UNE IMP EHE CWI ★QMIPS★ INR UER ZAR UTO

Quantitative Modeling In Parallel Systems

Probabilistic Branching in Markovian Process Algebras

Michael Rettelbach
University of Erlangen–Nürnberg, IMMD VII

D W1.T2.2b-5.v1

22 September 1995

Distribution Level: Public

Approved by: U. Herzog

Abstract We introduce immediate transitions as an extension of Stochastic Process Algebras (SPA). We distinguish two different classes of immediate transitions: probabilistic branching and management activities. We discuss both approaches and develop a theory for the probabilistic branching case. Although we use TIPP as a sample language within this paper, the theory can easily be adapted to other Stochastic Process Algebras and can therefore be seen as a general result for probabilistic branching in SPA.

The QMIPS Project: The QMIPS project is a collaborative research project supported by the CEC as ESPRIT-BRA project no 7269. It is being carried out by the following organisations: CWI (Amsterdam), EHEI (University of Paris V), Imperial College (London), INRIA (Sophia-Antipolis), University of Erlangen, University of Newcastle, University of Torino and University of Zaragoza.

1 Introduction and Motivation

Recent research on the algebraic theory for Stochastic Process Algebras (SPA) has concentrated on Markovian durations associated with each activity. We call this class of SPA Markovian Process Algebras (MPA). A complete theory for this class is available, including Markovian bisimulation as a notion of equivalence, which has been proven to be a congruence with respect to the calculus. An axiomatisation of this notion of equivalence was introduced which has been proven to be sound and complete [7].

1.1 Immediate activities

Incorporating immediate activities as a part of the language is a step towards *Generalised* MPA (in the sense of *Generalised* Stochastic Petri Nets). Immediate transitions are practical means for the modelling process. In general they are useful for

- probabilistic branching and
- management actions (e.g. the test if a resource is available).

This is particularly true if we choose a level of abstraction in which we do not want to explicitly model very short execution times rather than to assume them to be zero.

Consider for example a node in a communication network. Experience tells us that about 10% of incoming messages have to be delivered locally, whereas the rest are forwarded somewhere else. Local delivery happens with rate μ_l, forwarding with rate μ_f. The decision itself is assumed to consume almost no time. Currently, we can specify the behaviour of this node in two ways. One is to introduce a state which is left by two transitions (*branch*) with a very high rate ρ. The rate is split into the two transitions according to our branching probabilities.

$$\begin{aligned} Node \;&:=\; (get\_message, 1). \\ &\quad\;\; \big((branch, 0.9\rho).(forward, \mu_f).Node \\ &\quad\quad + \; (branch, 0.1\rho).(local, \mu_l).Node\big) \end{aligned}$$

Unfortunately, this description will be translated to a Markov chain in which the rates of the transitions differ by some orders of magnitude. Applying numerical solutions to such a Markov chain can easily lead to so-called *stiffness* problems. A second way to model our probabilistic branch is to make use of TIPP's synchronisation discipline: synchronisation with the environment via the action *get_message* will generate a transition with the product of the two synchronising transitions' rates. Thus we can come up with

$$\begin{aligned} Node' \;&:=\; (get\_message, 0.9).(forward, \mu_f).Node' \\ &\quad\; + (get\_message, 0.1).(local, \mu_l).Node' \end{aligned}$$

as a description for the communication node. We avoid transitions with very high rates but follow a rather tricky way of modelling, which – considering larger systems with lots of interactions between subcomponents – demands a deep insight into the whole system's behaviour. This contradicts the support of modular design.

Introducing an explicit construct for probabilistic branching will bring us into the position to make the *branch* action immediate. The corresponding immediate transition in the semantic model can be eliminated. The algorithm for this kind of elimination of probabilistic branching is widely used. The main idea is to collapse a Markovian transition with all immediately following probabilistic branchings (cf. Fig. 1). The transition system containing

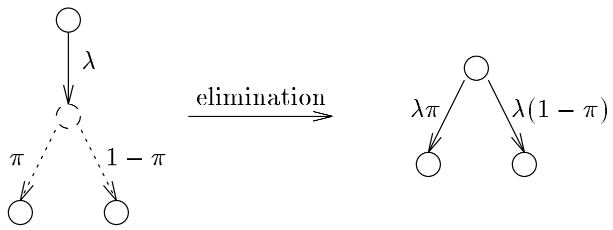

Figure 1: Example for the elimination of immediate transitions

Markovian transitions and probabilistic branching can be translated into a matrix, where the states are collected into tangible (states from which no probabilistic transition starts) and vanishing (all other states) states. The corresponding matrix, containing transition rates and probabilities, is of the following shape

$$\underline{Q} \;=\; \left(\begin{array}{c|c} \underline{M}_{tt} & \underline{M}_{tv} \\ \hline \underline{P}_{vt} & \underline{P}_{vv} \end{array}\right) \;.$$

where $\underline{M}_{tt}$ and $\underline{M}_{tv}$ contain the Markovian transitions (from tangible) to tangible and vanishing states respectively, and $\underline{P}_{vt}$ and $\underline{P}_{vv}$ contain the probabilistic transitions (from vanishing) to tangible and vanishing states respectively. The elimination algorithm defines a characteristic matrix

$$\underline{B} \;=\; \{b_{ij}\} \;=\; (\underline{I} - \underline{P}_{vv})^{-1}\,\underline{P}_{vt}$$

which contains the probabilities $b_{i,j}$, to reach the tangible state j from a vanishing state i via probabilistic transitions only. The transition rate matrix $\underline{Q}'$ not containing any probabilistic transitions is given by:

$$\underline{Q}' \;=\; \underline{M}_{tt} \;+\; \underline{M}_{tv}\underline{B}$$

1.2 Stochastic Process Algebras

In the world of Stochastic Process Algebras we want to adapt this elimination schema by defining a notion of equivalence (cf. Fig. 2). Functional analysis might be executed on the full state space, also containing immediate transitions; for the application of numerical solution methods, however, we choose an equivalent transition system not containing vanishing states. Therefore each of the equivalence classes, induced by the notion of equivalence on the set of transition systems, should contain at least one transition system without immediate transitions/vanishing states (as far as it is possible).

