
Mokdad AROUS 1+, Jean-Michel Ilié 2 and Djamel-Eddine Saïdouni 1

1 MISC Laboratory, Mentouri University, Constantine, 25000, Algeria.
2 LIP6 Laboratory, Paris6 University, Paris, France.

Abstract. A new Stochastic Process Algebra called S-LOTOS is investigated, it extends LOTOS in order to specify the durations of actions in terms of generally distributed functions. We present its operational semantics and its underlying semantic model, called Maximality-based Labeled Stochastic Transition System (MLSTS). With regards to performance properties, we show that MLSTS and ST-semantics (Start-Termination) based models are equivalent, but the former brings more compact structure.

Keywords: Performance Evaluation, Maximality Semantics, ST-semantics, Stochastic Process Algebra, Stuttering Equivalence, (Stochastic) Labeled Transition System.

1. Introduction

The importance of studying the stochastic temporal behaviors of concurrent (stochastic) systems is widely recognized, e.g. [1, 12, 15]. In stochastic systems, the durations of actions are expressed probabilistically through probability distribution functions. Among the specification languages, the extension of the Process Algebras called Stochastic Process Algebras (SPAs) take advantage from its compositionality (model a system as the interaction of its components) and abstraction aspects (build up complex models from detailed components but disregarding internal behavior when it is appropriate to do so), whereas providing a formal description context.

Two main approaches have been adopted for expressing random time properties of stochastic systems. The first one considers exponential distributions for action durations. Markovian Process Algebras (MPAs) are SPAs where expressiveness is limited to exponential distributions for specifying the durations of actions, but are useful for many applications. MPA models accord with the interleaving semantics [1, 6, 9], such that the parallel execution of two actions a and b (see the expression E in Figure 1) is assumed to be equivalent to their interleaving execution (see the expression F). The semantic model of a MPA is a transition system wherein each transition is labeled with a pair (a, λ) representing the execution of the action a and the rate λ of the exponential distribution function governing the duration of a. Because of the memoryless property, exponential distributions yield analytically tractable models in the form of Continuous Time Markov Chains (CTMCs) [2, 16].

In the second approach, the durations of actions are specified by general distributions. In fact, exponential distribution appears to be not realistic in many concrete phenomena. It is only appropriate whether the mean values of random variables are known. This is not the case, for instance, when only the minimum and maximum values of the random variables are known.

* Corresponding author. Tel.: (00213) 662 736 739
E-mail address: mokdad.a@gmail.com
Contrary to exponential laws, general distributions allow handling the residual durations of running actions. In Figure 1, it turns out that the two presented expressions are not behaviorally equivalent with regards to the temporal properties, although the same graph structure. This distinction is not possible under the memoryless property of the exponential distribution, since both models yield the same CTMC hence the same performance properties.

For specifying generally distributed durations, the choice of true concurrency semantics appears to be more appropriate [7]. Actually, the system behaviors are never more represented like totally ordered expressions that can specify the studied system, according to the following syntax of expressions:

\[ E ::= \text{stop} \mid \text{exit} \mid (a, f) \cdot E \mid (i, f) \cdot E \mid E \parallel E \mid E \parallel [L] E \mid E \cdot \text{hide} \cdot E \mid E \cdot \text{hide} \cdot L \cdot E \mid E \cdot [b_1/a_1, \ldots, b_n/a_n] \]

In S-LOTOS, stochastic time is handled by using arbitrary distribution functions. An action is represented by a pair \((a, f)\), where \(a\) is the action name and \(f\) is the probability distribution function that governs the duration of \(a\).
The semantic model of an S-LOTOS expression is a Labeled Transitions System (LTS), called Maximality-based Labeled Stochastic Transition System (MLSTS). Within this model, each transition only represents the start of an action execution. Since actions are not considered as atomic, the concurrent execution of multiple actions can be represented, and distinguishing between sequential and parallel executions is possible.

