# Reactive, Generative, and Stratified Models of Probabilistic Processes

Rob van Glabbeek
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
*robvg@cwi.nl*

Scott A. Smolka*
Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794-4400
USA
*sas@sbcs.sunysb.edu*

Bernhard Steffen
Aarhus University
Datalogisk Afdeling
Ny Munkegade 116 - 8000 Aarhus C
Denmark
*bus@daimi.dk*

Chris M. N. Tofts
LFCS
University of Edinburgh
Edinburgh EH9 3JZ
Scotland
*cmnt@lfcs.ed.ac.uk*

## 1  Introduction

In the *reactive model* [Pnu85] of classical concurrency theory, a process reacts to stimuli presented by its environment. A mechanistic view of the reactive model has been given by Milner [Mil80] in terms of *button pushing experiments*. The environment or *observer* experiments on a process by attempting to depress one of several buttons that the process possesses as its interface to the outside world. The experiment *succeeds* if the button is unlocked and therefore goes down; otherwise the experiment *fails*. In response to a successful experiment, the process makes an internal state transition and is then ready for further experimentation.

The reactive model has been adopted by Larsen and Skou [LS89] for *probabilistic processes*: a button-pressing experiment succeeds, with probability 1, or else fails. If successful, the process makes an internal state transition according to a probability distribution associated with the depressed button.

In the probabilistic case, it is interesting to consider a more "probabilistic" form of experimentation we call the *generative model*. In this setting, an observer may attempt to depress *more than one button at a time*.

Now the process is more or less on equal footing with its environment, and will decide, according to a prescribed probability distribution, which button if any will go down. In response to a successful outcome, this same probability distribution, conditioned by the process's choice of button, will govern the internal state transition made by the process.

For example, consider the reactive process $P$ and the generative process $Q$ given by:

$$P = \tfrac{1}{4}a + \tfrac{3}{4}a.(a+b) + b.c$$
$$Q = \tfrac{1}{6}a + \tfrac{1}{2}a.(\tfrac{1}{2}a + \tfrac{1}{2}b) + \tfrac{1}{3}b.c$$

$P$ and $Q$ have as semantics the *probabilistic labeled transition systems* depicted in Figure 1. For $P$, an $a$- or $b$-experiment will succeed with probability 1, whereas a $c$-experiment will fail. In the case of an $a$-experiment, $P$ will branch left with probability $\tfrac{1}{4}$ and right with probability $\tfrac{3}{4}$. Note that no information is given about the relative probability of performing an $a$-action versus a $b$-action in $P$'s initial state.

For the generative process $Q$, if the observer simultaneously attempts to depress the $a$ and $b$ buttons, $Q$ will unlock its $a$-button with probability $\tfrac{2}{3}$ and its $b$-button with probability $\tfrac{1}{3}$. In the former case, $Q$ will branch left with probability $\tfrac{1}{4}$ and right with probability $\tfrac{3}{4}$,

Figure 1: Reactive process $P$ and generative process $Q$.

which is precisely $P$'s reaction to an $a$-experiment. In fact, for any *single-button experiment*, $P$ and $Q$ behave the same. Thus $Q$ contains strictly more information than $P$, and, in a broader sense, the reactive model is an *abstraction* of the generative model.

In this paper we also consider the *stratified model* of probabilistic processes, which captures the branching structure of the *purely probabilistic choices* made by a process. For example, consider an operating system in which there are $n$ processes to be multiprogrammed. One of these is the garbage collector which requires exactly one third of the CPU cycles to function properly. The other $n - 1$ processes are user processes and should equally share the remaining two thirds of the CPU. For the case $n = 3$, a plausible specification of a scheduler for these processes would be

$$Sc = fix_X(\tfrac{1}{3}a.X + \tfrac{1}{3}b.X + \tfrac{1}{3}c.X)$$

where the action $a$ identifies the garbage collector, and $b$ and $c$ the user processes. But consider the *restriction context* in which user $c$ is denied further access to the machine. What would happen to its share of the CPU? Because of the symmetry in the above specification, we would naturally arrive at the expression

$$fix_X(\tfrac{1}{2}a.X + \tfrac{1}{2}b.X)$$

Now, however, the garbage collector is granted one half of the CPU which is *different* from our original intent. An exact specification of the scheduler can be obtained through the use of *nested* expressions of probabilistic choice:

$$Sc' = fix_X(\tfrac{1}{3}a.X + \tfrac{2}{3}(\tfrac{1}{2}b.X + \tfrac{1}{2}c.X))$$

which, in the stratified model, yields the leftmost probabilistic labeled transition system of Figure 2. If user $c$ were now denied access we would obtain

$$fix_X(\tfrac{1}{3}a.X + \tfrac{2}{3}b.X)$$

as desired. Thus, in the stratified model, the intended relative frequencies are preserved in a level-wise fashion in the presence of restriction.

Note that the probabilistic labeled transition system of $Sc'$ in the generative model is simply the right one of Figure 2. Thus, in the generative model, $Sc$ is (unfortunately) equivalent to $Sc'$. We shall see that, in a broader sense, the generative model is an abstraction of the stratified model, in which the branching structure of probabilistic choices has been "flattened."

The extremal case of nested probabilistic choice in the stratified model, in which zero probabilities are permitted, yields a general notion of *process priority*. For example, the expression

$$1P + 0(1Q + 0R)$$

gives priority to process $P$ over $Q$ and $R$, and priority to $Q$ over $R$. Thus process $R$ can only be executed in a restriction context that excludes $P$ and $Q$. Zero probabilities are not considered in this paper, but their role in modeling priority is examined carefully in [SS90].

## Summary of Technical Results

We will be working within the framework of PCCS, a specification language for probabilistic processes introduced in [GJS90]. PCCS is derived from Milner's SCCS [Mil83] by replacing the operator of nondeterministic process summation with a probabilistic counterpart. Several PCCS expressions have appeared above, which should give the flavor of the language.

For each of the three models we present the following:

- a *structural operational semantics* of PCCS, given as a set of inference rules in the style of Plotkin [Plo81] and Milner [Mil89]. For each model, these inference rules constitute a *semantic mapping* from the set of PCCS process expressions, $Pr$, to a particular domain of probabilistic labeled transition systems. We denote these mappings as $\varphi_R$, $\varphi_G$, and $\varphi_S$, respectively. (As discussed in Section 3, the summation and relabeling operators of PCCS are not compatible with the reactive model. Therefore, $\varphi_R$ applies only to a sublanguage PCCS$_R$ of PCCS, in which summations
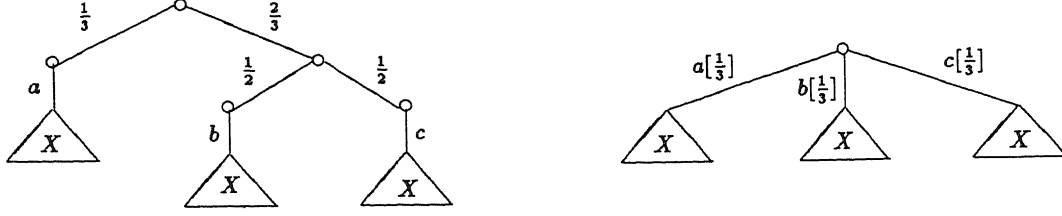
Figure 2: Stratified and generative transition systems of $Sc'$.

are both probability- *and* action-guarded, and re-labeling is excluded.)