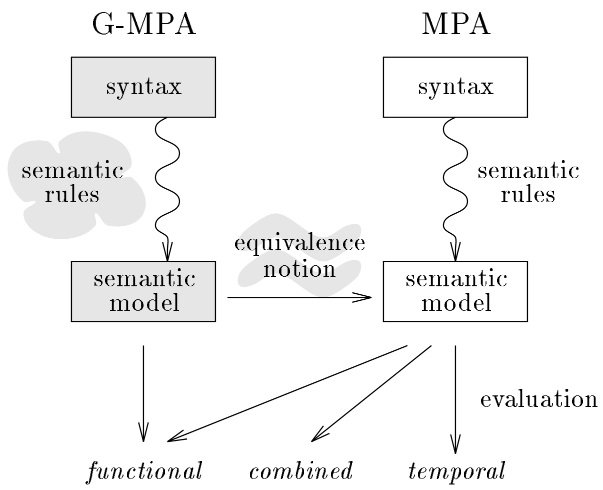

Figure 2: Introducing and eliminating immediate transitions

Additionally we want this equivalence to be a congruence. Thus elimination of immediate transitions and vanishing states can be done within subsystems *before* they are composed[1] and state space explosion can be avoided. Furthermore a congruence can be characterised on the syntactic level in terms of an axiomatisation.

1.3 Related work

[2] proposes to introduce immediate transitions into MPA. In connection with that he defines a probabilistic choice operator which is translated to a branching between two immediate, invisible actions ($\underline{\tau}$ in our notation). We will learn about the difference from our approach in Section 4. No algebraic theory is investigated for his approach.

[1] introduce immediate transitions with weights into MPA. Thus nondeterministic branching becomes probabilistic by converting the weights of the involved immedate transitions into probabilities. They propose to investigate the compositionality of their approach.

There is a large number of approaches to including branching probabilities into classical (non-timed) process algebras as well as those with deterministic transition times. An overview on these approaches can be found for example in [5]. The way [12] introduce probabilities into a process calculus is the approach that is closest to the one presented in this paper.

1.4 Overview

In this paper we will investigate possibilities for introducing branching probabilities into MPA in Section 2. Section 3 defines a class of transition systems as well as a notion of equivalence for these transition systems. We extend TIPP to PM-TIPP by introducing probabilistic

[1] This property is one of the main advantages we gain by constructivity.

choice in Section 4 and give a set of semantic rules for this language. In Section 5 we show that the equivalence property is preserved by the operators of PM-TIPP which means that the notion of equivalence is a congruence. Section 6 concludes this paper and gives an overview of further work.

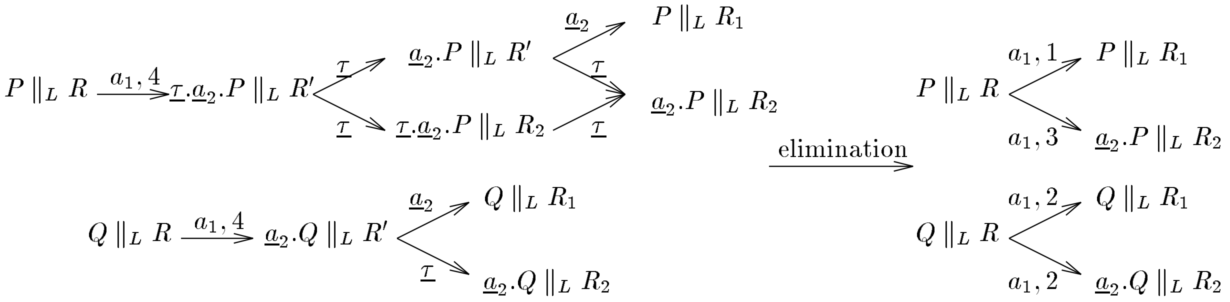

Figure 3: Different behaviour of the two processes P and Q under composition ($R' := \underline{a_2}.R_1 + \underline{\tau}.R_2$)

2 Probabilities in MPA

2.1 First attempt

We want to introduce probabilistic branching by assigning weights to immediate transitions. Thus nondeterministic branching between immediate transitions [8] becomes probabilistic. Like in that paper we use pairs $(a, \lambda), (b, \mu), \ldots$ composed of an action name and a rate for Markovian actions, with τ being the name of invisible Markovian actions, and $\underline{a}, \underline{b}, \ldots$ to denote immediate actions, where $\underline{\tau}$ is the invisible immediate action.

$P := (a_1, \lambda).\underline{\tau}.\underline{a_2}.P$ and $Q := (a_1, \lambda).\underline{a_2}.Q$ are two processes which have to be considered equivalent in our sense, because of the following consideration. No matter which weight we associate to the immediate transitions $\underline{\tau}$ or $\underline{a_2}$ we will always observe that each of them is executed with probability 1, whenever it is enabled. Applying the elimination schema will collapse both processes to $\hat{P} := (a_1, \lambda).\hat{P}$. This fact forces us to regard systems P and Q as equivalent, because both should be in the same equivalence class together with the model without vanishing states $\hat{P}$. Further composition, however, might distinguish the systems. For composition we use the process $R := (a_1, 1).(\underline{a_2}.R_1 + \underline{\tau}.R_2)$, and synchronise via the actions in $L := \{a_1, \underline{a_2}\}$. The composed processes do not show the same behaviour concerning branching probabilities, assuming that all immediate transitions have the same weight (cf. Fig. 3). Because of that, we would not be able to reduce P in a larger environment $P \parallel_L R$ to Q, in general, because $Q \parallel_L R$ is not necessarily equivalent to $P \parallel_L R$. In other words: the congruence property is violated.

2.2 Solution

The question is: how much of our goal can be achieved under the condition that the congruence property is preserved? Our investigations showed that a promising way is to strictly distinguish two kinds of immediate activities: *management*[2] activities and *probabilistic* branching (cf. Fig. 4). Management activities are allowed to synchronise with each other, but do

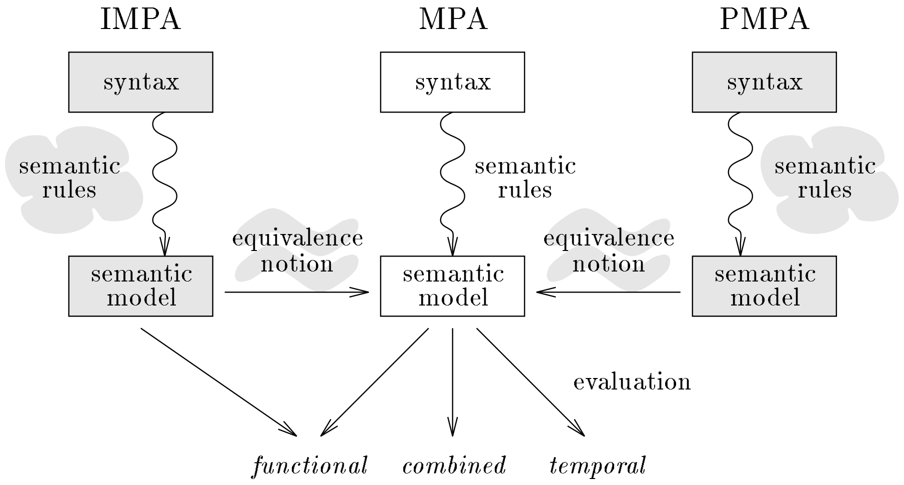

Figure 4: Global idea for the elimination of immediate transitions

not include probabilistic branching. Thus a choice between two management activities is nondeterministic and can only be resolved by an appropriate composition with an environment, but definitely does not denote a probabilistic branching as implicitly in our example. Probabilistic branching transitions on the other hand side must not synchronise individually.

