Approximating Labeled Markov Processes

Josée Desharnais*
School of Computer Science
McGill University
Montreal, Quebec, H3A 2A7, Canada
desharna@cs.mcgill.ca

Radha Jagadeesan†
Dept. of Math. and Computer Sciences
Loyola University-Lake Shore Campus
Chicago IL 60626, USA
radha@math.luc.edu

Vineet Gupta
PurpleYogi.com
201 Ravendale Drive
Mountain View CA 94043, USA
vineet@purpleyogi.com

Prakash Panangaden‡
School of Computer Science
McGill University
Montreal, Quebec, H3A 2A7, Canada
prakash@cs.mcgill.ca

Abstract

We study approximate reasoning about continuous-state labeled Markov processes. We show how to approximate a labeled Markov process by a family of finite-state *labeled Markov chains. We show that the collection of labeled Markov processes carries a Polish space structure with a countable basis given by finite state Markov chains with rational probabilities. The primary technical tools that we develop to reach these results are*

- *A finite-model theorem for the modal logic used to characterize bisimulation*
- *A categorical equivalence between the category of Markov processes (with simulation morphisms) with the ω-continuous dcpo* $\mathbf{Proc}$*, defined as the solution of the recursive domain equation* $\mathbf{Proc} = \prod_{\mathtt{Labels}} P_{\mathtt{Prob}}(\mathbf{Proc})$[1].

The correspondence between labeled Markov processes and $\mathbf{Proc}$ *yields a logic complete for reasoning about simulation for continuous-state processes.*

1. Introduction

Markov processes with continuous state spaces or continuous time evolutions (or both) arise naturally in several fields of physics, economics and computer science. Examples of such systems are brownian motion, gas diffusion, population growth models and changes in stock prices. In order to come to *computational* grips with such systems one needs a notion of discrete approximation. The purpose of this paper is to study *interacting* versions of such general Markov processes from the point of view of approximation.

Continuous time Markov processes arise naturally in performance modeling of computer systems.

Example 1.1 *A system resource scheduler may be modeled as a* queue *[33] — a Markov process in which jobs come in, and are processed by a server. The rate at which jobs arrive (or their inter-arrival times) and the rate at which they are serviced are given by various distributions.*

The next example shows that continuous state systems arise in even superficially discrete paradigms.

Example 1.2 *Consider a probabilistic process algebra with recursion. The natural model of a probabilistic process with recursion may have uncountably many states. For instance consider the process $P = a.P +_{\frac{1}{2}} b.P$, where the probabilistic choice operator takes either branch with equal probability. This process produces a uniform distribution over all infinite strings of a's and b's, so a continuous state space is inevitable.*

The next example comes from a real-life situation (currently under study by one of the authors in collaboration with an avionics software company) and shows the importance of concurrency and verification.

Example 1.3 *Consider the flight management system (FMS) of an aircraft. The FMS is responsible for periodically monitoring the state of the aircraft—the altitude, wind*

*Research supported in part by NSERC.
†Research supported in part by NSF and a visiting fellowship from UK SERC.
‡Research supported in part by NSERC.

[1] $P_{\mathtt{Prob}}(D)$ is the probabilistic powerdomain [28]

speed, direction, roll, yaw etc. It also monitors navigational data from satellites and makes corrections, as needed, by issuing commands to the engines and the wing flaps. The physical system is a complex continuous real-time stochastic system; stochastic because the response of the physical system to commands cannot be completely deterministic and also because of unexpected situations like turbulence. From the point of view of the FMS, however, the system is a discrete-time continuous-space system with the time unit being the sampling rate. Furthermore the system will have to react to inputs from the pilots, who may change the flight plan in mid-flight.

Example 1.4 *The fundamental example of a continuous stochastic system is brownian motion. This was observed by Brown in 1827 and studied systematically by Einstein [20], Smoluchowski [46] and others. The basic phenomenon that Brown observed is that grains of pollen moved in an apparently random fashion in the air as a result of collisions with air molecules. Mathematically, one can think of a stochastic process as a "random variable extended in time." The stochastic process underlying brownian motion is the Weiner process W: a function from reals (representing time) to random variables satisfying:*

- *Gaussian property: $W(t_1), \dots, W(t_n)$ is a multivariate normal distribution;*
- *$W(t+s) - W(s)$ has a normal distribution with mean 0 and variance proportional to t;*
- *$W(t+s) - W(s)$ is independent of $W(t'+s') - W(s')$ whenever $s' > t + s$;*
- *the sample paths of W can be chosen to be continuous with probability 1.*

This mathematical model is the starting point of many stochastic systems; see the book "Stochastic Differential Equations" by Øksendal [41] for a survey.

In a series of previous papers [7, 13] we studied Markov processes with continuous state spaces, called *partial labeled Markov processes* or **PLMP**s. We gave a definition of bisimulation between **PLMP**s, and gave a logical characterization of this bisimulation. We have also explored the expressiveness and semantics of probabilistic programming languages with continuous state spaces [23]. In this paper our aim is to establish the foundations of computing with these processes, with the aim of computing approximations to the observations that one makes on continuous stochastic systems.

Observations What are the observations of interest in such systems? At the outset we wish to emphasize the importance of quantitative properties and the need for techniques to compute these quantities.

- In the performance modeling example, we want to know the expected length of time a job will need to wait, the average number of jobs in the queue, the probability that the number of jobs in the queue exceeds a certain number, the expected fraction of time that the server is busy—these numbers guide decisions on the amount of resource needed, resource bottlenecks etc.
- In the process algebra example, we are interested in computing various conditional probabilities relating occurrences of events, for example see [23]. Recursion forces us to use measure-theoretic apparatus such as the Radon-Nikodym theorem for this purpose.
- In the FMS example we would want to know average response times of components, drift probabilities and large deviation bounds.
- In the Wiener process example, we wish to know properties like the distribution of times when $W(t) > c$, the asymptotic distribution, if a steady state asymptotic distribution exists, and the distribution after a certain time.

An important reason for the focus on "quantitative" rather than "logical" properties for observations is the incoherence between the notions of approximation and logical reasoning. In section 3.4, we present an example of a continuous state system P, a logical formula ϕ and a chain of finite state approximants to P such that

- ϕ is satisfied infinitely often in the approximating sequence.
- ϕ is invalidated infinitely often in the approximating sequence.

The problem is that with a two-valued logic one has to make a sharp decision whereas, in general, approximation schemes approach the correct answer in an incremental fashion. If one uses approximation techniques to determine the result of a boolean answer one will not be able to tell when the approximation has converged. This general kind of example of the subtle interaction between approximation and logic leads us to revisit Kozen's seminal ideas on logics in the context of probability [34, 35]—moving from truth valued boolean functions to real valued measurable functions. In this viewpoint, the notion of approximation acquires a precise quantitative character — we are demanding the result of an observation (e.g. an integral) within some error bound ϵ. In this world-view, the wild instability of logical reasoning (as shown by the example in section 3.4) in the context of approximations is replaced by the more stable decrease in the permitted error ϵ.

Our results. Our main result is a systematic approximation scheme for labeled Markov processes. We have essentially given the collection of labeled Markov processes the kind of structure enjoyed by the real numbers, namely that there is a countable collection of elements, the rationals, which approximate all the elements and determine the behavior of all continuous functions. Our primary technical result is:

> The set of **PLMP**s is a Polish space[2] with a countable dense subset given by finite rational **PLMP**s.