In the semantic model of S-LOTOS, the running actions are represented at the states level. Each instance of running actions is called a maximal event and is identified by a distinct name. In fact, each state of the system is featured by a unique configuration [3]. The configuration of a state \( s \) is denoted \( M[E] \) s.t. \( M \) is the set of maximal events in \( s \) and \( E \) is the behavior expression of \( s \). In addition, a distinct clock is associated with each maximal event of \( M \), to represent the time evolving. Every transition defined from \( s \) is labeled by \( \varepsilon(a, f) \), whenever \( a \) is an action that can be activated from \( E \) iff. the maximal events of the subset \( C \subseteq M \) are terminated. Further \( C \) is called the causality set of the transition. The symbol \( x \) is the name identifying the start event of the new execution of \( a \). The event identification is required to avoid confusion since several instances of running actions can have the same action name.

To illustrate the principles of S-LOTOS and both concepts of maximality and configuration, consider the following two behavior expressions: \( E = (a; f); \text{stop} [] [(b, g)]; \text{stop} \quad \text{and} \quad F = (a; f); (b, g); \text{stop} [] [(b, g)]; (a; f); \text{stop} \). Their respective MLSTSs, obtained by applying the maximality semantics, are represented in Figure 2. Initially, no action has yet been executed, then the set of maximal events is empty, and the initial configurations associated with \( E \) and \( F \) respectively, are \( \varnothing[E] \) and \( \varnothing[F] \). By assuming that the action \( a \) happens first from \( E \) and \( F \), the corresponding transitions are respectively:

\[
\varnothing[E] \xrightarrow{\varepsilon(a, f)} \{x\}[\text{stop}] [] [(b, g)]; \text{stop} \\
\varnothing[F] \xrightarrow{\varepsilon(a, f)} \{x\}[\text{stop}] [] [(b, g)]; \text{stop}.
\]

\( x \) is the event name identifying the starting of \( a \), and it represents a counting down clock which is initially set according to the distribution function \( f \) of the duration of \( a \). In both new resulting configurations, \( x \) is said maximal.

From the new state of \( E \) (according to the semantic of the parallel operator []) the following transition occurs in case \( b \) starts: \( \{x\}[\text{stop}] [] [(b, g)]; \text{stop} \xrightarrow{\varepsilon(b, g)} \{x\}[\text{stop}] [] [(b, g)]; \text{stop} \), where \( y \) is the maximal event name identifying the start of \( b \), and it represents a clock set according to the distribution function \( g \) of the duration of \( b \). In the resulting state, the clock of \( y \) starts counting down while the clock of \( x \) continues recording the time of \( a \).

From the new state \( \{x\}[\text{stop}] [] [(b, g)]; \text{stop} \) of \( F \) and the semantic of the prefix operator \( (\cdot) \) expressing the sequentiality in execution, we deduce that the start of \( b \) is constrained by the causality dependence against \( x \). Actually, it is submitted to the end of the execution of \( a \), inducing that the clock of \( x \) has expired. This results in the following transition: \( \{x\}[\text{stop}] [(b, g)]; \text{stop} \xrightarrow{\varepsilon(b, g)} \{y\}[\text{stop}] \), In the resulting state, the only maximal event is the one identified by \( y \), representing the start of \( b \), and the value of the associated clock is set according to the distribution function \( g \) of the duration of \( b \).

![Fig. 2: Behaviors of \( E \) and \( F \) according to the maximality semantics](image)

As this can be seen in Figure 2, the behaviors represented by the configurations \( \{y\}[\text{stop}] \) and \( \{x\}[\text{stop}] [] [(b, g)]; \text{stop} \) are rather different, in particular there is a single maximal event (identified by \( y \)) in the first one, whereas two maximal events appear in the second (identified by \( x \) and \( y \)). Observe that a symmetric scenario happens when the action \( b \) happens first.
3. Formal Aspects of S-Lotos

We briefly recall the definition of configurations [3], and then an operational semantics is defined for S-LOTOS to derive the possible transitions linking the configurations.

3.1. Configurations

Let $M$ be the set of event names, ranged over $x, y$ … Further, the notation $2^{M}_{f_n}$ represents the set of finite subsets of a set $X$.

