URL: http://www.elsevier.nl/locate/entcs/volume7.html 20 pages

Domain Equations for Probabilistic Processes (Extended Abstract)

Christel Baier

Fakultät für Mathematik & Informatik
Universität Mannheim
68131 Mannheim, Germany

Marta Kwiatkowska [1]

School of Computer Science
University of Birmingham
Edgbaston, Birmingham B15 2TT, UK

Abstract

In this paper we consider Milner's calculus *CCS* enriched by a probabilistic choice operator. The calculus is given operational semantics based on probabilistic transition systems. We define operational notions of preorder and equivalence as probabilistic extensions of the simulation preorder and the bisimulation equivalence respectively. We extend existing category-theoretic techniques for solving domain equations to the probabilistic case and give two denotational semantics for the calculus. The first, "smooth", semantic model arises as a solution of a domain equation involving the probabilistic powerdomain and solved in the category $CONT_\perp$ of continuous domains. The second model also involves appropriately restricted probabilistic powerdomain, but is constructed in the category *CUM* of complete ultra-metric spaces, and hence is necessarily "discrete". We show that the domain-theoretic semantics is fully abstract with respect to the simulation preorder, and that the metric semantics is fully abstract with respect to bisimulation.

1 Introduction

To deal with situations involving uncertainty, which arise, for example, in randomized distributed systems, fault-tolerant systems and communication protocols, it has become necessary to extend existing techniques to the probabilistic case. Concurrent calculi such as *CCS* [35] and *CSP* [20] can successfully serve as high-level specification languages for compositional design and analysis of distributed systems; their theoretical foundations, i.e. the underlying transition systems, associated operational notions and logics, as well as the

[1] Supported in part by EPSRC grant GR/K42028.

©1997 Published by Elsevier Science B. V.

denotational models, are considered to be well understood, which makes them eminently suitable to serve as basis for formulating probabilistic calculi.

In this paper we consider the (full) calculus of *CCS* [35], which we enrich with the probabilistic choice operator, instead of probabilistic choice replacing the usual non-deterministic choice. Non-determinism is used to model situations where two possibilities arise, and it is not known which would be taken. While genuinely useful in cases such as underspecification, possibly to be removed by further refinements, non-deterministic choice is not amenable when *quantitative* analysis is needed, for example, when the (probabilistic) distribution of possible continuations is known. By means of the probabilistic choice we add to *CCS* it is possible to state that the process, having executed an action, becomes another process *with probability* λ. Our framework allows to independently combine both the quantitative and qualitative forms of choice. This means that it is possible in a given state to non-deterministically choose between two or more probabilistic distributions on the successor states. The distinction between non-deterministic and probabilistic choices is necessary since concurrent systems contain states that are inherently non-deterministic. The non-determinism may arise e.g. from the asynchronicity of certain subprocesses. It is conventioned that a scheduler decides which of the subprocesses performs its next step, and that no (probabilistic) assumptions about these decisions, which any scheduler has to obey, are or can be made.

Research on probabilistic concurrent calculi has so far concentrated mainly on the operational notions and the associated logics. [28] defined probabilistic transition systems and probabilistic bisimulation. [40] introduced the notion of probabilistic simulation. Probabilistic extensions of *CCS* are introduced in [14,18,42,47,46] and Probabilistic *CSP* in [30]. While the calculi in [14,42] are based on *SCCS* [34] and replace the non-deterministic choice operator by a probabilistic choice operator, like [47,46] we deal with a calculus that is based on (asynchronous) *CCS* and allows for both non-deterministic and probabilistic choice. A "metric" for ϵ-bisimulation is introduced in [14], but, unlike our approach, does not satisfy the axioms of a metric.

In this paper we focus on extending to the probabilistic setting existing category-theoretic techniques for providing a (non-probabilistic) calculus with fully abstract denotational semantics. Denotational semantics, being compositional, provides the theory that underpins system decomposition; in addition, if fully abstract, i.e., if the inherent order or equality in the model precisely corresponds to the operational (pre)order or equivalence, the denotational semantics can provide additional insight into the nature of operational notions, and eventually serve as an intermediate link between the operational semantics and an appropriate logic. In contrast to the fully abstract models for probabilistic processes, which are based on testing equivalences (see e.g. [10,24,45]), we take as basis Probabilistic *CCS* together with the probabilistic bisimulation equivalence [28] and the probabilistic simulation preorder [40]. We include both the domain-theoretic and the metric-space approach in our investigations. Starting from domain equations for (non-probabilistic) synchronization trees, we derive domain equations involving the probabilistic powerdomain construction which, when solved respectively in the categories $CONT_\perp$ of continuous domains and CUM of complete ultra-metric spaces, give rise to semantic "domains" of probabilistic processes. We are able to show that the inherent order on the domain of processes corresponds to the probabilistic simulation, while the equality on the metric space to the probabilistic bisimulation.

The paper is organised as follows. Section 2 introduces probabilistic transition systems, together with probabilistic simulation and bisimulation. In Sections 3 we give domain equations for probabilistic processes and the denotational models they give rise to. Finally, in Section 4 we discuss operational and denotational semantics for Probabilistic *CCS* and state the corresponding full abstraction results. The full version of this paper is available as [5].

2 Probabilistic transition systems

In this section we give a brief overview of the notions of probabilistic distributions, probabilistic transition systems and the probabilistic bisimulation and simulation.

Notation 2.1 *Let X be a set. We write $\mathcal{D}_1(X)$ to denote the set of all (probabilistic)* distributions *on X, i.e. the set of all functions $\mu : X \to [0,1]$ such that $\mu(x) \neq 0$ for at most countably many $x \in X$ and $\sum_{x\in X} \mu(x) = 1$. A distribution μ is called* simple *iff there exists $z \in X$ with $\mu(z) = 1$ (and $\mu(x) = 0$ for all $x \neq z$). For $z \in X$, μ_z^1 denotes the unique simple distribution on X with $\mu_z^1(z) = 1$. We extend a distribution μ to a function $2^X \to [0,1]$, $U \mapsto \mu[U]$, which assigns to each subset U of X the probability $\mu[U] \stackrel{def}{=} \sum_{x\in U} \mu(x)$.*

Our model of probabilistic transition systems is based on the "simple probabilistic automaton" of [40]. It generalizes the "concurrent Markov chains" considered e.g. in [12,43] and the "alternating model" of [18]. In what follows, we suppose *Act* to be a nonempty and countable set of actions.

Definition 2.2 *A* probabilistic transition system *is a tuple $(S, \to)$ where S is a set of states and $\to \subseteq S \times Act \times \mathcal{D}_1(S)$ such that for each $s \in S$ and $\alpha \in Act$ the set $\{(\alpha,\mu) : s \stackrel{\alpha}{\to} \mu\}$ is finite.*[2] *A* probabilistic process *is a tuple $(S, \to, s)$ consisting of a probabilistic transition system $(S, \to)$ and an* initial state *$s \in S$.*

Intuitively, the transition relation $\to$ represents the non-deterministic alternatives in each state: given a state s, a scheduler chooses a transition $s \stackrel{\alpha}{\to} \mu$. Then, the action α is performed and with probability $\mu(t)$ the state t is reached afterwards. Terminal states are those for which $\{(\alpha,\mu) : s \stackrel{\alpha}{\to} \mu\}$ is empty.

Non-probabilistic transition systems (where the transition relation $\to$ is a subset of $S \times Act \times S$) arise as special cases of probabilitistic transition systems by identifying each "non-probabilistic" transition $s \stackrel{\alpha}{\to} t$ with the "probabilistic" transition $s \stackrel{\alpha}{\to} \mu_t^1$.

[2] We write $s \stackrel{\alpha}{\to} \mu$ instead of $(s, \alpha, \mu) \in \to$.

Definition 2.3 ***(cf. [28,40])*** *Let $\mathcal{P}_i = (S_i, \rightarrow_i, s_i)$, $i = 1, 2$, be probabilistic processes where w.l.o.g. $S_1 \cap S_2 = \emptyset$. A* bisimulation *for $(\mathcal{P}_1, \mathcal{P}_2)$ is an equivalence relation R of $S_1 \cup S_2$ which contains (s_1, s_2) and such that for all $(t_1, t_2) \in R$: If $t_1 \xrightarrow{\alpha}_1 \mu_1$ then there is a transition $t_2 \xrightarrow{\alpha}_2 \mu_2$ with $\mu_1[A] = \mu_2[A]$ for all $A \in (S_1 \cup S_2)/R$.*[3] *$\mathcal{P}_1$ and $\mathcal{P}_2$ are called* bisimilar *(denoted by $\mathcal{P}_1 \sim \mathcal{P}_2$) iff there exists a bisimulation for $(\mathcal{P}_1, \mathcal{P}_2)$.*

[23] give an alternative characterization of bisimulation equivalence which uses weight functions.

Definition 2.4 *Let X_1, X_2 be sets, $R \subseteq X_1 \times X_2$ and $\mu_i \in \mathcal{D}_1(X_i)$, $i = 1, 2$. A* weight function *for (μ_1, μ_2) w.r.t. R is a function $\delta : X_1 \times X_2 \rightarrow [0, 1]$ such that for all $x_1 \in X_1$, $x_2 \in X_2$:*

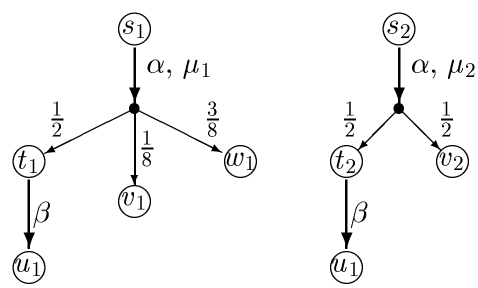
1. *$\delta(y_1, y_2) \neq 0$ for at most countably many $(y_1, y_2) \in X_1 \times X_2$.*
2. *$\sum_{y \in X_2} \delta(x_1, y) = \mu_1(x_1)$, $\sum_{y \in X_1} \delta(y, x_2) = \mu_2(x_2)$*
3. *If $\delta(x_1, x_2) > 0$ then $(x_1, x_2) \in R$.*

We write $\mu_1 \preceq_R \mu_2$ if there exists a weight function for (μ_1, μ_2) w.r.t. R.