A very similar approach can be found in [12]. They, however, introduce probabilistic branching into a standard functional process algebra not incorporating Markovian transitions.

2.3 Available results

The results concerning the management transitions are presented in [8]. The aspect of probabilistic branching is elaborated within this paper.

3 Probabilistic and Markovian Transition Systems and Equivalence

In this section we define a new type of transition system involving two transition relations. The one covering Markovian transitions is labelled by actions taken from the set Act_M,

[2]In the latter we will use the term 'immediate' for management activities only.

rates from set $\mathcal{R}$ and an additional word taken from the set *Lab*, which will be explained in Section 4. The other (probabilistic transitions) is labelled by probabilities (Π) and also with words from *Lab*.

Definition 3.1 *A* probabilistic and Markovian transition system *(PMTS) is a 4-tuple*

$$(\mathcal{S}, s_0, \longrightarrow, \cdots\!\!\blacktriangleright) \quad ,$$

where $\mathcal{S}$ is a set of states, s_0 is the initial state and

$$\longrightarrow\ \subseteq \mathcal{L}_{PM} \times \mathsf{Act}_M \times \mathcal{R} \times Lab \times \mathcal{L}_{PM}$$

and

$$\cdots\!\!\blacktriangleright\ \subseteq \mathcal{L}_{PM} \times \Pi \times Lab \times \mathcal{L}_{PM}$$

are transition relations denoting Markovian and probabilistic transitions, respectively. Additionally the following holds:

1. *For* every state *the sum of probabilities of outgoing probabilistic transitions is 1.*
2. *Every probabilistic transition ends in a state t from which exactly one probabilistic transition is possible, namely the one labelled with the probability 1 back to state t.*
3. *For each of the two relations, there are no two transitions starting at the same state, which are labelled with the same additional word (from the set Lab).*

Property 1 ensures that probabilistic transitions show the probabilistic behaviour of a state including the degenerate case when there is no probabilistic branching, realised by the 1-loops. Property 2 guarantees that there cannot be two subsequent vanishing states within the relation $\cdots\!\!\blacktriangleright$ or more formally $\cdots\!\!\blacktriangleright\ \subseteq \mathcal{S} \times \Pi \times Lab \times \mathcal{T}$, with $\mathcal{T}$ being the set of tangible states. The meaning of property 3 is explained in Section 4.

3.1 Notion of equivalence

Summarizing our claims, we are searching for a notion of equivalence with the following properties:

1. It shall be sound with respect to the standard elimination algorithm and is coarse enough (contains enough pairs) to assign *at least one* transition system without probabilistic transitions to each one containing them.

2. It shall be sound with respect to our central notion of equivalence for purely exponential processes (Markovian Bisimulation), and therefore also preserves lumpability and does not affect functional behaviour.

3. It shall be a congruence with respect to the operators of PM-TIPP, the language we will define in Section 4.

We will come up with an appropriate definition now and show later that it coincides with these properties. First of all we need to define some auxiliary functions.

Definition 3.2 *$\gamma^M : \mathcal{S} \times \mathsf{Act}_M \times 2^{\mathcal{S}} \longrightarrow \mathcal{R}$ computes the sum of rates of all Markovian transitions that start from a state s, are labelled with action a and lead to a state in C, for any given state s and action a:*[3]

$$\gamma^M(s,a,C) := \sum \{\!| \lambda \mid \exists s' \in C, \exists w \in Lab\colon\ s \xrightarrow{a,\lambda,w} s' |\!\}$$

$\gamma^P : \mathcal{S} \times 2^{\mathcal{S}} \longrightarrow \mathcal{R}$ computes the sum of probabilities for all probabilistic transitions that start from a given state s and end in a state that is an element of the set C:

$$\gamma^P(s,C) := \sum \{\!| \pi \mid \exists s' \in C,\ \exists w \in Lab:\ s \overset{[\pi],w}{\cdots\cdots\blacktriangleright} s' |\!\}$$

$\gamma^{PM} : \mathcal{S} \times \mathsf{Act}_M \times 2^{\mathcal{S}} \longrightarrow \mathcal{R}$ computes the sum over all rates of Markovian transitions that start from a given state s and are labelled by a given action a, each weighted by the probability of a directly following probabilistic transition. Additionally the state that is finally reached has to be an element of C:

$$\gamma^{PM}(s,a,C) \ := \ \sum_{t \in \mathcal{S}} \gamma^M(s,a,\{t\}) \cdot \gamma^P(t,C)$$

Using this definition we can formalise our notion of equivalence. We adopt the definition schema that was already used for the definition of the Markovian bisimulation [6, 7] initially introduced by [9] for a probabilistic calculus.

Definition 3.3 *An equivalence relation $\mathcal{B}$ is called PM-bisimulation, if the following holds:*

$$\begin{array}{l} \forall s,s' \in \mathcal{S},\ \forall a \in \mathsf{Act}_M,\ \forall C \in \mathcal{S}/_{\mathcal{B}} : \\ (s,s') \in \mathcal{B} \iff \\ \quad (i)\quad \gamma^P(s,C) = \gamma^P(s',C) \\ \quad (ii)\quad s \in \mathcal{T} \wedge s' \in \mathcal{T} \implies \\ \qquad\qquad \gamma^{PM}(s,a,C) = \gamma^{PM}(s',a,C) \end{array}$$

where $\mathcal{T} := \{s \mid \exists w \in Lab:\ s \overset{[1],w}{\cdots\cdots\blacktriangleright} s\}$ is the set of tangible states.

Two states s,s' are called PM-bisimulation equivalent ($s \overset{PM}{\sim} s'$), if there exists a PM-bisimulation $\mathcal{B}$, such that $(s,s') \in \mathcal{B}$, or equivalently $\overset{PM}{\sim} = \bigcup \mathcal{B}$.