This shows that finite rational **PLMP**s can be used to approximate any **PLMP**— we provide a construction of these finite approximants for any **PLMP**. Furthermore, our approximation results allow us to approximate integrals of continuous functions by computing them on finite rational **PLMP**s, using standard techniques such as:

- Parthasarathy [42] shows that the space of all measures on a Polish space X is itself a Polish space. Furthermore, the set of measures, whose supports are finite subsets of the countable set dense in X, is itself dense in the space of measures. Finally, if $\mu_n \to \mu$, then $\int f d\mu_n \to \int f d\mu$ for all bounded continuous functions f.
- Edalat [16, 17, 19, 18] has exploited domain theoretic methods to define R-integration and show that R-integrability w.r.t. a bounded Borel measure extends Riemann integrability to compact metric spaces. The R-integral of f with respect to measure μ is approximated by the sums of f wrt simple valuations less than μ.

These two ways of approximating integrals can both be used by us since we have shown that we have a (compact) Polish space of **PLMP**s. However, the results of the present paper do not *use* the results of Edalat or Parthasarathy.

The route to our results is based first on the observation that the category of **PLMP**s is a natural "timed" extension of the poset $([0,1], \leq)$. Since the usual Polish space structure of $[0,1]$ is recovered from the poset $[0,1]$ by the Lawson topology, we study the domain theoretic structure of **PLMP**s. Indeed, we establish the following equivalence of categories:

$$\textbf{PLMP} \simeq \mathbf{Proc}$$

where **PLMP** is the category with objects **PLMP**s and with morphisms simulations; and $\mathbf{Proc}$ is the solution to the recursive domain equation

$$\mathbf{Proc} \simeq \prod_{\texttt{Labels}} \mathcal{P}_{\texttt{Prob}}(\mathbf{Proc}).$$

[2] A Polish space is a topological space underlying a complete separable metric space.

Since $\mathbf{Proc}$ is a (compact) ω-continuous dcpo, a standard construction (its Lawson topology) yields a Polish space inducing the required Polish space structure on **PLMP**. Moreover, the equivalence **PLMP** $\simeq$ $\mathbf{Proc}$ can also be viewed as a full abstraction result relating an operational model (**PLMP**) and a denotational model ($\mathbf{Proc}$).

The proof of this result breaks up naturally into two parts:

- We develop the notion of " finite " approximants to a **PLMP** and analyze the logical properties of the set of "finite" approximants to a **PLMP**. For any **PLMP**, we explicitly provide a (countable) sequence of approximants to it such that:

  1. For every logical property satisfied by a process, there is a element of the chain that also satisfies the property.

  2. The sequence is a chain in the simulation ordering.

  The key technical tools in this development are the logical characterization of bisimulation [13] and a construction (in Section 3) of a class of approximants.

- We perform an analysis of $\mathbf{Proc}$ in the spirit of "A domain equation for bisimulation" [1]. In particular, we show that there is a perfect match between simulation/bisimulation on the one hand and the partial order/equality in $\mathbf{Proc}$ on the other hand. The key technical tool in this development is showing that a variant of the logic of [13] is the "internal" logic of $\mathbf{Proc}$.

Some new and interesting consequences of the equivalence are:

- The equivalence endows **PLMP** with least upper bounds of ω-chains (wrt the simulation ordering). This shows that **PLMP** can be used as the target of interpretation of a syntax that includes recursion.

- The internal logic of $\mathbf{Proc}$ is a logic complete for reasoning about simulation of **PLMP**s.

Summary of paper In Section 2 we recall background material on labeled Markov processes. Section 3 develops the class of approximants, and studies its properties. In section 4, we analyze the solution of the recursive domain equation $D \simeq \Pi_{l \in \texttt{Labels}} \mathcal{P}_{\texttt{Prob}}(D)$. Finally, in section 5 we prove the equivalence **PLMP** $\simeq$ $\mathbf{Proc}$. In section 6, we place our work in the context of extant research.

In this extended abstract, we omit all proofs.

2. Background: Labeled Markov Processes

In this section we recapitulate the definitions of labeled Markov processes from [14, 7]. However, the presentation of bisimulation has been reformulated in a more relational style very close to the original definition of Larsen and Skou [36]. We assume that the reader is familiar with process algebra and labeled transition systems [40] and the concept of bisimulation in the nonprobabilistic setting.

Labeled Markov processes are probabilistic versions of labeled transition systems. For each label there is a transition probability distribution which gives the probability distribution of the possible final states given the initial state. This is the "reactive" model due to Larsen and Skou [36] who used it in a discrete state-space setting. We extended this to continuous-state systems, thus forcing our formalism to be couched in measure-theoretic terms. For instance, we cannot ask for the transition probability to any set of states — we need to restrict ourselves to measurable sets. In fact, we need to assume that the state space has an analytic space structure — these generalize Polish spaces. However, in this paper we focus on more combinatorial issues and suppress some of the finer points of measure theory, see [14] for details.

A key ingredient in the theory is the *transition probability function* (also called stochastic kernel [21] or Markov kernel [35]).

Definition 2.1 *A* ***transition (sub-)probability function*** *on a measurable space (S, Σ) is a function $\tau : S \times \Sigma \to [0, 1]$ such that for each fixed $s \in S$, the set function $\tau(s, \cdot)$ is a (sub-)probability measure, and for each fixed $X \in \Sigma$ the function $\tau(\cdot, X)$ is a measurable function.*

One interprets $\tau(s, X)$ as the probability of the process starting in state s making a transition into one of the states in X. We will work with *sub-probability* functions; i.e. with functions where $\tau(s, S) \leq 1$ rather than $\tau(s, S) = 1$. We view processes where the transition functions are sub-probabilities as being *partially defined*, opening the way for a notion of approximation.

Definition 2.2 *A* ***partial labeled Markov process*** *$\mathcal{S}$ with label set $\mathcal{A}$ is a structure $(S, i, \Sigma, \{\tau_a \mid a \in \mathcal{A}\})$, where S is the set of states forming an analytic space, i is the initial state, and Σ is the Borel σ-field on S, and*

$$\forall a \in \mathcal{A}, \tau_a : S \times \Sigma \longrightarrow [0, 1]$$

is a transition sub-probability function.

We will fix the label set to be $\mathcal{A}$ once and for all and will write (S, i, Σ, τ) for partial labeled Markov processes. Also we will just say labeled Markov process rather than *partial* labeled Markov process.

We give a few examples to illustrate the ideas.

Example 2.3 *(From [14])* Consider a process with two labels $\{a, b\}$. The state space is the real plane, $\mathbf{R}^2$. When the process makes an a-move from state (x_0, y_0), it jumps to (x, y_0), where the probability distribution for x is given by the density $K_\alpha e^{-\alpha(x-x_0)^2}$, where $K_\alpha = \sqrt{\alpha/\pi}$. When it makes a b-move it jumps from state (x_0, y_0) to (x_0, y), where the distribution of y is given by the density function $K_\beta e^{-\beta(y-y_0)^2}$. Thus the probability of jumping from (x_0, y_0) to a state with x-coordinate in the interval $[s, t]$ under an a-move is $\int_s^t K_\alpha e^{-\alpha(x-x_0)^2} dx$. Note that the probability of jumping to any given point is, of course, 0. In this process the interaction with the environment controls whether the jump is along the x-axis or along the y-axis but the actual extent of the jump is governed by a probability distribution. If there were just a single label we would have an ordinary (time-independent) Markov process.