Lemma 2.5 ***(cf. [23])*** *Let $\mathcal{P}_i = (S_i, \rightarrow_i, s_i)$, $i = 1, 2$, be probabilistic processes. Then, $\mathcal{P}_1 \sim \mathcal{P}_2$ iff there exists a relation $R \subseteq S_1 \times S_2$ which contains (s_1, s_2) and satisfies:*

- *If $(t_1, t_2) \in R$ and $t_1 \xrightarrow{\alpha}_1 \mu_1$ then there is a transition $t_2 \xrightarrow{\alpha}_2 \mu_2$ with $\mu_1 \preceq_R \mu_2$.*
- *If $(t_1, t_2) \in R$ and $t_2 \xrightarrow{\alpha}_2 \mu_2$ then there is a transition $t_1 \xrightarrow{\alpha}_1 \mu_1$ with $\mu_1 \preceq_R \mu_2$.*

Example 2.6 *The probabilistic processes shown on the right are bisimilar. A weight function for (μ_1, μ_2) w.r.t. $\sim$ can be obtained by combining parts of bisimilar states. Clearly, $t_1 \sim t_2$ and v_1, $w_1 \sim v_2$. Thus, $\delta(t_1, t_2) = 1/2$, $\delta(v_1, v_2) = 1/8$, $\delta(w_1, v_2) = 3/8$ (and $\delta(\cdot) = 0$ in all other cases) yields a weight function for (μ_1, μ_2) w.r.t. $\sim$.* ■



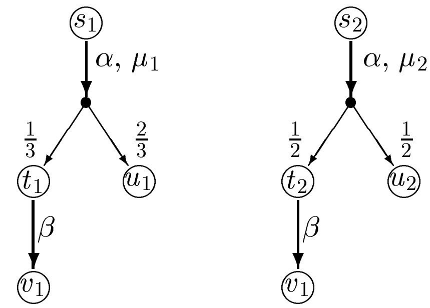
As in the non-probabilistic case, simulation can be viewed as "uni-directional bisimulation" in the sense that a process $\mathcal{P}_2$ "simulates" another process $\mathcal{P}_1$ if each step of $\mathcal{P}_1$ can be "simulated" by a step of $\mathcal{P}_2$. In that case, $\mathcal{P}_1$ can be viewed as an "implementation" of $\mathcal{P}_2$ as each step of $\mathcal{P}_1$ is "allowed" by the "specification" $\mathcal{P}_2$. The notion of a simulation can be obtained from the notion of a bisimulation by dropping the symmetry. Formally, in the characterization of a bisimulation in Lemma 2.5 we give up the requirement for R to be an equivalence relation.

Definition 2.7 ***(cf. [40])*** *Let $\mathcal{P}_i = (S_i, \rightarrow_i, s_i)$, $i = 1, 2$, be probabilistic processes. W.l.o.g. $S_1 \cap S_2 = \emptyset$. A* simulation *for $(\mathcal{P}_1, \mathcal{P}_2)$ is a subset R of $S_1 \times S_2$ such that $(s_1, s_2) \in R$ and for all $(t_1, t_2) \in R$: If $t_1 \xrightarrow{\alpha}_1 \mu_1$ then there is a*

[3] If X is a set and R an equivalence on X then X/R denotes the quotient space, i.e. the set of equivalence classes w.r.t. R.

transition $t_2 \xrightarrow{\alpha}_2 \mu_2$ *with* $\mu_1 \preceq_R \mu_2$. *We say* $\mathcal{P}_1$ implements $\mathcal{P}_2$ *and* $\mathcal{P}_2$ simulates $\mathcal{P}_1$ *(denoted by* $\mathcal{P}_1 \sqsubseteq_{sim} \mathcal{P}_2$*) iff there exists a simulation for* $(\mathcal{P}_1, \mathcal{P}_2)$. $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *are called* similar *(denoted by* $\mathcal{P}_1 \sim_{sim} \mathcal{P}_2$*) iff* $\mathcal{P}_1 \sqsubseteq_{sim} \mathcal{P}_2$ *and* $\mathcal{P}_2 \sqsubseteq_{sim} \mathcal{P}_1$.

Example 2.8 *For the probabilistic processes* $\mathcal{P}_1$, $\mathcal{P}_2$ *(with initial states* s_1 *and* s_2 *respectively) shown on the right, we have* $\mathcal{P}_1 \sqsubseteq_{sim} \mathcal{P}_2$. *Clearly,* $u_1 \sqsubseteq_{sim} u_2$ *and* $u_1, t_1 \sqsubseteq_{sim} t_2$. *A weight function for* (μ_1, μ_2) *can be obtained by combining certain parts of* t_1 *(*u_1*) with certain parts of* t_2 *(*u_2 *and* t_2*). The weight function* δ *for* (μ_1, μ_2) *w.r.t.* $\sqsubseteq_{sim}$ *is given by:* $\delta(t_1, t_2) = 1/3$, $\delta(u_1, t_2) = 1/6$, $\delta(u_1, u_2) = 1/2$. ■



In the non-probabilistic case the above notion of a (bi-)simulation agrees with Milner's notion of a (bi-)simulation [35].[4] It is easy to see that $\sqsubseteq_{sim}$ is a preorder and its kernel $\sim_{sim}$ is strictly coarser than bisimulation equivalence $\sim$. In the case of reactive processes, i.e. probabilistic processes $(S, \rightarrow, s)$ where for each state t and action α there is at most one transition $t \xrightarrow{\alpha} \mu$, simulation and bisimulation equivalence coincide. This can be viewed as the probabilistic counterpart to the well-known result that simulation equivalence and bisimulation equivalence are the same for "deterministic" (non-probabilistic) transition systems.

3 Domain equations for probabilistic processes

We now turn our attention to the construction of semantic domains which can serve as fully abstract denotational models for languages with probabilistic transitions and recursion. Full abstraction w.r.t. bisimulation means that the semantic domain identifies exactly those probabilistic processes which are bisimilar. Full abstraction w.r.t. simulation requires that the semantic domain be equipped with a partial order which reflects the simulation preorder $\sqsubseteq_{sim}$. As in the non-probabilistic case, where fully abstract models for bisimulation and simulation are given on domains that are defined by recursive domain equations (e.g. [9,17,8,1,4]), we define the semantic domains by recursive equations soluble in suitable categories of complete metric spaces and domains by the methods of [38], resp. [2]. The central idea is to represent a probabilistic process by a *set of pairs* (α, μ) where α is an action and μ a distribution on probabilistic processes. More formally, our aim is to solve domain equations of the form $X \cong \wp_*(Act \times \mathcal{D}_1(X))$ where $\wp_*(\cdot)$ denotes a suitable powerdomain construction. Unfortunately, the standard methods of [31,32,3,33,38] in the metric case and [29,41,2] for partial order fail for the above equation since

---

[4] Note that the only weight function for (μ_1, μ_2), where $\mu_1 = \mu^1_{t_1}$, $\mu_2 = \mu^1_{t_2}$ are simple distributions, is $\delta(u_1, u_2) = 0$ if $(u_1, u_2) \neq (t_1, t_2)$ and $\delta(t_1, t_2) = 1$. Hence, if $\mathcal{P}_i = (S_i, \rightarrow_i, s_i)$ are non-probabilistic processes, $i = 1, 2$, and $R \subseteq S_1 \times S_2$ then R is a simulation in the sense of Definition 2.7 if and only if R is a simulation in the sense of Milner [35].

the operator $\mathcal{D}_1(\cdot)$ does not preserve completeness (cf. Remarks 3.6 and 3.10). Nevertheless, the equation $X \cong \wp_{fin}(Act \times \mathcal{D}_1(X))$ has a final solution in *SET*, the category of sets and functions. Here, $\wp_{fin}(\cdot)$ denotes the collection of finite subsets of $(\cdot)$. In order to obtain denotational models of probabilistic processes we switch from $\mathcal{D}_1(\cdot)$ to the probabilistic powerdomain $\mathcal{E}_1(\cdot)$ of probabilistic evaluations in the sense of [22], that is, we solve domain equations of the form $X \cong \wp_*(Act \times \mathcal{E}_1(X))$ by the methods of [3,38] and [41,2]. In the metric approach we work with the powerdomain operator $\wp_{comp}(\cdot)$ which assigns to each metric space M the set of compact subsets of M. When working with domains we use the Hoare powerdomain $\wp_{Hoare}(\cdot)$ since the language does not have divergence.

3.1 The set-theoretic approach

$\mathcal{D}_1$ can be viewed as an endofunctor of the category *SET*. Here, for each function $f : X \to Y$, we define $\mathcal{D}_1(f) : \mathcal{D}_1(X) \to \mathcal{D}_1(Y)$ by $\mathcal{D}_1(f)(\mu)(y) \stackrel{def}{=} \mu[f^{-1}(y)]$. If X is a set then $\wp_{fin}(X)$ denotes the set of finite subsets of X and $\mathcal{A}ct(X) \stackrel{def}{=} Act \times X$. If $f : X \to Y$ is a function then we define $\wp_{fin}(f) : \wp_{fin}(X) \to \wp_{fin}(Y)$ and $\mathcal{A}ct(f) : \mathcal{A}ct(X) \to \mathcal{A}ct(Y)$ by $\wp_{fin}(f)(U) \stackrel{def}{=} f(U)$, $\mathcal{A}ct(f)(\alpha, x) \stackrel{def}{=} (\alpha, f(x))$. Hence, $\wp_{fin}$ and $\mathcal{A}ct$ can be considered as endofunctors of *SET*.