Two transition systems $(\mathcal{S}_1, s_1, \longrightarrow_1, \cdots\blacktriangleright_1)$ and $(\mathcal{S}_2, s_2, \longrightarrow_2, \cdots\blacktriangleright_2)$ are PM-bisimulation equivalent, if $s_1 \overset{PM}{\sim} s_2$.

Proposition 3.1 *$\overset{PM}{\sim}$ is the largest PM-bisimulation.*

Proof: A detailed proof of this proposition can be found in [11].

The sketch of the proof is as follows. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two PM-bisimulations and

$$\mathcal{B} = (\mathcal{B}_1 \cup \mathcal{B}_2)^* \quad .$$

[3] $\{\!|$ and $|\!\}$ are used as multiset brackets.

It is easy to show that $\mathcal{B}$ is an equivalence relation (reflexivity and transitivity are given by the definition, symmetry by the fact that both $\mathcal{B}_1$ and $\mathcal{B}_2$ are symmetric).

We know that $\mathcal{B}_1 \subseteq \mathcal{B}$. Thus for each $C \in \mathcal{S}/_{\mathcal{B}}$ we can find a number n of equivalence classes $C_i^1 \in \mathcal{S}/_{\mathcal{B}_1}$ $(1 \leq i \leq n)$ of $\mathcal{B}_1$ such that

$$C = \bigcup_{i=1}^{n} C_i^1 \quad .$$

We can show that for arbitrary $a \in \mathsf{Act}_M$ and $s \in \mathcal{S}$ the following holds:

$$\begin{array}{ll} i) & \gamma^P(s, C) = \sum_{i=1}^{n} \gamma^P(s, C_i^1) \qquad\qquad (*) \\ ii) & \gamma^{PM}(s, a, C) = \sum_{i=1}^{n} \gamma^{PM}(s, a, C_i^1) \end{array}$$

The same holds for $\mathcal{B}_2$ instead of $\mathcal{B}_1$.

From this result we can derive that $\mathcal{B}$ is a PM-bisimulation, because both $\mathcal{B}_1$ and $\mathcal{B}_2$ are PM-bisimulations.

By induction we can show that each element $(s, s') \in (\mathcal{B}_1 \cup \mathcal{B}_2)^r$, for a non negative natural number r, preserves the conditions (i) and (ii) specified in definition 3.3. $r = 0$ means that $s \equiv s'$ which satisfies the conditions. Now let us assume that arbitrary pairs $(s, s'') \in (\mathcal{B}_1 \cup \mathcal{B}_2)^r$ fulfil the conditions (i) and (ii). For $(s, s') \in (\mathcal{B}_1 \cup \mathcal{B}_2)^{r+1}$ we can derive that at least one of the two following propositions holds:

$$\begin{array}{ll} 1 & \exists s'' \in \mathcal{S}: \ (s, s'') \in (\mathcal{B}_1 \cup \mathcal{B}_2)^r \ \wedge \ (s'', s') \in \mathcal{B}_1 \\ 2 & \exists s'' \in \mathcal{S}: \ (s, s'') \in (\mathcal{B}_1 \cup \mathcal{B}_2)^r \ \wedge \ (s'', s') \in \mathcal{B}_2 \end{array}$$

Without loss of generality we restrict ourselves to show proposition 1. Condition (i) of definition 3.3 is preserved because we know that

$$\gamma^P(s'', C^{\mathcal{B}_1}) = \gamma^P(s', C^{\mathcal{B}_1})$$

holds for each equivalence class $C^{\mathcal{B}_1} \in \mathcal{S}/_{\mathcal{B}}$, because $(s'', s') \in \mathcal{B}_1$. Hence we can use $(*)$ to show that

$$\gamma^P(s'', C^{\mathcal{B}}) = \gamma^P(s', C^{\mathcal{B}})$$

for $C^{\mathcal{B}} \in \mathcal{S}/_{\mathcal{B}}$, because $\mathcal{B}_1 \subseteq \mathcal{B}$. By induction we know

$$\gamma^P(s, C^{\mathcal{B}}) = \gamma^P(s'', C^{\mathcal{B}})$$

from which we can derive

$$\gamma^P(s, C^{\mathcal{B}}) = \gamma^P(s', C^{\mathcal{B}})$$

directly. Analogously we can show condition (ii), where additionally $t \xrightarrow{[1],w} t$ is claimed for each $t \in \{s, s', s''\}$, because otherwise nothing needs to be shown.

Thus we know that $\mathcal{B}$ is a PM-bisimulation.

From this result we can derive directly that $\overset{PM}{\sim}$ is a PM-bisimulation, which includes that it is the largest one, because of the way it is defined. $\square$

Let us now turn towards the properties we demanded from our notion of equivalence.

3.2 $\overset{PM}{\sim}$ and the standard elimination algorithm for immediate transitions

For PMTS we have to extend the representation as matrices with respect to the actions used. Therefore we need one matrix ${}^a\underline{M}_{tt}$ and ${}^a\underline{M}_{tv}$ for each action $a \in \mathsf{Act}_M$. An entry ${}^a m_{ij}^{tt}$ within that matrix for example represents the sum of rates of transitions labelled with a leading from state i to state j. Consequently there are also individual matrices

$$ {}^a\underline{Q} \;=\; \left(\begin{array}{c|c} {}^a\underline{M}_{tt} & {}^a\underline{M}_{tv} \\ \hline \underline{P}_{vt} & \underline{P}_{vv} \end{array}\right) \qquad {}^a\underline{Q}' \;=\; {}^a\underline{M}_{tt} \;+\; {}^a\underline{M}_{tv}\underline{B} $$

Because of property 2 of definition 3.1 it holds that $\underline{P}_{vv} = \underline{0}$. This follows from the fact that no probabilistic transition ends in a vanishing state. Therefore the computation of $\underline{B}$ is simplified to

$$ \underline{B} \;=\; (\underline{I} - \underline{0})^{-1}\,\underline{P}_{vt} \;=\; \underline{P}_{vt} $$

Hence it follows that

$$ {}^a\underline{Q}' \;=\; \{{}^a q_{ij}\} \;=\; {}^a\underline{M}_{tt} \;+\; {}^a\underline{M}_{tv}\underline{P}_{vt} \quad . $$