- a notion of *bisimulation semantics*. In [LS89], Larsen and Skou introduced *probabilistic bisimulation*, a natural and elegant extension of strong bisimulation [Par81, Mil83] for reactive processes. We likewise define probabilistic bisimulation on generative and stratified processes. In each model, the largest probabilistic bisimulation (under set inclusion), denoted $\overset{R}{\sim}$, $\overset{G}{\sim}$, and $\overset{S}{\sim}$, respectively, determines the model's bisimulation semantics.

- We prove that $\overset{R}{\sim}$ is a congruence with respect to $\text{PCCS}_R$, and $\overset{G}{\sim}$ and $\overset{S}{\sim}$ are congruences with respect to PCCS.

We then inter-relate the models, ultimately showing that they form a hierarchy: the generative model is an abstraction of the stratified model, and the reactive model is an abstraction of the generative model. This reflects the stepwise reduction of "observational power"; *i.e.* starting from the stratified model, we first abstract from the probabilistic branching structure, and then from the relative probabilities among different actions. We proceed as follows:

- We add to the generative and stratified operational semantics *inter-model abstraction rules*, which allow the inference of reactive probabilistic transitions from generative ones; and, likewise, the inference of generative probabilistic transitions from stratified ones. These rules determine mappings between domains of probabilistic labeled transition systems, and are denoted as $\varphi_{GR}$ and $\varphi_{SG}$, respectively.

- We obtain the following *commutativity results*, which establish the hierarchy among the models. For any $\text{PCCS}_R$ expression $P$ and restriction-free PCCS expression $Q$,

$$\varphi_R(P) = \varphi_{GR}(\varphi_G(P)) \quad \text{and}$$
$$\varphi_G(Q) = \varphi_{SG}(\varphi_S(Q))$$

Additionally, we show that the latter commutativity result does not hold in the presence of restriction.

- We then show that the equivalence induced on the stratified (generative) model via abstraction to the generative (reactive) model is *not* a congruence with respect to PCCS. This demonstrates the need for refining the bisimulation semantics when moving to a less abstract model. More precisely, we exhibit a pair of PCCS processes $P$, $Q$ and a context $C[\,]$ such that

$$\varphi_{SG}(\varphi_S(P)) \overset{G}{\sim} \varphi_{SG}(\varphi_S(Q)) \quad \text{and}$$
$$\varphi_{SG}(\varphi_S(C[P])) \overset{G}{\not\sim} \varphi_{SG}(\varphi_S(C[Q]))$$

Similarly for the reactive vs. generative case.

The interdependencies between the different models are summarized in Figure 3. Here the upper part reflects the commutativity results, and the dashed arrows below indicate the bisimulations that are induced on the stratified (generative) model via abstraction to the generative (reactive) model.

We conclude the paper with an interesting open problem concerning an equivalence relation $\overset{M}{\sim}$ (*mixed bisimulation*) that, in terms of its distinguishing strength, falls strictly between $\overset{G}{\sim}$ and $\overset{S}{\sim}$, and is still a congruence in the stratified model. We conjecture that $\overset{M}{\sim}$ is the largest congruence contained in $\overset{G}{\sim}$.

## Related Work

Larsen and Skou [LS89] have examined the reactive model in the setting of *testing*. They exhibit a testing algorithm that, with probability $1 - \epsilon$, where $\epsilon$ is arbitrarily small, can distinguish processes that are not probabilistically bisimilar. Bloom and Meyer
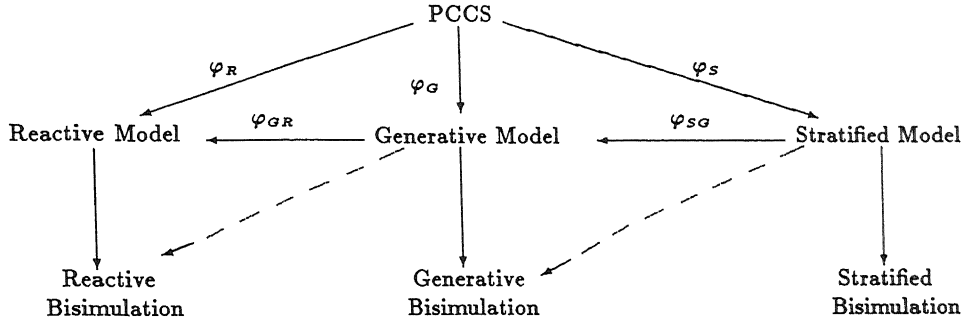
Figure 3: Interdependencies between the models.

[BM89] further show that if nondeterministic bounded-branching processes $P$ and $Q$ are bisimilar, then there is an assignment of probabilities to the edges of $P$ and $Q$, yielding reactive processes $P'$ and $Q'$ such that $P'$ and $Q'$ are probabilistically bisimilar; and $P'$ and $Q'$ have the same probability of producing a given outcome under every test.

Christoff [Chr89] also considers the testing of probabilistic processes. He proposes three probabilistic testing equivalences and outlines an algorithm for the verification of these equivalences. Finally, Jones and Plotkin [JP89] investigate a probabilistic powerdomain of evaluations which they use to give the semantics of a language with a probabilistic parallel construct.

## 2 Syntax of PCCS

As in SCCS, the *atomic actions* of PCCS form a multiplicative structure $(Act, \cdot)$ that is generated freely from the set $\Lambda$ of *particulate actions*. Unlike SCCS, where $Act$ is an abelian monoid, we assume neither commutativity nor associativity for action product $(\cdot)$. Thus all elements of $Act$ are of the form $a$ or $(\alpha, \beta)$, where $a \in \Lambda$ and $\alpha, \beta \in Act$. One can think of the atomic action $(\alpha, \beta)$, which we sometimes write as $\alpha\beta$, as the *simultaneous ordered occurrence* of actions $\alpha$ and $\beta$.