Theorem 3.1 *The set $\boldsymbol{P}$ of bisimulation equivalence classes of probabilistic processes*[5] *is the final coalgebra (and hence the final fixed point) of the functor $\wp_{fin} \circ \mathcal{A}ct \circ \mathcal{D}_1 : SET \to SET$.*

Since $\boldsymbol{P}$ is the final coalgebra we get a "final semantics" in the sense of [38,37]. Let $\mathcal{P} = (S, \to, s)$ be a probabilistic process. Then, (S, k) is a coalgebra of $\mathcal{K} \stackrel{def}{=} \wp_{fin} \circ \mathcal{A}ct \circ \mathcal{D}_1$ where $k : S \to \mathcal{K}(S)$ is given by $k(t) \stackrel{def}{=} \{(\alpha, \mu) : t \stackrel{\alpha}{\to} \mu\}$. Let $F : S \to \boldsymbol{P}$ be the unique function with $\mathcal{K}(F) \circ k = e \circ F$ where $e : \boldsymbol{P} \to \mathcal{K}(\boldsymbol{P})$ is the unique isomorphism such that $(\boldsymbol{P}, e)$ is the final coalgebra of $\mathcal{K}$. Then, the "final semantics" $\varphi(\mathcal{P})$ of $\mathcal{P}$ in $\boldsymbol{P}$ is the element $F(s)$. It is easy to see that $\varphi(\mathcal{P})$ is the bisimulation equivalence class of $\mathcal{P}$ and that φ is fully abstract in the sense that it identifies two states if and only if they are bisimilar and that it preserves the simulation preorder $\sqsubseteq_{sim}$. We associate with $\boldsymbol{P}$ a probabilistic transition system $(\boldsymbol{P}, \to)$ as follows. If $\mathcal{P} = (S, \to, s)$ is a probabilistic process then $\varphi(\mathcal{P}) \stackrel{\alpha}{\to} \nu$ iff there is a transition $s \stackrel{\alpha}{\to} \mu$ where $\nu = \mathcal{D}_1(\varphi \circ \pi)(\mu)$. Here, π is the function which associates with each state $t \in S$ the probabilistic process $\pi(t) = (S, \to, t)$.

Example 3.2 *Consider the probabilistic processes $\mathcal{P}_1$ and $\mathcal{P}_2$ of Example 2.8. The "final semantics" $\varphi(\mathcal{P}_1)$ and $\varphi(\mathcal{P}_2)$ of $\mathcal{P}_1$ and $\mathcal{P}_2$ (as elements of $\boldsymbol{P} =$*

[5] In order to see that $\boldsymbol{P}$ is really a set consider a fixed set S_0 of cardinality ω and define $\boldsymbol{Q}$ to be the set of bisimulation equivalence classes of all probabilistic processes whose states belong to S_0. Then, each probabilistic process is bisimilar to some probabilistic process whose states belong to S_0 (note that we only consider finitely branching systems). Hence, each object of $\boldsymbol{P}$ corresponds to an object of $\boldsymbol{Q}$, and thus $\boldsymbol{P}$ and $\boldsymbol{Q}$ can be identified.

$\wp_{fin}(\mathcal{A}ct(\mathcal{D}_1(\boldsymbol{P}))))$ are $\varphi(\mathcal{P}_1) = \{(\alpha, \nu_1)\}$, $\varphi(\mathcal{P}_2) = \{(\alpha, \nu_2)\}$ where $\nu_1(x) = 1/3$, $\nu_1(\emptyset) = 2/3$, $\nu_2(x) = \nu_2(\emptyset) = 1/2$ and $x = \{(\beta, \mu^1_\emptyset)\}$. ■

3.2 Probabilistic powerdomains

Following [22] we define an *evaluation* on a topological space X as a function $E : \Omega(X) \to \mathbb{R}_{\geq 0}$ on the open sets $\Omega(X)$ of X such that:

1. Whenever $U \subseteq U'$ then $E(U) \leq E(U')$.
2. $E(U \cap U') + E(U \cup U') = E(U) + E(U')$.
3. $E(\emptyset) = 0$.

An evaluation E on X is said to be *continuous* iff $E(\bigcup_i U_i) = \sup_i E(U_i)$ for each directed family $(U_i)_i$ of open sets. $\mathcal{E}(X)$ denotes the set of all continuous evaluations on X. Continuity of evaluations is needed to ensure the asymmetry axioms of order and metric. E is called a *probabilistic evaluation* iff $E(X) = 1$. The *probabilistic powerdomain* $\mathcal{E}_1(X)$ of X is the set of probabilistic continuous evaluations on X. In what follows, by an evaluation we mean a probabilistic continuous evaluation. If X, X' are topological spaces and $f : X \to X'$ is a continuous function then we define the operator $\mathcal{E}_1(f) : \mathcal{E}_1(X) \to \mathcal{E}_1(X')$ by $\mathcal{E}_1(f)(E)(U) \stackrel{def}{=} E(f^{-1}(U))$. Thus, $\mathcal{E}_1$ can be considered as a functor $TOP \to SET$ where TOP denotes the category of topological spaces and continuous functions. The following lemma shows that for each topological space X, the distributions on X induce evaluations on X.

Lemma 3.3 *Let X be a topological space and $\mu \in \mathcal{D}_1(X)$. Then, $E_\mu : \Omega(X) \to [0,1]$ where $E_\mu(U) \stackrel{def}{=} \mu[U]$ is an evaluation on X.*

Whether the function $\mathcal{D}_1(X) \to \mathcal{E}_1(X)$, $\mu \mapsto E_\mu$, is injective (and hence can be considered as an embedding) depends on the underlying topology on X. For example, for the topology $\{\emptyset, X\}$ on a set X which contains at least two points it is easy to see that this function is not injective. In our applications – where X is equipped with an ultrametric or a complete partial order – $\mathcal{D}_1(X)$ can be considered as a subspace of $\mathcal{E}_1(X)$.

Remark 3.4 *Let $eval_X : \mathcal{D}_1(X) \to \mathcal{E}_1(X)$ be the function $eval_X(\mu) \stackrel{def}{=} E_\mu$. Then, $eval_Y \circ \mathcal{D}_1(f) = \mathcal{E}_1(f) \circ eval_X$ for every continuous function $f : X \to Y$, i.e. for each distribution $\mu \in \mathcal{D}_1(X)$, $E_{\mathcal{D}_1(f)(\mu)} = \mathcal{E}_1(f)(E_\mu)$. Hence, eval is a natural transformation $\mathcal{D}_1 \to \mathcal{E}_1$ where $\mathcal{D}_1$ is considered as a functor $SET \to TOP$ (where $\mathcal{D}_1(X)$ is equipped with the discrete topology).* ■

3.3 The domain-theoretic approach

Using the methods of [2,41] we obtain a domain-theoretic equation for probabilistic processes.

We suppose the reader to be familiar with basic notions concerning domain theory and metric spaces; for details see e.g. [2,15]. By a *domain* we mean a directed-complete partial order with bottom. Along the lines of [22] we define a locally continuous endofunctor on $CONT_\perp$, the category of continuous domains and strict and continuous functions, which assigns to each continuous domain D its probabilistic powerdomain $\mathcal{E}_1(D)$.[6] Here, we suppose a dcpo D to be equipped with the Scott-topology, that is, $\Omega(D)$ consists of all subsets U of D where U is upward-closed and $D \setminus U$ is lub-closed.[7] In contrast to the approach of [22] we work with the normalised powerdomain $\mathcal{E}_1(\cdot)$ of probabilistic evaluations instead of $\mathcal{E}(\cdot)$. If D is a dcpo then $\mathcal{E}_1(D)$ is a dcpo where we use the same partial order as in [22]: $E_1 \sqsubseteq E_2$ iff $E_1(U) \leq E_2(U)$ for all $U \in \Omega(D)$. The bottom element $\bot_{\mathcal{E}_1(D)}$ of $\mathcal{E}_1(D)$ is $E_{\mu^1_{\bot_D}}$ (where $\bot_D$ is the bottom element of D). Thus, $\bot_{\mathcal{E}_1(D)}$ is different from the bottom element of $\mathcal{E}(D)$. If $(E_i)_i$ is a directed family of evaluations then the least upper bound $E = \bigsqcup E_i$ in $\mathcal{E}_1(D)$ is given by $E(U) = \sup_i E_i(U)$ (which is at the same time the least upper bound of $(E_i)_i$ in $\mathcal{E}(D)$). Extending the argument in Section 4.2 of [21] it is immediate that the functor $\mathcal{E}_1 : CONT_\bot \to CONT_\bot$ is locally continuous, and hence continuous.

The following theorem shows that, for each dcpo D, the function $\mathcal{D}_1(D) \to \mathcal{E}_1(D)$, $\mu \mapsto E_\mu$, is an order preserving embedding where $\mathcal{D}_1(D)$ is equipped with the ordering $\preceq$ which is given by $\mu_1 \preceq \mu_2$ iff there exists a weight function δ for (μ_1, μ_2) w.r.t. the partial order $\sqsubseteq$ on D.[8] Thus, each distribution μ on D can be identified with the evaluation E_μ.

Theorem 3.5 *If D is a dcpo then $\preceq$ is a partial order on $\mathcal{D}_1(D)$. Moreover, $\mu_1 \preceq \mu_2$ iff $E_{\mu_1} \sqsubseteq E_{\mu_2}$ for all μ_1, $\mu_2 \in \mathcal{D}_1(D)$.*

Remark 3.6 *In general, $\mathcal{D}_1(D)$ is not complete. Consider the dcpo $D = \{0,1\}^\infty$ of all (finite or infinite) words built from 0 and 1 equipped with the prefix ordering. Let μ_k be the distribution with $\mu_k(x) = 1/2^k$ if x is a word of length k, $\mu_k(x) = 0$ in all other cases. Then, $(\mu_k)_{k \geq 1}$ is a monotone sequence in $\mathcal{D}_1(D)$ which does not have an upper bound in $\mathcal{D}_1(D)$.* ■

We now demonstrate that the equation $X \cong \wp_{Hoare}(Act \times \mathcal{E}_1(X))$ can indeed be solved with the help of the fixed point theorem of [2,41]. Here, $\wp_{Hoare}(\cdot)$ denotes the Hoare powerdomain of $(\cdot)$. If D is a dcpo then $\wp_{Hoare}(D)$ is the dcpo of nonempty and Scott-closed subsets of D ordered by inclusion.[9] For a nonempty subset A of D, A^* denotes the Scott-closure of A, i.e. the smallest Scott-closed subset containing A. We define $\emptyset^* \stackrel{def}{=} \{\bot\}$. $\mathcal{A}ct_\bot(D)$ denotes the dcpo $\{\bot\} \cup Act \times D$ ordered by $\bot \sqsubseteq x$ for all x and $(\alpha, x) \sqsubseteq (\beta, y)$ iff $\alpha = \beta \wedge x \sqsubseteq y$. If D, D' are dcpo's and $f : D \to D'$ is strict and continuous then $\wp_{Hoare}(f) : \wp_{Hoare}(D) \to \wp_{Hoare}(D')$

[6] Alternatively, we could work with $DCPO_\bot$, the category of dcpo's and strict and continuous functions.

