Metrics for Labeled Markov Systems

Josée Desharnais*
School of Computer Science
McGill University
Montreal, Quebec, Canada

Vineet Gupta†
Autonomous Systems Group
NASA Ames Research Center
Moffett Field CA 94035, USA

Radha Jagadeesan
Dept. of Math. and Computer Sciences
Loyola University-Lake Shore Campus
Chicago IL 60626, USA

Prakash Panangaden*
School of Computer Science
McGill University
Montreal, Quebec, Canada

February 26, 1999

Abstract

Partial Labeled Markov Chains are simultaneously generalizations of process algebra and of traditional Markov chains. They provide a foundation for interacting discrete probabilistic systems, the interaction being synchronization on labels as in process algebra. Existing notions of process equivalence are too sensitive to the exact probabilities of various transitions. This paper addresses contextual reasoning principles for reasoning about more robust notions of "approximate" equivalence between concurrent interacting probabilistic systems.

- We develop a family of metrics between partial labeled Markov chains to formalize the notion of distance between processes.
- We show that processes at distance zero are bisimilar.
- We describe a decision procedure to compute the distance between two processes.
- We show that reasoning about approximate equivalence can be done compositionally by showing that process combinators do not increase distance.
- We introduce an asymptotic metric to capture asymptotic properties of Markov chains; and show that parallel composition does not increase asymptotic distance.

Keywords: Concurrency, probability, Markov chains, logics of programs, metrics.

*Research supported in part by NSERC.
†Caelum Research Corporation

1 Introduction

Probability, like nondeterminism, is an abstraction mechanism used to hide inessential or unknown details. Statistical mechanics — originated by Boltzmann, Gibbs, Maxwell and others — is the fundamental successful example of the use of the probabilistic abstraction. Computer science, and process algebraic theories in particular, are focussed on providing compositional reasoning techniques. Our investigations are concerned with the development of contextual reasoning principles for concurrent interacting probabilistic systems. Consider the following paradigmatic examples.

Example 1.1 *[AJKvO97] analyzes a component (say c) of the Lucent Technologies' 5ESS® telephone switching system that is responsible for detecting malfunctions on the hardware connections between switches*[1]*. This component responds to alarms generated by another complicated system that is only available as a black-box. A natural model to consider for the black-box is a stochastic one, which represents the timing and duration of the alarm by random variables with a given probability distribution. [AJKvO97] then shows that the desired properties hold with extremely high probability, showing that the component being analyzed approximates the desired idealized behavior (say i) with sufficient accuracy.*

Example 1.2 *Consider model-based diagnosis settings. Often information about* failure models *and their associated probabilities is obtained from field studies and studies of manufacturing practices. Failure models can be incorporated by assigning a variable, called the* mode *of the component, to represent the physical state of the component, and associating a failure model with each value of the mode variable. Probabilistic information can be incorporated by letting the mode vary according to the given probability distribution [dKW89]. The diagnostic engine computes the most probable diagnostic hypothesis, given observations about the current state of the system.*

These examples illustrate the modes of contextual reasoning that interest us. In the first example, we are interested in exploring whether c can substitute for i in arbitrary program contexts; *i.e.* for some context $C[]$, does $C[c]$ continue to approximate $C[i]$. Similarly, in the second example, we are looking to see the extent to which systems with similar failure behaviors are intersubstitutable. Such a question perforce generalizes the study of congruences elaborated by the theory of concurrency. The theory of concurrency performs a study of "exactly intersubstitutable" processes with temporal behavior. In the probabilistic context, the extant notions of bisimulation (or any process equivalence for that matter) are too sensitive to the probabilities; a slight perturbation of the probabilities would make two systems non-bisimilar. The examples motivate a shift to the study of the more robust notion of "approximately intersubstitutable".

The next example illustrates a deeper interaction of the temporal and probabilistic behavior of processes.

Example 1.3 *Consider a producer and a consumer process connected by a buffer, where the producer is say a model of a network. Examples of this kind are studied extensively in the performance modeling of systems. In a model of such a system, probability serves to abstract the details of the producer (resp. consumer) process by considering rates of production (resp. consumption) of data based on empirical information. This model can be analyzed to calculate the number of packets lost as a function of the probabilities and the buffer size. The analysis aids in tuning system parameters, e.g. to optimize the buffer size. These studies are often couched in terms of asymptotic/stationary behavior to abstract over the transient behavior associated with system initialization (such as large bursts of communication) evident when the system begins execution.*

Such examples motivate the study of equality notions based on "eventually approximately intersubstitutable" processes.

[1] For another instance of modeling a complex environment that is best done statistically, see [Gat95].

1.1 Our results

Partial labeled Markov chains (`plMc`) are the discrete probabilistic analogs of labeled transition systems. In this model "internal choice" is modeled probabilistically and the so-called "external choice" is modeled by the indeterminate actions of the environment. The starting point of our investigation is the study of strong bisimulation for `plMc`. This study was initiated by [LS91] for `plMc` in a style similar to the queuing theory notion of "lumpability". This theory has been extended to continuous state spaces and continuous distributions [BDEP97, DEP98]. These papers showed:

- Bisimulation is an equivalence relation.
- The logic $\mathcal{L}$ given by $\phi ::= \mathsf{T} \mid \phi_1 \wedge \ \phi_2 \mid \langle a \rangle_q \phi$ is complete for bisimulation[2]

In the context of the earlier discussion, we note that probabilistic bisimulation is too "exact" for our purposes — intuitively, two states are bisimilar only if the probabilities of outgoing transitions match exactly, motivating the search for a relaxation of the notion of equivalence of probabilistic processes. Jou and Smolka [JS90] note that the idea of saying that processes that are close should have probabilities that are close does not yield a transitive relation, as illustrated by an example of van Breugel [Bre]. This leads them to propose that the correct formulation of the "nearness" notion is via a metric.

A metric d is a function that yields a real number distance for each pair of processes. It should satisfy the usual metric conditions: $d(P,Q) = 0$ implies P is bisimilar to Q, $d(P,Q) = d(Q,P)$ and $d(P,R) \leq d(P,Q) + d(Q,R)$. Inspired by the Hutchinson metric on probability measures [Hut81], we demand that d be "Lipschitz" with respect to probability numbers, an idea best conveyed via a concrete example.

Example 1.4 *Consider the family of* `plMc`*s* $\{P_\epsilon \mid 0 \leq \epsilon < r\}$ *where* $P_\epsilon = a_{r-\epsilon}.Q$, i.e. P_ϵ *is the* `plMc` *that does an* a *with probability* $r - \epsilon$ *and then behaves like* Q*. We demand that:*

$$d(P_{\epsilon_1}, P_{\epsilon_2}) \leq |\epsilon_1 - \epsilon_2|.$$

This implies that P_ϵ *converges to* P_0 *as* ϵ *tends to 0.*

Metrics on `plMc`s. Our technical development of these intuitions is based on the key idea expounded by Kozen [Koz85] to generalize logic to handle probabilistic phenomena.