As discussed in Section 3, the free structure of our action algebra is necessary to be able to define synchronous product in the reactive model. For any SCCS-like action monoid or group, the corresponding synchronization merge can be expressed in our calculus by a combination of product and relabeling. For example, the group structure of SCCS can be obtained through relabelings like: $(\alpha, \overline{\alpha}) \rightarrow 1$ and $(\overline{\alpha}, \alpha) \rightarrow 1$, where $1$ is the unit or idle action of SCCS.

Let $X$ be a variable, $A$ a subset of $Act$, and $f : Act \rightarrow Act$. The syntax of PCCS is given by:

$$E ::= 0 \mid X \mid \alpha.E \mid \sum_{i \in I} [p_i] E_i \text{ where } p_i \in (0, 1],$$

$$\sum_{i \in I} p_i = 1 \mid E \times F \mid E \restriction A \mid E[f] \mid fix_X E$$

An expression having no free variables is called a *process*, and $Pr$ is the set of all PCCS processes. Intuitively, $0$ is the *zero process* having no transitions, while $\alpha . E$ performs action $\alpha$ with probability 1 and then behaves like $E$. The expression $\sum[p_i] E_i$ offers a probabilistic choice among its constituent behaviors $E_i$. $E \times F$ represents synchronized product, and the restricted expression $E \restriction A$ can perform actions only from the set $A$. Finally, $E[f]$ specifies a relabeling of actions, and $fix_X E$ defines a recursive process.

For this paper, all summation expressions are required to be finite. Also, we write the binary version of process summation as $[p] E + [1 - p] F$, and often omit the square brackets around the probabilities.

## 3 The Reactive Model

The reactive model of probabilistic processes was introduced by Larsen and Skou in [LS89]. In this section, we consider the reactive model within the context of $PCCS_R$, a sublanguage of PCCS tailored to describing reactive processes. We begin by presenting the reactive operational semantics for $PCCS_R$ that defines a probabilistic transition system for every $PCCS_R$ process. We then equip the model with a notion of probabilistic bisimulation, also due to Larsen and Skou, and show that the resulting equivalence relation is a congruence with respect to $PCCS_R$.

The syntax of $PCCS_R$ is identical to that of PCCS

except for:

1. Summation expressions are required to be both probability- *and* action-guarded; i.e., they are of the form:

$$\sum_{i \in I} [p_i]\, \alpha_i \,.\, E_i \quad \text{where } \alpha_i \in Act,\ p_i \in (0,1] \text{ and}$$

$$\sum_{i \in I} p_i = 1$$

2. The relabeling operator is eliminated.

The first restriction is a consequence of the reactive viewpoint (see Section 1): for each action symbol $\alpha$, a reactive process either possesses no $\alpha$-transitions, or else the sum of the probabilities of all $\alpha$-transitions is 1. This would immediately suggest a syntax for summation expressions of the form:

$$\sum_{i \in I}\sum_{j \in J} [p_{ij}]\, \alpha_i \,.\, E_{ij} \quad \text{such that } \forall i \in I,\ \sum_{j \in J} p_{ij} = 1$$

However, such summations are not syntactically valid in PCCS, as the syntax of PCCS is inherently generative. Therefore, instead, we resort to PCCS summations of the form $E = \sum[p_i]\alpha_i \,.\, E_i$, and the probability that $E$ will perform the transition encoded by the $i$th summand will be $p_i$, conditioned by the assumption $\{\alpha_i\}$. For example, the expression $\frac{1}{4}a.X + \frac{1}{2}a.Y + \frac{1}{4}b.Z$ has an $a$-transition to $X$ with associated probability $\frac{1}{3}$, an $a$-transition to $Y$ with associated probability $\frac{2}{3}$, and a $b$-transition to $Z$ with associated probability 1.

We note that it is possible to provide a reactive semantics to general PCCS summations. However, the resulting models lack key algebraic properties such as substitutivity in the bisimulation semantics, or commutativity from the generative model.

The second restriction on the syntax of $PCCS_R$ is due to an inherent incompatibility between the operation of relabeling and the reactive viewpoint. For example, consider process $P = \frac{1}{2}a.X + \frac{1}{2}b.Y$. $P$ has a probability-1 $a$-transition to $X$ and a probability-1 $b$-transition to $Y$. However, if the relabeling that maps $a$ to itself and $b$ to $a$ is applied to $P$, then we end up with a "nonsensical" object having two probability-1 $a$-transitions. Relabeling could be defined in the reactive model if an appropriate normalization procedure were applied. But, in this case, we would lose substitutivity and commutativity as for general summation. Of course, injective relabelings can be added without problem.

## 3.1 Operational Semantics of $PCCS_R$

The reactive operational semantics of $PCCS_R$ is given in Figure 4 as a set of inference rules. These rules collectively define the semantic mapping $\varphi_R$ from $Pr_R$, the closed expressions of $PCCS_R$, to the domain of reactive probabilistic labeled transition systems. (This domain will be defined more formally in Section 3.2.) Reactive transitions are of the form

$$P \xrightarrow{\alpha[p]} P'$$

meaning that $P$, with probability $p$, can perform an $\alpha$-transition to become $P'$.

The first rule, which has no premise and is thus an axiom, defines the probabilistic transitions of a probability- and action-guarded summation (henceforth referred to simply as a "guarded summation"). Here, $r_i$ is the normalization factor used to compute the conditional probabilities of the guarded sum under the assumption $\{\alpha_i\}$. Note that action-prefixing is now a special case of guarded sum in which the index set $I$ is of cardinality 1. The rest of the rules are straightforward adaptations of their SCCS counterparts.

Unlike SCCS, all probabilistic transitions are indexed. The purpose of the indices is to distinguish different *occurrences* of the same probabilistic transition, and are constructed so that every probabilistic transition of an expression has a unique index. (The indices will be used in the next section to define *cumulative* probability distributions.) The following example is illustrative:

$$([\tfrac{1}{2}]a\,.\,0 + [\tfrac{1}{2}]a\,.\,0) \xrightarrow{a[\frac{1}{2}]}_1 0$$

$$([\tfrac{1}{2}]a\,.\,0 + [\tfrac{1}{2}]a\,.\,0) \xrightarrow{a[\frac{1}{2}]}_2 0$$

## 3.2 Reactive Bisimulation

We now consider *reactive bisimulation*, a notion of probabilistic bisimulation for reactive processes due to Larsen and Skou [LS89]. By definition, all reactive bisimulations are equivalence relations. Intuitively, two processes $P,Q$ are probabilistically bisimilar in the reactive model if, for each action symbol, they derive reactive bisimulation classes with equal cumulative probability.