[7] A subset X of a dcpo is upward-closed iff $\{y \in D : x \sqsubseteq y\} \subseteq X$ for all $x \in X$. X is lub-closed iff $\bigsqcup Y \in X$ for each directed subset Y of X.

[8] With the notations of Definition 2.4, $\preceq \, = \, \preceq_{\sqsubseteq}$.

[9] Recall that Scott-closedness means lub- and downward-closedness.

and $\mathcal{A}ct_\perp(f) : \mathcal{A}ct_\perp(D) \to \mathcal{A}ct_\perp(D')$ are given by

$$\wp_{Hoare}(f)(A) \stackrel{def}{=} f(A)^*, \quad \mathcal{A}ct_\perp(f)(x) \stackrel{def}{=} \begin{cases} \perp & : \text{if } x = \perp, \\ (\alpha, f(y)) & : \text{if } x = (\alpha, y). \end{cases}$$

It is easy to see that $\wp_{Hoare}$ and $\mathcal{A}ct_\perp$ can be viewed as endofunctors on $CONT_\perp$, and that they are locally continuous. Hence, the functor $\mathcal{F}_{cont} \stackrel{def}{=} \wp_{Hoare} \circ \mathcal{A}ct_\perp \circ \mathcal{E}_1 : CONT_\perp \to CONT_\perp$ is locally continuous. By the results of [2], $\mathcal{F}_{cont}$ has an initial fixed point which is simultaneously the initial algebra and final coalgebra of $\mathcal{F}_{cont}$ as shown by [38]. Let $\boldsymbol{D}$ denote the initial fixed point of $\mathcal{F}_{cont}$; in other words, $\boldsymbol{D}$ is the initial solution of

$$D \;\cong\; \wp_{Hoare}(\{\perp\} \cup Act \times \mathcal{E}_1(D)).$$

In what follows we deal with the isomorphism as an equality, i.e. if $(\boldsymbol{D}, j)$ is the initial fixed point of $\mathcal{F}_{cont}$ then we suppose $\boldsymbol{D} = \mathcal{F}_{cont}(\boldsymbol{D})$ and $j = id_{\boldsymbol{D}}$. Note that the partial order on $\boldsymbol{D}$ is the inclusion. The bottom element $\perp_{\boldsymbol{D}}$ in $\boldsymbol{D}$ is $\{\perp\}$ where $\perp$ denotes the bottom element in $\mathcal{A}ct_\perp(\boldsymbol{D})$. If $(x_i)_{i \in I}$ is a directed family of elements in $\boldsymbol{D} \setminus \{\perp_{\boldsymbol{D}}\}$ then the least upper bound $\bigsqcup x_i$ is $(\bigcup x_i)^*$, the Scott-closure of $\bigcup x_i$ in $\mathcal{A}ct_\perp(\boldsymbol{D})$. The following theorem shows the connection between $\boldsymbol{P}$ and $\boldsymbol{D}$.

Theorem 3.7 *There exists a unique function $\imath_{\boldsymbol{D}} : \boldsymbol{P} \to \boldsymbol{D}$ with $\imath_{\boldsymbol{D}}(x) = \{(\alpha, E_{\mathcal{D}_1(\imath_{\boldsymbol{D}})(\nu)}) : x \stackrel{\alpha}{\to} \nu\}^*$. This function $\imath_{\boldsymbol{D}}$ satisfies:*

$$\varphi(\mathcal{P}_1) \sqsubseteq_{sim} \varphi(\mathcal{P}_2) \text{ iff } \imath_{\boldsymbol{D}}(\varphi(\mathcal{P}_1)) \subseteq \imath_{\boldsymbol{D}}(\varphi(\mathcal{P}_2))$$

for all probabilistic processes $\mathcal{P}_1$, $\mathcal{P}_2$.

By Theorem 3.7, the "final semantics" of Section 3 yields a semantics $\mathcal{P} \mapsto \imath_{\boldsymbol{D}}(\varphi(\mathcal{P}))$ on $\boldsymbol{D}$ which is fully abstract w.r.t. simulation. Thus, the element $\imath_{\boldsymbol{D}}(\varphi(\mathcal{P}))$ can be identified with the simulation equivalence class of $\mathcal{P}$.

Example 3.8 *The probabilistic processes $\mathcal{P}_1$ and $\mathcal{P}_2$ of Example 3.2 and 2.8 are represented in $\boldsymbol{D}$ by $\imath_{\boldsymbol{D}}(\varphi(\mathcal{P}_i)) = \{(\alpha, E_{\nu_i})\}^* = \{\perp\} \cup \{(\alpha, E) : E \in \mathbf{E}_{p_i}\}$, $i = 1, 2$, where $\nu_1 = \mu_{\frac{2}{3}}$, $\nu_2 = \mu_{\frac{1}{2}}$, $p_1 = \frac{2}{3}$, $p_2 = \frac{1}{2}$, $\mathbf{E}_p = \{E_{\mu_q} : p \le q \le 1\}$ where μ_q is the unique distribution on $\boldsymbol{D}$ with $\mu_q(\perp_{\boldsymbol{D}}) = q$ and $\mu_q(\{(\beta, E_\mu)\}^*) = 1 - q$ where $\mu = \mu^1_{\perp_{\boldsymbol{D}}}$.* ■

3.4 The metric approach

We show that the equation $X \;\cong\; \wp_{comp}(Act \times \mathcal{E}_1(M))$ can be solved in the category of complete ultrametric spaces by the methods of [38,3] where $\wp_{comp}(\cdot)$ denotes the collection of all compact subsets of $(\cdot)$.

We assume familiarity with basic notions of metric spaces, see e.g. [13]. We suppose that the underlying distance on an ultrametric space – which we always denote by d – satisfies $d \le 1$. The topology on an ultrametric space M is given by taking the open balls $B(x, \rho)$, $x \in M$, $\rho > 0$, as its basic opens. $\mathcal{B}$ denotes the set of all open balls, $\mathcal{B}_\rho$ the set of open balls with radius $\ge \rho$, i.e. open balls of the form $B(x,r)$ where $r \geq \rho$. *CUM* denotes the category of complete ultrametric spaces and non-expansive functions.

We show that the probabilistic powerdomain construction $\mathcal{E}_1(\cdot)$ can be considered as an endofunctor on *CUM* which is locally non-expansive in the sense of [38]. Recall that, for each ultrametric space M, $\Omega(M)$ is the collection of all subsets U of M such that, for each $x \in M$, there is some $B \in \mathcal{B}$ with $x \in B \subseteq U$. It is easy to see that for each distribution μ on M, $\mu(x) = \inf\{E_\mu(U) : x \in U \in \Omega(M)\}$. Thus, whenever μ, $\mu' \in \mathcal{D}_1(M)$ with $E_\mu = E_{\mu'}$ then $\mu = \mu'$. Hence, the set $\mathcal{D}_1(M)$ of distributions on M can be considered as a subspace of $\mathcal{E}_1(M)$. We suppose $\mathcal{E}_1(M)$ to be endowed with the distance

$$d(E_1, E_2) \stackrel{def}{=} \inf \{ \rho > 0 \ : \ E_1(B) = E_2(B) \ \ \forall \ B \in \mathcal{B}_\rho \}.$$

Theorem 3.9 *If M is a complete ultrametric space then $\mathcal{E}_1(M)$ is a complete ultrametric space. In this case, $\mathcal{E}_1(M)$ is the completion of $\mathcal{D}_1(M)$ (considered as a subspace of $\mathcal{E}_1(M)$).*

Whenever $f : M \to M'$ is a non-expansive function between ultrametric spaces M and M' then the function $\mathcal{E}_1(f)$ is non-expansive. Hence, $\mathcal{E}_1$ can be considered as an endofunctor on *CUM*. It is easy to see that the functor $\mathcal{E}_1 : CUM \to CUM$ is locally non-expansive in the sense of [38].

Remark 3.10 *Similarly to Remark 3.6 the set $\{0,1\}^\infty$ equipped with the natural distance $d(x,y) \stackrel{def}{=} \inf\{1/2^n : x[n] \neq y[n]\}$ (where $z[n]$ denotes the n-th prefix of z) yields an example for a complete metric space M where $\mathcal{D}_1(M)$ is not complete (i.e. $\mathcal{D}_1(M)$ viewed as a subspace of $\mathcal{E}_1(M)$ is not closed). Consider the sequence $(\mu_k)_k$ which is defined as in Remark 3.6. Then, $d(\mu_k, \mu_i) \leq 1/2^i$ for all $k \geq i$. Thus, $(\mu_k)_k$ is a Cauchy sequence in $\mathcal{D}_1(M)$ which does not have a limit in $\mathcal{D}_1(M)$.* ■

If M is a complete ultrametric space then $\wp_{comp}(M)$ equipped with the Hausdorff metric

$$d(X,Y) \stackrel{def}{=} \max \left\{ \sup_{x \in X} d(x,Y), \ \ \sup_{y \in Y} d(y,X) \right\}$$

(where $d(w,Z) \stackrel{def}{=} \inf_{z \in Z} d(w,z)$) is a complete ultrametric space (see [26]). If $f : M \to M'$ is a non-expansive function then $\wp_{comp}(f) : \wp_{comp}(M) \to \wp_{comp}(M')$, $\wp_{comp}(f)(X) \stackrel{def}{=} f(X)$ is non-expansive. Hence, we get a functor $\wp_{comp} : CUM \to CUM$ which is locally non-expansive (see [38]). We define the functor $\mathcal{A}ct_{\frac{1}{2}} : CUM \to CUM$ as follows. If M is a complete ultrametric space then $\mathcal{A}ct_{\frac{1}{2}}(M)$ is $Act \times M$ endowed with the distance

$$d((\alpha,x), \ (\beta,y)\ ) \stackrel{def}{=} \begin{cases} 1 & : \text{if } \alpha \neq \beta \\ \frac{1}{2} \cdot d(x,y) & : \text{if } \alpha = \beta. \end{cases}$$