Classical logic	Generalization
Truth values $\{0,1\}$	Interval $[0,1]$
Propositional function	Measurable function
State	Measure
Evaluation of prop. functions	Integration

Following these intuitions, we consider a class $\mathcal{F}$ of functions that assign a value in the interval $[0,1]$ to states of a `plMc`. These functions are inspired by the formulas of $\mathcal{L}$ — the result of evaluating these functions at a state corresponds to a quantitative measure of the extent to which the state satisfies a formula of $\mathcal{L}$. The identification of this class of functions is a key contribution of this paper, and motivates a metric d:

$$d(P,Q) = \sup\{|f(s_P) - f(s_Q)| \mid f \in \mathcal{F}\}.$$

In section 4, we formalize the above intuitions to define a family of metrics $\{d^c \mid c \in (0,1]\}$. These metrics support the spectrum of possibilities of relative weighting of the two factors that contribute to the

[2] a is a label, q is a rational. $\langle a \rangle_q \phi$ holds in a state s if s has probability $> q$ of making an a-transition to the set of states satisfying ϕ. Note that such a characterization of bisimulation using a negation-free logic is a new result even for discrete systems.

distance between processes: the complexity of the functions distinguishing them versus the amount by which each function distinguishes them. d^1 captures only the differences in the probability numbers; probability differences at the first transition are treated on par with probability differences that arise very deep in the evolution of the process. In contrast, d^c for $c < 1$ give more weight to the probability differences that arise earlier in the evolution of the process, i.e. differences identified by simpler functions. As c approaches 0, the future gets discounted more.

As is usual with metrics, the actual numerical values of the metric are less important than the notions of convergence that they engender [3]. Our justification of the metrics will rely on properties like the significance of zero distance, relative distance of processes, contractivity and the notion of convergence rather than a detailed justification of the exact numerical values.

Example 1.5 *Consider the* `plMc` P *with two states, and a transition going from the start state to the other state with probability* p*. Let* Q *be a similar process, with the probability* q*. Then in section 4, we show that* $d^c(P,Q) = c|p-q|$*. Now if we consider* P' *with a new start state, which makes a* b *transition to* P *with probability 1, and similarly* Q' *whose start state transitions to* Q *on* b *with probability 1, then* $d^c(P',Q') = c^2|p-q|$*, showing that the next step is discounted by* c*.*

Each of these metrics agree with bisimulation:

$$d^c(P,Q) = 0, \text{ iff } P \text{ and } Q \text{ are bisimilar.}$$

For $c < 1$, we show how to evaluate $d^c(P,Q)$ to within an ϵ-error for finite state processes P, Q.

An "asymptotic" metric on `plMc`**.** The d^c metric (for $c < 1$) is heavily influenced by the initial transitions of a process — processes which can be differentiated early are far apart. For each $c \in (0,1]$, we define a dual metric d^c_∞ (Section 6) on `plMc`*s* to capture the idea that processes are close if they have the same behavior "eventually", thus disregarding their initial behavior. Informally, we proceed as follows. Let P **after** s stand for the `plMc` P after exhibiting a trace s. Then, the j'th distance d^c_j between P, Q after exhibiting traces of length j is given by

$$\sup\{d^c(P \textbf{ after } s, Q \textbf{ after } s) \mid length(s) = j\}.$$

The asymptotic distance between P, Q is given by the appropriate limit of the d^c_j's:

$$d^c_\infty(P,Q) = \lim_{i\to\infty} \sup_{j>i} d^c_j(P,Q).$$

A process algebra of probabilistically determinate processes. In order to illustrate the properties of the metrics via concrete examples, we use an algebra of probabilistically determinate processes and a (bounded) buffer example coded in the algebra (Section 5). This process algebra has input and output prefixing, parallel composition and a probabilistic choice combinator. We do not consider hiding since this paper focuses on strong (as opposed to weak) probabilistic bisimulation.

We show that bisimulation is a congruence for all these operations. Furthermore, we generalize the result that bisimulation is a congruence, by showing that process combinators do not increase distance in any of the d^c metrics. Formally, let $d^c(P_i, Q_i) = \epsilon_i$. For every n-ary process combinator $C[X_1, \ldots, X_n]$, we have

$$d^c(C(P_1, \ldots, P_n), C(Q_1, \ldots, Q_n)) \le \sum_i \epsilon_i.$$

[3] We take the uniformity view of metrics, e.g. see [Bou89]. Intuitively, a uniformity captures relative distances, *e.g.* if x is closer to z than y; it ignores the numerical distances. For example, a uniformity on a metric space M is induced by the collection of sets $K_\epsilon = \{(x,y) \in M \times M \mid d(x,y) < \epsilon\}$ – note that different metrics may yield the same uniformity.

We show that the prefixing and parallel composition combinators do not increase the asymptotic distance d^c_∞. However, the probabilistic choice combinator is not contractive for d^c_∞.

Continuous systems. While this paper focuses on systems with a countable number of states, all the results extend to systems with continuous state spaces. The technical development of continuous systems requires measure theory apparatus to develop analogs of the results in section 3[4] and will be reported in a separate paper.

Related and future work. In this paper, we deal with probabilistic nondeterminism. In a probabilistic analysis, quantitative information is recorded and used in the reasoning. In contrast, a purely qualitative nondeterministic analysis does not require and does not yield quantitative information. In particular when one has no quantitative information at all, one has to work with indeterminacy — using a uniform probability distribution is not the same as expressing complete ignorance about the possible outcomes.

The study of the interaction of probability and nondeterminism, largely in the context of exact equivalence of probabilistic processes, has been explored extensively in the context of different models of concurrency. Probabilistic process algebras add a notion of randomness to the process algebra model and have been studied extensively in the traditional framework of (different) semantic theories of (different) process algebras (to name but a few, see [HJ90, JY95, LS91, HS86, BBS95, vGSS95, CSZ92]) *e.g.* bisimulation, theories of (probabilistic) testing, relationship with (probabilistic) modal logics etc. Probabilistic Petri nets [Mar89, VN92] add Markov chains to the underlying Petri net model. This area has a well developed suite of algorithms for performance evaluation. Probabilistic studies have also been carried out in the context of IO Automata [Seg95, WSS97].

In contrast to the above body of research the primary theme of this paper is the the study of intersubstitutivity of (eventually) (approximately) equivalent processes. The ideas of approximate substitutivity in this paper are inspired by the work of Jou and Smoka [JS90] referred to earlier and the ideas in the area of performance modeling as exemplified in on the work on process algebras for compositional performance modeling (see for example [Hil94]). The extension of the methods of this paper to systems which have both probability and traditional nondeterminism remains open and will be the object of future study.

The verification community has been active in developing model checking tools for probabilistic systems, for example [BLL$^+$96, BdA95, BCHG$^+$97, CY95, HK97]. Approximation techniques in the spirit of those of this paper have been explored for hybrid systems [GHJ97]. In future work, we will explore efficient algorithms and complexity results for our metrics.

Our work on the asymptotic metric is closely related to, at least in spirit, the work of Lincoln, Mitchell, Mitchell and Scedrov [LMMS98] in the context of security protocols. Both [LMMS98] and this paper consider the asymptotic behavior of a single process, rather than the limiting behavior of a probabilistically described family of processes as is performed in some analysis performed in Markov theory.

Organization of this paper The rest of this paper is organized as follows. First, in section 2, we review the notions of `plMc` and probabilistic bisimulation and associated results to make the paper self-contained. We next present (section 3) an alternate way to study processes using real-valued functions and show that this view presents an alternate characterization of probabilistic bisimulation. In section 4, we define a family of metrics, illustrate with various examples and describe a decision procedure to evaluate the metric. The following section 5 describes a process algebra of probabilistically determinate processes. We conclude with a section 6 on the asymptotic metric.