To define reactive bisimulation, we first need to define the *cumulative probability distribution function* (cPDF) which computes the total probability by which a process derives a set of processes. Using $\mathcal{P}$ as the

$$\sum_{i \in I} [p_i] \alpha_i . E_i \xrightarrow{\alpha_i[p_i/r_i]}_i E_i \qquad \left( r_i = \sum_{i \in I,\ \alpha_j = \alpha_i} p_j \right)$$

$$E \xrightarrow{\alpha[p]}_i E', \ F \xrightarrow{\beta[q]}_j F' \quad \Longrightarrow \quad E \times F \xrightarrow{\alpha\beta\ [p \cdot q]}_{(i,j)} E' \times F'$$

$$E \xrightarrow{\alpha[p]}_i E' \qquad\qquad \Longrightarrow \quad E \!\restriction\! A \xrightarrow{\alpha[p]}_i E' \!\restriction\! A \qquad (\alpha \in A)$$

$$E\{fix_X E/X\} \xrightarrow{\alpha[p]}_i E' \quad \Longrightarrow \quad fix_X E \xrightarrow{\alpha[p]}_i E'$$

Figure 4: Reactive Operational Semantics of $PCCS_R$

powerset operator and adopting the convention that the empty sum of probabilities is 0, we have:

**Definition 1 (Reactive cPDF)** $\mu_R \colon (Pr_R \times Act \times \mathcal{P}(Pr_R)) \longrightarrow [0,1]$ *is the total function given by :* $\forall \alpha \in Act, \ \forall P \in Pr_R, \ \forall S \subseteq Pr_R,$

$$\mu_R(P, \alpha, S) = \sum_i \{ p_i \,|\, P \xrightarrow{\alpha[p_i]}_i Q \text{ and } Q \in S \}$$

For an equivalence relation $\mathcal{R}$ over $Pr_R$, we write $Pr_R/\mathcal{R}$ to denote the set of equivalence classes induced by $\mathcal{R}$. Reactive bisimulation can now be defined as follows:

**Definition 2 ([LS89])** *An equivalence relation* $\mathcal{R} \subseteq Pr_R \times Pr_R$ *is a reactive bisimulation if* $(P, Q) \in \mathcal{R}$ *implies :* $\forall S \in Pr_R/\mathcal{R}, \ \forall \alpha \in Act,$

$$\mu_R(P, \alpha, S) = \mu_R(Q, \alpha, S)$$

*Two processes* $P, Q$ *are reactive bisimulation equivalent (written* $P \overset{R}{\sim} Q$*) if there exists a reactive bisimulation* $\mathcal{R}$ *such that* $(P, Q) \in \mathcal{R}$.

We will sometimes consider reactive bisimulation equivalence of transition systems rather than expressions, e.g., $\varphi_R(P) \overset{R}{\sim} \varphi_R(Q)$. Definition 2 can be used with no change in this case.

Similar to the case of classical bisimulation, if $\mathcal{R}_1$ and $\mathcal{R}_2$ are reactive bisimulations, then so is their transitive closure $(\mathcal{R}_1 \cup \mathcal{R}_2)^*$. Furthermore, reactive bisimulation equivalence is the largest reactive bisimulation, i.e.

$$\overset{R}{\sim} = \bigcup \{ \mathcal{R} \mid \mathcal{R} \text{ is a reactive bisimulation} \}$$

and can be found by a straightforward adaptation of the fixed-point iteration technique of [Mil89].

Like strong bisimulation does for SCCS or CCS, reactive bisimulation equivalence provides a compositional notion of semantics for $PCCS_R$ that is consistent with the operational semantics defined in the last section. Specifically:

**Theorem 1 (Congruence)** *For* $P, Q \in Pr_R$:

$$P \overset{R}{\sim} Q \text{ implies } \forall \ PCCS_R \text{ contexts } C[\ ] \colon C[P] \overset{R}{\sim} C[Q]$$

**Proof:** The proof is by induction on the structure of the context. The case for the empty context is trivial. Thus, it remains to investigate only the top-most operator of the context. The proof for product is essentially the same as the one given in Section 4.2 for the generative model. The case of restriction is straightforward, and for *fix*, the proof is an adaptation of the same case in the congruence proof for strong bisimulation [Mil83].

We are left with the case of guarded summation. Let $C[\ ]$ be a summation context in which exactly one summand $E_i$ is replaced with a hole. It is sufficient to show that $P \overset{R}{\sim} Q$ implies $C[P] \overset{R}{\sim} C[Q]$. Let $(P, Q)$ belong to the reactive bisimulation $\mathcal{R}_0$. It is enough to show that the reflexive, symmetric and transitive closure $\mathcal{R}$ of $\mathcal{R}_1 = \mathcal{R}_0 \cup \{(C[P], C[Q])\}$ is a reactive bisimulation. This is straightforward. $\square$

We close this section by pointing out how the cumulative PDF $\mu_R$ can be used to define the semantic mapping $\varphi_R$ from $PCCS_R$ to the domain of reactive transition systems. In particular, $\varphi_R$ is of the form

$$\varphi_R : Pr_R \longrightarrow (Pr_R, (Pr_R \times Act \times Pr_R) \longrightarrow [0,1])$$

such that $\varphi_R(P)$ represents the probabilistic transition system having start state $P$ and (cumulative) probability $\mu_R(P, \alpha, \{Q\})$ of transiting to state $Q$ via action

$\alpha$. Note that the cumulative PDF provides us with a transition-system semantics in which the indices of the previous section, as defined by the inference rules of Figure 4, have been abstracted away.

# 4 The Generative Model

In contrast to the reactive model, which is defined only over the sublanguage $PCCS_R$ of PCCS, the generative model is defined over full PCCS. In this section, we provide PCCS with a generative operational semantics. We then extend the notion of reactive bisimulation to the generative case and show that the resulting equivalence is a congruence with respect to PCCS.

## 4.1 Operational Semantics of PCCS

The generative operational semantics of PCCS is given in Figure 5. We use a different kind of arrow (non-hooked) to distinguish generative transitions from reactive ones. As in the reactive case, generative transitions are indexed to distinguish multiple occurrences of the same probabilistic transition.