If $f : M \to M'$ is a non-expansive function between complete ultrametric spaces then we define $\mathcal{A}ct_{\frac{1}{2}}(f)$ to be the function $\mathcal{A}ct(f)$. Then, the functor

$\mathcal{F}_{cum} \stackrel{def}{=} \wp_{comp} \circ \mathcal{A}ct_{\frac{1}{2}} \circ \mathcal{E}_1 : CUM \to CUM$ is locally contracting in the sense of [38]. By the results of [38], $\mathcal{F}_{cum}$ has a unique fixed point which is at the same time the final coalgebra and initial algebra. [10] Let $\boldsymbol{M}$ denote the unique fixed point of $\mathcal{F}_{cum}$, i.e.

$$\boldsymbol{M} \;\cong\; \wp_{comp}(Act \times \frac{1}{2}\mathcal{E}_1(\boldsymbol{M})).$$

In what follows, we deal with the isomorphism as an equality, i.e. if $(\boldsymbol{M}, j)$ is the unique fixed point of $\mathcal{F}_{cum}$ then we suppose $\boldsymbol{M} = \mathcal{F}_{cum}(\boldsymbol{M}))$ and $j = id_{\boldsymbol{M}}$.

Theorem 3.11 *$\boldsymbol{P}$ is a dense subset of $\boldsymbol{M}$. More precisely, there exists a unique function $\imath_{\boldsymbol{M}} : \boldsymbol{P} \to \boldsymbol{M}$ such that for $x \in \boldsymbol{M}$, $\imath_{\boldsymbol{M}}(x) = \{(\alpha, E_{\mathcal{D}_1(\imath_{\boldsymbol{M}})(\nu)}) : x \stackrel{\alpha}{\to} \nu\}$. This function $\imath_{\boldsymbol{M}}$ is injective and $\imath_{\boldsymbol{M}}(\boldsymbol{P})$ is a dense subspace of $\boldsymbol{M}$. Moreover, $\mathcal{P}_1 \sim \mathcal{P}_2$ iff $\imath_{\boldsymbol{M}}(\varphi(\mathcal{P}_1)) = \imath_{\boldsymbol{M}}(\varphi(\mathcal{P}_2))$ for all probabilistic processes $\mathcal{P}_1$, $\mathcal{P}_2$.*

The result that $\boldsymbol{P}$ can be viewed a dense subspace of $\boldsymbol{M}$ (Theorem 3.11) should be contrasted to the domain-theoretic setting where $\imath_{\boldsymbol{D}}(\boldsymbol{P})$ is *not* a basis of $\boldsymbol{D}$. For instance, it can be shown that the element $\{(\alpha, E)\}^*$ is not of the form $\bigsqcup X$ for some directed subset X of $\imath_{\boldsymbol{D}}(\boldsymbol{P})$. Here, the evaluation E is defined as follows. Let $\mu = \mu^1_{\perp_{\boldsymbol{D}}}$, $y = \{(\gamma, E_\mu)\}^*$, $x_p = \{(\beta, E_{\mu_p})\}^*$, where μ_p is the unique distribution on $\boldsymbol{D}$ with $\mu_p(y) = p$ and $\mu_p(\perp_{\boldsymbol{D}}) = 1 - p$, and $E(U) \stackrel{def}{=} \sup\{p : x_{1-p} \in U\}$ where $\sup \emptyset \stackrel{def}{=} 0$.

4 Semantics for Probabilistic *CCS*

In this section we consider a probabilistic extension of Milner's *CCS* [35] which is based on the calculi of [47,46,18] that allow for non-deterministic and probabilistic choice. In our calculus, called *PCCS* (Probabilistic *CCS*), the usual prefixing operator $s \mapsto \alpha.s$ is replaced with the (guarded) *probabilistic choice* operator $\sum_i \alpha_{p_i}.s_i$ where p_i is a real number between 0 and 1 denoting the probability that after performing α the above process becomes s_i (i ranges over an at most countable index set I). The above syntax derives from Milner's notion of an infinite summation $\sum_v \alpha_v.s_v$, where α_v denotes the input action which transmits the value v to a program variable x, and the behaviour of the continuation process might be dependent on the value v. The difference between our operator and Milner's infinite sum is that we suppose that there exists a distribution on the possible values v; p_i is then the sum of the probabilities for input values v for which the continuation process s_v equals s_i. Instead of Milner's process equations, for convenience we use declarations to model recursive behaviour.

In what follows, *Var* is a set of (process) variables and *Act* a set of atomic actions which contains an internal action τ (representing internal computa-

[10] [38] deal with the category *CMS* of complete metric spaces instead of the subcategory *CUM*. It is easy to see that the fixed point theorem of [38], and also those of [3], carry over to the category *CUM*.

tions of a process which are not visible for the environment) and which is equipped with a function $Act \to Act$, $\alpha \mapsto \overline{\alpha}$, where $\overline{\tau} = \tau$ and $\overline{\overline{\alpha}} = \alpha$. $\overline{\alpha}$ is called the complementary action of α. Synchronization of processes is only possible by performing complementary actions α and $\overline{\alpha}$, $\alpha \neq \tau$. The result of a synchronization is the internal action τ. The syntax of *PCCS* (Probabilistic *CCS*) is as follows.[11] *Statements* are built from the production system

$$s \;::=\; nil \mid x \mid \sum_{i\in I} \alpha_{p_i}.s_i \mid s_1 \oplus s_2 \mid s_1 \parallel s_2 \mid s[\lambda] \mid s \setminus L$$

where $\alpha \in Act$, $x \in Var$, L is a subset of $Act \setminus \{\tau\}$ with $\overline{L} = L$ (where $\overline{L} = \{\overline{\alpha} : \alpha \in L\}$), $\lambda : Act \to Act$ is a relabelling function (i.e. $\lambda(\overline{\alpha}) = \overline{\lambda(\alpha)}$ and $\lambda(\tau) = \tau$) and I is a nonempty countable indexing set and $(p_i)_{i\in I}$ a family of real numbers $p_i \in [0,1]$ such that $\sum_{i\in I} p_i = 1$. For finite indexing set $I = \{i_1, \ldots, i_n\}$ we also write $\alpha_{p_{i_1}}.s_{i_1} + \ldots + \alpha_{p_{i_n}}.s_{i_n}$ instead of $\sum_{i\in I} \alpha_{p_i}.s_i$. *Guarded statements* are built from the production system

$$g \;::=\; nil \mid \sum_{i\in I} \alpha_{p_i}.s_i \mid g_1 \oplus g_2 \mid g_1 \parallel g_2 \mid g[\lambda] \mid g \setminus L$$

Stmt denotes the set of all statements. A *declaration* is a function $\sigma : Var \to Stmt$. A declaration σ is called guarded iff $\sigma(x)$ is guarded for all $x \in Var$. A *program* is a pair $\mathcal{P} =< \sigma, s >$ consisting of a declaration σ and a statement s. *PCCS* denotes the set of all programs, and *GPCCS* the subset of guarded programs, i.e. all programs $< \sigma, s >$ where σ is a guarded declaration. The intended meaning of programs $\mathcal{P} =< \sigma, s >$ is that the behaviour of $\mathcal{P}$ is given by the statement s where each occurrence of a variable x in s is considered as a recursive call of the procedure $\sigma(x)$. In other words, a declaration σ corresponds to the family of process equations $\sigma(x) = x$, $x \in Var$. *nil* stands for a process which does not perform any action. $\oplus$ models nondeterministic choice (denoted by $+$ in *CCS*): $s_1 \oplus s_2$ either behaves like s_1 or like s_2. $\parallel$ denotes the usual *CCS* parallel composition. The operators $s \mapsto s \setminus L$, $s \mapsto s[\lambda]$ model restriction and relabelling: $s \setminus L$ behaves like s as long as s does not perform an action $\alpha \in L$. $s[\lambda]$ behaves like s where each action α is replaced by $\lambda(\alpha)$.

4.1 Operational semantics for Probabilistic CCS

We give an operational semantics for *PCCS* based on probabilistic transition systems. Let σ be a declaration. We define a probabilistic transition system $(Stmt, \to_\sigma)$ where $\to_\sigma \;\subseteq\; Stmt \times Act \times \mathcal{D}_1(Stmt)$ is the smallest relation satisfying the following rules. (Here, we write $s \xrightarrow{\alpha}_\sigma \mu$ instead of $(s, \alpha, \mu) \in \to_\sigma$.)

1. $\displaystyle\sum_{i\in I} \alpha_{p_i}.s_i \xrightarrow{\alpha}_\sigma \mu$ where $\displaystyle\mu(s) = \sum_{\substack{i\in I\\ s_i = s}} p_i$

[11] Our language *PCCS* is contained in the calculus considered in [47] which allows probabilistic choice between arbitrary processes, and also the extended version of [46] which adds aspects of time and uses a generalized probabilistic choice operator that prescribes intervals of probabilities rather than the exact probabilities.