[4] In particular the results on finite detectability of logical satisfaction.

2 Background

This section on background briefly recalls definitions from previous work [BDEP97, DEP98, LS91] on partial labeled Markov processes and sets up the basic notations and framework for the rest of the paper. Our definitions are for discrete spaces, see [BDEP97] for the continuous space definitions.

Definition 2.1 *A* partial labeled Markov chain *(*`plMc`*) with a label set L is a structure $(S, \{k_l \mid l \in L\}, s)$, where S is a countable set of states, s is the start state, and $\forall l \in L. k_l : S \times S \longrightarrow [0, 1]$ is a transition function such that $\forall s \in S. \sum_t k_l(s, t) \leq 1$.*

A `plMc` *is finite if S is finite.*

There is no finite branching restriction on a `plMc`; $k_l(s, t)$ can be non-zero for countably many t's. k_l is extended to a function $S \times \mathcal{P}(S) \longrightarrow [0, 1]$ by defining: $k_l(s, A) = \sum_{t \in A} k_l(s, t)$. Given a `plMc` $P = (S, \{k_l \mid l \in L\}, s)$, we shall refer to its state set, transition probability and initial state as S_P, k_l^P and s_P respectively, when necessary.

We could have alternatively presented a `plMc` as a structure $(S, \{k_l \mid l \in L\}, \mu)$ where μ is an initial distribution on S. This notion of initial distribution is no more general than the notion of initial state. Given a `plMc` with initial distribution P, one can construct an equivalent `plMc` with initial state Q as follows. $S_Q = S_P \cup \{u\}$ where u is a new state not in S_P. u will be the start state of Q. $k_l^Q(s, t) = k_l^P(s, t)$ if $s, t \in S_P$; $k_l^Q(s, u) = 0$, and $k_l^Q(u, t) = \sum k_l^P(s, t)\mu^P(s)$. We will freely move between the notions of initial state and initial distribution. For example, when a transition on label l occurs in a `plMc` P, there is a new initial distribution given by $\mu'(t) = \sum k_l(s, t) \times \mu(s)$.

We recall the definition of bisimulation on `plMc` from [LS91].

Definition 2.2 *An equivalence relation, R, on the set of states of a* `plMc` *P is a* ***bisimulation*** *if whenever two states s_1 and s_2 are R-related, then for any label l and any R-equivalence class of states T, $k_l(s_1, T) = k_l(s_2, T)$.*

Two `plMc`*s P, Q are bisimilar if there is a bisimulation R on the disjoint union of P, Q such that $s_P \; R \; s_Q$.*

In [DEP98] it is shown that bisimulation can be characterized using a *negation free* logic $\mathcal{L}$: $\mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$, where a is an label from the set of labels L and $q \in [0, 1)$ is a rational number. Given a `plMc` $P = (S, \Sigma, k_a, s)$ we write $t \models_P \phi$ to mean that the state t satisfies the formula ϕ. The definition of the relation $\models$ is given by induction on formulas.

$$
\begin{array}{lll}
t \models_P \mathsf{T} & & \\
t \models_P \phi_1 \wedge \phi_2 & \Leftrightarrow & t \models_P \phi_1, \quad t \models_P \phi_2 \\
t \models_P \langle a \rangle_q \phi & \Leftrightarrow & \exists A \subseteq S. (\forall t' \in A. t' \models_P \phi) \wedge (q < k_a(t, A)).
\end{array}
$$

In words, $t \models_P \langle a \rangle_q \phi$ if the system P in state t can make an a-move to a set of states that satisfy ϕ with probability strictly greater than q. We write $[\![\phi]\!]_P$ for the set $\{s \in S_P | s \models \phi\}$. We often omit the P subscript when no confusion can arise. The results of [DEP98] relevant to the current paper are:

- Two `plMc`s are bisimilar if and only if their start states satisfy the same formulas.
- [DEP98] also shows how to construct the maximal autobisimulation on a given system. In the finite state case, this yields a state minimization construction.

The following example helps to illustrate some of the key aspects of the logic.

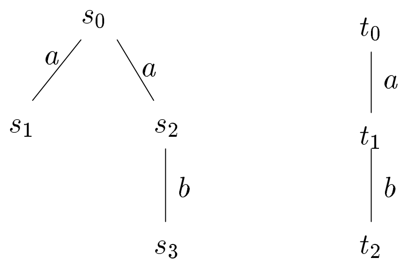

Figure 1: Two processes which cannot be distinguished without negation in HML.

Example 2.3 *(Example from [DEP98]) Consider the processes shown in figure 1. They are both nonprobabilistic processes. It is well known that they cannot be distinguished by a negation-free formula of Hennessy-Milner logic; the process on the left satisfies $\langle a \rangle \neg \langle b \rangle \top$ while the process on the right does not. However, for no assignment of probabilities are the two processes going to be bisimilar. Suppose that the two a-labeled branches of the left hand process are given probabilities p and q, assume that the b-labeled transitions have probability 1. Now if the right hand process has its a-labeled transition given a probability anything other than $p + q$, say $r > p + q$ we can immediately distinguish the two processes by the formula $\langle a \rangle_{p+q} \top$ which will not be satisfied by the left hand process. If $r = p + q$ then we can use the formula $\langle a \rangle_{r'} \langle b \rangle_{1/2} \top$, where $q < r' < r$. The left hand process cannot satisfy this formula but the right hand one does unless $p = 0$ in which case the processes are bisimilar.*

3 An alternate characterization of probabilistic bisimulation

In this section, following Kozen [Koz85], we present an alternate characterization of probabilistic bisimulation using functions into the reals instead of the logic $\mathcal{L}$. We first show that for countably infinite `plMc`s, we can work with their finite sub-`plMc`s. Then we define a set of functions which are sufficient to characterize bisimulation. It is worth clarifying our terminology here. We define a set of *functional expressions* by giving an explicit syntax. A functional expression becomes a function when we interpret it in a system. Thus we may loosely say “the same function” when we move from one system to another. What we really mean is the “same functional expression”; obviously it cannot be the same function when the domains are different. This is no different from having syntactically defined formulas of some logic which become boolean-valued functions when they are interpreted on a structure.

Logical satisfaction is finitely detectable

Definition 3.1 *P is a sub-*`plMc` *of Q if $S_P \subseteq S_Q$ and $(\forall l)$ $[k_l^P(s,t) \leq k_l^Q(s,t)]$*

Thus, a sub `plMc` of a `plMc` has fewer states and lower probabilities. The logic $\mathcal{L}$, since it does not have negation, satisfies a basic monotonicity property with respect to substructures.