Our second example is from queuing theory.

Example 2.4 We first show how to "label" states with values.

- To label state s with an integer n, add a self loop labeled f_n to s by defining a transition $\tau_{f_n}(s, X) = 1$ if $s \in X$, 0 otherwise.
- To label s with a real number r, add self loops labeled d_q on s for all rational $q < r$, and self-loops labeled e_q for all rational $q > r$. Thus the value on state s is $\inf\{q \mid \tau_{e_q}(s, S) > 0\} = \sup\{q \mid \tau_{d_q}(s, S) > 0\}$.

Now, consider a process that is a single queue with countably many buffers. The process is modeled as a **PLMP**. The state set is $\mathbf{N} \times \mathbf{R}$. Intuitively, the first number will represent the number of buffers filled and the second will represent the length of time to be spent in this state.

Each state (k, t) has a transition labeled a to states $(k + 1, t')$, and if $k > 0$, to $(k - 1, t')$. The probabilities on these are given by the arrival and departure probabilities for the queue — for example for $M/M/1$ queues these are given by exponential distributions. Besides these, we label each state with k and t, as described above.

The next example is from stochastic process theory.

Example 2.5 We represent a Wiener process with drift C. The set of states is $\mathbf{R}$, with the usual σ-algebra. The start state is 0. For every rational $q > 0$ we have a label a_q, and for every state s we have an a_q labeled transition whose probability distribution is normal, with mean $s + C \times q$ and variance q. Each state is also labeled with its real number, as described above. Each label a_q intuitively represents the passage of q time units. Now we can check that this process satisfies the Kolmogorov consistency conditions [41, p.9], so it corresponds to the continuous time Wiener process—the a_q correspond to rational time observations of the underlying continuous process.

2.1. Bisimulation and Logic

The fundamental process equivalence that we consider is *strong probabilistic bisimulation* or just "bisimulation" for the present paper. Our definition is adapted from the definition given by Larsen and Skou [36], with extra conditions to deal with measure-theoretic issues. Probabilistic bisimulation means matching the moves and probabilities *exactly*—thus each system must be able to make the same transitions with the same probabilities as the other.

Let R be a binary relation on a set S. We say a set $X \subseteq S$ is *R-closed* if $R(X) \stackrel{d}{=} \{t | \exists s \in X, sRt\}$ is a subset of X. If R is reflexive, this becomes $R(X) = X$. If R is an equivalence relation, a set is R-closed iff it is a union of equivalence classes.

Definition 2.6 *Let $\mathcal{S} = (S, i, \Sigma, \tau)$ be a labeled Markov process. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-closed measurable set $X \in \Sigma$, $\tau_a(s, X) = \tau_a(s', X)$. Two states are bisimilar if they are related by a bisimulation relation.*

The intuition of this definition is that the relation R relates those states that can be "lumped" together. Bisimulation is the largest such relation. Bisimulation is an equivalence relation [14, 11].

The logic $\mathcal{L}$ has the following syntax:

$$\mathsf{T} \mid \phi_1 \wedge \phi_2 \mid \langle a \rangle_q \phi$$

where a is a label and q is a rational number. This is the basic logic with which we establish the logical characterization. In later sections we will work with this logic augmented with disjunction, $\mathcal{L}_\vee$:

$$\mathcal{L} \mid \phi_1 \vee \phi_2.$$

Given a **PLMP** $\mathcal{S} = (S, i, \Sigma, \tau)$ we write $s \models \phi$ to mean that the state s satisfies the formula ϕ. The definition of the relation $\models$ is given by induction on formulas. The definition is obvious for the propositional constant T, conjunction and disjunction. We say $s \models \langle a \rangle_q \phi$ iff $\exists X \in \Sigma.(\forall s' \in X.s' \models \phi) \wedge (\tau_a(s, X) > q)$. We write $[\![\phi]\!]_{\mathcal{S}}$ for the set $\{s \in S \mid s \models \phi\}$, and $\mathcal{S} \models \phi$ if $i \models \phi$. We often omit the subscript on $[\![\phi]\!]_{\mathcal{S}}$ when no confusion can arise. The main theorem relating $\mathcal{L}$ and bisimulation is the following [13, 14].

Theorem 2.7 *Let (S, i, Σ, τ) be a labeled Markov process. Two states $s, s' \in S$ are bisimilar iff they satisfy the same formulas of $\mathcal{L}$.*

The notion of simulation is the natural unidirectional version of the definition of bisimulation. We also introduce a concept called *strict simulation* which will correspond to the "way-below" relation. Our definition of simulation follows [12].

Definition 2.8 *Let $\mathcal{S} = (S, i, \Sigma, \tau)$ be a labeled Markov process. A reflexive and transitive relation (a preorder) R on S is a **simulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-closed measurable set $A \in \Sigma$, $\tau_a(s, A) \leq \tau_a(s', A)$.*

*R is a **strict simulation** if there is an $\epsilon > 0$ such that for all R-closed $A \in \Sigma$, we have $\tau_a(s, A) < \tau_a(s', A) - \epsilon$ whenever $0 < \tau_a(s', A)$. We say s is simulated (strictly simulated) by s' if sRs' for some simulation (resp. strict simulation) relation R.*

Note that strict simulation is quite stringent and the requirement is really something that is commonly found only with finite-state processes.

The notion of simulation meshes properly with the logic in the sense of the following proposition. Later in this paper, we prove its converse.

Proposition 2.9 *If s is simulated by s', then for all formulas $\phi \in \mathcal{L}_\vee$, $s \models \phi$ implies $s' \models \phi$.*

Although $\mathcal{L}$ is enough to characterize bisimulation, characterization of simulation needs (finite) disjunction — the full paper contains an example to illustrate this. Notice that since $\mathcal{L}$ has no negation, the logical characterization of bisimulation implies that two-way simulation is bisimulation.

3. Approximations and Finite Models

In this section we develop the basic approximation result — for any labeled Markov process $\mathcal{S}$, we build a chain (in the strict simulation ordering) of *finite acyclic labeled Markov processes* (FAMPs — finite-state processes with acyclic transition graphs) $\{\mathcal{S}_i\}$ such that:

1. For any formula ϕ satisfied by $\mathcal{S}$, there is an i such that $\mathcal{S}_i$ satisfies ϕ — this is a finite model theorem since it gives finite models for any formula satisfied by the system being approximated.

2. $\mathcal{S}_i$ is simulated by $\mathcal{S}$, for all i.

Finally, we also present the example that illustrates the subtleties of logical reasoning in the presence of approximations.

3.1. Finite-state approximation

The key tool in our analysis is the construction of some approximants via an "unfolding" construction. There are two parameters to the approximation, one is a natural number n, and the other is a positive rational ϵ. The number n gives the number of successive transitions possible from the start state. The number ϵ measures the accuracy with which the probabilities approximate the transition probabilities of the original process.