Considering each state $s_x \in \mathcal{T}$ being represented on the xth position in the matrices ${}^a\underline{Q}'$ and $s_i, s_j \in \mathcal{T}$ the following equation holds for arbitrary actions a:

$$ \begin{aligned} &\gamma^{PM}(s_i, a, \{s_j\}) \\ &\quad= \sum_{s_k \in \mathcal{S}} \gamma^{M}(s_i, a, \{s_k\}) \cdot \gamma^{P}(s_k, \{s_j\}) \\ &\quad= \gamma^{M}(s_i, a, \{s_j\}) + \\ &\qquad \sum_{s_k \in \mathcal{V}} \gamma^{M}(s_i, a, \{s_k\}) \cdot \gamma^{P}(s_k, \{s_j\}) \\ &\quad= {}^a m_{ij}^{tt} \;+\; \sum_{s_k \in \mathcal{V}} {}^a m_{i(k-t)}^{tv} p_{(k-t)j}^{vt} \\ &\quad= {}^a q'_{ij} \end{aligned} $$

where $t \in I\!N$ is the number of tangible states and $(k-t)$ is the horizontal/vertical position of a vanishing state s_k within the matrices ${}^a\underline{M}_{tv}$ and $\underline{P}_{vt}$, respectively, and $\mathcal{V} := \mathcal{S} \setminus \mathcal{T}$ is the set of vanishing states. This guarantees that there is an equivalent transition system with a transition $s'_i \xrightarrow{a,\gamma^{PM}(s_i,a,\{s_j\}),w} s'_j$ in which s_k is eliminated according to the usual algorithm.

The generator matrix for the whole system can be built by summing up all matrices ${}^a Q'$ and replacing the main diagonal elements by the negative row sum.

The following proposition ensures, that we can eliminate all probabilistic branchings but those starting at the initial state, using the PM-bisimulation[4].

Proposition 3.2 *Each equivalence class $C \in \mathcal{S}/_{\overset{PM}{\sim}}$ containing at least one tangible state, also contains a state that has* no *transition to a vanishing state.*

Thus by recursion we can build up a transition system, equivalent to our initial one, but not containing immediate transitions. This is done by choosing the special state mentioned

[4] The formal proof can be found in [11].

above from each equivalence class as a representative. The only probabilistic branching that cannot be eliminated this way, is one starting at the initial state of the transition system. Such branching in the beginning denotes the initial weights that have to be given to the states of the underlying Markov chain when doing temporal analysis.

3.3 $\overset{PM}{\sim}$ and the Markovian bisimulation

The following proposition determines the relation between the Markovian and the PM-bisimulation:

Proposition 3.3 *Let $T_1 = (\mathcal{S}_1, s_1, \longrightarrow_1)$ and $T_2 = (\mathcal{S}_2, s_2, \longrightarrow_2)$ be two transition systems and $\dashrightarrow_{[1]}$ a special probabilistic transition relation satisfying: $\forall s \in \mathcal{S}_1 \cup \mathcal{S}_2,\ \exists w \in Lab$: $(s, [1], w, s) \in \dashrightarrow_{[1]}$. Then the following proposition holds*

$$T_1 \sim_M T_2 \Longleftrightarrow T_1' \overset{PM}{\sim} T_2'$$

where $T_1' = (\mathcal{S}_1, s_1, \longrightarrow_1, \dashrightarrow_{[1]})$ and $T_2' = (\mathcal{S}_2, s_2, \longrightarrow_2, \dashrightarrow_{[1]})$ are PMTS.

Proof: The proposition follows from definition 3.2 and definition 3.3. Condition (i) is satisfied for all states of both transitions systems, because for all states s of both T_1' and T_2' $\gamma^P(s, \{s\}) = 1$ holds; the precondition of (ii) is satisfied because of the special structure of the relation $\dashrightarrow_{[1]}$. Therefore both conditions are reduced to $\gamma^{PM}(s, a, C) = \gamma^{PM}(s', a, C)$. Furthermore the following holds:

$$\begin{aligned}
&\gamma^{PM}(s, a, C)\\
&\quad = \sum_{t \in \mathcal{S}} \gamma^M(s, a, \{t\}) \cdot \gamma^P(t, a, C)\\
&\quad = \sum_{t \in \mathcal{S} \setminus C} \gamma^M(s, a, \{t\}) \cdot 0 \ + \ \sum_{t \in C} \gamma^M(s, a, \{t\}) \cdot 1\\
&\quad = \sum_{t \in C} \gamma^M(s, a, \{t\})\\
&\quad = \gamma^M(s, a, C)
\end{aligned}$$

The definition of $\gamma^M(\ldots)$ is identical to the one of $\gamma(\ldots)$ defined in [7], that was used for the definition of $\sim_M$. This completes the proof. □

Thus it is guaranteed that the PM-bisimulation contains the Markovian bisimulation. This means that all pairs of processes not containing probabilistic branching, which are elements of the PM-bisimulation, are also elements of the Markovian bisimulation.

4 PM-TIPP — Syntax and Semantics

4.1 Syntax

PM-TIPP is an extension from our Markovian Process Algebra TIPP. Aside of the standard operators for TIPP [7] we introduce an operator for probabilistic choice. We use the notation

$$\sum([p_1].P_1, \ldots, [p_n].P_n) \quad ,$$

if we focus on a special process term, and adapt the notation of [3]

$$\sum_{i=1}^{n}[p_i].P_i \quad ,$$

if we consider probabilistic choice in common and do not want to emphasise single values of p_i or P_i.

Definition 4.1 *Let* Act_M *be a set of actions,* $\mathcal{R} = I\!R^+$ *the set of rates valid as parameter for the exponential distribution, and* $\Pi = (0 \ldots 1]$ *a set of probabilities. We define a set of expressions by the following grammar:*

$$\begin{array}{rcl} \mathsf{P} & ::= & \mathit{Stop} \quad | \quad (a,\lambda).\mathsf{P} \quad | \quad \mathsf{P}{+}\mathsf{P} \quad | \\ & & \sum([p_1].\mathsf{P},\ldots,[p_n].\mathsf{P}) \quad | \quad \mathsf{P} \parallel_L \mathsf{P} \quad | \\ & & \mathsf{P}\backslash L \quad | \quad \mathsf{P}[f] \quad | \quad X \quad | \quad \mathit{rec}X.\mathsf{P} \quad | \end{array}$$

where $a \in \mathsf{Act}_M$*,* $\lambda \in \mathcal{R}$*,* $p_i \in \Pi$*,* $\sum_{i=1}^{n} p_i = 1$*,* $L \subseteq \mathsf{Act}_M \setminus \{\tau\}$ *and* $f : \mathsf{Act}_M \longrightarrow \mathsf{Act}_M$ *is a function with* $f(\tau) = \tau$ *and* $f(a) \neq \tau$ *for* $a \neq \tau$*.* X *is a process variable.*

The set $\mathcal{L}_{PM}$ *of valid PM-TIPP-expressions contains all terms defined by that grammar that additionally satisfy the following three conditions.*

1. *The term is closed.*
2. *No recursion occurs over static operators (* $\parallel_L$ *,* $\backslash$*,* $[f]$*).*
3. *Process variables occur temporally guarded within sub-terms*[5] $\mathit{rec}X : P(X)$*, only.*

Closed terms are terms in which every occurrence of a process variable X can only be found in sub-terms $\mathit{rec}X : P(X)$, which means that every process variable is bounded. Recursion over static operators potentially leads to infinite state spaces which are not considered within the scope of this paper. Finally *temporally guarded* means, that every occurrence of a process variable X in a sub-term $\mathit{rec}X.P(X)$ is within a sub-term $(a,\lambda).P'(X)$ of $P(X)$. Thus condition 3 excludes terms like $\mathit{rec}X : X$ that make no sense and avoids loops of probabilistic transitions.

