procedure of the proof of Theorem 3.8 with the initial state $P \parallel_{\emptyset,\emptyset} P$. We have:

$$P \parallel_{\emptyset,\emptyset} P = f_1^+.(P' \parallel_{\emptyset,\{f:(1,l_1)\}} P) + f_1^+.(P' \parallel_{\emptyset,\{f:(1,l_1)\}} P)$$

by applying (Par), $(LM4)$ and $(SM3)$. From this equation we derive:

$$P \parallel_{\emptyset,\emptyset} P = <f_1^+, 2>.(P' \parallel_{\emptyset,\{f:(1,l_1)\}} P)$$

by applying $(Prob)$. Then, we have:

$$P' \parallel_{\emptyset,\{f:(1,l_1)\}} P = f_2^+.(P' \parallel_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}} P')$$

by applying (Par), $(LM4)$, $(LM5)$, $(SM3)$ and $(Pri2)$. Then, we have:

$$P' \parallel_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}} P' = f_1^-.(P \parallel_{\emptyset,\{f:(2,r_1)\}} P') + f_2^-.(P \parallel_{\emptyset,\{f:(1,r_1)\}} P')$$

by applying (Par), $(LM5)$ and $(SM3)$. From this equation we derive:

$$P' \parallel_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}} P' = f_1^-.(P \parallel_{\emptyset,\{f:(2,r_1)\}} P') + f_2^-.(P' \parallel_{\emptyset,\{f:(1,l_1)\}} P)$$

by applying (Par), $(A1)$ and $(SM1)$ to $P \parallel_{\emptyset,\{f:(1,r_1)\}} P'$. Finally we have:

$$P \parallel_{\emptyset,\{f:(2,r_1)\}} P' = f_1^+.(P' \parallel_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}} P')$$

by applying (Par), $(LM4)$, $(LM5)$, $(SM3)$ and $(Pri2)$. Now we perform the second part of the procedure where we generate recursive processes and we substitute states with equivalent terms. We start with $P \parallel_{\emptyset,\{f:(2,r_1)\}} P'$. Since the state does not occur in its equivalent term we do not have to generate any recursion. Substituting the state with its equivalent term in the other equations generates the new equation:

$$P' \parallel_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}} P' = f_1^-.f_1^+.(P' \parallel_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}} P') + f_2^-.(P' \parallel_{\emptyset,\{f:(1,l_1)\}} P).$$

Then we consider the state $P' \parallel_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}} P'$. We change its equation by generating a recursion as follows:

$$P' \parallel_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}} P' = recY.(f_1^-.f_1^+.Y + f_2^-.(P' \parallel_{\emptyset,\{f:(1,l_1)\}} P)).$$

Substituting the state with its equivalent term in the remaining equations generates the new equation:

$$P' \parallel_{\emptyset,\{f:(1,l_1)\}} P = f_2^+.recY.(f_1^-.f_1^+.Y + f_2^-.(P' \parallel_{\emptyset,\{f:(1,l_1)\}} P)).$$

Now we consider the state $P' \parallel_{\emptyset,\{f:(1,l_1)\}} P$. We change its equation by generating a recursion as follows:

$$P' \parallel_{\emptyset,\{f:(1,l_1)\}} P = recX.(f_2^+.recY(f_1^-.f_1^+.Y + f_2^-.X)).$$

Substituting the state with its equivalent term in the remaining equations generates the new equation:

$$P \parallel_{\emptyset,\emptyset} P = <f_1^+, 2>.recX.(f_2^+.recY(f_1^-.f_1^+.Y + f_2^-.X)).$$

Therefore we have turned our initial system $recX.f.X \parallel_\emptyset recX.f.X$ into the recursive sequential process $<f_1^+, 2>.recX.(f_2^+.recY(f_1^-.f_1^+.Y + f_2^-.X))$. Note that the operational semantics of this process generates the labelled transition system of Fig. 5. ■

4 Deriving the Performance Model

In this section we show how to formally derive a *GSMP* from a system specification. In particular this transformation is possible only if the specification of the system is *complete* both from the *interaction* and from the *performance* point of view.

A specification is complete from the interaction viewpoint if the system specified is not a part of a larger system which may influence its behavior, hence when every standard action appearing in its semantic model is an *internal* τ action. Note that the states of the semantic model of such a system can be classified as follows:

- *choice states*: states whose outgoing transitions are all (weighted) delay starts
- *timed states*: states whose outgoing transitions are all delay terminations
- *silent states*: states whose outgoing transitions are all τ actions

A specification is complete from the performance viewpoint if all the choices in which the specified system may engage are quantified probabilistically. This means that the semantic model must not include silent states with a non-deterministic choice among different future behaviors. In other words a silent state either must have only one outgoing τ transition, or all its outgoing τ transitions must lead to equivalent behaviors. This notion can be formally defined as follows: A semantic model is complete w.r.t. performance if it can be reduced, by aggregating weakly bisimilar states (see Sect. 3), to a model without silent states.