Given a labeled Markov process $\mathcal{S} = (S, i, \Sigma, \tau)$, an integer n and a rational number $\epsilon > 0$, we define $\mathcal{S}(n, \epsilon)$ to be an n-step unfolding approximation of $\mathcal{S}$. Its state-space is divided into $n+1$ levels which are numbered $0, 1, \dots, n$. A state is a pair (X, l) where $X \in \Sigma$ and $l \in \{0, 1, \dots, n\}$. *At each level*, the sets that define states form a partition of S. The initial state of $\mathcal{S}(n, \epsilon)$ is at level n and transitions only occur between a state of one level to a state of one lower level. Thus, in particular, states of level 0 have no outgoing transitions. In the following we omit the curly brackets around singletons.

Definition 3.1 *Let (S, i, Σ, τ) be a labeled Markov process, $n \in \mathbf{N}$ and ϵ a positive rational. We denote the finite-state approximation by $\mathcal{S}(n, \epsilon) = (P, p, \rho)$ where P is a subset of $\Sigma \times \{0, \dots, n\}$. For $n \in \mathbf{N}$ and $\epsilon > 0$, $\mathcal{S}(n, \epsilon)$ has $n + 1$ levels. States are defined by induction on their level. Level 0 has one state $(S, 0)$. Now, given the sets from level l, we define states of level $l + 1$ as follows. Suppose that there are m states at level l, then let $(B_j)_{j \in I}$ stand for the following partition of $[0, 1]$: $\{\{0\}, (0, \epsilon/m], (\epsilon/m, 2\epsilon/m], \dots\}$. States of level $l + 1$ are obtained by the partition of S that is generated by the sets $\tau_a(\cdot, C)^{-1}(B_j)$, for (C, l) a state at level l and every label $a \in \{a_1, \dots, a_n\}$. Thus, if a set X is in this partition of S, $(X, l + 1)$ is a state of level $l + 1$. Transitions can happen from a state of level $l + 1$ to a state of level l, and the transition probability function is given by*

$$\rho_a((X, k), (B, l)) = \begin{cases} \inf_{t \in X} \tau_a(t, B)) & \text{if } k = l + 1, \\ 0 & \text{otherwise.} \end{cases}$$

The initial state p of $\mathcal{S}(n, \epsilon)$ is the unique state (X, n) such that $i \in X$.

Suppose that $B = \cup B_j$, is a (finite) union of sets at the same level (i.e. $(B_j, l) \in \mathcal{S}(n, \epsilon)$). We write (B, l) instead of $\{(B_1, l), (B_2, l), ldots, (B_n, l)\}$. In this situation, we have that

$$\rho_a((X, l + 1), (B, l)) = \sum_{j \in I} \rho_a((X, l + 1), (B_j, l)),$$

since sets at the same level form a partition. If $s \in S$, we denote by (X_s, l) the unique state (at level l) such that $s \in X_s$.

It turns out that every state (X, l) in $\mathcal{S}(n, \epsilon)$ is simulated in S by every state $s \in X$. The next theorem is the main result of this section.

Theorem 3.2 *If a state $s \in S$ satisfies a formula $\phi \in \mathcal{L}_\vee$, then there is some approximation $\mathcal{S}(n, \epsilon)$ such that $(X_s, n) \models \phi$.*

Proof (Outlined). The proof is by induction on the structure of formulas. Let $depth(\phi)$ be the maximum nesting level of modal operators in ϕ. We prove that for all formulas ϕ and every $l \geq depth(\phi)$ there is an increasing sequence $(X_n)_{n \geq l}$ of sets in Σ which satisfy:

(i) $\cup_{n \geq l} X_n = [\![\phi]\!]_{\mathcal{S}}$;

(ii) $\exists (C_j, l) \in \mathcal{S}(n, 1/2^n), j = 1, \dots, m$, such that $X_n = \cup_{j=1}^m C_j, n \geq l$;

(iii) the states (C_j, l) satisfy ϕ in $\mathcal{S}(n, 1/2^n)$.

This implies the stated theorem. ■

It is possible to explicitly reconstruct the original process from the approximants. See the full paper for details.

3.2. Example

We compute a few approximations of a simple continuous process (Figure 1). States are from the set $\{s, t\} \cup [0, 3]$, the initial state is 1 and transitions are as follows:

- Let $x \in [0, 1]$, $0 \leq y \leq 1$.

$$\begin{aligned} p_a(x, [0, y)) &= \tfrac{x+y}{4}, \\ p_a(x, \{1\}) &= \tfrac{1-x}{4}, \\ p_a(x, (1, 1 + y]) &= \tfrac{y}{4}, \\ p_a(x, (2, 2 + y]) &= \tfrac{xy}{4}, \end{aligned}$$

- if $x \in (1, 2]$, $p_a(x, s) = 1$.
- if $x \in (2, 3]$, $p_b(x, t) = 1$.

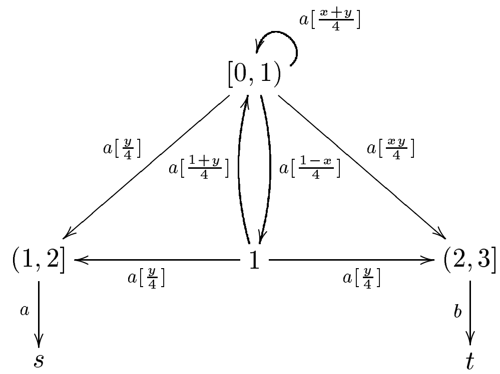


Figure 1. A simple continuous process

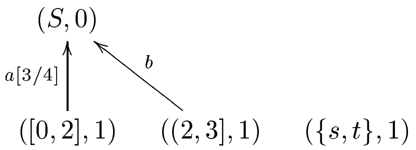
Let us compute the approximation $\mathcal{S}(2, 1/2)$. At level 0, we have state $(S, 0)$. At level 1, we partition S according to the partition of $[0, 1]$ into intervals of size $1/2$: $\{\{0\}, (0, 1/2], (1/2, 1]\}$. If $x \in [0, 1]$, then $p_a(x, S) = \frac{x+1}{4} + \frac{1-x}{4} + \frac{1}{4} + \frac{x}{4} = \frac{3+x}{4}$ hence

$$p_a(x, S) \begin{cases} = 0 & \texttt{if } x \notin [0, 2] \\ \in (1/2, 1] & \texttt{if } x \in [0, 2] \end{cases}$$

and

$$p_b(x,S) = \begin{cases} 0 & \texttt{if } x \notin (2,3] \\ 1 & \texttt{if } x \in (2,3]. \end{cases}$$

This yields, for level 1, the sets $[0,2]$, $(2,3]$ and $\{s,t\}$. The transitions are as in the following picture which represents the approximation $\mathcal{S}(1,1/2)$.



The initial state is $([0,2],1)$ and satisfies the formula $\langle a\rangle_{3/4-\epsilon}\mathsf{T}$, for all $\epsilon > 0$.