With the exception of restriction, all rules are straightforward adaptations of their SCCS counterparts. The restriction rule defines the probabilistic transitions of $E \upharpoonright A$ in terms of the *conditional probabilities* of $E$ under the assumption $A$. In this rule, the function $\nu_G$ computes the *generative normalization factor* such that $\nu_G(E, A)$ is the sum of the probabilities of the transitions of $E$ labeled by symbols from $A$. Using $\{\!\!\{,\}\!\!\}$ as multi-set brackets, the formal definition of $\nu_G$ is given by

$$\nu_G(E, A) = \sum \{\!\!\{ p_i \mid E \xrightarrow{\alpha[p_i]}_i E_i , \alpha \in A \}\!\!\}$$

To illustrate the generative operational semantics, consider the expression

$$E = (a \, . \, 0) \times ([\tfrac{1}{3}]b \, . \, X + [\tfrac{1}{3}]c \, . \, Y + [\tfrac{1}{3}]0)$$

We have:

$$E \xrightarrow{(a,b)[\frac{1}{3}]}_{(1,1.1)} 0 \times X \qquad E \xrightarrow{(a,c)[\frac{1}{3}]}_{(1,2.1)} 0 \times Y$$

As $\nu_G(E, \{(a,b)\}) = \tfrac{1}{3}$, we also have:

$$E \upharpoonright \{(a,b)\} \xrightarrow{(a,b)[1]}_{(1,1.1)} (0 \times X) \upharpoonright \{(a,b)\}$$

A generative process is said to be *stochastic* if the sum of the probabilities of its derivations is 1. Otherwise, when this sum is strictly less than 1, the process is said to be *substochastic*, and therefore possesses a non-zero probability of deadlock. With the exception of restriction, the inference rules of Figure 5 *preserve stochasticity*: if the processes in the assumptions are stochastic, then so are the defined processes. In the case of restriction, the defined process may have no derivations at all.

The normalization factor $\nu_G(E, A)$ used in the restriction rule of Figure 5 is such that a non-zero substochastic process placed in a restriction context becomes stochastic. Alternatively, the relative probability of deadlock in a substochastic process can be preserved by normalizing by the quantity $r = \nu_G(E, A) + 1 - \nu_G(E, Act)$. The term $1 - \nu_G(E, Act)$ represents the probability of deadlock in $E$. To illustrate, we would have in the above example that $\nu_G(E, Act) = \tfrac{2}{3}, r = \tfrac{2}{3}$, and thus:

$$E \upharpoonright \{(a,b)\} \xrightarrow{(a,b)[\frac{1}{2}]}_{(1,1.1)} (0 \times X) \upharpoonright \{(a,b)\}$$

## 4.2 Generative Bisimulation

The extension of reactive bisimulation to the generative model is straightforward. The definition of the generative cPDF $\mu_G$ is the same as Definition 1 except that it is defined over $Pr$ and in terms of indexed generative transitions. Likewise, the definition of a generative bisimulation and of $\overset{G}{\sim}$ are the same as in Definition 2, except that they are defined over $Pr$ and in terms of $\mu_G$. Also, the semantic mapping $\varphi_G$ from PCCS to the domain of generative transition systems is defined exactly as $\varphi_R$.

Similar to the reactive case, $\overset{G}{\sim}$ is substitutitive in PCCS. For an equivalence relation $\mathcal{R}$ and a process $P$, we write $[P]_{\mathcal{R}}$ to denote the equivalence class induced by $\mathcal{R}$ of which $P$ is a member.

**Theorem 2 (Congruence)** *For $P, Q \in Pr$:*

$$P \overset{G}{\sim} Q \text{ implies } \forall \text{ PCCS contexts } C[\ ] : C[P] \overset{G}{\sim} C[Q]$$

**Proof:** Again, the proof is by induction on the structure of the context. We present the cases of product and restriction.

**Product:**

$$\alpha.E \xrightarrow{\alpha[1]}_1 E$$

$$E_j \xrightarrow{\alpha[q]}_k E' \quad\Longrightarrow\quad \sum_{i\in I}[p_i]E_i \xrightarrow{\alpha[p_j\cdot q]}_{j.k} E' \quad (j\in I)$$

$$E \xrightarrow{\alpha[p]}_i E',\ F \xrightarrow{\beta[q]}_j F' \quad\Longrightarrow\quad E\times F \xrightarrow{\alpha\beta\,[p\cdot q]}_{(i,j)} E'\times F'$$

$$E \xrightarrow{\alpha[p]}_i E' \quad\Longrightarrow\quad E\restriction A \xrightarrow{\alpha[p/r]}_i E'\restriction A \qquad (\alpha\in A,\ r=\nu_G(E,A))$$

$$E \xrightarrow{\alpha[p]}_i E' \quad\Longrightarrow\quad E[f] \xrightarrow{f(\alpha)[p]}_i E'[f]$$

$$E\{fix_X E/X\} \xrightarrow{\alpha[p]}_i E' \quad\Longrightarrow\quad fix_X E \xrightarrow{\alpha[p]}_i E'$$

Figure 5: Generative Operational Semantics of PCCS

We show $P \overset{G}{\sim} Q$ implies $P \times R \overset{G}{\sim} Q \times R$. Let $(P, Q)$ belong to the generative bisimulation $\mathcal{R}_0$. It is enough to show that the reflexive closure $\mathcal{R}$ of

$$\mathcal{R}_1 = \{\,(P \times R, Q \times R) \mid (P,Q) \in \mathcal{R}_0, R \in Pr\,\}$$

is a generative bisimulation. First note that $\mathcal{R}$ is an equivalence relation. Now suppose $(P \times R, Q \times R) \in \mathcal{R}$ and $\mu_G(P \times R, \gamma, [P' \times R']_\mathcal{R}) = r$. Then, by the structure of $Act$ and the fact that $[P' \times R']_\mathcal{R} = [P']_{\mathcal{R}_0} \times R'$, we must have $\mu_G(P, \alpha, [P']_{\mathcal{R}_0}) = p$, $\mu_G(R, \beta, R') = q$, $\gamma = \alpha \cdot \beta$, and $r = pq$. But then, since $P \overset{G}{\sim} Q$, $\mu_G(Q, \alpha, [P']_{\mathcal{R}_0}) = p$ and $\mu_G(Q \times R, \gamma, [P' \times R']_\mathcal{R}) = r$. $\mathcal{R}$ is therefore a generative bisimulation.

**Restriction:**

We show $P \overset{G}{\sim} Q$ implies $P\restriction A \overset{G}{\sim} Q\restriction A$, for $A \subseteq Act$. Let $(P,Q)$ belong to the generative bisimulation $\mathcal{R}_0$. It is enough to show that the relation

$$\mathcal{R} = \{\,(P\restriction A, Q\restriction A) \mid (P,Q) \in \mathcal{R}_0\,\} \cup Id_{Pr}$$

is a generative bisimulation. This follows almost immediately from the following fact, whose proof is straightforward:

$$P \overset{G}{\sim} Q \text{ implies } \forall A \subseteq Act : \nu_G(P, A) = \nu_G(Q, A)$$

$\square$

# 5 The Stratified Model

The treatments of the reactive and generative models are extended here to the stratified case.