Provided that a system $P \in IGSMP_g$ satisfies these two conditions, we now present a formal procedure for deriving the *GSMP* representing the performance behavior of P from its integrated semantic model $\mathcal{I}[\![P]\!] = (S_P, L, \longrightarrow_P, P)$.

4.1 Elimination of τ Actions

The first phase is to *minimize* the state space S_P by aggregating states that are equivalent according to the notion of weak bisimulation defined in Sect. 3.

Since we supposed that the system P satisfies the two conditions above, a side effect of this minimization is that all τ actions disappear from $\mathcal{I}[\![P]\!]$.

We denote the resulting *LTS* with $(S_{P,m}, L_m, \longrightarrow_{P,m}, P)$, where m stands for "minimal". We have $L_m = \mathit{TAct}^+ \cup \mathit{TAct}^-$, hence $S_{P,m}$ includes only choice states and timed states.

4.2 Solution of Choice Trees

The second phase is the transformation of every *choice tree* present in the semantic model into a single probabilistic choice. A choice tree is formed by the possible choice paths that go from a given choice state (the root of the tree) to a timed state (a leaf of the tree). Note that such trees cannot include loops composed of one or more transitions, because after each delay start the number of delays in execution strictly increases. To be precise, such trees are directed acyclic graphs (*DAGs*) with root, since a node may have multiple incoming arcs. The choice trees are flattened into a single choice that goes directly from the root to the leaves of the tree, with the following inductive procedure.

Initially (at step 0) we transform our semantic model by turning all weights into the corresponding probability values. We denote the resulting *LTS* with $(S_{P,p,0}, L_p, \longrightarrow_{P,p,0}, P)$, where p stands for "probabilistic", defined by:

- $S_{P,p,0} = S_{P,m}$
- $L_p = \mathbb{R}_{[0,1]} \cup \mathit{TAct}^-$, where positive real numbers represent probabilities
- $\longrightarrow_{P,p,0} = (\longrightarrow_{P,m} \cap\, S_{P,m} \times \mathit{TAct}^- \times S_{P,m}) \cup \{(s, \mathit{prob}, s') \mid$
$$\exists f, i, w : s \xrightarrow{<f_i^+, w>}_{P,m} s' \wedge \mathit{prob} = w / \mathit{TotalW}(s)\}$$
  where:
$$\mathit{TotalW}(s) = \sum \{\!|\, w \mid \exists f, i, s' : s \xrightarrow{<f_i^+, w>}_{P,m} s' \,|\!\}$$

Hence now we have a semantic model with delay termination transitions and *probabilistic* transitions labeled by a probability *prob*. Note that delay start events are removed from transition labels. The occurrence of such events becomes implicit in the representation of system behavior similarly as in *GSMPs*.

At the k-th step, beginning from the *LTS* $(S_{P,p,k-1}, L_p, \longrightarrow_{P,p,k-1}, P)$ we eliminate a node in a choice tree, thus reducing its size. This is done by considering a choice state $s \in S_{P,p,k-1}$ with incoming probabilistic transitions. Such transitions are removed and replaced by a new set of probabilistic transitions which are determined in the following way. Each incoming probabilistic transition is divided into multiple transitions, one for each probabilistic transition that leaves the state s. Its probability is distributed among the new transitions in parts that are proportional to the probabilities of the transitions that leave s. Moreover, if s has no incoming delay termination transitions, then s is eliminated together with its outcoming probabilistic transitions. Therefore the resulting *LTS* $(S_{P,p,k}, L_p, \longrightarrow_{P,p,k}, P)$, is defined by:

- $S_{P,p,k} = \begin{cases} S_{P,p,k-1} & \text{if } \exists f, i, s' : s' \xrightarrow{f_i^-}_{P,p,k-1} s, \\ S_{P,p,k-1} - \{s\} & \text{otherwise} \end{cases}$

- $\longrightarrow_{P,p,k} = (\longrightarrow_{P,p,k-1} \cap S_{P,p,k-1} \times \mathit{TAct}^- \times S_{P,p,k-1}) \cup$
  $\{(s', prob, s'') \mid s', s'' \in S_{P,p,k} \wedge s'' \neq s \wedge$
  $\mathit{MultiProb}_s(s', s'') \neq \emptyset \wedge prob = \sum \mathit{MultiProb}_s(s', s'')\}$

  where:

  $$\begin{aligned} \mathit{MultiProb}_s(s', s'') = \{\!| prob \mid s' \xrightarrow{prob}_{P,p,k-1} s'' \vee \\ (\exists prob', prob'' : s' \xrightarrow{prob'}_{P,p,k-1} s \wedge \\ s \xrightarrow{prob''}_{P,p,k-1} s'' \wedge prob = prob' \cdot prob'') |\!\} \end{aligned}$$

The algorithm terminates when we reach k for which $\nexists s \in S_{P,p,k-1}$ with s choice state with incoming probabilistic transitions. Since at every step we eliminate a node in a choice tree of the initial semantic model, thus reducing its size, we are guaranteed that this will eventually happen. Let t be such k.