4.2 Semantics

Definition 4.2 *The semantics* $[\![P]\!]$ *of a process term* P *is a labelled transition system (with two transition relations)* $(\mathcal{L}_{PM}, P, \longrightarrow, \dashrightarrow)$*, where* $\longrightarrow \subseteq \mathcal{L}_{PM} \times \mathsf{Act}_M \times \mathcal{R} \times \mathit{Lab} \times \mathcal{L}_{PM}$ *and* $\dashrightarrow \subseteq \mathcal{L}_{PM} \times \Pi \times \mathit{Lab} \times \mathcal{L}_{PM}$ *are defined as the smallest relations satisfying the rules in Fig. 5 (in the style of [10]).*

The use of the additional words as the third and second label of the labelled transition relations $\longrightarrow$ and $\dashrightarrow$, respectively, is necessary in order not to lose essential information. See [4] for a more detailed explanation. In fact it is guaranteed that two transitions within

[5]The term $P(X)$ is an abbreviation for an arbitrary process term containing the process variable X.

$[\textsc{pref}^M]$
$$\frac{}{(a,\lambda).P \xrightarrow{a,\lambda,\varepsilon} P}$$

$[+^M_L]$
$$\frac{P \xrightarrow{a,\lambda,w} P'}{P+Q \xrightarrow{a,\lambda,+_l w} P'}$$

$[+^M_R]$
$$\frac{Q \xrightarrow{a,\lambda,w} Q'}{P+Q \xrightarrow{a,\lambda,+_r w} Q'}$$

$[\|^M_L]$
$$\frac{P \xrightarrow{a,\lambda,w} P'}{P \parallel_L Q \xrightarrow{a,\lambda,\|_l w} P' \parallel_L Q} \quad (a \notin L)$$

$[\|^M_R]$
$$\frac{Q \xrightarrow{a,\lambda,w} Q'}{P \parallel_L Q \xrightarrow{a,\lambda,\|_r w} P \parallel_L Q'} \quad (a \notin L)$$

$[\|^M_S]$
$$\frac{P \xrightarrow{a,\lambda,w} P' \quad Q \xrightarrow{a,\mu,v} Q'}{P \parallel_L Q \xrightarrow{a,\lambda\mu,(w\|v)} P' \parallel_L Q'} \quad (a \in L)$$

$[rec^M]$
$$\frac{P\{(recX : P)/X\} \xrightarrow{a,\lambda,w} P'}{recX \ : \ P \xrightarrow{a,\lambda,w} P'}$$

$[\backslash^M_{no}]$
$$\frac{P \xrightarrow{a,\lambda,w} P'}{P\backslash L \xrightarrow{a,\lambda,w} P'\backslash L} \quad (a \notin L)$$

$[\backslash^M_{yes}]$
$$\frac{P \xrightarrow{a,\lambda,w} P'}{P\backslash L \xrightarrow{\tau,\lambda,w} P'\backslash L} \quad (a \in L)$$

$[f^M]$
$$\frac{P \xrightarrow{a,\lambda,w} P'}{P[f] \xrightarrow{f(a),\lambda,w} P'[f]}$$

$[0^P]$
$$\frac{}{Stop \overset{[1],\varepsilon}{\dashrightarrow} Stop}$$

$[\textsc{pref}^P]$
$$\frac{}{(a,\lambda).P \overset{[1],\varepsilon}{\dashrightarrow} (a,\lambda).P}$$

$[+^P]$
$$\frac{P \overset{[p],w}{\dashrightarrow} P' \quad Q \overset{[p'],v}{\dashrightarrow} Q'}{P+Q \overset{[p\cdot p'],(w+v)}{\dashrightarrow} P'+Q'}$$

$[\Sigma^P]$
$$\frac{P_i \overset{[q],w}{\dashrightarrow} P_i'}{\sum[p_i].P_i \overset{[p_i\cdot q],i(w)}{\dashrightarrow} P_i'}$$

$[\|^P]$
$$\frac{P \overset{[p],w}{\dashrightarrow} P' \quad Q \overset{[p'],v}{\dashrightarrow} Q'}{P \parallel_L Q \overset{[p\cdot p'],(w\|v)}{\dashrightarrow} P' \parallel_L Q'}$$

$[rec^P]$
$$\frac{P\{(recX : P)/X\} \overset{[p],w}{\dashrightarrow} P'}{recX \ : \ P \overset{[p],w}{\dashrightarrow} P'}$$

$[\backslash^P]$
$$\frac{P \overset{[p],w}{\dashrightarrow} P'}{P\backslash L \overset{[p],w}{\dashrightarrow} P'\backslash L}$$

$[f^P]$
$$\frac{P \overset{[p],w}{\dashrightarrow} P'}{P[f] \overset{[p],w}{\dashrightarrow} P'[f]}$$

Figure 5: SOS of PM-TIPP

the same transition relation and with the same starting state are labelled with two different additional words. This will become important later.

The notion of equivalence can also be lifted to the level of processes: $P \overset{PM}{\sim} Q :\Longleftrightarrow [\![P]\!] \overset{PM}{\sim} [\![Q]\!]$.