## 5.1 Operational Semantics of PCCS

The stratified operational semantics of PCCS is comprised of two types of transition relations: *action transitions* (as in SCCS) and *probability transitions*. Action transitions are of the form $P \xrightarrow{\alpha} Q$. Probability transitions are of the form $P \xmapsto{p} Q$, meaning that $P$, with probability $p$, can behave as the process $Q$. This separation of action and probability in the stratified model permits the branching structure of the purely probabilistic choices to be captured explicitly. The inference rules for probability transitions appear in Figure 6; the rules for action transitions, being essentially the same as in SCCS [Mil83], are omitted. Note, however, that there is no action rule for process summation since, in the stratified model, the only choice mechanism is probabilistic. This bi-structured approach to operational semantics was (to our knowledge) first presented in [Tof90] to give a semantics for a timed version of CCS.

The inference rules for action and probability transitions define the semantic mapping $\varphi_S$ from $Pr$ to the domain of stratified probabilistic labeled transition systems. Such transition systems are *stochastic* in the sense that for each non-deadlocked state, the sum of the probabilities of its outgoing probability transitions is 1. A state with a probability transition to a dead-

$$\sum_{i \in I} [p_i] E_i \xmapsto{\ p_i\ }_i E_i$$

$$E \xrightarrow{\ \alpha\ } E' \qquad \Longrightarrow \qquad E \xmapsto{\ 1\ }_1 E$$

$$E \xmapsto{\ p\ }_i E', \ F \xmapsto{\ q\ }_j F' \qquad \Longrightarrow \qquad E \times F \xmapsto{\ p \cdot q\ }_{(i,j)} E' \times F'$$

$$E \xmapsto{\ p\ }_i E', \ \nu_S(E', A) \neq 0 \qquad \Longrightarrow \qquad E \restriction A \xmapsto{\ p/\nu_S(E,A)\ }_i E' \restriction A$$

$$E \xmapsto{\ p\ }_i E' \qquad \Longrightarrow \qquad E[f] \xmapsto{\ p\ }_i E'[f]$$

$$E\{fix_X E/X\} \xmapsto{\ p\ }_i E' \qquad \Longrightarrow \qquad fix_X E \xmapsto{\ p\ }_i E'$$

Figure 6: Stratified Operational Semantics of PCCS

locked state corresponds to a substochastic state in the generative model.

Similarly to the reactive and generative cases, the transitions defined in Figure 6 are indexed to distinguish multiple occurrences of the same probability transition. Except for the second and fourth rules, all of the inferences rules for probability transitions are straightforward adaptations of their SCCS counterparts. The second rule is needed to avoid deadlock in a synchronous product that is caused by a difference in depth of the purely probabilistic branching structures of the argument processes. For example, we do not want $(\frac{1}{2}a.0 + \frac{1}{2}b.0) \times c.0$ to deadlock simply because there does not exist a probability transition in the right hand argument. The deadlock is avoided by the second rule, which provides the missing 1-transition.

The fourth rule deals with the restriction operator, and expresses the probability transitions of $E \restriction A$ in terms of the conditional probabilities of $E$ under the assumption $A$. Intuitively, $E \restriction A$ behaves like $E$, where all probability transitions to subexpressions that necessarily require the execution of a restricted action are eliminated. The probabilities associated with these transition are evenly distributed among the remaining probability transitions.

The function $\nu_S$ calculates the *stratified normalization factor*. The condition $\nu_S(E', A) \neq 0$ in the rule premise means that derivative $E'$ of $E$ is capable of performing an action transition from the set $A$ of permitted actions.

$$\nu_S(E, A) = \begin{cases} 1 & \text{if } E \xrightarrow{\ \alpha\ } , \ \alpha \in A \\ 0 & \text{if } E \xrightarrow{\ \beta\ } , \ \beta \notin A; \quad \text{else} \\ \sum_i \{ p_i \mid E \xmapsto{\ p_i\ }_i E_i , \ \nu_S(E_i, A) \neq 0 \} \end{cases}$$

To illustrate the inference rule for restriction, con-

sider the process

$$P = \tfrac{1}{3} a.0 + \tfrac{2}{3} (\tfrac{1}{2} b.0 + \tfrac{1}{2} c.0)$$

In the following, $P$ is "evaluated" with respect to some relevant restriction contexts, resulting in the restriction-free processes on the right-hand side.

$$P \restriction \{b, c\} \ \rightsquigarrow \ 1 (\tfrac{1}{2} b.0 + \tfrac{1}{2} c.0)$$
$$P \restriction \{a, c\} \ \rightsquigarrow \ \tfrac{1}{3} a.0 + \tfrac{2}{3} 1 c.0$$
$$P \restriction \{c\} \ \rightsquigarrow \ 1 (1 c.0)$$

As in the generative case, the inference rule for restriction can be extended in order to preserve the relative probability of deadlock in a substochastic process. This is accomplished by augmenting the definition of $\nu_S$ with the clause

$$\nu_S(E, A) = 1 \quad \text{if } E \xmapsto{\ \ } \text{ and } E \xrightarrow{\ \ }$$

## 5.2 Stratified Bisimulation

Stratified bisimulation is similar to reactive and generative bisimulation in that processes are required to derive stratified bisimulation equivalence classes with equal cumulative probability. However, the separation of probability and action in the stratified operational semantics is reflected in the definition of stratified bisimulation.

To define stratified bisimulation, we need to: (1) define the function that computes the total probability by which a process can behave the same as as any process in a set of processes (the technique is analogous to the one in Definition 1, and thus the details are omitted); (2) lift, in the obvious way, the action relations to *sets* of derivative processes. The *stratified*

*cumulative PDF $\mu_S$* encorporates both (1) and (2) in an integrated fashion. In particular, $\mu_S$ is of the form

$$\mu_S : (Pr \times (Act \cup \{*\}) \times \mathcal{P}(Pr)) \longrightarrow [0,1]$$

where $*$ is a dummy symbol used to mark probability transitions. That is, for $\alpha \in Act$, $\mu_S(P, \alpha, S) \in \{0, 1\}$ indicates whether or not $P$ has an $\alpha$–transition to some process in $S$. Otherwise, $\mu_S(P, *, S) \in [0, 1]$ specifies the total probability by which $P$ may behave the same as any process in $S$.