2. $s_1 \oplus s_2 \stackrel{\alpha}{\rightarrow}_\sigma \mu$ if $s_1 \stackrel{\alpha}{\rightarrow}_\sigma \mu$ or $s_2 \stackrel{\alpha}{\rightarrow}_\sigma \mu$

3. $s_1 \parallel s_2 \stackrel{\alpha}{\rightarrow}_\sigma \mu$ if one of the following three conditions is satisfied:

   (i) $s_1 \stackrel{\alpha}{\rightarrow}_\sigma \mu_1$ and $\mu(s) = \begin{cases} \mu_1(s_1') : \text{if } s = s_1' \parallel s_2 \\ 0 \quad : \text{otherwise} \end{cases}$

   (ii) $s_2 \stackrel{\alpha}{\rightarrow}_\sigma \mu_2$ and $\mu(s) = \begin{cases} \mu_2(s_2') : \text{if } s = s_1 \parallel s_2' \\ 0 \quad : \text{otherwise} \end{cases}$

   (iii) $\alpha = \tau$, $s_1 \stackrel{\beta}{\rightarrow}_\sigma \mu_1$ and $s_2 \stackrel{\overline{\beta}}{\rightarrow}_\sigma \mu_2$ for some $\beta \in \mathit{Act} \setminus \{\tau\}$ with

$$\mu(s) = \begin{cases} \mu_1(s_1') \cdot \mu_2(s_2') : \text{if } s = s_1' \parallel s_2' \\ 0 \quad : \text{otherwise} \end{cases}$$

4. $s[\lambda] \stackrel{\alpha}{\rightarrow}_\sigma \mu$ if $s \stackrel{\beta}{\rightarrow}_\sigma \mu'$, $\lambda(\beta) = \alpha$ and $\mu(s) = \begin{cases} \mu'(s') : \text{if } s = s'[\lambda] \\ 0 \quad : \text{otherwise} \end{cases}$

5. $s \setminus L \stackrel{\alpha}{\rightarrow}_\sigma \mu$ if $s \stackrel{\alpha}{\rightarrow}_\sigma \mu'$, $\alpha \notin L$ and $\mu(s) = \begin{cases} \mu'(s') : \text{if } s = s' \setminus L \\ 0 \quad : \text{otherwise} \end{cases}$

6. $x \stackrel{\alpha}{\rightarrow}_\sigma \mu$ if $\sigma(x) \stackrel{\alpha}{\rightarrow}_\sigma \mu$

The operational semantics assigns to each program $< \sigma, s >$ the probabilistic process $(\mathit{Stmt}, \rightarrow_\sigma, s)$. In what follows, we identify each program $\mathcal{P}$ with its operational meaning.

Example 4.1 *The operational semantics of the programs $< \sigma, s_1 >$ and $< \sigma, s_2 >$ where*

$$s_1 = \alpha_{\frac{1}{3}}.\beta_1.nil + \alpha_{\frac{2}{3}}.nil, \quad s_2 = \alpha_{\frac{1}{2}}.\beta_1.nil + \alpha_{\frac{1}{2}}.nil$$

(and σ is an arbitrary declaration) are the probabilistic processes of Example 2.8. The picture below shows the operational semantics of the recursive program $\mathcal{P}_0 =< s_0, \sigma_0 >$ where $s_0 = s_1 \oplus x$ and $\sigma_0(x) = \alpha_1.x$ (and s_1 is as above).

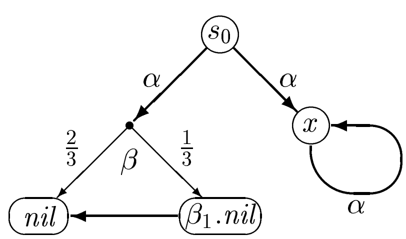


Figure 1 shows the pointed transition systems of the program $< \sigma, (s_1 \| s_2) \setminus L >$ where

$$s_1 \stackrel{def}{=} \alpha_{\frac{1}{4}}.\beta_1.nil + \alpha_{\frac{3}{4}}.\overline{\alpha}_1.nil, \; s_2 \stackrel{def}{=} (\overline{\alpha}_{\frac{1}{3}}.\overline{\beta}_1.nil + \overline{\alpha}_{\frac{2}{3}}.\alpha_1.nil) \oplus \gamma_1.nil,$$

$L \stackrel{def}{=} \{\alpha, \overline{\alpha}, \beta, \overline{\beta}\}$ and σ is an arbitrary declaration. In the picture, $t_1 \otimes t_2$ stands for $(t_1 \| t_2) \setminus L$. The γ-transition of $s_1 \otimes s_2$ represents the case where

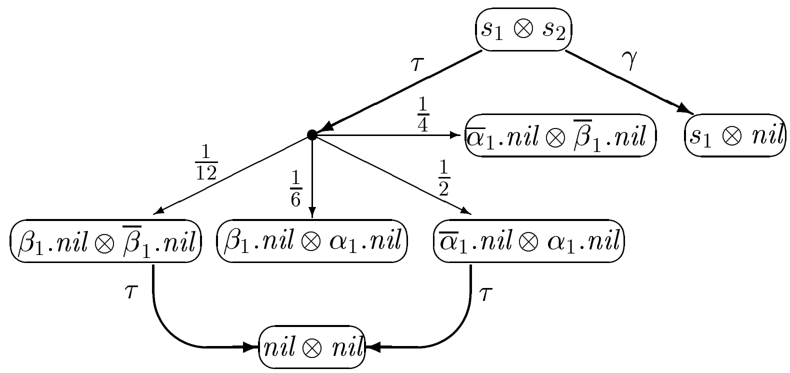


Fig. 1. The operational semantics of $< \sigma, (s_1 \| s_2) \setminus \{\alpha, \beta\} >$

the γ*-transition of* s_2 *is chosen non-deterministically. Thus,* $s_1 \otimes s_2$ *can make a* γ*-move where* s_1 *does not participate, i.e. does not change its local state. The* τ*-transition of* $s_1 \otimes s_2$ *stands for the synchronization of* α *and* $\overline{\alpha}$*. For instance, with probability* $\frac{1}{4} \cdot \frac{1}{3} = \frac{1}{12}$*,* s_1 *moves to the local state* $\beta_1.nil$ *and* s_2 *to* $\overline{\beta}_1.nil$*. In the global states* $s_1 \otimes nil$*,* $\beta_1.nil \otimes \alpha_1.nil$ *and* $\overline{\alpha}_1.nil \otimes \overline{\beta}_1.nil$ *no actions are possible because of the restriction operator while in the global states* $\beta_1.nil \otimes \overline{\beta}_1.nil$ *and* $\overline{\alpha}_1.nil \otimes \alpha_1.nil$ *further synchronizations take place.* ■

4.2 Denotational semantics on $\boldsymbol{M}$ *and* $\boldsymbol{D}$

We give a denotational semantics for *GPCCS* on $\boldsymbol{M}$, which is fully abstract w.r.t. bisimulation equivalence, and a denotational semantics for *PCCS* on $\boldsymbol{D}$, which is fully abstract w.r.t. the simulation preorder. For this we need non-expansive/contracting semantic operators on $\boldsymbol{M}$ and continuous semantic operators on $\boldsymbol{D}$.

In the sequel, $X = \boldsymbol{M}$ or $X = \boldsymbol{D}$. We use the star notation for subsets of $Act \times \boldsymbol{M}$ and for subsets of $\{\bot\} \cup Act \times \boldsymbol{D}$. When dealing with $\boldsymbol{D}$, A^* denotes the Scott-closure of A (if $A \neq \emptyset$) and $\emptyset^* = \bot_{\boldsymbol{D}}$, as before. When dealing with $\boldsymbol{M}$, we put $A^* \stackrel{def}{=} A$.

The process *nil* is modelled by $\emptyset$ in $\boldsymbol{M}$ and by $\bot_{\boldsymbol{D}} = \{\bot\}$ in $\boldsymbol{D}$. Nondeterministic choice on $\boldsymbol{M}$ and $\boldsymbol{D}$ is modelled by set-theoretic union (which, of course, is continuous as an operator on $\boldsymbol{D}$, and non-expansive when considered as an operator on $\boldsymbol{M}$).

(Guarded) probabilistic choice: Let $\alpha \in Act$ and $(p_i)_{i \in I}$ be a countable family of real numbers $p_i > 0$ with $\sum_{i \in I} p_i = 1$. Let $(x_i)_{i \in I}$ be a family in X. We put

$$\sum_{i \in I} \alpha_{p_i}.x_i \stackrel{def}{=} \{(\alpha, E_\mu)\}^* \quad \text{where } \mu \in \mathcal{D}_1(X) \text{ is given by } \mu(x) \stackrel{def}{=} \sum_{\substack{i \in I \\ x_i = x}} p_i.$$

Clearly, the operator $\sum$ is contracting on $\boldsymbol{M}$ and continuous on $\boldsymbol{D}$.

We define semantic operators for modelling restriction, relabelling and parallelism as fixed points of suitable operators. This reflects the recursive nature of restriction, relabelling and parallelism (cf. Milner's expansion law [35] for parallelism).

Relabelling and restriction: If λ is a relabelling function and $L \subseteq Act \setminus \{\tau\}$ with $L = \overline{L}$ then then F_λ^X, $F_L^X : (X \to X) \to (X \to X)$ are given by

$$F_\lambda^X(f)(x) \stackrel{def}{=} \{\ (\lambda(\alpha), \mathcal{E}_1(f)(E))\ :\ (\alpha, E) \in x\ \}^*,$$
$$F_L^X(f)(x) \stackrel{def}{=} \{\ (\alpha, \mathcal{E}_1(f)(E))\ :\ (\alpha, E) \in x,\ \alpha \notin L\ \}^*.$$

The unique/least fixed points of F_λ^X and F_L^X (whose existence can be shown using Banach's and Tarski's fixed point theorems) yield non-expansive, resp. continuous, semantic operators $x \mapsto x[\lambda]$ and $x \mapsto x \setminus L$, on $\boldsymbol{M}$ and $\boldsymbol{D}$ for modelling relabelling and restriction.