**Definition 1.** Configurations.

The set $C$ of configurations is given by:

- $\forall E \in B, \forall M \in 2^{M}_{f_n} M[E] \in C$
- $\forall P \in PN, \forall M \in 2^{M}_{f_n} : M[P] \in C$
- if $e \in C$ then : hide $L$ in $e \in C$
- if $e, F \in C$ then : $e \cup F \in C$, $e \llbracket L \rrbracket F \in C$
- if $e \in C$ and $\{a_1, \ldots, a_n\}$, $\{b_1, \ldots, b_m\} \in 2^{M}_{f_n}$, then: $e[b_1/a_1, \ldots, b_m/a_n] \in C$

3.2. Derivation Rules

Let $S$ be the set of states; transitions between states are also projected between configurations of these states. The transition relation between configurations is denoted $\rightarrow \subseteq S \times S$, where the set of atoms of support (ActxFD) is $Atm=2^M_{f_n}(ActxFD) \times M$. For any subset of event names $M \in 2^{M}_{f_n}$, $(a, f) \in (ActxFD)$ and $x \in M$, the atom $(M, (a, f), x)$ will be denoted $M(a, f)$. The choice of an event name can be realized deterministically by using any function $get: 2^M_{f_n} \setminus \{\emptyset\} \rightarrow M$ satisfying $get(M) \in M$, for all $M \in 2^M_{f_n} \setminus \{\emptyset\}$. The operational semantics of S-LOTOS is summarized in Table 1. Function $\psi: S \rightarrow 2^M_{f_n}$ is the function that associates every state with the finite set of its maximal events, and the predicate $Wait: 2^M_{f_n} \rightarrow \{true, false\}$ characterizes the termination of maximal actions: $Wait(M)=true$ if there is at least one running action referenced in $M$.

<table>
<thead>
<tr>
<th>Table 1: Operational Semantics of S-LOTOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi$ (stop) $\rightarrow$ $\psi$ (exit) $\quad x = get(M)$</td>
</tr>
<tr>
<td>$\psi$ (stop) $\rightarrow$ $\psi$ (exit) $\quad x = get(M)$</td>
</tr>
<tr>
<td>$e \downarrow \rightarrow e'$ $\quad a \notin L$</td>
</tr>
<tr>
<td>$\psi[L] \llbracket e \rrbracket \rightarrow \psi[L] \llbracket e' \rrbracket$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow \psi[L] \llbracket e' \rrbracket \llbracket F \llbracket e \rrbracket$ $\quad z = get(M - (\psi(F) \cup (\psi(F)))$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
<tr>
<td>$e \llbracket L \rrbracket F \llbracket e \rrbracket \rightarrow e'$ $\quad a \in L$</td>
</tr>
</tbody>
</table>

4. Models based on Maximality Semantics vs. ST-Semantics

We briefly present the principles of some well-known ST-semantic models, and then show its equivalence with our performance model working under the maximality semantics.

4.1. ST-Semantics
The ST-semantics is used to describe the behavior of a concurrent system in terms of running actions, each one for a specific duration. Contrary to standard interleaving semantics, actions are not considered as atomic. The running of an action is split into two distinguished events, representing the start of the running and the corresponding end. Since the same actions can be launched several times concurrently (auto-concurrency), supplementary information is included in the semantic models to avoid confusion: maintain the correspondence between each start event of an action and its termination. In that purpose, two naming strategies were proposed for events.

The first one is based on static names, where names are defined at compiling time, according to their syntactical position (left or right) in the description of the initial parallel process specifying the whole system. The second strategy is based on dynamic names, that is: under a fixed rule, dynamically assign a different name to each new action that becomes active, assuming the names of the currently active actions are known.

Whatever the technique used to name the events, the ST-semantic model appears to be an LTS wherein states are labeled by pairs \((E, X)\) mentioning for each state, the behavior expression \(E\) and the set \(X\) of activities under execution. Moreover, transitions between these states are of two types: start transitions labeled by \(a^+_x\) where \(x\) is the name associated to the running action \(a\), and termination transitions labeled by \(a^-_x\) where the name \(x\) determines exactly which action \(a\) is terminating.

### 4.2. Performance Equivalence

Regarding to any system specification \(P\), we now prove that there is a stuttering equivalence between its maximality semantic model \(M_P\) and ST-semantic model \(ST_P\), under the following hypothesis.