**Definition 3** *An equivalence relation $\mathcal{R} \subseteq Pr \times Pr$ is a stratified bisimulation if $(P, Q) \in \mathcal{R}$ implies $\forall S \in Pr/\mathcal{R}$, $\forall \alpha \in Act \cup \{*\}$,*

$$\mu_S(P, \alpha, S) = \mu_S(Q, \alpha, S)$$

*Two processes $P, Q$ are stratified bisimulation equivalent (written $P \overset{S}{\sim} Q$) if there exists a stratified bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.*

**Theorem 3 (Congruence)** *For $P, Q \in Pr$:*

$$P \overset{S}{\sim} Q \quad implies \quad \forall \ PCCS \ contexts \ C[\ ]: \quad C[P] \overset{S}{\sim} C[Q]$$

**Proof:** As before the proof is by induction on the structure of $C$. This time, however, each case is broken down into two subcases: cumulative probability transitions and action transitions. The proofs for the probability transitions are similar to the ones presented in Theorems 1 and 2. The proofs for the action transitions are similar to the standard SCCS proofs. □

# 6 Interrelating the Models

In this section, we show by means of the abstraction function $\varphi_{GR}$, that for $PCCS_R$ expressions, the reactive model is an abstraction of the generative model. Likewise, by means of the abstraction function $\varphi_{SG}$, we show that for restriction-free PCCS expressions, the generative model is an abstraction of the stratified model. These abstraction functions are defined in terms of *inter-model abstraction rules* (IMAR$_{GR}$ and IMAR$_{SG}$), which allow the inference of reactive transitions from generative ones, and generative transitions from stratified ones. The situation is summarized in Figure 3 of Section 1.

## 6.1 The Generative to Reactive Abstraction

Let $E, E' \in Pr$ be closed expressions of PCCS. Then IMAR$_{GR}$ is defined by

$$E \xrightarrow{\alpha[p]}_i E' \quad \Longrightarrow \quad E \xrightarrow{\alpha[p/\nu_G(E, \{\alpha\})]}_i E'$$

This rule uses the generative normalization function to convert generative probabilities to reactive ones, thereby abstracting away from the relative probabilities between different actions (see also the discussion in Section 3). We can now define $\varphi_{GR}(\varphi_G(P))$ as the reactive transition system that can be inferred from $P$'s generative transtion system via IMAR$_{GR}$.

**Lemma 1** *For $E, E' \in Pr_R$ and action $\alpha \in Act$,*

*1. $\mu_G(E, \alpha, E') = p > 0$, implies*
   *$\mu_R(E, \alpha, E') = p/\nu_G(E, \{\alpha\})$ and*

*2. $\mu_G(E, \alpha, E') = 0$ implies $\mu_R(E, \alpha, E') = 0$*

**Proof:** By structural induction over $E$. The cases of interest are guarded summation and product. For the former, the key observation is that $r_i = \nu_G(E, \{\alpha_i\})$, where $r_i$ is the normalization factor for the $i$th summand in the reactive axiom for guarded summation (Figure 4).

For product, the result hinges on the fact that, for $E = F \times H$ and $\alpha, \beta \in Act$,

$$\nu_G(F, \{\alpha\}) \cdot \nu_G(H, \{\beta\}) = \nu_G(E, \{(\alpha, \beta)\})$$

□

As an immediate consequence of Lemma 1 we have:

**Theorem 4 (Commutativity)** *For $P \in Pr_R$*

$$\varphi_{GR}(\varphi_G(P)) = \varphi_R(P)$$

Finally, we show that the equivalence induced on the generative model by reactive bisimulation is not a congruence. Consider the PCCS processes

$$P = \tfrac{1}{3}a \cdot 0 + \tfrac{2}{3}b \cdot c \cdot 0 \qquad Q = \tfrac{1}{2}a \cdot 0 + \tfrac{1}{2}b \cdot c \cdot 0$$

For $P, Q$ we have

$$\varphi_{GR}(\varphi_G(P)) \overset{R}{\sim} \varphi_{GR}(\varphi_G(Q))$$

However, the same is not true for $C[P]$ and $C[Q]$, where $C$ is the relabeling $[a \rightarrow a, b \rightarrow a, c \rightarrow c]$. In particular, $\mu_G(\varphi_{GR}(\varphi_G(C[P])), a, [c \cdot 0]_{\underset{\sim}{R}}) = \tfrac{2}{3}$ and $\mu_G(\varphi_{GR}(\varphi_G(C[Q])), a, [c \cdot 0]_{\underset{\sim}{R}}) = \tfrac{1}{2}$.

## 6.2 The Stratified to Generative Abstraction

Let $E, E'$ be PCCS expressions. Then $\text{IMAR}_{SG}$ is given by

$$E \xmapsto{p_0}_{i_0} E_0 \xmapsto{p_1}_{i_1} \cdots \xmapsto{p_n}_{i_n} E_n \xrightarrow{\alpha} E'$$
$$\implies E \xrightarrow{\alpha[p]}_i E'$$

$$\text{where } p = p_0 p_1 \cdots p_n, \ i = i_0 i_1 \cdots i_n$$
$$\text{and } (n = 0 \vee E_{n-1} \neq E_n)$$

The condition $(n = 0 \vee E_{n-1} \neq E_n)$ excludes paths that contain occurrences of a probability-1 transition from an expression that can perform an $\alpha$-action. These probability transitions are artificial in the sense that they have only been introduced for technical purposes (i.e., to deal with synchronous product).

$\text{IMAR}_{SG}$ has the effect of "flattening" trees of probability transitions with action transitions at the leaves, into a single-level structure of generative transitions. Indeed, we show that the generative transition system of a restriction-free PCCS process $P$ is isomorphic to the generative transition system that can be inferred from $P$'s stratified transition system via $\text{IMAR}_{SG}$. For example, let $P = \frac{1}{3}a.\mathbf{0} + \frac{2}{3}(\frac{1}{2}b.\mathbf{0} + \frac{1}{2}c.\mathbf{0})$. Then, by $\text{IMAR}_{SG}$

$$P \xrightarrow{a[\frac{1}{3}]}_1 \mathbf{0} \qquad P \xrightarrow{b[\frac{1}{3}]}_{2\cdot1} \mathbf{0} \qquad P \xrightarrow{c[\frac{1}{3}]}_{2\cdot2} \mathbf{0}$$

Except for the transition indices, these are precisely the transitions of $P$ in the generative model.

To derive the stratified-to-generative commutativity result, we need to extend the stratified cumulative PDF from single transitions to paths of transitions: for $n \geq 0$,

$$\mu_S(E_0 E_1 \cdots E_n, *) = \prod_{1 \leq i \leq n} \mu_S(E_{i-1}, *, \{E_i\})$$

By convention, the empty product is 1. Second, we define the notion of a path, possibly with probability transitions, from $E$ to $E'$ ending in an $\alpha$-transition, $\alpha \in Act$.

**Definition 4** *For expressions $E_0, E_1, \ldots, E_n, E'$ and action $\alpha \in Act$,*

$$E_0 \cdots E_n \in PATHS(E, \alpha, E') \text{ iff}$$

$$(E = E_0) \wedge (n = 0 \vee E_{n-1} \neq E_n) \wedge E_n \xrightarrow{\alpha} E'$$

As for $\text{IMAR}_{SG}$, the second conjunct in the definition excludes paths that contain artificial probability-1 transitions.