4.3 Properties of the semantics

Some things are worth mentioning concerning the semantic rules for the probabilistic transition relation ($\cdots\!\!\blacktriangleright$). The first one is that the rules $[0^P]$ and $[\text{PREF}^P]$ associate a direct probabilistic loop (with the probability 1) with every state which does not include probabilistic branching, as is required for PMTS. The usual choice operator (+) is resolved by the first action to happen. This is not true for probabilistic branching. Thus rule $[+^P]$ allows both processes to perform their probabilistic transitions and to decide the race afterwards. This property distinguishes our approach from the one proposed in [2]. Finally the effect of $[\Sigma^P]$ is a little tricky. The precondition tests for the probabilistic behaviour of the destination process P_i. There will be a probabilistic behaviour (at least a loop – see above). The two successive probabilistic transitions are amalgamated into one transition for each process P_i. Thus the semantic models of

$$\begin{array}{l} \sum([0.8].\sum([0.5].P_1, [0.5].P_2), [0.2].P_3) \quad \text{and} \\ \sum([0.4].P_1, [0.4].P_2, [0.2].P_3) \end{array}$$

are isomorphic.

Proposition 4.1 *If $P \in \mathcal{L}_{PM}$ then $[\![P]\!]$ is a PMTS.*

Proof: All three properties can be shown by induction over the length of the derivation tree for each transition [11]. □

5 Congruence Investigations and Axiomatisation

5.1 Congruence

Proposition 5.1 *Let $P, Q \in \mathcal{L}_{PM}$ be process terms. Then the following propositions hold*

$$P \overset{PM}{\sim} Q \Longrightarrow$$

$$\begin{array}{lrcl}
(1) & (a,\lambda).P & \overset{PM}{\sim} & (a,\lambda).Q \\
(2) & \sum_{i=1}^{n}[p_i].P_i & \overset{PM}{\sim} & \sum_{i=1}^{n}[p_i].Q_i \\
 & \multicolumn{3}{l}{P_1 \equiv P,\ Q_1 \equiv Q,\ \forall 2 \le i \le n:\ P_i \equiv Q_i} \\
(3) & P+R & \overset{PM}{\sim} & Q+R \\
(4) & P \parallel_L R & \overset{PM}{\sim} & Q \parallel_L R \\
(5) & P \backslash L & \overset{PM}{\sim} & Q \backslash L \\
(6) & P[f] & \overset{PM}{\sim} & Q[f]
\end{array}$$

as long as the considered process terms are valid expressions in $\mathcal{L}_{PM}$.

Proof: We will highlight only one special aspect of the proof [11], namely the probabilistic choice operator (2). We shall write $\sum P$ as an abbreviation for $\sum([p_1].P, [p_2].P_2, \ldots, [p_n].P_n)$ and $\sum Q$ for $\sum([p_1].Q, [p_2].P_2, \ldots, [p_n].P_n)$. Our goal is

$$P \overset{PM}{\sim} Q \implies \sum P \overset{PM}{\sim} \sum Q$$

We will show directly, that $\sum P$ and $\sum Q$ fulfil the conditions of Def. 3.3.

Assuming $P_1 \equiv P$ for a more compact notation we can deduce:

$$\begin{aligned}
&\gamma^P(\textstyle\sum P, C) \\
= \;& \textstyle\sum \{\!| \, \pi \mid \exists P' \in C,\ \exists w \in Lab:\ \sum P \xrightarrow{[\pi],w} P' \, |\!\} \\
= \;& \sum_{i=1}^{n} p_i \textstyle\sum \{\!| \, \pi \mid \exists P' \in C,\ \exists w \in Lab: \\
& \qquad\qquad\qquad w = i(w') \ \wedge\ P_i \xrightarrow{[\pi],w'} P' \, |\!\} \\
= \;& p_1 \textstyle\sum \{\!| \, \pi \mid \exists P' \in C,\ \exists w' \in Lab:\ P \xrightarrow{[\pi],w'} P' \, |\!\} + \\
& \sum_{i=2}^{n} p_i \textstyle\sum \{\!| \, \pi \mid \exists P' \in C,\ \exists w' \in Lab:\ P_i \xrightarrow{[\pi],w'} P' \, |\!\} \\
= \;& p_1 \cdot \textstyle\sum \gamma^P(P, C) \ + \ \sum_{i=2}^{n} \pi_i \cdot \gamma^P(P_i, C) \\
= \;& p_1 \cdot \textstyle\sum \gamma^P(Q, C) \ + \ \sum_{i=2}^{n} \pi_i \cdot \gamma^P(P_i, C) \\
= \;& \ldots \\
= \;& \gamma^P(\textstyle\sum Q, C)
\end{aligned}$$

This proves condition (i). For condition (ii) there is nothing to show, because the implication evaluates to true due to the false precondition. □

5.2 Axiomatisation

Figure 6 shows axioms concerning the probabilistic choice operator. The rule P1 reflects the fact that a probabilistic choice leaving the initial state of a transition system denotes the initial distribution of weights on the Markov chain. If there is no probabilistic choice, it is assumed that this weight is 1 for the initial state and 0 for all others. P2 includes commutativity and associativity of the probabilistic choice, allowing an arbitrary permutation h of all participating processes. P3 amalgamates two summands within the probabilistic choice operator. P4 flattens a hierarchical structure of several probabilistic choice operators. The axioms P5 – P8 show the behaviour of probabilistic choice in connection with the operators $+$, $\|_L$, $[f]$ and $\backslash L$, respectively. Finally P9 is the rule which describes the elimination of probabilistic branching from the model by amalgamating it to a preceding timed transition.

We can convince ourselves that these axioms are sound with respect to the PM-bisimulation. Additionally there is a strong conjecture that these axioms together with the axioms of the Markovian bisimulation [7, 8], shown in Fig. 7, form a complete set of axioms for sequential processes in PM-TIPP. This is due to the fact that every probabilistic choice operator (except for those originating in the initial state of the system) can be eliminated from any process term using these rules. In that way we obtain a process term taken from our basic Markovian Process Algebra for which we already proved the completeness of the set of axioms.

$$\sum [1].P = P \qquad \mathsf{P1}$$

$$h : \{1 \ldots n\} \xrightarrow{bij.} \{1 \ldots n\}$$

$$\sum_{i=1}^{n} [p_i].P_i = \sum_{i=1}^{n} [p_{h(i)}].P_{h(i)} \qquad \mathsf{P2}$$

$$\sum ([p_1].P_1, \ldots, [p_{n-1}].P_{n-1}, [p_n].P_{n-1}) = \sum ([p_1].P_1, \ldots, [p_{n-1} + p_n].P_{n-1}) \qquad \mathsf{P3}$$

$$\sum([p_1].(\sum_{j=1}^{m} [q_j].Q_j), [p_2].P_2, \ldots, [p_n].P_n) =$$
$$\sum ([p_1 \cdot q_1].Q_1, \ldots, [p_1 \cdot q_m].Q_m, [p_2].P_2, \ldots, [p_n].P_n) \qquad \mathsf{P4}$$