**Hypothesis 1:** The durations of transitions in semantic models are assumed to be null.

The considered stuttering equivalence consists in aggregating the states of \(ST_P\) up to preserve its temporal properties. Such aggregations emerge from the observation of the start events, without regarding the end ones. From a performance view point, when focusing on \(ST_P\), it appears that two successive states linked by a termination event are stutter, thus can be aggregated. Roughly speaking, when considering any transition \((E, X)\xrightarrow{t}(E', X')\), s.t. \(t\) is a terminated event, we deduce that both source and target states preserve the same temporal properties because \(E=E'\) and \(duration(t)=0\) (according to Hypothesis 1). This adjacency relation can be extended to transitions sequence, hence the application of the stuttering equivalence on words, as follows: two words are stuttering equivalent if both can be partitioned into \(n\) blocks (having possibly different lengths), so that the states in the \(k\)th block of one word are labeled the same as the states in the \(k\)th block of the other word.

As an example, Figure 3-a presents the ST-semantic model corresponding to a system that concurrently runs two actions \(a\) and \(b\), having \(f\) and \(g\) as respective duration distribution functions. The dashed lines bring out different groups of stutter states. A state aggregation w.r.t. these equivalences results in the graph of Figure 3-b, which can be obtained directly by considering the maximality semantics.

Considering the model \(ST_P\) of a concurrent system specification \(P\), let \(Start\) and \(End\) be the two disjoint subsets of start and end events with regards to the behavior of \(P\). Let \(L_P\) be the language generated by \(ST_P\) and \(w=s_0s_1...s_n\) be any word of \(L_P\) corresponding to the transition sequence \(s_0\rightarrow s_1...\rightarrow s_n\) enabled in \(P\). The notation \(w(i)\) and \(w(last)\) respectively represent the \(i\)th and the last state of \(w\). Further, we consider the observational language \(L^{start}_P\) of \(L_P\) built from the observation of the start transitions as follows:

**Definition 2:** Observational language for ST-semantic model

Let \(w = s_0s_1...s_n\) be a word in \(L_P\), \(\sigma=\sigma_0\sigma_1...\sigma_m\in L^{start}_P\) is the observation of \(w\) w.r.t. \(Start\) iff. each \(\sigma_i\) corresponds to a sub-word of \(w\), such that:

\[
\sigma_0(1) = s_0 \\
\sigma_i(j) \rightarrow \sigma_{i+1}(j + 1) \quad 1 \leq j, 0 \leq i \leq n, t \in End \\
\sigma_i(last) \rightarrow \sigma_{i+1}(1) \quad 0 \leq i < n, d \in Start
\]

**Lemma:** Given a system specification \(P\), let \(M_P\) and \(ST_P\) be the semantic models of \(P\) respectively based on the maximality and ST-semantics, then \(M_P\) and \(ST_P\) are stuttering equivalent with respect to the performance properties considered for \(P\).
We proceed the proof by induction on the length of transitions sequences of $M_p$ and $ST_p$. A word $w = s_0s_1\ldots s_n$ is a sequence of (labeled) states such that there is a transitions sequence $s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_n$ in the transition system. For sake of simplicity, we assimilate the performance properties that hold on some state to considering the next transitions in both models.

![Diagram](attachment:image.png)

**Fig. 3:** Example of ST-semantic model and the equivalent maximality-based semantic model.

**Proof:**

First, the property of stuttering equivalence straight fully holds, whether paths are reduced to their initial states, i.e. $\langle [P] \rangle$ in $M_p$ and $\langle (P, \{ \}) \rangle$ in $ST_p$.

Consider now that the following two words, $w_1 = s_0s_1\ldots s_n$ in $ST_p$ and $w_2 = s'_0s'_1\ldots s'_m$ in $M_p$, are stuttering equivalent with respect to the performance properties of $P$. From the fact that $w_1 (last)$ and $w_2 (last)$ are labeled by the same behavior expression, say $E$, we prove that the stuttering equivalence still holds after considering the next transitions in both models.

1) If the behavior expression $E$ allows the execution of an action $a$ with duration distributed by function $f$ yielding to a behavior expression $E'$: $E \xrightarrow{a} E'$, then we obtain in $ST_p$ the transition: $s_n \xrightarrow{a^+} s_{n+1}$, and in $M_p$ the transition: $s'_m \xrightarrow{s'\delta_{k}} s'_{m+1}$

Both new states $s_{n+1}$ and $s'_{m+1}$ are labeled by the same behavior expression $E'$. Therefore, both new paths: $w_1 = s_0s_1\ldots s_{n+1}$ and $w_2 = s'_0s'_1\ldots s'_{m+1}$ are stuttering equivalent, because we construct, in each path, a new block of one state labeled by the same behavior expression $E'$, and executed action have same duration.