Parallel composition: In the definition of the semantic parallel operators we use a result of Heckmann [19] stating the existence of a unique evaluation $E_1 * E_2 \in \mathcal{E}_1(X \times X)$ with $E_1(U) * E_2(V) = (E_1 * E_2)(U \times V)$ for all opens U, V of X and for all evaluations E_1, $E_2 \in \mathcal{E}_1(X)$. Here, X is an arbitrary topological space and $X \times X$ the product space (with the sets $U \times V$, U, $V \in \Omega(X)$, as its basic opens).[12] Clearly, if μ_1, $\mu_2 \in \mathcal{D}_1(X)$ then $E_{\mu_1} * E_{\mu_2} = E_{\mu_1 * \mu_2}$ where $\mu_1 * \mu_2 \in \mathcal{D}_1(X \times X)$ is given by $(\mu_1 * \mu_2)(x_1, x_2) \stackrel{def}{=} \mu_1(x_1) \cdot \mu_2(x_2)$. It is shown in [19] that for every dcpo D the product $*$ is a continuous operator $\mathcal{E}_1(D) \times \mathcal{E}_1(D) \to \mathcal{E}_1(D \times D)$.[13] We use the following notations. If $f : X \times X \to X$ is a function and x_0, $y_0 \in X$ then we define $f(x_0, \cdot), f(\cdot, y_0) : X \to X$ by $f(x_0, \cdot)(y) \stackrel{def}{=} f(x_0, y)$ and $f(\cdot, y_0)(x) \stackrel{def}{=} f(x, y_0)$. We define $F_\|^X : (X \times X \to X) \to (X \times X \to X)$ by

$$F_1^X(f)(x,y) \stackrel{def}{=} \{(\alpha, \mathcal{E}_1(f(\cdot, y))(E)) : (\alpha, E) \in x\},$$
$$F_2^X(f)(x,y) \stackrel{def}{=} \{(\alpha, \mathcal{E}_1(f(x, \cdot))(E)) : (\alpha, E) \in y\},$$
$$F_{syn}^X(f)(x,y) \stackrel{def}{=} \{(\tau, \mathcal{E}_1(f)(E_1 * E_2)) : \exists \alpha \neq \tau.(\alpha, E_1) \in x \wedge (\overline{\alpha}, E_2) \in y\},$$
$$F_\|^X(f)(x,y) \stackrel{def}{=} \left(F_1^X(f)(x,y) \cup F_2^X(f)(x,y) \cup F_{syn}^X(f)(x,y)\right)^*.$$

Using Banach's and Tarski's fixed point theorem respectively it can be shown that $F_\|^X$ has a unique/least fixed point which yields an operator $X \times X \to X$, $(x, y) \mapsto x \| y$. $\|$ is non-expansive as an operator on $X = \boldsymbol{M}$ and continuous as an operator on $X = \boldsymbol{D}$.

By standard methods, see e.g. [39,9,8,6], we extend the above to recursive programs $< \sigma, s >$ and obtain denotational semantics $Me^{\boldsymbol{D}} : PCCS \to \boldsymbol{D}$ and $Me^{\boldsymbol{M}}\ :\ GPCCS \to \boldsymbol{M}$ as follows:

$$Me^{\boldsymbol{D}}(< \sigma, s >) \stackrel{def}{=} f_\sigma^{\boldsymbol{D}}(s), \quad Me^{\boldsymbol{M}}(< \sigma, s >) \stackrel{def}{=} f_\sigma^{\boldsymbol{M}}(s)$$

[12] [19] deals with $\mathcal{E}(\cdot)$ instead of $\mathcal{E}_1(\cdot)$, but it is clear that $(E_1 * E_2)(X \times X) = 1$ for E_1, $E_2 \in \mathcal{E}_1(X)$. Thus, $E_1 * E_2 \in \mathcal{E}_1(X \times X)$.

[13] In the metric approach, where $\mathcal{E}_1(\boldsymbol{M})$ is a completion of $\mathcal{D}_1(\boldsymbol{M})$, the product of evaluations can be defined without using the result of [19], as the operator $* : \mathcal{E}_1(\boldsymbol{M}) \times \mathcal{E}_1(\boldsymbol{M}) \to \mathcal{E}_1(\boldsymbol{M} \times \boldsymbol{M})$ can be defined as the canonical extension of the non-expansive operator $\mathcal{D}_1(\boldsymbol{M}) \times \mathcal{D}_1(\boldsymbol{M}) \to \mathcal{D}_1(\boldsymbol{M} \times \boldsymbol{M})$, $(\mu_1, \mu_2) \mapsto \mu_1 * \mu_2$.

where $f_\sigma^{\boldsymbol{D}}(s)$ is the least, and $f_\sigma^{\boldsymbol{M}}(s)$ the unique, fixed point of an appropriately defined operator $\Psi_\sigma^X : (\mathit{Stmt} \to X) \to (\mathit{Stmt} \to X)$ where X is $\boldsymbol{D}$ or $\boldsymbol{M}$ respectively. The interested reader is referred to the full paper [5] for further details.

We now illustrate how recursion and the parallel operator is dealt with by means of examples.

Example 4.2 *The denotational semantics of the recursive program $\mathcal{P}_0$ of Example 4.1 in $\boldsymbol{M}$ is $Me^{\boldsymbol{M}}(\mathcal{P}_0) = x \cup y$ where y is $\{(\alpha, E_\mu)\}$ and μ the unique distribution with $\mu(\emptyset) = 2/3$, $\mu(y_\beta) = 1/3$, $y_\beta = \{(\beta, E_{\mu^1_\emptyset})\}$ and x is the unique element in $\boldsymbol{M}$ such that $x = \{(\alpha, E_{\mu^1_x})\}$. Similarly, $Me^{\boldsymbol{D}}(\mathcal{P}_0) = x' \cup y'$ where y' is $\{(\alpha, E_{\mu'})\}^*$ and μ' the unique distribution with $\mu'(\bot_{\boldsymbol{D}}) = 2/3$, $\mu'(y'_\beta) = 1/3$, $y'_\beta = \{(\beta, E_{\mu^1_{\bot_{\boldsymbol{D}}}})\}^*$ and x' is the unique element in $\boldsymbol{D}$ such that $x' = \{(\alpha, E_{\mu^1_{x'}})\}^*$.*[14] ■

Theorem 4.3 *The denotational semantics $Me^{\boldsymbol{M}} : \mathit{GPCCS} \to \boldsymbol{M}$ and $Me^{\boldsymbol{D}} : \mathit{PCCS} \to \boldsymbol{D}$ and are fully abstract w.r.t. bisimulation and simulation respectively. I.e. for all $\mathcal{P}$, $\mathcal{P}_1$, $\mathcal{P}_2 \in \mathit{GPCCS}$ and $\mathcal{P}'$, $\mathcal{P}'_1$, $\mathcal{P}'_2 \in \mathit{PCCS}$:*

(a) $Me^{\boldsymbol{M}}(\mathcal{P}) = \varphi(\mathcal{P})$

(b) $\mathcal{P}_1 \sim \mathcal{P}_2$ *iff* $Me^{\boldsymbol{M}}(\mathcal{P}_1) = Me^{\boldsymbol{M}}(\mathcal{P}_2)$

(c) $Me^{\boldsymbol{D}}(\mathcal{P}') = \imath_{\boldsymbol{D}}(\varphi(\mathcal{P}'))$

(d) $\mathcal{P}'_1 \sqsubseteq_{sim} \mathcal{P}'_2$ *iff* $Me^{\boldsymbol{D}}(\mathcal{P}'_1) \subseteq Me^{\boldsymbol{D}}(\mathcal{P}'_2)$.

Here, $\boldsymbol{P}$ is considered as a subspace of $\boldsymbol{M}$ (Theorem 3.11), $\imath_{\boldsymbol{D}}$ is as in Theorem 3.7 and φ denotes the "final semantics" for probabilistic processes.

Proof (Sketch) Using the statements (0)-(6) of above it can be shown by structural induction on the syntax of $s \in \mathit{Stmt}$ that $\Psi_\sigma^X(\imath_X \circ \varphi_\sigma)(s) = \varphi_\sigma(s)$. In the case $X = \boldsymbol{M}$, from the uniqueness of $f_\sigma^{\boldsymbol{M}}$ as a fixed point of $\Psi_\sigma^{\boldsymbol{M}}$ we get $f_\sigma^{\boldsymbol{M}} = \varphi_\sigma$ which yields (a). When dealing with $X = \boldsymbol{D}$ one can show that $\imath_{\boldsymbol{D}} \circ \varphi_\sigma$ is the least fixed point of $\Psi_\sigma^{\boldsymbol{D}}$. Thus, $f_\sigma^{\boldsymbol{D}} = \imath_{\boldsymbol{D}} \circ \varphi_\sigma$ which yields (c). (b) and (d) follow by (a), (c), Theorem 3.7 and Theorem 3.11. □

Clearly, $\boldsymbol{D}$ is more "abstract" than $\boldsymbol{M}$ as simulation equivalence is coarser than bisimulation equivalence. It can be shown that there is a unique function $\Phi : \boldsymbol{M} \to \boldsymbol{D}$ with $\Phi(x) = \{(\alpha, \mathcal{E}_1(\Phi)(x)) : (\alpha, E) \in x\}^*$ for all $x \in \boldsymbol{M}$. For this function Φ we obtain the "consistency result" $\Phi \circ Me^{\boldsymbol{M}} = Me^{\boldsymbol{D}}$.[15]

5 Conclusion

We have studied two general semantic frameworks, one rooted in domains and the other in complete ultrametric spaces, of probabilistic processes, and have shown that they are fully abstract with respect to natural probabilistic

[14] Formally, $x = \lim x_n$, $x' = \bigsqcup x'_n$ where $x_0 = \emptyset$, $x_{n+1} = \{(\alpha, E_{\mu^1_{x_n}})\}$, $x'_0 = \bot_{\boldsymbol{D}}$, $x'_{n+1} = \{(\alpha, E_{\mu^1_{x'_n}})\}^*$.

[15] For the notion of "consistency" see [7].

extensions of the simulation preorder and bisimulation equivalence. Furthermore, we obtained final semantics in the sense of [38]. Our frameworks allow to combine the qualitative non-deterministic choice together with the quantitative probabilistic choice in an independent fashion. The simpler, reactive, models can be obtained as special cases by replacing the non-deterministic choice with its deterministic counter-part. Moreover, it can be seen that the usual denotational semantics of the non-probabilistic *CCS* arises as a special case of the probabilistic version by setting the distributions to the simple distributions (point masses).