Lemma 3.2 *If P is a sub-*`plMc` *of Q, then $(\forall s \in S_P)$ $[s \models_P \phi \ \Rightarrow \ s \models_Q \phi]$*

Proof. The proof proceeds by induction on ϕ. It is immediate for $\top$ and conjunction. Let $s \models_P \langle a \rangle_q.\psi$. Then, we deduce:

$$\begin{array}{rcll} s \models_P \langle a \rangle_q.\psi & \Rightarrow & q < k_a^P(s, [\![\psi]\!]_P) & \\ & \Rightarrow & q < k_a^Q(s, [\![\psi]\!]_P) & P \text{ is a sub-plmc of } Q \\ & \Rightarrow & q < k_a^Q(s, [\![\psi]\!]_Q) & \text{by induction on } \psi \\ & \Rightarrow & s \models_Q \langle a \rangle_q.\psi. & \end{array}$$

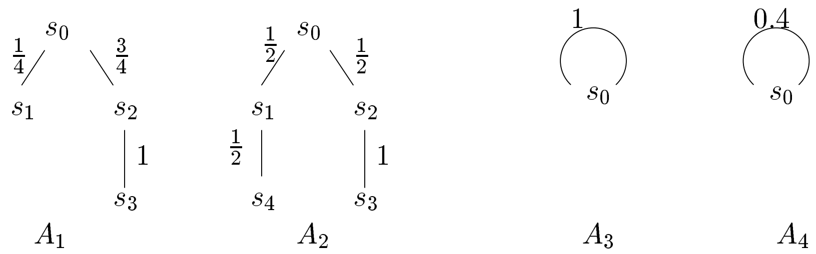

Figure 2: Examples of plMcs

∎

Every formula satisfied in a state of a plMc is witnessed by a finite sub-plMc.

Lemma 3.3 *Let P be a* plMc, *$s \in S_P$, such that $s \models_P \phi$. Then there exists a finite sub-*plMc *of P, Q^s_ϕ, such that $s \in S_Q$, and $s \models_Q \phi$.*

Proof. The proof is by induction on ϕ. For T, the one state plMc containing s suffices. For $\phi_1 \wedge \ \phi_2$, we take the union of the finite plMcs, $Q^s_{\phi_1}, Q^s_{\phi_2}$ given by the induction hypothesis. Note that lemma 3.2 ensures that the plMc so constructed satisfies $s \models \phi_1$ and $s \models \phi_2$.

Let $s \models_P \langle a \rangle_q.\psi$. Then, since $q < k^P_a(s, [\![\psi]\!]_P)$, there is a finite subset $U = \{s_1, \ldots, s_n\} \subseteq [\![\psi]\!]_P$, such that $q < k^P_a(s, U)$. The required finite plMc, $Q^s_{\langle a \rangle_q.\psi}$ is now constructed by taking the unions of the finite plMcs, $Q^{s_1}_\psi, \ldots, Q^{s_n}_\psi$, adding state s and the transitions from s to s_i for $i = 1 \ldots \ n$. ∎

We now give the class of functional expressions. First, some notation. Let $\lfloor r \rfloor_q = r - q$ if $r > q$, and 0 otherwise. $\lceil r \rceil^q = q$ if $r > q$, and r otherwise. Note that $\lfloor r \rfloor_q + \lceil r \rceil^q = r$.

For each $c \in (0, 1]$, we consider a family $\mathcal{F}^c$ of functional expressions generated by the following grammar. Here q is a rational in $[0, 1]$.

$$
\begin{array}{rcll}
f^c & ::= & \lambda s.1 & \text{Constant schema} \\
 & | & \lambda s. \min(f^c_1(s), f^c_2(s)) & \text{Min schema} \\
 & | & \lambda s.c \times \sum_{t \in S} k_a(s,t) f^c(t) & \text{Prefix schema} \\
 & | & \lambda s. \lfloor f^c(s) \rfloor_q \ | \ \lambda s. \lceil f^c(s) \rceil^q & \text{Conditional schema.}
\end{array}
$$

The functional expressions generated by these schemas will be written as $1, min(f_1, f_2), \langle a \rangle.f, \lfloor f \rfloor_q$ and $\lceil f \rceil^q$ respectively. One can informally associate functional expressions with every connective of the logic $\mathcal{L}$ in the following way — the precise formalization will be presented in lemma 3.7. T is represented by $\lambda s.1$ and conjunction by min. The contents of the connective $\langle a \rangle_q$ is split up into two expression schemas: the $\langle a \rangle.f$ schema that intuitively corresponds to prefixing and the conditional schema $\lfloor f \rfloor_q$ that captures the "greater than q" idea.

Given a plMc P, any expression $f^c \in \mathcal{F}^c$ induces a function $f^c_P : S_P \to [0, 1]$.

Example 3.4 *Consider the* plMcs *A_1 and A_2 of figure 2. All transitions are labeled a. The functional expression $(\langle a \rangle.1)^c$ evaluates to c at states s_0, s_2 of both A_1 and A_2; it evaluates to 0 at states s_1, s_3 of A_1 and s_3, s_4 of A_2, and it evaluates to $c/2$ at state s_1 of A_2. The functional expression $(\langle a \rangle.\langle a \rangle.1)^c$ evaluates to $3c^2/4$ at states s_0 of A_1, A_2 and to 0 elsewhere. The functional expression $(\langle a \rangle.\lfloor \langle a \rangle.1 \rfloor_{\frac{1}{2}})^c$ evaluates to $3c^2/8$ at state s_0 of A_1 and to $c^2/4$ at state s_0 of A_2.*

Example 3.5 *Consider the* $\texttt{plMc}$ A_3 *of figure 2. All transitions are labeled* a. *A functional expression of the form* $(\underbrace{\langle a\rangle.\ldots.\langle a\rangle}_{n}.1)^c$ *evaluates to* c^n *at state* s_0. *On state* s_0 *of* $\texttt{plMc}$ A_4 *the same functional expression evaluates to* $(c \times 0.4)^n$.

The following lemma is the functional analog of lemma 3.2.

Lemma 3.6 *If* P *is a sub-*$\texttt{plMc}$ *of* Q, *then* $(\forall s \in S_P)\ [f_P^c(s) \leq\ f_Q^c(s)]$.

The proof is a routine induction on the construction of the functional expression f^c and is omitted.

Lemma 3.7 *Given any* $\phi \in \mathcal{L}$ *and a finite* $\texttt{plMc}$ P, *and any* $c \in (0,1]$, *there is a functional expression* $f^c \in \mathcal{F}^c$ *such that*

1. $\forall s \in S_P.f_P^c(s) > 0$ *iff* $s \models_P \phi$.
2. *for any* $\texttt{plMc}$ Q, $\forall s \in S_Q.s \not\models_P \phi \Rightarrow f_Q^c(s) = 0$.

Proof. The proof is by induction on the structure of ϕ. If $\phi = \mathsf{T}$, the functional expression $\lambda s.1$ suffices. If $\phi = \psi_1 \wedge \psi_2$, let f_1^c and f_2^c be the functional expressions corresponding to ψ_1 and ψ_2. Then the functional expression $\lambda s.\min(f_1^c(s), f_2^c(s))$ satisfies the conditions.

If $\phi = \langle a\rangle_q.\psi$, let g^c be the functional expression corresponding to ψ yielded by induction. Let S_ψ be the set of states in P satisfying ψ, and let $x = \min\{g(s) \mid s \in S_\psi\}$. By induction hypothesis, $x > 0$. Consider the functional expression f^c given by $\lfloor\langle a\rangle.\lceil g\rceil^x\rfloor_{cxq}$. For all $t \in [\![\psi]\!]$, $(\lceil g\rceil^x)(t) = x$. Now for any state $s \in S_P$,

$$(\langle a\rangle.\lceil g\rceil^x)^c(s) = cx \sum_{t \in [\![\psi]\!]} k_a(s,t) = cxk_a(s,[\![\psi]\!]).$$

Now for each state $s \in [\![\phi]\!]$, $k_a(s,[\![\psi]\!]) > q$. Thus f^c satisfies the first condition.