The *LTS* that results from this second phase is denoted by $(S_{P,p}, L_p, \longrightarrow_{P,p}, P) = (S_{P,p,t}, L_p, \longrightarrow_{P,p,t}, P)$.

If the nodes of a choice tree are eliminated by following a breadth-first visit from the root, it can be easily seen that the time complexity of the algorithm above is just linear in the number of probabilistic transitions forming the tree (the *DAG*). This because by following this elimination ordering, each node to be eliminated has one ingoing probabilistic transition only.

4.3 Derivation of the GSMP

Now we show how to derive a generalized semi-Markov processes from the semantic model obtained at the end of the previous phase.

A *generalized semi-Markov process (GSMP)* [24] is a stochastic process defined on a set of states $\{s \mid s \in \mathcal{S}\}$ as follows. In each state s there is a set of active elements $ElSt(s)$ taken from a set El, that decay at the rate $C(e, s)$, $e \in El$. The set El is partitioned into two sets El' and El^* with $El = El' \cup El^*$. If $e \in El'$ the element e has an exponentially distributed lifetime, if instead $e \in El^*$ it has a generally distributed lifetime. Whenever in a state s an advancing element e dies, the process moves to the state $s' \in \mathcal{S}$ with probability $Pr(s, e, s')$.

A *GSMP* can be represented by a tuple:

$$(\mathcal{S}, El, ElPDF, ElSt, C, \longrightarrow, Pr, P_{init})$$

where:

- $\mathcal{S}$ is the set of the states of the *GSMP*.
- El is the set of the elements of the *GSMP*.
- $ElPDF : El \longrightarrow PDF^+$ is a function that associates with each element the distribution of its lifetime.
- $ElSt : \mathcal{S} \longrightarrow El$ is a function that associates with each state the set of its active elements.
- $C : El \times \mathcal{S} \mathrel{-\!\!\!\circ\!\!\!\rightarrow} \mathbb{R}^+$ is a partial function that associates a decay rate with each active element of each state. C is partial because for each $s \in \mathcal{S}$ it is defined only for the (e, s) such that $e \in ElSt(s)$.
- $\longrightarrow \subseteq \mathcal{S} \times El \times \mathcal{S}$ is a relation that represents the transitions between the states of the *GSMP*. They are labeled by the element $e \in El$ that terminates. We include only transitions for which $Pr(s, e, s') > 0$.
- Pr is a function that associates a (non zero) probability with each transition of the *GSMP* (relation $\longrightarrow$). The meaning of Pr is: if in s an element e terminates, with probability $Pr(s, e, s')$ the process moves into state s'. For what we said in the previous item, Pr is never zero over its domain, whilst it is considered as zero outside.
- $P_{init} : \mathcal{S} \longrightarrow \mathbb{R}_{[0,1]}$ is a function that associates with each state the probability that it is the initial state.

Note that, given a tuple defining a *GSMP*, the sets El' and El^* are derived in the following way (where $Exp(\lambda)$ is the exponential distribution with rate λ):

- $El' = \{e \in El \mid \exists \lambda : ElPDF(e) = Exp(\lambda)\}$
- $El^* = \{e \in El \mid \nexists \lambda : ElPDF(e) = Exp(\lambda)\}$

With respect to the general definition of a *GSMP* given above, we have that in an *IGSMP* all elements (delays) decay at rate 1, i.e. they all advance uniformly with time at the same speed.

The *performance semantic model* $\mathcal{P}[\![P]\!]$ of $P \in IGSMP_g$ is derived from the *LTS* $(S_{P,p}, L_p, \longrightarrow_{P,p}, P)$ as follows.

The elements of the *GSMP* are the "identified" delays f_i labeling the transitions of $\longrightarrow_{P,p}$. The states of the *GSMP* are the timed states of $S_{P,p}$. A transition leaving a state of the *GSMP* is derived beginning from a delay termination transition leaving the corresponding timed state of $S_{P,p}$ and, in the case this transition leads to a choice state, from a probabilistic transition leaving this state. The timed state of $S_{P,p}$ reached in this way is the state of the *GSMP* the derived transition leads to. Note that we are certain to reach a timed state because all choice trees have been solved and, consequently (see Sect. 4.2) choice states cannot have incoming probabilistic transitions. Each transition of the *GSMP* is labeled by the element f_i terminating in the corresponding termination transition. The probability associated with a transition of the *GSMP* (function Pr) is the probability of the corresponding probabilistic transition (or probability 1 if the transition is derived from a delay termination transition leading directly to a timed state).

The performance semantics of an *IGSMP* process P is defined as follows.