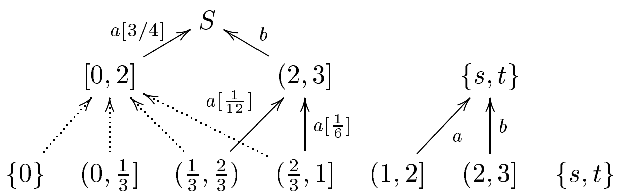
Now for level two of $\mathcal{S}(2,1/2)$, we must consider the partition of $[0,1]$ into intervals of size $1/2 \times 1/3$ since there are 3 states at level 1.

$$p_a(x,\{s,t\}) = \begin{cases} 0 & \texttt{if } x \notin (1,2] \\ 1 & \texttt{if } x \in (1,2] \end{cases}$$
$$p_b(x,\{s,t\}) = \begin{cases} 0 & \texttt{if } x \notin (2,3] \\ 1 & \texttt{if } x \in (2,3] \end{cases}$$

Now for $x \in [0,1]$, we have

$$p_a(x,[0,3]) = \tfrac{3+x}{4} \in \begin{cases} (4/6,5/6], & x \in [0,1/3] \\ (5/6,1], & x \in (1/3,1] \end{cases}$$
$$p_b(x,[0,3]) = 0$$
$$p_a(x,[0,2]) = \tfrac{x+1}{4} + \tfrac{1-x}{4} + \tfrac{1}{4} = 3/4$$
$$p_a(x,(2,3]) = \tfrac{x}{4} \in \begin{cases} (0,1/6] & \texttt{if } x \in (0,2/3] \\ (1/6,2/6] & \texttt{if } x \in (2/3,1], \end{cases}$$

The sets formed by unions of the set $\{s,t\}$ with other sets of level one are not useful and we ignore them. Thus the sets constituting the partition of S at level 2 are $\{0\}$, $(0,1/3]$, $(1/3,2/3]$, $(2/3,1]$, $(1,2]$, $(2,3]$, $\{s,t\}$. Transitions are illustrated in the following picture, where we omit the levels. In order not to clutter up the picture, we have not labeled the dotted lines that represent a-transitions with probability $3/4$.



The initial state is $((2/3,1],2)$ and satisfies the following formulas:
$\langle a\rangle_{\frac{11}{12}-\epsilon}\mathsf{T}$, $\langle a\rangle_{3/4-\epsilon}\langle a\rangle_0\mathsf{T}$, and $\langle a\rangle_{1/6-\epsilon}\langle b\rangle_0\mathsf{T}$ where $\epsilon > 0$. Note that the set $(1,2]$ will never be split in any approximations, since it contains only bisimilar states.

3.3. A countable basis for labeled Markov processes

The space of all labeled Markov processes appears too large to be used in a computational way. In fact this space has a countable subset which serves to approximate all labeled Markov processes. For brevity, in this section, we will just say "rational tree" when we mean a finite-state process with a tree-like transition graph and rational transition probabilities.

Lemma 3.3 *Let $\mathcal{T}$ be a rational tree that is strictly simulated by a labeled Markov process $\mathcal{S}$. Then there is a finite approximation $\mathcal{S}(n,\epsilon)$ strictly simulating $\mathcal{T}$.*

The following theorem shows that rational trees that are strictly simulated by a process form a directed set, establishing one of the properties needed to show that labeled Markov processes form a continuous domain.

Theorem 3.4 *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rational trees that are strictly simulated by a labeled Markov process $\mathcal{S}$. Then there is a rational tree which is strictly simulated by $\mathcal{S}$ and also strictly simulates both $\mathcal{T}$ and $\mathcal{T}'$.*

3.4. Incoherence of logic+approximation: An example

We consider the logic **PCTL*** [3], used for defining probabilistic properties on Markov processes. State $s \models a$ if s is labeled a. A sequence of states $\pi = (s_0, s_1, \dots)$ satisfies $X\psi$ if $s_1 \models \psi$. A measure is imposed on the set of sequences that start from a state, in the usual way. A state $s \models Pr_{>c}\psi$ if the measure of all sequences which start from the state and satisfy ψ is greater than c. We consider the **PCTL*** property

$$\phi = Pr_{\geq 1}X((Pr_{>0.2}Xa) \vee (Pr_{<0.2}Xb))$$

Suppose we want to check $s \models \phi$ on the following **PLMP** P: its state set is $\{s\} \cup [0,1] \cup [0,1]^2$. There is a transition from s to $[0,1]$, with uniform probability. Also $\forall x \in [0,1]$, there is a transition from x to $\{x\} \times [0,1]$ with uniform probability. Furthermore, state $(x,y) \in [0,1]^2$ is labeled a if $y \geq 0.8 - 0.2x$. Also, $(x,y), y \leq 0.2 + 0.2x$ is labeled b. It is easy to see that this is a **PLMP**, and that $s \models \phi$ — for all states $x \in (0,1]$, we have that $x \models Pr_{>0.2}Xa$.

Now consider the finite approximations to P given intuitively by partitioning $[0,1]^2$ with a finite grid (see figure 2). Each rectangle in the grid is a state, and is labeled a if all states in it are labeled a in P, and similarly for b. The grid also partitions the x-axis, and each interval on it corresponds to a state also. Each such interval has transitions to the rectangles directly above it, with probability given by

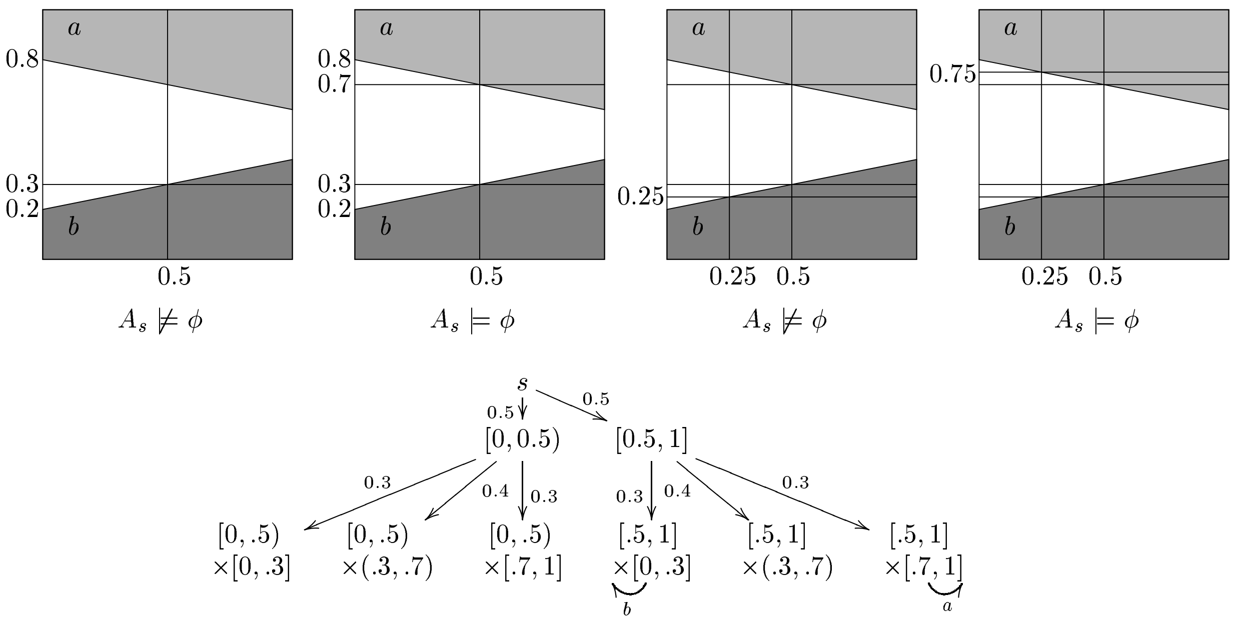


Figure 2. (a) 4 successive grids, showing how further refinement can toggle the status of a property. The states labeled a and b are shaded. (b) The finite PLMP corresponding to the second grid.

the height of the rectangle. The start state has a transition to each interval state, with probability given by the size of the interval. It is easy to check that each grid induces a finite approximation to P.