The second condition holds because for any state s in Q, $(\langle a\rangle.\lceil g\rceil^x)(s) \leq cxk_a(s,[\![\psi]\!]_Q)$, so if $k_a(s,[\![\psi]\!]_Q) \leq q$ then $(\lfloor\langle a\rangle.\lceil g\rceil^x\rfloor_{cxq})(s) = 0$. ∎

Corollary 3.8 *For any* $\texttt{plMc}$ P *and state* $s \in S_P$, *if* $s \models_P \phi$ *then there exists* $f^c \in\ \mathcal{F}^c$ *such that* $f_P^c(s) > 0$ *and* $(\forall \texttt{plMc}\ R)\ (\forall s \in S_R)\ f_R^c(s) > 0 \Rightarrow\ s \models_R \phi$.

Proof. Let s be a state in $\texttt{plMc}$ P such that $s \models_P \phi$. By lemma 3.3, there is a finite sub-$\texttt{plMc}$ Q of P such that $s \models_Q \phi$. By lemma 3.7, $\exists f^c \in \mathcal{F}^c$ such that $f_Q^c(s) > 0$ and for any $\texttt{plMc}$ R, $\forall s \in S_R.s \not\models \phi \Rightarrow f_R^c(s) = 0$. By lemma 3.6, $f_P^c(s) > 0$, so f^c satisfies the conditions required by the lemma. ∎

Theorem 3.9 *For any* $\texttt{plMc}$ P, $(\forall c \in (0,1])$, $\forall s, s' \in S_P$

$$[(\forall \phi \in \mathcal{L})\ s \models_P \phi \Leftrightarrow s' \models_P \phi] \Leftrightarrow (\forall f \in \mathcal{F}^c)\ [f_P^c(s) = f_P^c(s')].$$

Proof. Let $(\forall \phi \in \mathcal{L})\ s \models_P \phi \Leftrightarrow s' \models_P \phi$. Then, by the results of [DEP98], there is a bisimulation R such that s, s' are in the same equivalence class. We now show that for any bisimulation R, sRs' implies that $(\forall f \in \mathcal{F}^c)\ [f_P^c(s) = f_P^c(s')]$. The proof proceeds by induction on the structure of the function expression f^c. The key case is when f^c is of the form $(\langle a\rangle.g)^c$. Let E_i be the R-equivalence classes. Then:

$$\begin{array}{rcll} f_P^c(s) & = & c \times \sum_{t \in S} k_a(s,t)g^c(t) & \\ & = & c \times \sum_i \sum_{t \in E_i} k_a(s,t)g^c(t) & \\ & = & c \times \sum_i [g^c(E_i) \times k_a(s,E_i)] & \text{by induction, } g^c \text{ is constant on } E_i \\ & = & c \times \sum_i [g^c(E_i) \times k_a(s',E_i)] & \text{from } s\ R\ s' \\ & = & c \times \sum_i \sum_{t \in E_i} k_a(s',t)g^c(t) & = f_P^c(s'). \end{array}$$

For the converse, let ϕ be such that $s \models_P \phi$ and $s' \not\models_P \phi$. By corollary 3.8, there is a functional expression f^c such that $f_P^c(s) > 0$ and $f_P^c(s') = 0$. ∎

Example 3.10 *Consider the* `plMcs` *A_1, A_2 of figure 2. The calculations of example 3.4 show that the s_0 states of A_1, A_2 are distinguishable. Furthermore, the states are indistinguishable if we use only the function schemas Constant, Min and Prefixing. Thus, example 3.4 shows that the conditional functional expressions are necessary.*

4 A Metric on Processes

Each collection of functional expression $\mathcal{F}^c$ be the set of all such expressions induces a distance function as follows:

$$d^c(P,Q) = \sup_{f^c \in \mathcal{F}^c} |f^c_P(s_P) - f^c_Q(s_Q)|.$$

Theorem 4.1 *For all $c \in (0,1]$, d^c is a metric.*

Proof. The transitivity and symmetry of d^c is immediate. $d^c(P,Q) = 0$ iff P and Q are bisimilar follows from theorem 3.9. ∎

Example 4.2 *The analysis of example 3.10 yields $d^c(A_1, A_2) = c^2/8$.*

Example 4.3 *Example 3.5 shows the fundamental difference between the metrics $d^c, c < 1$ and d^1. For $c < 1$, $d^c(A_3, A_4)$ is witnessed by $(\langle a \rangle.1)^c$ and is given by $d^c(A_3, A_4) = 0.6c$. In contrast, $d^1(A_3, A_4) = \sup\{1 - (0.4)^n \mid n = 0, 1, \dots\} = 1$. What this shows is that the notion of convergence is different for the two metrics. If we had a family of processes like $A4$ with the transition probability given by $1 - \frac{1}{m}$ the distance of these processes from $A3$ would always be 1, hence they would not converge to $A3$ in the d^1 metric, but they would converge to $A3$ in any d^c metric with $c < 1$. Thus the d^1 metric defines a different topology than do the other metrics.*

Example 4.4 *(Analysis of Example 1.4) Consider the family of* `plMcs` *$\{P_\epsilon \mid 0 \le \epsilon < r\}$ where $P_\epsilon = a_{r-\epsilon}.Q$,* i.e. *$P_\epsilon$ is the* `plMc` *that does an a with probability $r - \epsilon$ and then behaves like Q. The function expression $(\langle a \rangle.1)^c$ evaluates to $(r - \epsilon)c$ at P_ϵ. This functional expression witnesses the distance between any two P's (other functions will give smaller distances). Thus, we get $d(P_{\epsilon_1}, P_{\epsilon_2}) = c|\epsilon_1 - \epsilon_2|$. This furthermore ensures that P_ϵ converges to P_0 as ϵ tends to 0.*

Example 4.5 *(from [DEP98]) Consider the* `plMcs` *P (left) and Q (right) of figure 3. Q is just like P except that there is an additional transition to a state which then has an a-labeled transition back to itself. The probability numbers are as shown. If both* `plMcs` *have the same values on all functional expressions we will show that $q_\infty = 0$,* i.e. *it really cannot be present. The functional expression $(\langle a \rangle.1)^c$ yields $c(\sum_{i \ge 0} p_i)$ on P and $c(q_\infty + \sum_{i \ge 0} q_i)$ on Q. The functional expression $(\langle a \rangle.\langle a \rangle.1)^c$ yields $c^2(\sum_{i \ge 1} p_i)$ on P and $c^2(q_\infty + \sum_{i \ge 2} q_i)$ on Q. Thus, we deduce that $p_0 = q_0$. Similarly, considering functional expressions $(\langle a \rangle.\langle a \rangle.\langle a \rangle.1)^c$ etc, we deduce that $p_n = q_n$. Thus, $q_\infty = 0$.*

A decision procedure for $d^c, c < 1$. Given finite `plMcs` P, Q, we now provide a decision procedure for computing $d^c(P,Q)$ for $c < 1$ to any desired accuracy c^n, where n is a natural number. We do this by computing $sup_F|f^c(s_P) - f^c(s_Q)|$ for a finite set of functions F, and then show that for this F, $d^c(P,Q) - sup_F|f^c(s_P) - f^c(s_Q)| \le c^n$.