Definition 4.1 The *performance semantics* $\mathcal{P}[\![P]\!]$ of a process $P \in IGSMP_g$ is a *GSMP* represented by the tuple:

$$\mathcal{P}[\![P]\!] = (\mathcal{S}, El, ElPDF, ElSt, C, \longrightarrow, Pr, P_{init})$$

where:

- $\mathcal{S} = \{s \in S_{P,p} \mid \nexists prob, s' : s \xrightarrow{prob}_{P,p} s'\}$
- $El = \{f_i \mid \exists s, s' : s \xrightarrow{f_i^-}_{P,p} s'\}$
- $\forall f_i \in El.\, ElPDF(f_i) = f$
- $\forall s \in \mathcal{S}.\, ElSt(s) = \{f_i \mid \exists s' : s \xrightarrow{f_i^-}_{P,p} s'\}$
- $\forall s \in \mathcal{S}, \forall f_i \in ElSt(s).\ C(f_i, s) = 1$
- We let $\longrightarrow \; = \; \longrightarrow_1 \cup \longrightarrow_2$ where:
  - $\cdot$ $\longrightarrow_1 \; = \{(s, f_i, s') \mid s \xrightarrow{f_i^-}_{P,p} s' \wedge s' \in \mathcal{S}\}$
  - $\cdot$ $\longrightarrow_2 \; = \{(s, f_i, s') \mid \exists s'', prob : s \xrightarrow{f_i^-}_{P,p} s'' \wedge s'' \xrightarrow{prob}_{P,p} s'\}$ [8]
- $\forall s, s' \in \mathcal{S}; f_i \in ElSt(s)$ such that $s \xrightarrow{f_i} s'$
  - $\cdot$ if $s \xrightarrow{f_i}_1 s'$ let $Pr(s, f_i, s') = 1$
  - $\cdot$ if $s \xrightarrow{f_i}_2 s'$ let $Pr(s, f_i, s')$ be the unique *prob* such that:
  $$\exists s'' : s \xrightarrow{f_i^-}_{P,p} s'' \;\wedge\; s'' \xrightarrow{prob}_{P,p} s'$$
- P_{init} is defined as follows:
  - $\cdot$ if $P \in \mathcal{S}$ then:
  $$P_{init}(P) = 1$$

[8] The transition relations $\longrightarrow_1$ and $\longrightarrow_2$ are disjoint. This because the only transition in $\longrightarrow_{P,p}$ that leaves state s and is labeled by a given element f_i, leads either to a choice state or to a timed state.

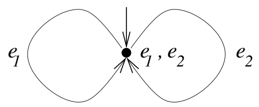
$P_{init}(s) = 0 \quad \forall s \in \mathcal{S}, s \neq P$

· if $P \notin \mathcal{S}$ then $\forall s \in \mathcal{S}$:

$P_{init}(s) = prob \quad$ if $P \xrightarrow{prob}_{P,p} s$

$P_{init}(s) = 0 \quad$ if $\not\exists prob : P \xrightarrow{prob}_{P,p} s$ ■

Example 4.2 In Fig. 7 we show the *GSMP* derived, by applying the formal translation we have presented, from the integrated semantic model of Fig. 4. In particular the *GSMP* is obtained, as described above, from the minimal model of Fig. 5. Since such model does not include standard action transitions the system considered is complete both from the interactive and the performance viewpoints. In the *GSMP* of Fig. 7 the states are labeled by the active elements and the transitions with the terminating elements. The probability Pr associated to each transition of the *GSMP* that is shown in the picture is 1. Moreover P_{init} is 1 for the unique state of the *GSMP* (it is pointed by the arrow to outline the fact that it is the initial state). The elements e_1 and e_2 represent the delays f_1 and f_2 respectively, and the probability distribution function of both is given by function f. ■



$$ElPDF(e_1) = f, ElPDF(e_2) = f$$

Fig. 7. Derived GSMP

5 Example: Queueing Systems G/G/1/q

In this section we present an example of specification with *IGSMP*. In particular we concentrate on Queuing Systems (*QSs*) $G/G/1/q$, i.e. *QSs* which have one server and a FIFO queue with q-1 seats and serve a population of unboundedly many customers. In particular the *QS* has an interarrival time which is generally distributed with distribution f and a service time which is generally distributed with distribution g.

Such a system can be modeled with *IGSMP* as follows. Let a be the action representing the fact that a new customer arrives at the queue of the service center, d be the action representing that a customer is delivered by the queue to the server. The process algebra specification is the following one: [9]

[9] In the specification we use process constants, instead of the operator "*recX*", to denote recursion. The reason being that the use of constants is suitable for doing

- $QS_{G/G/1/q} \triangleq \mathit{Arrivals} \parallel_{\{a\}} (\mathit{Queue}_0 \parallel_{\{d\}} \mathit{Server})$

  · $\mathit{Arrivals} \triangleq f.a.\mathit{Arrivals}$
  · $\mathit{Queue}_0 \triangleq a.\mathit{Queue}_1$
  $\mathit{Queue}_h \triangleq a.\mathit{Queue}_{h+1} + d.\mathit{Queue}_{h-1} \qquad 0 < h < q-1$
  $\mathit{Queue}_{q-1} \triangleq a.\mathit{Queue}_{q-1} + d.\mathit{Queue}_{q-2}$
  · $\mathit{Server} \triangleq d.g.\mathit{Server}$

We have specified the whole system as the composition of the arrival process, the queue and the server which communicate via action types a and d. Then we have separately modeled the arrival process, the queue, and the server. As a consequence if we want to modify the description by changing the interarrival time distribution f or the service time distribution g , only component *Arrivals* or *Server* needs to be modified while component *Queue* is not affected. Note that the role of actions a and d is defining interactions among the different system components. Such actions have zero duration and they are neglected from the performance viewpoint.