Consider the first grid — we have a state labeled b, and thus the right interval does not satisfy $Pr_{<0.2}Xb$, so the property ϕ is not satisfied. However refining the grid further by adding the line $y = 0.7$ makes $Pr_{>0.2}Xa$ be true of this state, so the property is satisfied. Continuing in this way, if at the $2n$'th stage we add the lines $x = 2^{-n}$ and $y = 0.2(1 + 2^{-n})$, the property is violated, but at the $(2n + 1)$st stage if we add $y = 0.2(4 - 2^{-n})$, the property is satisfied.

Note that these are not approximants that are produced by our concrete construction. If one carries out that construction on this example one will see a different anomaly. The formula ϕ is never true in the approximants though it is true for the system being approximated. Thus there is a failure on continuity even with the approximants that we have constructed. The message is that one cannot sensibly approximately reason about the truth of formulas. Our measurable functions approach allows one to speak of coming closer and closer to the truth.

4. A domain of processes

So far we have defined processes in terms of standard concepts of probability theory. In this section we describe how to view the collection of processes as a dcpo. The domain of processes will be constructed by solving a recursive domain equation giving a dcpo very close, in spirit, to the domain of synchronization trees constructed by Abramsky [1]. Finally, we show that simulation in the domain is characterized by $\mathcal{L}_\vee$.

4.1. Background

A partial order is a dcpo, if it is closed under limits of directed sets. We will only consider dcpos with a bottom element. $b \ll x$, if for all directed sets S such that $x \sqsubseteq \bigsqcup S$, $(\exists s_i \in S)\ b \sqsubseteq s_i$. A dcpo D is continuous, if for all $d \in D$, the set $\{b \mid b \ll d\}$ is directed and has lub d. A dcpo D is ω-continuous, if there is a countable subset S such that for all $d \in D$, the set $\{b \in S \mid b \ll d\}$ is directed and has lub d. Such a set S is called a basis.

Define $b{\Uparrow} \stackrel{d}{=} \{x \mid b \ll x\}$ and $(b){\uparrow} \stackrel{d}{=} \{x \mid b \sqsubseteq x\}$. The Scott topology on a dcpo D, written $\sigma(D)$, consists of all sets U satisfying $(U){\uparrow} = U$ and for all directed sets $S \subseteq D$, $\sup S \in U$ implies $S \cap U \neq \emptyset$. In any ω-continuous dcpo, $\{b{\Uparrow} \mid\ b \in$ basis for $D\}$ is a base for the Scott topology.

A valuation ν on a dcpo D is a monotone and continuous function from $(\sigma(D), \subseteq)$ to $([0, 1], \leq)$ that satisfies: $(\forall U, V \in \sigma(D)\ [\nu(U \cup V) = \nu(U) + \nu(V) - \nu(U \cap V)]$. For any $x \in D$, the point valuation $\eta_x(U) = 1$ if $x \in U$, 0 otherwise. The probabilistic powerdomain of D, written $\mathcal{P}_{\mathtt{Prob}}(D)$, is the set of all valuations on D ordered pointwise.

Definition 4.1 *The probabilistic powerdomain of D is the set of all valuations on D ordered by $\nu \sqsubseteq \mu \Leftrightarrow (\forall U \in \sigma(D))\ [\nu(U) \leq \mu(U)]$.*

If D is ω-continuous, so is $\mathcal{P}_{\texttt{Prob}}(D)$ with a countable base given by valuations of the form: $r_1 \times \eta_{x1} + \ldots + r_n \times \eta_{xn}$ where r_i are rationals. There is a unique extension of valuations to measures on the Borel sets associated with the Scott topology [44, 28, 2, 37].

The Lawson topology on a dcpo D, written $\lambda(D)$, is generated by the base $U \setminus (F)\uparrow$, where $U \in \sigma(D)$, and F is a finite subset of D. In any ω-continuous dcpo, a countable base for the Lawson topology is given by $\{b \Uparrow \setminus (\{b_1, \cdots, b_n\})\uparrow \mid b, b_i \in \text{ a basis for } D\}$. For ω-continuous dcpos D, the Borel algebra generated by $\lambda(D)$ is the same as that generated by $\sigma(D)$. The Lawson topology on ω-continuous dcpos is separable and metrizable. In the case that the Lawson topology is also compact, we get a Polish space for ω-continuous dcpos. The following result [32] relates Scott compactness and Lawson closedness.

Lemma 4.2 *In a ω-continuous dcpo, every Scott-compact upper set A is Lawson closed and is expressible as the countable intersection of Scott open sets.*

We use this lemma and the fact that all measures on metric spaces are regular to characterize the ordering relation on valuations of a lawson compact D.

Lemma 4.3 *If $\lambda(D)$ is compact, $\forall \nu_1, \nu_2 \in \mathcal{P}_{\texttt{Prob}}(D)$:*

- $\nu_1 \sqsubseteq \nu_2 \Rightarrow (\forall$ *upper* $A)[\nu_1(A) \leq \nu_2(A)]$
- $(\forall$ *Lawson closed upper* $A)[\nu_1(A) \leq \nu_2(A)] \Rightarrow \nu_1 \sqsubseteq \nu_2]$

Lawson compactness is stable under inverse limits:

Lemma 4.4 (A.Jung) *ω-continuous Lawson compact dcpo's are closed under inverse limits.*

4.2. The domain $\mathbf{Proc}$

We fix a (countable) set $Labels$ of labels and use the Jones-Plotkin probabilistic powerdomain [29, 28]. For notational convenience, we write $Labels \to D$ for the product $\prod_{Labels} D$. Processes are given by the recursive domain equation:

$$\mathbf{Proc} = Labels \to \mathcal{P}_{\texttt{Prob}}(\mathbf{Proc}).$$

We will write $\sqsubseteq$ for the partial order in the domain.

Proposition 4.5 *The above domain equation can be solved in the category of ω-continuous, Lawson-compact dcpos.*

First, we define some rudimentary transition probabilities.

Definition 4.6 *$\tau_a(p, U) \stackrel{d}{=} p(a)(U)$ for $p \in \mathbf{Proc}$, and U a Scott-open set in $\mathbf{Proc}$.*

Definition 4.7 *An equivalence relation R on $\mathbf{Proc}$, is a* ***domain-bisimulation*** *if: sRt implies that for all labels a, and all Scott-open R-closed C, $s(a)(C) = t(a)(C)$. We say that s is domain-bisimilar to t if there exists a domain-bisimulation R such that sRt.*

Domain-bisimulation is an equivalence relation.