**Lemma 2** *For restriction-free PCCS expressions $E, E'$ and action $\alpha \in Act$,*

$$\mu_G(E, \alpha, \{E'\}) = \sum_{\pi \in PATHS(E, \alpha, E')} \mu_S(\pi, *)$$

**Proof:** The proof is based on the algebraic properties of $\mu_G$ and PATHS, which can be specified equationally. For example, the equations for summation are:

$$\mu_G(\sum_{i \in I}[p_i]E_i, \alpha, \{E\}) = \sum_{i \in I} p_i \cdot \mu_G(E_i, \alpha, \{E\})$$

$$PATHS(\sum_{i \in I}[p_i]E_i, \alpha, E) =$$

$$\bigcup_i \{(\sum_{i \in I}[p_i]E_i)P \mid P \in PATHS(E_i, \alpha, E)\}$$

For product we have

$$\mu_G(E \times F, \gamma, \{E' \times F'\}) =$$
$$\mu_G(E, \alpha, \{E'\}) \cdot \mu_G(F, \beta, \{F'\})$$

$$PATHS(E \times F, \gamma, E' \times F') =$$
$$PATHS(E, \alpha, E') \times PATHS(F, \beta, F')$$

where $\alpha \cdot \beta = \gamma$. Together with similar equations for the other operators and with the defining properties of $\mu_S$, a proof by structural induction over $E$ is now straightforward. $\square$

As an immediate consequence of this lemma, we obtain the following commutativity result:

**Theorem 5 (Commutativity)**
*Let $P \in Pr$ be a restriction-free PCCS process. Then*

$$\varphi_{SG}(\varphi_S(P)) = \varphi_G(P)$$

Theorem 5 does not hold for arbitrary PCCS processes. Consider the process

$$P = \frac{1}{3}a.\mathbf{0} + \frac{2}{3}(\frac{1}{2}b.\mathbf{0} + \frac{1}{2}c.\mathbf{0}) \upharpoonright \{a, b\}$$

$\varphi_G(P)$ is equal to $\frac{1}{2}a.\mathbf{0} + \frac{1}{2}b.\mathbf{0}$ while $\varphi_{SG}(\varphi_S(P))$ is equal to $\frac{1}{3}a.\mathbf{0} + \frac{2}{3}b.\mathbf{0}$.

Finally, we show that the equivalence induced on the stratified model by generative bisimulation is not a congruence. Consider processes $Sc$ and $Sc'$ of Section 1 (the scheduler specifications). We have $\varphi_{SG}(\varphi_S(Sc)) \overset{G}{\sim} \varphi_{SG}(\varphi_S(Sc'))$ but, as discussed in Section 1, $\varphi_{SG}(\varphi_S(Sc \upharpoonright \{a, b\})) \overset{G}{\not\sim} \varphi_{SG}(\varphi_S(Sc' \upharpoonright \{a, b\}))$.

# 7 Conclusions

In this paper we have examined three models of probabilistic processes. In so doing, we have seen that generative bisimulation ($\overset{G}{\sim}$) is not a congruence in the stratified model, while stratified bisimulation ($\overset{S}{\sim}$) is. However, $\overset{S}{\sim}$ is not the *largest* congruence contained in $\overset{G}{\sim}$ (it is too fine). For example, consider $P = [1][1]a$ and $Q = [1]a$. $\varphi_S(P) \overset{S}{\not\sim} \varphi_S(Q)$ yet $\varphi_{SG}(\varphi_S(C[P])) \overset{G}{\sim} \varphi_{SG}(\varphi_S(C[Q]))$, for any context $C[]$.

It is interesting, therefore, to ask what is the largest congruence contained in $\overset{G}{\sim}$. We can show that, in terms of its distinguishing strength, the following equivalence relation falls strictly between $\overset{G}{\sim}$ and $\overset{S}{\sim}$, and is still a congruence in the stratified model.

**Definition 5** *An equivalence relation $\mathcal{R} \subseteq Pr \times Pr$ is a mixed bisimulation if $(P, Q) \in \mathcal{R}$ implies $\forall S \in Pr/\mathcal{R}$,*

$$\mu_S(P, *, S) = \mu_S(Q, *, S) \quad \text{and}$$
$$\forall \alpha \in Act, \mu'_G(P, \alpha, S) = \mu'_G(Q, \alpha, S)$$

*where $\mu'_G$ (for expressions with restriction) is defined in the style of the rhs of the formula in Lemma 2. Two processes $P, Q$ are mixed bisimulation equivalent (written $P \overset{M}{\sim} Q$) if there exists a mixed bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.*

Mixed bisimulation essentially allows an $\alpha$-transition in one process to be matched by an $\alpha$-transition preceded by a number of probability-1 transitions in the other process (the second clause). At the same time, probability-1 transitions may be significant in a product context, and must therefore be taken into account (the first clause). We close with the following:

**Conjecture (Full Abstraction)** In the stratified model, $\overset{M}{\sim}$ is the largest congruence contained in $\overset{G}{\sim}$.

# References

[BM89] B. Bloom and A. R. Meyer. A remark on bisimulation between probabilistic processes. In Meyer and Tsailin, editors, *Logik at Botik*, Springer-Verlag, 1989.

[Chr89] Ivan Christoff. Distinguishing probabilistic processes through testing. In *Proceedings of Nordic Workshop on Program Correctness*, Uppsala, Sweden, 1989.

[GJS90] 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*, IFIP TC 2, Sea of Gallilee, Israel, April 1990.

[JP89] C. Jones and G. D. Plotkin. A probabilistic powerdomain of evaluations. In *Proceedings of 4th Annual Symposium on Logic in Computer Science*, 1989.

[LS89] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. In *Proceedings of 16th Annual ACM Symposium on Principles of Programming Languages*, 1989.

[Mil80] R. Milner. *A Calculus of Communicating Systems*. Volume 92 of *Lecture Notes in Computer Science*, Springer-Verlag, 1980.

[Mil83] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[Mil89] R. Milner. *Communication and Concurrency*. International Series in Computer Science, Prentice Hall, 1989.

[Par81] D. M. R. Park. Concurrency and automata on infinite sequences. In *Proceedings of 5th G.I. Conference on Theoretical Computer Science*, pages 167–183, Springer-Verlag, 1981.

[Plo81] G. D. Plotkin. *A Structural Approach to Operational Semantics*. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[Pnu85] A. Pnueli. Linear and branching structures in the semantics and logics of reactive systems. In *Proceedings of 12th ICALP*, pages 15–32, Springer Verlag, 1985.

[SS90] S. A. Smolka and B. Steffen. Priority as extremal probability. 1990. To appear.

[Tof90] Chris M. N. Tofts. *Proof Methods and Pragmatics for Parallel Programming*. PhD thesis, LFCS, University of Edinburgh, 1990.