In Fig. 8 we show $\mathcal{I}[\![QS_{G/G/1/q}]\!]$. In this picture A stands for *Arrivals*, A' stands for $f^-.a.\mathit{Arrivals}$, A'' stands for $a.\mathit{Arrivals}$. Similarly, S stands for *Server*, S' stands for $g.\mathit{Server}$, S'' stands for $g^-.\mathit{Server}$. Moreover, Q_h stands for Queue_h, for any h. We omit parallel composition operators in terms, so, e.g., AQ_hS stands for $\mathit{Arrivals} \parallel_{\{a\}} (\mathit{Queue}_h \parallel_{\{d\}} \mathit{Server})$.

In order to derive the performance model of the system $QS_{G/G/1/q}$ we have to make sure that it is complete both from the interaction and the performance viewpoints. In Fig. 8 we have visible actions a and d, therefore the behavior of the system can be influenced by interaction with the environment and is not complete. We make it complete by considering $QS_{G/G/1/q}/\{a,d\}$ so that every action in the semantic model of Fig. 8 becomes a τ action.

As far as completeness w.r.t. performance is concerned, we present in Fig. 9 the minimal version of $\mathcal{I}[\![QS_{G/G/1/q}/\{a,d\}]\!]$, obtained by aggregating weakly bisimilar states (see Sect. 3). Since in the minimal model there are no longer internal τ actions, we have that our system is complete also w.r.t. performance.

By applying the formal procedure defined in Sect. 4, hence by solving choice trees in the minimal model of Fig. 9, we finally obtain the *GSMP* of Fig. 10. The probability Pr associated to each transition of the *GSMP* is 1. P_{init} is 1 for the state pointed by the arrow and 0 for all the other states. The elements e_1 and e_2 represent the delays f and g.

---

specifications, while the use of operator "*recX*" is preferable when dealing with axiomatizations. The two constructs are shown to be completely equivalent in [25].