Definition 4.8 *A preorder R on $\mathbf{Proc}$ is a* ***domain-simulation*** *if sRt implies that for all labels a, and all Scott-open R-closed sets C, $s(a)(C) \leq t(a)(C)$.*

Domain-simulation is a preorder. Consider $\mathcal{L}_\vee$ interpreted over $\mathbf{Proc}$.

$$\begin{array}{ll} p \models T & \\ p \models \phi_1 \wedge \phi_2 & \text{if } p \models \phi_1 \text{ and } p \models \phi_2 \\ p \models \phi_1 \vee \phi_2 & \text{if } p \models \phi_1 \text{ or } p \models \phi_2 \\ p \models \langle a \rangle_q \phi & \text{if } p(a)[\![\phi]\!] > q \end{array}$$

where $[\![\phi]\!] = \{p \mid p \models \phi\}$. $[\![\phi]\!]$ is a Scott-open subset of $\mathbf{Proc}$. Using lemma 4.8 of [28]:

Lemma 4.9 *The following are equivalent:*

1. *$p \sqsubseteq q$,*
2. *p is simulated by q,*
3. *$p \models \phi$ implies $q \models \phi$.*

Corollary 4.10 *p is domain-bisimilar to q iff $p = q$ iff $(\forall \phi \in \mathcal{L}_\vee)\ p \models \phi \Leftrightarrow q \models \phi$.*

5. Relating $\mathbf{Proc}$ and LMP

In the preceding sections we have developed the theory of probabilistic processes from two points of view. Now, we connect these viewpoints. In this section, we use $\models_D$ for the domain-theoretic notion of satisfaction and $\models_M$ for the **PLMP** concept of satisfaction.

From $\mathbf{Proc}$ to LMP. Let p be an element in the probabilistic powerdomain $\mathbf{Proc}$. Consider the putative labeled Markov process $\mathcal{U}_0 = (|\mathbf{Proc}|, p, \tau_a)$ where we are considering the elements of $\mathbf{Proc}$ under the Lawson topology (yielding a Polish space) [38], and τ_a is given by the unique extension of valuations to measures on the Borel sets associated with the Lawson topology. Using the fact that $\tau_a(., U)$ is upper semicontinuous for each Scott-open U, we prove that $\tau_a(., E)$ is a measurable function for measurable $E \subseteq |\mathbf{Proc}|$. This makes $\mathcal{U} = (|\mathbf{Proc}|, p, \tau_a)$ a labeled Markov process such that $(\forall \phi)\ [p \models_D \phi \Leftrightarrow \mathcal{U} \models_M \phi]$.

Furthermore, using lemma 4.3, we show that $\sqsubseteq$ order on $\mathbf{Proc}$ is a simulation between the corresponding labeled Markov processes. Overall, this construction yields a 1-1 functor $[\![\cdot]\!]^{\mathbf{LMP}}_{\mathbf{Proc}} : \mathbf{Proc} \to \mathbf{LMP}$.

From LMP to $\mathbf{Proc}$. By induction on the height of the DAG, we can map any finite acyclic labeled Markov processes (FAMP) $\mathcal{P}$ to a $p \in \mathbf{Proc}$ such that for any formula ϕ: $\mathcal{P} \models_M \phi \iff p \models_D \phi$.

For general **PLMP**s $\mathcal{S}$, section 3 yielded us a family of FAMP's that are increasing in the simulation order. This chain of FAMP's when embedded in $\mathbf{Proc}$ yields a chain in $\mathbf{Proc}$, and we define $[\![\mathcal{S}]\!]^{\mathbf{Proc}}_{\mathbf{LMP}} = p$ as the lub of this chain. In combination with the results from section 3, we deduce that $p \models_D \phi \iff \mathcal{S} \models_M \phi$. Overall, this construction yields a functor $[\![\cdot]\!]^{\mathbf{Proc}}_{\mathbf{LMP}} : \mathbf{LMP} \to \mathbf{Proc}$.

Theorem 5.1 $[\![\cdot]\!]^{\boldsymbol{LMP}}_{\mathbf{Proc}}, [\![\cdot]\!]^{\mathbf{Proc}}_{\boldsymbol{LMP}}$ *form an equivalence between* ***LMP*** *and* $\mathbf{Proc}$.

Combined with the completeness of $\mathcal{L}_\vee$ for domain-simulation, we get:

Corollary 5.2 $\mathcal{L}_\vee$ *gives a logical characterization of simulation in* ***LMP***.

6. Conclusions and Related Work

The present paper exploits the remarkable interplay between measure theory and domain theory to yield a theory of approximation for probabilistic processes. On the one hand, we need our earlier results on the logical characterization of bisimulation to set up the correspondence. On the other hand, domain theory is crucial to get the completeness of the logic for simulation and establish the required Polish space structure on labeled Markov processes.

The foundational work on the use of probability in semantics is due to Kozen [34, 35] and Saheb-Djahromi [43, 44]. These are concerned with domain theory and programming languages rather than with process equivalences, but they both introduced nontrivial measure-theoretic ideas. Kozen's paper [35] introduced a probabilistic dynamic logic — however our logic derives from the modal logic of Larsen and Skou [36].

The first paper with an abstract categorical approach to stochastic processes is by Giry [22], though it did not deal with process equivalences. The study of the interaction of probability and nondeterminism, largely in the context of exact equivalence of probabilistic processes, has been explored extensively using different models of concurrency. Probabilistic process algebras add randomness to the process algebra models—see for example [25, 24, 30, 31, 36, 26, 4, 47, 9, 8]. Probabilistic Petri nets [39, 48] add Markov chains to the underlying Petri net model. Probabilistic extensions of IO Automata [45, 49] have also been developed.

By and large, the above work focuses on discrete state systems. An early investigation into continuous state spaces was by deVink and Rutten [10]. While their work applies only to *ultrametric* spaces, it should be interesting to explore whether their coalgebraic approach can be extended to ordinary metric spaces such as the reals.

Most work on continuous-time systems has focused on discrete-space continuous time systems of the sort one sees in queuing theory. The pioneering work is by Jane Hillston [27] on developing a process algebra for performance evaluation, using a variant of bisimulation for process equivalence. We are currently investigating the applicability of our approximation techniques in this area, while looking at distributions other than the exponential distribution.

Baier, Katoen and Hermanns [5] have used approximate reasoning for continuous-time Markov chains for verifying **PCTL*** formulas. While they point out the problems in using approximation for logical reasoning, they do not provide an explicit example. In terms of the domain theory, the work of Baier and Kwiatkowska [6] is closest to this paper. They study the solution of a similar recursive domain equation in a variety of categories. The focus of their paper is the interaction of nondeterminism and probability in the discrete state-space setting — they do not consider the issues of continuous state spaces and approximations.

We have already mentioned the relevance of the foundational work of Edalat and of classical measure theory to our work in the introduction.

In previous work we had developed a theory of metrics between **PLMP**s [15] for robust reasoning on probabilistic processes. Intuitively, the metrics of [15] measured the closeness of the probability numbers of processes. The metric distance in this paper is unrelated to the closeness of probability numbers and have very different convergence properties, eg. some of the metrics of [15] do not yield Polish spaces.

7. Acknowledgements

We have benefited from discussions with Samson Abramsky, Franck van Breugel, Martin Escardo, Reinhold Heckmann, Achim Jung and Gordon Plotkin. We thank Falk Bartels for pointing out an error in an earlier draft.

References

[1] S. Abramsky. A domain equation for bisimulation. *Information and Computation*, 92(2):161–218, 1991.

[2] M. Alvarez-Manilla, A. Edalat, and N. Saheb-Djahromi. An extension result for continuous valuations. *Electronic Notes in Theoretical Computer Science*, 13, 1998.

[3] A. Aziz, V. Singhal, F.Balarin, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. It usually works:the temporal logic of stochastic systems. In *Proceedings of the Conference on Computer-Aided Verification*, 1995.