2) If some running action $a$ terminates, this simply has an effect to retiring the corresponding event from the set of running events. We have the following transition in $ST_p$: $s_n \xrightarrow{a^-} s_{n+1}$, where state $s_{n+1}$ is labeled by the behavior expression $E$. However, this transition is not represented explicitly in $M_p$. Hence, both paths $w_1$ and $w_2$ still stuttering equivalent, because we have not change their blocks.

Generally, If the behavior expression $E$ have to terminate some actions $b_i$, $1 \leq i \leq k$, before the start event of an action $a$ with duration distributed by function $f$ yielding to a behavior expression $E'$,

$$E \xrightarrow{b_1^-} E \xrightarrow{b_2^-} \ldots \xrightarrow{b_k^-} E \xrightarrow{a^+} E'$$

then we have in the $ST_p$ the transitions:

$$s_n \xrightarrow{b_1^-} s_{n+1} \xrightarrow{b_2^-} \ldots \xrightarrow{b_k^-} s_{nk} \xrightarrow{a^+} s_{nk+1}$$

and in the $M_p$ the transitions:

$$s'_m \xrightarrow{\{b_1^-, \ldots, b_k^-\} \delta_{k}} s'_{m+1}$$

where the new states $s_{nk+1}$, $1 \leq i \leq k$, are labeled by the behavior expression $E$ (because, according to the ST-semantics, termination events does not alter the behavior expression), and the states $s_{nk+1}$ and $s'_{m+1}$ are
both labeled by the behavior expression $E'$. Therefore, the new paths $w_1' = s_0 s_1 \ldots s_{m+1}$ and $w_2' = s'_0 s'_1 \ldots s'_{m+1}$ stay stuttering equivalent because we increase the last block of $w_1$ by states $s_{m+1}$, $1 \leq i \leq k$, which are labeled by the same behavior expression than states in this block, and we construct a new block in each path of one state labeled by $E'$, and the duration added to total time of execution is the same in both models because termination event have null duration.

Consequently, models $M_P$ and $ST_P$ for a specification expression $P$ are stuttering equivalent, where blocks in a path of $M_P$ are constructed each one of one state, and the corresponding blocks in $ST_P$ are constructed according to the observational language on start transitions. The following transitions (1) in $M_P$ and (2) in $ST_P$ indicate the sequence of states of blocks constructed in both models. Termination transitions which are explicitly represented in $ST_P$ (2) are abstracted in $M_P$. These transitions are in fact implicitly formalized in $M_P$ through the notions of maximal events attached to states (i.e. a running action can terminate), nevertheless the transition can give the information about the termination of some running action if this last one belongs to the causality set $C$ of the transition (1).

\[
\mu [E] \overset{s \in C}{\longrightarrow} M [E] \\
\{E, M\} \overset{\delta}{\longrightarrow} \{E, M - \{b_1\}\} \overset{\delta}{\longrightarrow} \ldots \overset{\delta}{\longrightarrow} \{E, M - C\} \overset{\delta}{\longrightarrow} \{E', M'\} \quad C = \{b_1, b_2, \ldots\}, M' = M - C + \{x\}
\]

In order to obtain its equivalent $M_P$, for some $ST_P$, we only observe the start transitions, thus aggregating states which are connected by termination transitions. In fact, the equivalence between points (1) and (2) defines the basis of the formal transformation from one model to the other.

### 4.3. Conclusion and Perspectives

The stochastic algebra named S-LOTOS extends existing LOTOS specification language by introducing general distribution functions to govern the action durations. By adopting the maximality-based semantics, we gave to S-LOTOS the ability to conform with the true concurrency paradigm of concurrent systems.

We showed that there is a stuttering equivalence between our Maximality-based Stochastic Labeled Transition System (MLSTS) and the existing ST-semantics based models. This allows one to re-use existing techniques and tools for performance evaluations with general distribution functions.

Our next perspective is to derive the performance properties from the MLSTS representations directly. Actually, the fact that these representations are less prone to the state space explosion problem due to the fact that the terminations of actions is no more explicit, made them attractive to deal with stochastic model checking problem.

### 5. References