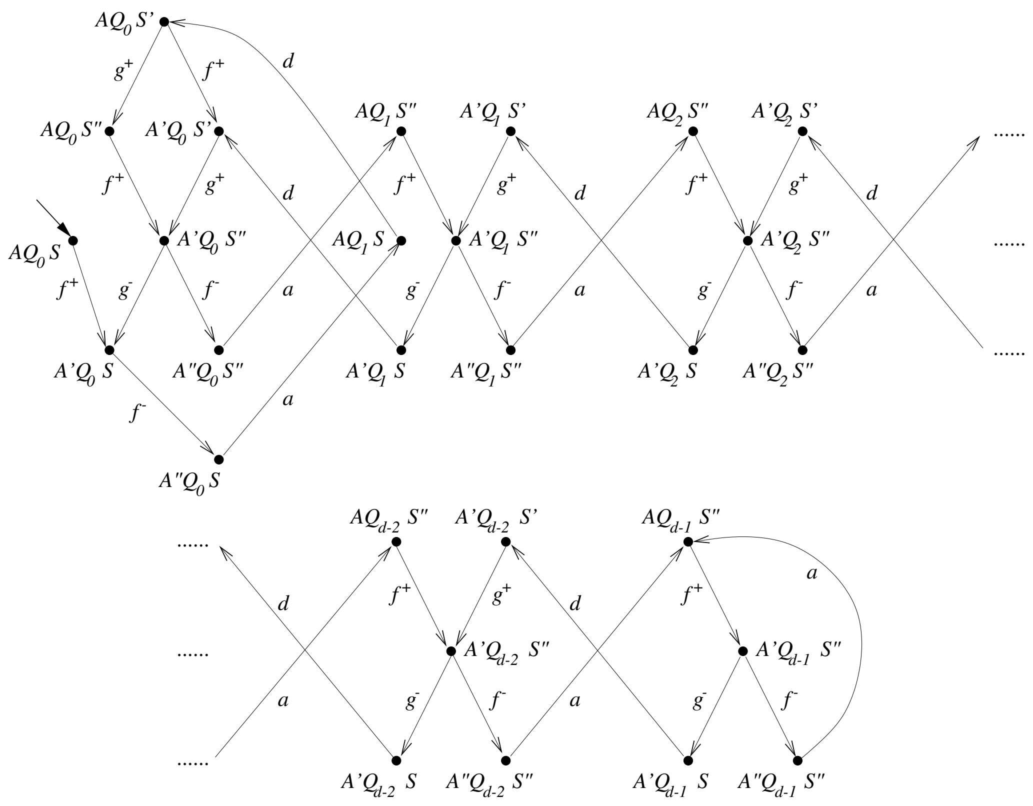


Fig. 8. Integrated Semantic Model

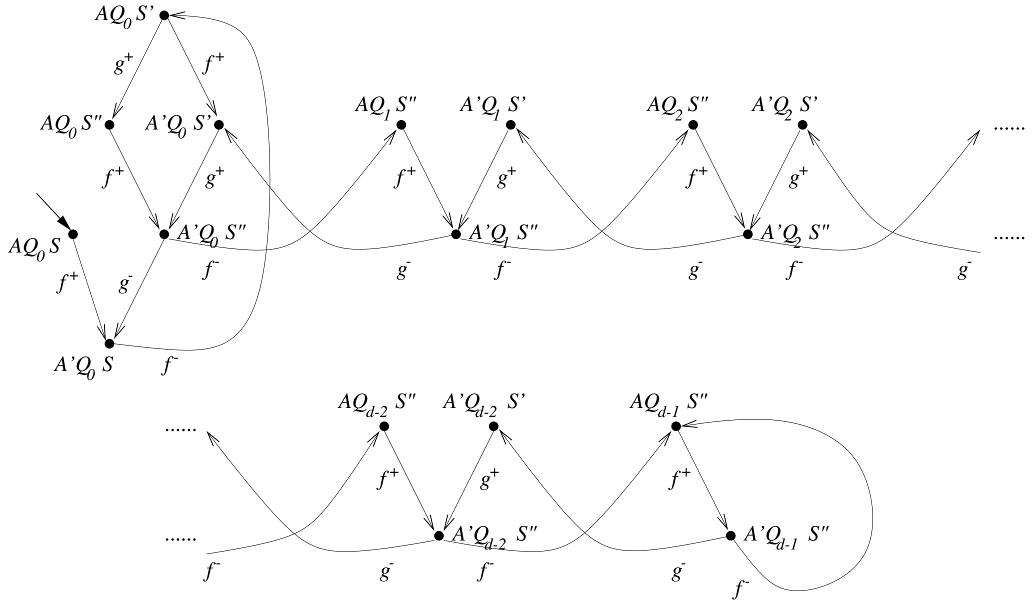
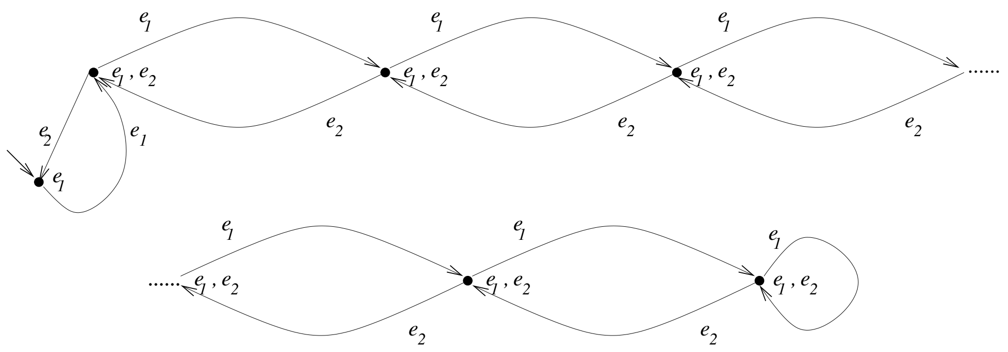


Fig. 9. Minimal Semantic Model

6 Conclusion

In this section we present some related work and we outline some open problems left for future research.



$$ElPDF(e_1) = f, \quad ElPDF(e_2) = g$$

Fig. 10. Derived GSMP

6.1 Related work

Several algebraic languages which express generally distributed durations like *IGSMP* have been previously developed. We start from the languages that follow a completely different approach in representing the behavior of systems.

In [10] a *truly concurrent approach* to modeling systems is proposed which employs general distributions. From a term of the algebra presented in [10] a *truly concurrent* semantic model (a stochastic extension of a bundle event structure) is derived that represents statically the concurrency of the system by expressing the components of the system and the causal relationships among them. Therefore the behavior of the system is not described by representing explicitly all possible global system states as it happens in labeled transition systems. In this way a very concise semantic model is obtained where duration distributions can be statically associated with durational actions. The drawback of this approach is that the semantic models produced must nevertheless be translated to a transition system form before their performance can be evaluated. This because in *GSMPs* the evolution of a stochastic process is represented in such a form. Another algebraic approach to modeling systems with general distributions is the *discrete event simulation approach* [16,11]. For example in [16] an algebra is developed that extends CCS with temporal and probabilistic operators in order to formally describe discrete event simulations. Such algebra employs actions with null duration (events) and delays similarly as in *IGSMP*. The states produced by the operational semantics include explicitly the residual durations of delays (as real numbers), hence the semantic models of systems are *not* finite. This approach excludes a priori the possibility of making mathematical analysis of such models by means of established theoretical results such as analytical solution methods for (insensitive) *GSMPs* [24].