[4] J.C.M. Baeten, J.A. Bergstra, and S.A. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities. *Information and Computation*, 121(2):234–255, 1995.

[5] C. Baier, J-P. Katonen, and H. Hermans. Approximative symbolic model checking of continuous-time Markov chains. In J. C. M. Baeten and S. Mauw, editors, *Proceedings of the 10th International Conference on Concurrency, CONCUR99*, number 1664 in Lecture Notes In Computer Science, pages 146–161. Springer-Verlag, 1999.

[6] C. Baier and M. Kwiatkowska. Domain equations for probabilistic processes. Available from URL http://www.cs.bham.ac.uk/ mzk/, March 1996.

[7] R. Blute, J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. In *Proceedings of the Twelfth IEEE Symposium On Logic In Computer Science, Warsaw, Poland.*, 1997.

[8] R. Cleaveland. Fully abstract characterizations of testing preorders for probabilistic processes. In *CONCUR 94*, number 836 in Lecture Notes In Computer Science, pages 497–512. Springer-Verlag, August 1994.

[9] R. Cleaveland, S. Smolka, and A. Zwarico. Testing preorders for probabilistic processes. In *Proceedings of the International Colloquium On Automata Languages And Programming 1992*, number 623 in Lecture Notes In Computer Science. Springer-Verlag, 1992.

[10] E. de Vink and J. J. M. M. Rutten. Bisimulation for probabilistic transition systems: A coalgebraic approach. In *Proceedings of the 24th International Colloquium On Automata Languages And Programming*, 1997.

[11] J. Desharnais. *Labelled Markov Processes*. PhD thesis, McGill University, November 1999.

[12] J. Desharnais. Logical characterization of simulation for Markov chains. In M. Kwiatkowska, editor, *Probmiv99, Proceedings of the Second International Workshop on Probabilistic Methods in Verification*, pages 33–48. The University of Birmingham, 1999. Available as TR: CSR-99-8.

[13] J. Desharnais, A. Edalat, and P. Panangaden. A logical characterization of bisimulation for labeled Markov processes. In *proceedings of the 13th IEEE Symposium On Logic In Computer Science, Indianapolis*, pages 478–489. IEEE Press, June 1998.

[14] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labeled Markov processes. *Information and Computation*, 2000.

[15] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labeled Markov processes. In *Proceedings of CONCUR99*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

[16] Abbas Edalat. Domain of computation of a random field in statistical physics. In C. Hankin, I. Mackie, and R. Nagarajan, editors, *Theory and Formal Methods 1994: Proceedings of the second Imperial College Department of Computing Workshop on Theory and Formal Methods*, pages 11–14. IC Press, 1994.

[17] Abbas Edalat. Domain theory and integration. *Theoretical Computer Science*, 151:163–193, 1995.

[18] Abbas Edalat. Domain theory in stochastic processes. In *Proceedings of the Tenth Annual IEEE Symposium On Logic In Computer Science*. IEEE Computer Society Press, 1995.

[19] Abbas Edalat. Dynamical systems, measures and fractals via domain theory. *Information and Computation*, 120(1):32–48, 1995.

[20] A. Einstein. The theory of the brownian movement. *Ann. der Physik*, 17:549, 1905.

[21] W. Feller. *An Introduction to Probability Theory and its Applications II*. John Wiley and Sons, 2nd edition, 1971.

[22] M. Giry. A categorical approach to probability theory. In B. Banaschewski, editor, *Categorical Aspects of Topology and Analysis*, number 915 in Lecture Notes In Mathematics, pages 68–85. Springer-Verlag, 1981.

[23] V. Gupta, R. Jagadeesan, and P. Panangaden. Stochastic processes as concurrent constraint programs. In *Proceedings of the 26th Proceedings Of The Annual ACM Symposium On Principles Of Programming Languages*, 1999.

[24] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[25] Hans A. Hansson. *Time and Probability in Formal Design of Distributed Systems*, volume 1 of *Real-time Safety-critical Systems*. Elseiver, 1994.

[26] S. Hart and M. Sharir. Probabilistic propositional temporal logics. *Information and Control*, 70:97–155, 1986.

[27] J. Hillston. *A Compositional Approach to Performance Modelling*. PhD thesis, University of Edinburgh, 1994. To be published as a Distinguished Dissertation by Cambridge University Press.

[28] C. Jones. *Probabilistic Non-determinism*. PhD thesis, University of Edinburgh, 1990. CST-63-90.

[29] C. Jones and G. D. Plotkin. A probabilistic powerdomain of evaluations. In *Proceedings of the Fourth Annual IEEE Symposium On Logic In Computer Science*, pages 186–195, 1989.

[30] B. Jonsson and K. Larsen. Specification and refinement of probabilistic processes. In *Proceedings of the 6th Annual IEEE Symposium On Logic In Computer Science*, 1991.

[31] B. Jonsson and W. Yi. Compositional testing preorders for probabilistic processes. In *Proceedings of the 10th Annual IEEE Symposium On Logic In Computer Science*, pages 431–441, 1995.

[32] A. Jung. *Cartesian Closed Categories of Domains*. PhD thesis, Technischen Hochschule Darmstadt, 1988.

[33] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Van Nostrand, 1960.

[34] D. Kozen. Semantics of probabilistic programs. *Journal of Computer and Systems Sciences*, 22:328–350, 1981.

[35] D. Kozen. A probabilistic PDL. *Journal of Computer and Systems Sciences*, 30(2):162–178, 1985.

[36] K. G. Larsen and A. Skou. Bisimulation through probablistic testing. *Information and Computation*, 94:1–28, 1991.

[37] J. D. Lawson. Valuations on continuous lattices. In R.-E. Hoffman, editor, *Continuous lattices and related topics*, volume 27 of *Mathematik Arbeitspapiere*, pages 204–225. Universität Bremen, 1982.

[38] J. D. Lawson. Spaces of maximal points. *Mathematical Structures in Computer Science*, 7(5):543–555, October 1997.

[39] M. Ajmone Marsan. Stochastic petri nets: an elementary introduction. In *Advances in Petri Nets 1989*, pages 1–29. Springer-Verlag, 1989.

[40] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[41] B. Øksendal. *Stochastic Differential Equations*. Springer-Verlag, 1995.

[42] K. R. Parthasarathy. *Probability Measures on Metric Spaces*. Academic Press, 1967.

[43] N. Saheb-Djahromi. Probabilistic LCF. In *Mathematical Foundations Of Computer Science*, number 64 in Lecture Notes In Computer Science. Springer-Verlag, 1978.

[44] N. Saheb-Djahromi. Cpos of measures for nondeterminism. *Theoretical Computer Science*, 12(1):19–37, 1980.

[45] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, Dept. of Electrical Engineering and Computer Science, 1995. Also appears as technical report MIT/LCS/TR-676.

[46] M. Smoluchowski. Brownian motion. *Ann. der Physik*, 21:756, 1906.

[47] R. van Glabbeek, S.A. Smolka, and B.U. Steffen. Reactive, generative, and stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, 1995.

[48] N. Viswanadham and Y. Narahari. *Performance Modeling of Automated Manufacturing Systems*. Prentice-Hall Inc, 1992.

[49] S.-H. Wu, S.A. Smolka, and E. W. Stark. Composition and behaviors for probabilistic I/O automata. *Theoretical Computer Science*, 176(1–2):1–36, April 1997.