Define the depth of a functional expression inductively as follows: $depth(\lambda s.1) = 0$, $depth(min(f^c_1, f^c_2)) = max(depth(f^c_1), depth(f^c_2))$ and $depth(\lfloor f^c \rfloor_q) = depth(\lceil f^c \rceil^q) = depth(f^c)$, $depth(\langle a \rangle.f^c) = depth(f^c)+$

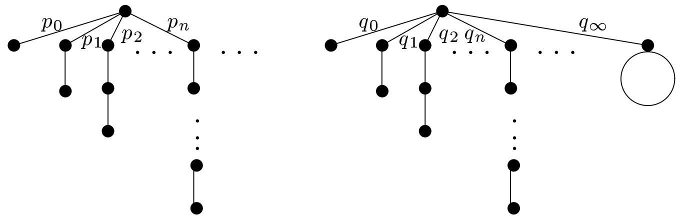

Figure 3: Probability and countable branching

1. Then it is clear that $|f^c(s_P) - f^c(s_Q)| \leq c^{depth(f)}$. Now if we include in F all functions of depth $\leq n$, then $d^c(P,Q) - sup_F|f^c(s_P) - f^c(s_Q)| \leq c^n$.

However there are infinitely many functional expressions of depth $\leq n$. We now construct a finite subset of these, such that the above inequality still holds. Let $A^i = \{\frac{k}{3^{m+1+n-i}} \mid k = 0, \ldots 3^{m+1+n-i}\}$, where $1/3^m < c^n$. We construct the set of functions inductively as follows. Let F^i be the set of all functions of depth $\leq i$. Define:

$$\begin{aligned} F_1^{i+1} &= \{\langle a \rangle.f \mid f \in F^i\} \\ F_2^{i+1} &= \{\lfloor f \rfloor_q \mid f \in F_1^{i+1}, q \in A^{i+1}\} \\ F_3^{i+1} &= F^n \cup \{\lceil f \rceil^q \mid f \in F_2^{i+1}, q \in A^{i+1}\} \end{aligned}$$

F^{i+1} is defined by closing F_3^{i+1} under pairwise mins.

We can prove that for any $f^c \in \mathcal{F}^c$ of depth $\leq n$, there is a function in F^n that approximates it closely enough.

Lemma 4.6 *Let $f^c \in \mathcal{F}^c$ be of depth $i \leq n$. Then, there exists $g_f^c \in F^i$ such that:*

$$(\forall \texttt{plMc}\ P)\ (\forall s \in S_P)\ [|f^c(s) - g_f^c(s)| \leq \frac{1}{3^{m+n-i}}].$$

Proof. The proof proceeds by induction on i. In this extended abstract, we only sketch the two basic ideas of the proof for the inductive step.

(1) The following identities show that repeating steps 2, 3, 4 on F^{i+1} does not get any new functions.

$$\begin{array}{ll} \lfloor \lfloor f \rfloor_q \rfloor_r = \lfloor f \rfloor_{q+r} & \lceil \lceil f \rceil^q \rceil^r = \lceil f \rceil^{min(q,r)} \\ \lfloor \lceil f \rceil^q \rfloor_r = \lceil \lfloor f \rfloor_r \rceil^{q-r} & \\ \lfloor min(f_1, f_2) \rfloor_r = min(\lfloor f_1 \rfloor_r, \lfloor f_2 \rfloor_r) & \lceil min(f_1, f_2) \rceil^r = min(\lceil f_1 \rceil^r, \lceil f_2 \rceil^r). \end{array}$$

(2) Define f_1, f_2 to be ϵ-close if for all states $s \in S_P \cup S_Q$, $|f_1(s) - f_2(s)| < \epsilon$. Then if f_1 and f_2 are ϵ-close, then $\langle a \rangle.f_1$ and $\langle a \rangle.f_2$ are ϵ-close, and so are $\lfloor f_1 \rfloor_q$ and $\lfloor f_2 \rfloor_q$, and also $\lceil f_1 \rceil^q$ and $\lceil f_2 \rceil^q$. In addition if f_1' and f_2' are also ϵ-close, then $min(f_1, f_1')$ and $min(f_2, f_2')$ are also ϵ-close. Furthermore,

$$|q_1 - q_2| \leq \epsilon \Rightarrow \sup\{|\lfloor f \rfloor_{q_1}(x) - \lfloor f \rfloor_{q_2}(x)|\} \leq \epsilon.$$

Similarly, for $\lceil f \rceil^{(\cdot)}$. ∎

5 Examples of metric reasoning principles

In this section, we use a process algebra and an example coded in the process algebra to illustrate the type of reasoning provided by our study.

5.1 A process algebra

The process algebra describes probabilistically determinate processes. The processes are input-enabled [LT89, Dil88, Jos92] in a weak sense ($(\forall s \in S_P)\ (\forall a \in L)\ k_{a?}(s, S_P) > 0$) and communication is via CSP style broadcast. The process combinators that we consider are parallel composition, prefixing and probabilistic choice. We do not consider hiding since this paper focuses on strong probabilistic bisimulation. Though we do not enforce the fact that output actions do not block, this assumption can safely be added to the algebra to make it an IO calculus [Vaa91]; this change does not alter the results of this section.

We assume an underlying set of labels $\mathcal{A}$. Let $L? = \{a? \mid a \in \mathcal{A}\}$ be the set of input labels, and $L! = \{a! \mid a \in \mathcal{A}\}$ the set of output labels. The set of labels are given by $L = L? \cup L!$. Every process P is associated with a subset of labels: $P_O \subseteq L!$, the set of relevant output labels. This signature is used to constrain parallel composition.

Prefixing. $P = a?_r.Q$ where r is a rational number, is the process that accepts input a and then performs as Q. The number r is the probability of accepting $a?$. With probability $(1-r)$ the process $P = a?_r.Q$ will *block* on an $a?$ label. S_P is given by adding a new state, q to S_Q. Add a transition labeled $a?$ from q to the start state of Q with probability r. For all other labels l, add a $l?$ labeled self-loop at q with probability 1. q is the start state of P.

Output prefixing, $P = a!_r.Q$, where r is a rational number, is the process that performs output action $a!$ and then functions as Q, is defined analogously. In this case, $P_O = Q_O \cup \{a!\}$.

Probabilistic choice. $P = Q +_r Q'$ is the probabilistic choice combinator [JP89] that chooses between Q, Q'; Q is chosen with probability r and Q' is chosen with probability $1-r$.

$P_O = Q_O \cup Q'_O$. $S_P = S_Q \uplus S_{Q'}$. Now $k^P_l(q, A \uplus A') = k^Q_l(q, A)$ if $q \in S_Q$, and $k^P_l(q, A \uplus A') = k^{Q'}_l(q, A')$ if $q \in S_{Q'}$. In this case, we define an initial distribution μ: $\mu(\{s_Q\}) = r, \mu(\{s_{Q'}\}) = 1-r$, referring the reader to section 2 for a way to convert the initial distribution to an initial state.

Parallel composition. $P = Q \mid\mid Q'$ is permitted if the output actions of Q, Q' are disjoint, *i.e.* $Q_O \cap Q'_O = \emptyset$. The parallel composition synchronizes on all labels in $Q_L \cap Q'_L$.