Other languages have been previously developed, that are more similar to *IGSMP*, in that they represent the behavior of specified systems via labeled transition systems which are (in many cases) finite. According to these approaches [12,28,22] the execution of durational actions is represented in an abstract way, as in *IGSMP*, without including explicitly their residual durations (as real numbers) in states. In [12] the technique of "start reference" is employed in order to have a pointer to the system state where an action begins its execution. In [28], instead, information about causality relations among actions is exploited in order to establish the starting point of actions. In a recent work [22] a methodology for obtaining finite semantic models from the algebra of [16] is defined, which is based on symbolic operational semantics. Such semantics generates symbolical transition systems which abstract from time values by representing operations on values as symbolic expressions. The drawback of these approaches is that the structure of the semantic models generated is very different from that of *GSMPs*. It is therefore not always clear how to derive a performance model for a specified system and [22] only provides a (quite involved) procedure for deriving a *GSMP* from systems belonging to a certain class.

The languages that are closest to *IGSMP*, in that they produce semantic models which represent probabilistic durations as in *GSMPs* are those of [1,5,11]. In particular such semantic models represent the performance behavior of systems by means of (some kind of) clocks with probabilistic duration which can be easily seen as the elements of a *GSMP*. With the language of [1], performance models are derived from terms specifying systems by applying to them a preliminary procedure that gives a different name to each durational action of the term. In this way, each name represents a different clock in the semantic model of the system. In the approach of [1] the events of action starts are not explicitly expressed in the semantic models and choices are resolved via the race policy (alternative actions are executed in parallel and the first action that terminates wins) instead of the preselection policy as in *IGSMP*. The language of [11] is endowed with an abstract semantics which may generate finite intermediate semantic models (from these models it is then possible to derive the infinite models which are used for discrete event simulation). With this language clock names must be explicitly expressed in the term that specify the system and the fact that a different name is used for each clock is ensured by imposing syntactical restrictions in terms. As in *IGSMP* the execution of a clock is represented by the events of clock start and clock termination, but here these two events must be explicitly expressed in the term specifying a system and they are not automatically generated by the operational semantics. Unfortunately the language of [11] can only express choices between events of clock terminations (which are resolved through race policy) and cannot express probabilistic choices which are a basic ingredient of *GSMPs*. With the language we have previously developed in [5], clock names and events of start and termination are automatically generated, as in

IGSMP, by the operational semantics. In this way system specifications can simply express general distributed delays as probability distribution functions and we do not have to worry about clock names and events. A drawback of the approaches of [1,5,11] w.r.t. *IGSMP* is that there is no easy way to decide equivalence of systems (hence to minimize their state space). This is because in order to establish the equivalence of two systems it is necessary to associate in some way the names of the clocks used by one system with the names of the corresponding clocks used by the other one. Trying to extend the notion of bisimulation in this way turns out to be rather complex especially in the presence of probabilistic choices (see [5]). In *IGSMP*, instead, names of clocks are dynamically generated by the operational semantics with a fixed rule. In this way equivalent systems get the same names for clocks and there is no need to associate names of clocks for establishing equivalence. We can, therefore, rely on standard (probabilistic) bisimulation and we have the opportunity to reuse existing results and tools.

6.2 Future research

As far as future work is concerned we are trying to extend our approach in two main directions.

The capability of *IGSMP* to express general distributions should allow us to extend the notion of observational congruence as follows. The idea is that we could be "weak" also on delays. A delay could be seen as a "timed τ" and we could equate a sequence of timed τ with a single timed τ provided that distribution of durations are in the correct relationship. For example a sequence (or a more complex pattern) of exponential τ could be equated by a phase-type distributed τ. The solution of this problem seems quite involved.

Moreover we are investigating the possibility of introducing in *IGSMP* an operator for delay interruption. This requires the introduction of a special event in semantic models representing "delay interruption" instead of "delay termination". An interruption operator would greatly enhance the expressive power of *IGSMP*, since a preemption mechanism is needed to model many real systems.

Acknowledgements

We thank the anonymous referees for their helpful comments. This research has been partially funded by MURST progetto TOSCA.