To construct the denotational models we have generalised to the probabilistic setting the established category-theoretic techniques for solving domain equations for non-probabilistic processes (notably, synchronization trees). The generalised domain equations involved appropriately adjusted probabilistic powerdomain constructions. Solving the domain equation in the category of continuous domains yields a "smooth" construction, in the sense that, for example, the probabilistic powerdomain $\mathcal{E}_1(S)$ of a two-point space S is the real interval $[0,1]$. Thus, limits can be approximated by approaching them arbitrarily close. The probabilistic powerdomain of [22] is also "smooth" in this sense. On the other hand, in the ultrametric case we obtain a "discrete" construction, in the sense that the two-point space lifted to the probabilistic case gives the real interval $[0,1]$ with the discrete topology. In particular, it is not possible to get arbitrarily close to a limit. We should emphasise though that the methodology we used to derive an ultrametric model is consistent with the established methodology (in particular, the metric satisfies the intuitive property $d(x,y) \leq \frac{1}{2^n}$ iff x and y agree up to the nth step, and we obtain full abstraction for bisimulation), and that an attempt to obtain a "smooth" metric construction might mean having to go beyond the known techniques, see [27,36].

The framework proposed in this paper can, with some modifications, be used to derive semantic models for a variety of probabilistic calculi known from the literature. For example, to give a denotational semantics for a calculus that allows for general probabilistic choice (rather than the guarded probabilistic choice considered here), e.g. Probabilistic *CCS* as introduced in [47], whose operational semantics is based on a variant of probabilistic transition systems that distinguish between action- and probability-labelled transitions, we have to solve recursive domain equations of the form $X \cong \wp_*(Act \times X \cup \mathcal{E}_1(X))$. If we consider a probabilistic extension of *SCCS* in which a probabilistic choice operator replaces the non-deterministic choice operator, e.g. the calculi of [14,42], then the natural choice for the operational semantics is that of a "generative" probabilistic transition system (i.e. a Markov chain with additional labelling with actions). The equations of the form $X \cong \mathcal{E}_1(Act \times X)$ are appropriate in this case and yield denotational models that are fully abstract w.r.t. bisimulation and simulation. Finally, given a language with operational semantics based on "reactive" transition systems [16], full abstraction w.r.t. (bi-)simulation can be obtained by solving domain equations of the form $X \cong Act \rightarrow \mathcal{E}_1(X)$.

As a possible future direction we should mention considering a language with divergence and finding an appropriate logic for probabilistic processes.

Acknowledgement

We would like to acknowledge Achim Jung for suggesting improvements to the paper.

References

[1] S. Abramsky: A Domain Equation for Bisimulation, Information and Computation, Vol. 92, pp 161-218, 1991.

[2] S. Abramsky, A. Jung: Domain Theory, In S. Abramsky, D.M. Gabbay and T.S.E. Maibaum, editors, Handbook of Logic in Computer Science, Vol. 3, Clarendon Press, pp 1-168, 1994.

[3] P. America, J.J.M.M. Rutten: Solving Recursive Domain Equations in a Category of Complete Metric Spaces, Journal of Computer and System Sciences, Vol. 39, No. 3, pp 343-375, 1989.

[4] C. Baier: Trees and Semantics, Theoretical Computer Science, Vol. 179, pp 217-250, 1997.

[5] C. Baier, M.Z. Kwiatkowska: Domain Equations for Probabilistic Processes, Technical Report CSR-97-7, University of Birmingham, 1997. Submitted for publication.

[6] C. Baier, M.E. Majster-Cederbaum: Denotational Semantics in the CPO and Metric Approach, Theoretical Computer Science, Vol. 135, pp 171-220, 1994.

[7] C. Baier, M.E. Majster-Cederbaum: How to Interpret and Establish Consistency Results for Semantics of Concurrent Programming Languages, Fundamenta Informaticae, Vol. 29, No. 3, pp 225-256, 1997.

[8] J.W. de Bakker, J.J.Ch. Meyer: Metric Semantics for Concurrency, Report CS-R8803, Centre for Mathematics and Computer Science, Amsterdam, 1988.

[9] J.W. de Bakker, J.I.Zucker: Processes and the Denotational Semantics of Concurrency, Information and Control, Vol.54, No. 1/2, pp 70-120, 1982.

[10] I. Christoff: Testing Equivalences and Fully Abstract Models for Probabilistic Processes, Proc. CONCUR'90, LNCS 458, pp 126-140, 1990.

[11] R. Cleaveland, S. Smolka, A. Zwarico: Testing Preorders for Probabilistic Processes, Proc. ICALP'92, LNCS 623, pp 708-719, 1992.

[12] C. Courcoubetis, M. Yannakakis: The Complexity of Probabilistic Verification, Journal of the ACM, Vol. 42, No. 4, pp 857-907, 1995.

[13] R. Engelking: General Topology, Sigma Series in Pure Mathematics, Vol. 6, Heldermann Verlag Berlin, 1989.

[14] A. Giacalone, C. Jou, S. Smolka: Algebraic Reasoning for Probabilistic Concurrent Systems, Proc. IFIP TC2 Working Conference on Programming Concepts and Methods, 1990.

[15] G. Gierz, H. Hofmann, K. Keimel, J. Lawson, M. Mislove, D. Scott: A Compendium of Continuous Lattices, Springer-Verlag, 1980.

[16] R. van Glabbeek, S. Smolka, B. Steffen, C. Tofts: Reactive, Generative, and Stratified Models for Probabilistic Processes, Proc. LICS'90, pp 130-141, 1990.

[17] W. Golson, W.C. Rounds: Connection Between Two Theories of Concurrency: Metric Spaces and Synchronisation Trees, Information and Control, Vol. 57, pp 102-124, 1983.

[18] H. Hansson, B. Jonsson: A Calculus for Communicating Systems with Time and Probabilities, in Proc. 11th IEEE Symposium on Real Time Systems, 1990.

[19] R. Heckmann: Spaces of Valuations, Technical Report A 09/95, FB 14 Informatik, Universität des Saarlandes, 1995.

[20] C.A.R. Hoare: Communicating Sequential Processes, Prentice Hall, 1985.

[21] C. Jones: Probabilistic Non-Determinism, PhD Thesis, University of Edinburgh, 1990.

[22] C. Jones, G.D. Plotkin: A Probabilistic Powerdomain on Evaluations, Proc. LICS'89, pp 186-195, 1989.

[23] B. Jonsson, K.G. Larsen: Specification and Refinement of Probabilistic Processes, Proc. LICS'91, pp 266-277, 1991.

[24] B. Jonsson, W. Yi: Compositional Testing Preorders for Probabilistic Processes, Proc. LICS'95, pp 431-443, 1995.

[25] C.-C. Jou, S. Smolka: Equivalences, congruences and complete axiomatizations for probabilistic processes, Proc. CONCUR'90, LNCS 458, pp 367-383, 1990.

[26] K. Kuratowski: Sur une méthode de métrisation complète des certains espaces d'ensembles compacts, Fundamentae Mathematicae 43, pp 114-138, 1956.

[27] M.Z. Kwiatkowska, G. Norman: Probabilistic Metric Semantics for a Simple Language with Recursion, Proc. MFCS'96, LNCS 1113, pp 419-430, 1996.

[28] K. Larsen, A. Skou: Bisimulation through Probabilistic Testing, Information and Computation, Vol. 94, pp 456-471, 1991.

[29] D. Lehmann: Categories for Mathematical Semantics, Proc. FOCS'76, IEEE Press, 1976.

[30] G. Lowe: Probabilities and Priorities in Timed CSP, D.Phil thesis, Oxford, 1991.

[31] M. Majster-Cederbaum: On the Uniqueness of Fixed Points of Endofunctors in a Category of Complete Metric Spaces, Information Processing Letters, Vol. 29, pp 277-281, 1988.

[32] M. Majster-Cederbaum: The Contraction Property is Sufficient to Guarantee the Uniqueness of Fixed Points of Endofunctors in a Category of Complete Metric Spaces, Information Processing Letters, Vol. 33, pp 15-19, 1988.

[33] M. Majster-Cederbaum, F. Zetzsche: Towards a Foundation for Semantics in Complete Metric Spaces, Information and Computation, Vol. 90, No. 2, pp 217-243, 1991.

[34] R. Milner: Calculi for Synchrony and Asynchrony, Theoretical Computer Science, Vol. 25, pp 269-310, 1983.

[35] R. Milner: Communication and Concurrency, Prentice Hall, 1989.

[36] G. Norman: Metric Semantics for Reactive Probabilistic Processes, PhD Thesis, University of Birmingham, forthcoming.

[37] D. Turi, G. Plotkin: Towards a Mathematical Operational Semantics, Proc. LICS'97, IEEE Computer Society Press, 1997.

[38] J.J.M.M. Rutten, D. Turi: On the Foundations of Final Semantics: Non-Standard Sets, Metric Spaces and Partial Orders, Proc. REX'92, LNCS 666, pp 477-530, 1993.

[39] J. Stoy: Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory, MIT Press, Cambridge, 1977.

[40] R. Segala, N. Lynch: Probabilistic Simulations for Probabilitsic Processes, Proc. CONCUR'94, LNCS 836, pp 481-496, 1994.

[41] M.B. Smyth, G.D. Plotkin: The Category-Theoretic Solution of Recursive Equations, SIAM J. Comput., Vol. 11, pp 761-783, 1982.

[42] C. Tofts: Processes with Probabilities, Formal Aspects of Computing, Vol. 6, No. 5, 1994.

[43] M. Vardi: Automatic Verification of Probabilistic Concurrent Finite-State Programs, Proc. 26th Symp. on Foundations of Computer Science, pp 327-338, 1985.

[44] G. Winskel: Synchronisation Trees, Theoretical Computer Science, Vol. 34, pp 33-82, 1984.

[45] S. Yuen, R. Cleaveland, Z. Dayar, S. Smolka: Fully Abstract Characterizations of Testing Preorders for Probabilistic Processes, Proc. CONCUR'94, LNCS 836, pp 497-512, 1994.

[46] W. Yi: Algebraic Reasoning for Real-Time Probabilistic Processes with Uncertain Information, Proc. FTRTFT'94, LNCS 863, pp 680-693, 1994.

[47] W. Yi, K. Larsen: Testing Probabilistic and Nondeterministic Processes, Proc. Protocol, Specification, Testing, Verification XII, pp 47-61, 1992.