$P_O = Q_O \uplus Q'_O$. $S_P = S_Q \times S_{Q'}$. The k^P_l definition is motivated by the following idea. Let s (resp. s') be a state of Q (resp. Q'). We expect the following synchronized transitions from the product state (s, s').

$$\frac{s \xrightarrow{c?} t \quad s' \xrightarrow{c?} t'}{(s,s') \xrightarrow{c?} (t,t')} \qquad \frac{s \xrightarrow{c!} t \quad s' \xrightarrow{c?} t'}{(s,s') \xrightarrow{c!} (t,t')} \qquad \frac{s \xrightarrow{c?} t \quad s' \xrightarrow{c!} t'}{(s,s') \xrightarrow{c!} (t,t')}.$$

The disjointness of the output labels of Q, Q' ensures that there is no non-determinism. Formally, if $l = a! \in Q_O$, then $k^P_{a?}((s,s'),(t,t')) = k^P_{a!}((s,s'),(t,t')) = k^Q_{a!}(s,t) \times k^{Q'}_{a?}(s',t')$. The case when $a! \in Q'_O$ and $l = a?$ is similar.

We now show that each of the operations of the process algebra are contraction mappings with respect to the metric defined above. Since theorem 3.9 shows that $d(P,Q) = 0$ iff $P \approx Q$, this shows that bisimulation is a congruence with respect to these operations.

Theorem 5.1 *The following hold:*

1. $d^c(l_r.P, l_r.Q) \leq c d^c(P,Q)$ *for any label* l.
2. $d^c(P +_r R, Q +_r R) \leq d^c(P,Q)$ *for any* R.
3. $d^c(P \mid\mid R, Q \mid\mid R) \leq d^c(P,Q)$ *for any* R *for which the processes on the left are defined.*

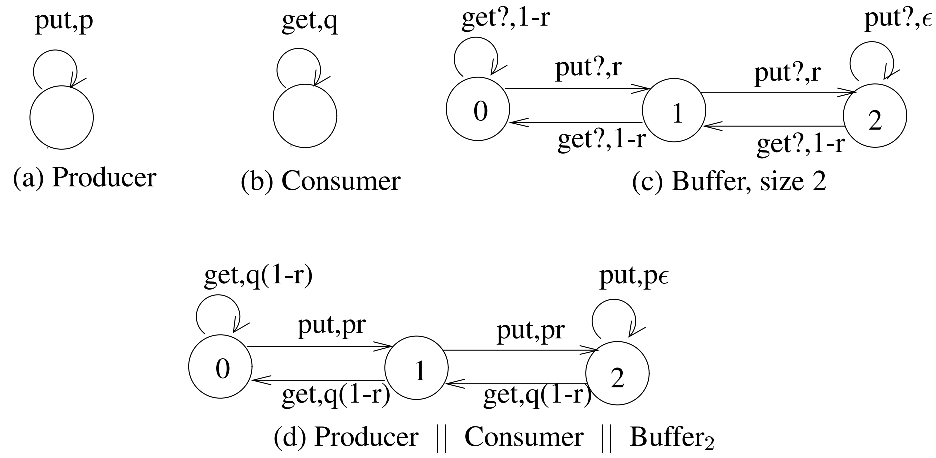

Figure 4: The producer consumer example.

Proof. The proof proceeds by induction on functional expressions. Let $d^c_{f^c}(P, Q)$ be the distance using f^c — $d^c(P, Q) = \sup_{f^c} d_{f^c}(P, Q)$. We show that for any f^c $d^c_{f^c}$ of the LHS is less than or equal to some $d^c_{g^c}$ of the RHS. In this extended abstract, we omit the detailed calculations. The key case is $f^c = \langle a \rangle.h^c$, sketched below for the parallel composition. If $a = b!$, and $b! \in R_O$, then by induction, that we know that $d^c_{h^c}(P' \mid\mid R', Q' \mid\mid R') \leq d^c_{g^c}(P', Q')$, where P', Q', R' are the same as P, Q, R but with the start distribution obtained by making a $b!$ transition on the start state in the case of R, and a $b?$ transition in the case of P, Q. Now $d^c_{f^c}(P \mid\mid R, Q \mid\mid R) = c \times d^c_{h^c}(P' \mid\mid R', Q' \mid\mid R') \leq c \times d^c_{g^c}(P', Q') = d^c_{\langle a \rangle . g^c}(P, Q)$. ∎

Lemma 5.2 *The following properties are true of our metric:*

1. $d^c(a_r.P, a_s.P) \leq c \mid r - s \mid$.
2. $d^c(P +_r Q, P +_s Q) \leq \mid r - s \mid d^c(P, Q)$.
3. $d^c(P +_r Q, P' +_r Q) \leq r d^c(P, P')$.

5.2 A bounded buffer example

We specify a producer consumer process with a bounded buffer (along the lines of [PS85]). The producer is specified by the 1 state finite automaton shown in Figure 4(a) — it outputs a put, corresponding to producing a packet, with probability p (we omit the ! in the labels). To keep the figure uncluttered, we also omit the input-enabling arcs, all of which have probability 1. The consumer (Figure 4(b)) is analogous — it outputs a get with probability q, corresponding to consuming a packet. The buffer is an n-state automaton, the states are merely used to count the number of packets in the buffer, while the probabilities code up the probability of scheduling either the producer or the consumer (thus the producer gets scheduled with probability r, and then produces a packet with probability p). Upon receiving a put in the last state, the buffer accepts it with a very small probability ϵ, modeling a blocked input. The parallel composition of the three processes is shown in Figure 4(d).

As the buffer size increases, the distance between the bounded buffer and the unbounded buffer decreases to 0. Let P_k = Producer || Consumer || Buffer$_k$, where Buffer$_k$ denotes the process Buffer with k states. Then by looking at the structure of the process, we can compute that $d(P_k, P_\infty) \propto (cpr)^k$. This allows us to conclude the following:

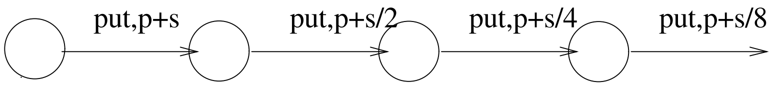

Figure 5: A producer with transient behavior

- As the bounded buffer becomes larger, it approximates an infinite buffer more closely: if $m > k$ then $d^c(P_k, P_\infty) > d^c(P_m, P_\infty)$.
- As the probability of a put decreases, the bounded buffer approximates an infinite buffer more closely. Thus if $p < p'$, $d^c(P^p, P^p_\infty) < d^c(P^{p'}, P^{p'}_\infty)$, where the superscripts indicate the producer probability.
- Similarly, as the probability of scheduling the Producer process (r) decreases, the buffer approximates an infinite buffer more closely.

6 The asymptotic metric

Let P be a `plMc`. Then P **after** a is the same `plMc` but with start distribution given by $\nu(t) = k_a(s,t)$. We perform some normalization based on the total probability of the resulting initial configuration $\nu(S)$: If $\nu(S) > 0$, it is normalized to be 1; if $\nu(S) = 0$, it is left untouched.

This definition extends inductively to P **after** s, where s is a finite sequence of labels $(a_0, a_1, a_2, \dots, a_k)$. Note that P **after** s is identical to P except that its initial configuration may be different.