References

[1] M. Ajmone Marsan, A. Bianco, L. Ciminiera, R. Sisto, A. Valenzano, *"A LOTOS Extension for the Performance Analysis of Distributed Systems"*, in IEEE/ACM Trans. on Networking 2:151-164, 1994

[2] L. Aceto, *"On "Axiomatising Finite Concurrent Processes" "* in SIAM Journal on Computing 23(4):852-863, 1994

[3] R. Alur, C. Courcoubetis, D. Dill *"Model-Checking in Dense Real-Time"*, in Information and Computation 104:2-34, 1993

[4] L. Aceto, M. Hennessy, *"Adding Action Refinement to a Finite Process Algebra"*, in Information and Computation 115:179-247, 1994

[5] M. Bravetti, M. Bernardo, R. Gorrieri, *"Towards Performance Evaluation with General Distributions in Process Algebras"*, in Proc. of the *9th Int. Conf. on Concurrency Theory (CONCUR '98)*, LNCS 1466:405-422, Nice (France), 1998

[6] M. Bernardo, R. Gorrieri, *"A Tutorial on EMPA: A Theory of Concurrent Processes with Nondeterminism, Priorities, Probabilities and Time"*, in Theoretical Computer Science 202:1-54, 1998

[7] M. Bravetti, R. Gorrieri, *"Deciding and Axiomatizing ST Bisimulation for a Process Algebra with Recursion and Action Refinement"*, Technical Report UBLCS-99-1, University of Bologna (Italy), February 1999. An extended abstract appeared in Proc. of the *6th Int. Workshop on Expressiveness in Concurrency (EXPRESS '99)*, ENTCS 27, Eindhoven (The Netherlands), August 1999

[8] M. Bravetti, R. Gorrieri, *"A Complete Axiomatization for Observational Congruence of Prioritized Finite-State Behaviors"*, to appear in Proc. of the *27th International Colloquium on Automata, Languages and Programming (ICALP 2000)*, Geneva (Switzerland), July 2000

[9] N. Busi, R.J. van Glabbeek, R. Gorrieri, *"Axiomatising ST-Bisimulation Equivalence"*, in Proc. of the *IFIP Working Conf. on Programming Concepts, Methods and Calculi (PROCOMET '94)*, pp. 169-188, S. Miniato (Italy), 1994

[10] E. Brinksma, J.P. Katoen, R. Langerak, D. Latella, *"A Stochastic Causality-Based Process Algebra"*, in Computer Journal 38:553-565, 1995

[11] P.R. D'Argenio, J.P. Katoen, E. Brinksma, *"An Algebraic Approach to the Specification of Stochastic Systems (extended abstract)"*, in Proc. of the *IFIP Working Conf. on Programming Concepts, Methods and Calculi (PROCOMET '98)*, New York (USA), 1998

[12] N. Götz, U. Herzog, M. Rettelbach, *"TIPP - A Stochastic Process Algebra"*, in Proc. of the *1st Workshop on Process Algebras and Performance Modelling (PAPM '93)*, pp. 31-36, Edinburgh (UK), 1993

[13] R.J. van Glabbeek, F.W. Vaandrager, *"Petri Net Models for Algebraic Theories of Concurrency"*, in Proc. of the *Conf. on Parallel Architectures and Languages Europe (PARLE '87)*, LNCS 259:224-242, Eindhoven (The Netherlands), 1987

[14] P.W. Glynn, *"A GSMP formalism for discrete event simulation"*, in Proc. of the IEEE, 77(1): 14-23, 1989

[15] J.F. Groote, *"Transition System Specifications with Negative Premises"*, in Theoretical Computer Science 118(2):263-299, 1993

[16] P.G. Harrison, B. Strulo, *"Stochastic Process Algebra for Discrete Event Simulation"*, in *Quantitative Methods in Parallel Systems*, Springer, pp. 18-37, 1995

[17] H. Hermanns, *"Interactive Markov Chains"*, Ph.D. Thesis, Universität Erlangen-Nürnberg (Germany), 1998

[18] H. Hermanns, *"An Operator for Symmetry Representation and Exploitation in Stochastic Process Algebras"*, in Proc. of the *5th Workshop on Process Algebras and Performance Modeling*, pp. 55-70, Twente (The Netherlands), 1997

[19] H. Hermanns, M. Rettelbach, T. Weiss, *"Formal Characterisation of Immediate Actions in SPA with Nondeterministic Branching"*, The Computer Journal (38)7, 1995

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

[21] C.A.R. Hoare, *"Communicating Sequential Processes"*, Prentice Hall, 1985

[22] K. Kanani, *"A Unified Framework for Systematic Quantitative and Qualitative Analysis of Communicating Systems"*, Ph.D. Thesis, Imperial College (UK), 1998

[23] K.G. Larsen, A. Skou, *"Bisimulation through Probabilistic Testing"*, in Information and Computation 94:1-28, 1991

[24] K. Matthes, *"Zur Theorie der Bedienungsprozesse"*, in Trans. of the *3rd Prague Conf. on Information Theory, Stat. Dec. Fns. and Random Processes*, pp. 513-528, 1962

[25] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989

[26] R. Milner, *"A complete axiomatization for observational congruence of finite-state behaviours"*, in Information and Computation 81:227-247, 1989

[27] X. Nicollin, J. Sifakis, *"An Overview and Synthesis on Timed Process Algebras"*, in Real-Time: Theory in Practice, LNCS 600, 1991

[28] C. Priami, *"Stochastic π-Calculus with General Distributions"*, in *Proc. of the 4th Workshop on Process Algebras and Performance Modelling (PAPM '96)*, CLUT, pp. 41-57, Torino (Italy), 1996