$$\delta : \{1 \ldots m\} \times \{1 \ldots n\} \xrightarrow{bij.} \{1 \ldots m \cdot n\}$$

$$\sum_{i=1}^{m} [p_i].P_i + \sum_{j=1}^{n} [q_j].Q_j = \sum_{\delta(i,j)=1}^{m \cdot n} [p_i \cdot q_j].(P_i + Q_j) \qquad \mathsf{P5}$$

$$\delta : \{1 \ldots m\} \times \{1 \ldots n\} \xrightarrow{bij.} \{1 \ldots m \cdot n\}$$

$$\sum_{i=1}^{m} [p_i].P_i \parallel_L \sum_{j=1}^{n} [q_j].Q_j = \sum_{\delta(i,j)=1}^{m \cdot n} [p_i \cdot q_j].(P_i \parallel_L Q_j) \qquad \mathsf{P6}$$

$$\left(\sum_{i=1}^{n} [p_i].P_i\right)[f] = \sum_{i=1}^{n} [p_i].P_i[f] \qquad \mathsf{P7}$$

$$\left(\sum_{i=1}^{n} [p_i].P_i\right) \backslash L = \sum_{i=1}^{n} [p_i].P_i \backslash L \qquad \mathsf{P8}$$

$$(a, \lambda). \sum_{i=1}^{n} [p_i].P_i = (a, \lambda p_1).P_1 + \ldots + (a, \lambda p_n).P_n \qquad \mathsf{P9}$$

Figure 6: Axioms for probabilistic choice

$$(\alpha, \lambda).P + (\alpha, \mu).P = (\alpha, \lambda + \mu).P \quad \mathsf{M0} \qquad P + Stop = P \quad \mathsf{M1}$$
$$P + (Q + R) = (P + Q) + R \quad \mathsf{M2} \qquad P + Q = Q + P \quad \mathsf{M3}$$

Figure 7: Axioms for the basic Markovian calculus

6 Conclusion and Further Work

In this paper we showed that one can eliminate probabilistic transitions in an algebraically exact manner by the use of an appropriate equivalence notion. We elaborated an extension of a standard MPA and a notion of equivalence for the elimination of probabilistic transitions. A complete theory (congruence investigations, axiomatisation) is introduced.

The newest results define extended transition systems including three different transition relations, namely the Markovian, the probabilistic, and additionally the immediate transitions as they are defined in [8]. The concepts are orthogonal in the sense that all axioms presented for both language extensions are sound with the combined calculus. A complete axiomatisation can be obtained by adding *one* new axiom relating immediate actions and probabilistic branching, namely:

$$\underline{\alpha}.\sum[p_i].P_i = \sum[p_i].\underline{\alpha}.P_i \qquad \mathsf{PI}$$

More details about this work can be found in [11].

One drawback of the presented theory is a lack of expressiveness in our language. One particular situation we cannot model up to now is a choice between two activities in a process P_1 that is to be resolved by probabilities *if synchronisation with the environment (P_2) is possible*, otherwise the behaviour of P_1 is deterministic. This situation can easily be expressed as a Petri net (cf. Fig. 8). If t_1 is the first transition to fire, then there is no choice

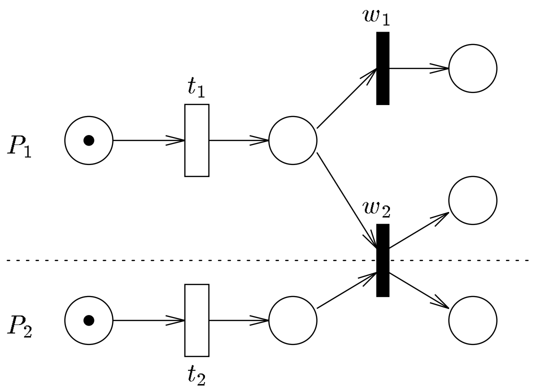

Figure 8: Sample Petri Net

since only one immediate transition is enabled. It will fire immediately. If t_2 fires first, then there will be a probabilistic choice between the two immediate transitions after t_1 fired. This choice will be resolved according to the given weights. To achieve this expressiveness we will introduce new concepts like priorities and new operators.

Acknowledgements The author wishes to thank the anonymous reviewers for their useful comments as well as Jane Hillston, Stephen Gilmore and Holger Hermanns for proof reading.

References

[1] M. Bernardo, L. Donatiello, and R. Gorrieri. MPA: A Stochastic Process Algebra. Technical report UBLCS-94-10, Laboratory for Computer Science, University of Bologna, 1994.

[2] P. Buchholz. On a Markovian Process Algebra. Technical Report 500, Informatik IV, Universität Dortmund, 1994.

[3] A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of Working Conference on Programming Concepts and Methods*, Sea of Gallilee, Israel, April 1990. IFIP TC 2.

[4] N. Götz, U. Herzog, and M. Rettelbach. Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebras. In *Proc. of the 16th Int'l Symposium on Computer Performance Modelling, Measurement and Evaluation, PERFORMANCE '93*. Springer, 1993. LNCS 729.

[5] H. Hansson. *Time and Probability in Formal Design of Distributed Systems*, volume 1 of *Real-Time Safety Critical Systems*. Elsevier, 1994.

[6] H. Hermanns. *Semantics of Process Languages for Performance Evaluation*. Diploma Thesis, Universität Erlangen-Nürnberg, 1993. (in German).

[7] H. Hermanns and M. Rettelbach. Syntax, Semantics, Equivalences, and Axioms for MTIPP. In U. Herzog and M. Rettelbach, editors, *Proceedings of the 2nd Workshop on Process Algebra and Performance Modelling*, Martensstr. 3, D-91085 Erlangen, November 1994. IMMD VII, Universität Erlangen-Nürnberg.

[8] H. Hermanns, M. Rettelbach, and T. Weiß. Formal Characterisation of Immediate Actions in SPA with Nondeterministic Branching. In *Proceedings of the 3rd Workshop on Process Algebra and Performance Modelling*, 1995. this issue.

[9] K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. In *Proc. of 16th ACM Symposium on Principles of Programming Languages*, pages 344–352, 1989.

[10] G. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Computer Science Department, Aarhus University, September 1981.

[11] M. Rettelbach. *Immediate Transitions in Stochastic Process Algebras — Theory and Application*. PhD thesis, Universität Erlangen-Nürnberg, 1995. (in German).

[12] Y. Wang and K.G. Larsen. Testing probabilistic and nondeterministic processes. In *International Workshop on Protocol Specification, Testing and Verification XII (Lake Buena Vista, Florida)*. North-Holland, 1992. IFIP TC-6 International Workshop.