Define the j distance between P, Q, $d^c_j(P,Q) = \sup\{d^c(P \textbf{ after } s, Q \textbf{ after } s) \mid length(s) = j\}$. We define the asymptotic distance between processes P and Q, $d^c_\infty(P,Q)$ to be

$$d^c_\infty(P,Q) = \lim_{i \to \infty} \sup_{j > i} d^c_j(P,Q).$$

The fact that d^c_∞ satisfies the triangle inequality and is symmetric immediately follows from the same properties for d.

Example 6.1 *For any* `plMc` *P, $d^c_\infty(a_r.P, a_s.P) = 0$, where $r, s > 0$. Consider A_3 from Figure 2. Without the normalization in the definition of A_3* **after** *s, we would have got $d^c_\infty(a_r.A_3, a_s.A_3) = c|r - s|$*

Example 6.2 *Consider the producer process P_2 shown in Figure 5. This is similar to the producer P_1 in Figure 4, except that initially the probability of producing put is more than p, however as more put's are produced, it asymptotically approaches p. If we consider the asymptotic distance between these two producers, we see that $d^c(P_2 \textbf{ after } put^n, P_1 \textbf{ after } put^n) \propto 2^{-(n+1)}$. Thus $d^c_\infty(P_1, P_2) = 0$. Now by using the compositionality of parallel composition (see below) , we see that $d^c_\infty(P_1 \mid\mid Consumer \mid\mid Buffer_k, P_2 \mid\mid Consumer \mid\mid Buffer_k) = 0$, which is the intuitively expected result.*

Parallel composition and prefixing in the process algebra are contraction mappings with respect to the metric defined above — this will show that asymptotic equivalence is preserved by these operations.

Theorem 6.3 *The following hold:*

1. $d^c_\infty(l_r.P, l_r.Q) \le d^c_\infty(P,Q)$ *for any label l.*
2. $d^c_\infty(P \mid\mid R, Q \mid\mid R) \le d_\infty(P,Q)$.

For the key case of parallel composition, the proof is based on: $(P \mid\mid Q)$ **after** $s = (P$ **after** $s_1) \mid\mid (Q$ **after** $s_2)$, where s_1 has those $a!$ labels of s replaced by $a?$ where $a! \notin P_O$, and similarly for s_2.

Acknowledgements. We have benefited from discussions with Franck van Breugel about the Hutchinson metric.

References

[AJKvO97] R. Alur, L. J. Jagadeesan, J. J. Kott, and J. E. von Olnhausen. Model-checking of real-time systems: A telecommunications application. In *Proceedings of the 19th International conference on Software Engineering*, pages 514–524, 1997.

[BBS95] 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.

[BCHG+97] C. Baier, E. Clark, V. Hartonas-Garmhausen, M. Kwiatkowska, and M. Ryan. Symbolic model checking for probabilistic processes. In *Proceedings of the 24th International Colloquium On Automata Languages And Programming*, Springer Verlag LNCS vol 1256, pages 430–440, 1997.

[BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In P. S. Thiagarajan, editor, *Proceedings of the 15th Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, Springer Verlag LNCS vol 1026, pages 499–513, 1995.

[BDEP97] 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.

[BLL+96] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi. Uppaal: A tool suite for automatic verification of real-time systems. In R. Alur, T. Henzinger, and E. Sontag, editors, *Hybrid Systems III*, Springer Verlag LNCS vol 1066, pages 232–243, 1996.

[Bou89] N. Bourbaki. *Elements of Mathematics: General Topology Chapters 1-4*. Springer-Verlag, 1989.

[Bre] F.van Breugel. private communication.

[CSZ92] R. Cleaveland, S. A. Smolka, and A. Zwarico. Testing preorders for probabilistic processes. *Lecture Notes in Computer Science*, 623, 1992.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[DEP98] 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*. IEEE Press, June 1998.

[Dil88] D. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits*. ACM Distinguished Dissertations. MIT Press, 1988.

[dKW89] Johan de Kleer and B. C. Williams. Diagnosis with behavioral modes. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1324–1330, August 1989.

[Gat95] Erann Gat. Towards principled experimental study of autonomous mobile robots. *Autonomous Robots*, 2:179–189, 1995.

[GHJ97] V. Gupta, T. A. Henzinger, and R. Jagadeesan. Robust timed automata. In Oded Maler, editor, *Hybrid and Real-Time Systems*, LNCS Vol 1201, pages 331–345. Springer Verlag, March 1997.

[Hil94] Jane Hillston. *A Compositional Approach to Performance Modelling*. PhD thesis, University of Edinburgh, 1994. To be published as a Distinguished Dissertation by Cambridge University Press.

[HJ90] H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *Proceedings of the 11th IEEE Real-Time Systems Symposium*, pages 278–287. IEEE Computer Society Press, 1990.

[HK97] M. Huth and M. Kwiatkowska. Quantitative analysis and model checking. In *proceedings of the 12 IEEE Symposium On Logic In Computer Science*, pages 111–122. IEEE Press, 1997.

[HS86] S. Hart and M. Sharir. Probabilistic propositional temporal logics. *Information and Control*, 70:97–155, 1986.

[Hut81] J. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30:713–747, 1981.

[Jos92] M. B. Josephs. Receptive process theory. *Acta Informatica*, 29(1):17–31, February 1992.

[JP89] C. Jones and G. D. Plotkin. A probabilistic powerdomain of evaluations. In *Proceedings, Fourth Annual Symposium on Logic in Computer Science*, pages 186–195, Asilomar Conference Center, Pacific Grove, California, 1989.

[JS90] C. Jou and S. Smolka. Equivalences, congruences and complete axiomatizations for probabilistic processes. In *CONCUR 90*, Springer Verlag LNCS vol 458, 1990.

[JY95] B. Jonsson and W. Yi. Compositional testing preorders for probabilistic processes. In *Proceedings, Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 431–441, San Diego, California, 1995.

[Koz85] D. Kozen. A probabilistic PDL. *Journal of Computer and Systems Sciences*, 30(2):162–178, 1985.

[LMMS98] P. Lincoln, J. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic poly-time framework for protocol analysis. In *ACM Computer and Communication Security (CCS-5)*, 1998.

[LS91] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, September 1991.

[LT89] N. A. Lynch and M. R. Tuttle. An introduction to input/output automata. *CWI Quarterly*, 2(3):219–246, 1989.

[Mar89] M. Ajmone Marsan. Stochastic petri nets: an elementary introduction. In *Advances in Petri Nets 1989*, pages 1–29. Springer, June 1989.

[PS85] J L Peterson and A Silberschatz. *Operating System Concepts*. Addison-Wesley Inc., 1985.

[Seg95] 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.

[Vaa91] F. W. Vaandrager. On the relationship between process algebra and input/output automata. In *Proceedings, Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 387–398, Amsterdam, The Netherlands, 15–18 July 1991. IEEE Computer Society Press.

[vGSS95] 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.

[VN92] N. Viswanadham and Y. Narahari. *Performance Modeling of Automated Manufacturing Systems*. Prentice-Hall Inc., 1992.

[WSS97] S.-H. Wu, S.A. Smolka, and E. Stark. Composition and behaviors for probabilistic i/o automata. *Theoretical Computer Science*, 176(1–2):1–36, April 1997.