Bisimulation for Labelled Markov Processes

José Desharnais[*] — School of Computer Science, McGill University, Montreal, Quebec, Canada, desharna@cs.mcgill.ca

Abbas Edalat[†] — Department of Computing, Imperial College, London, UK, ae@doc.ic.ac.uk

Prakash Panangaden[‡] — School of Computer Science, McGill University, Montreal, Quebec, Canada, prakash@cs.mcgill.ca

Abstract

In this paper we introduce a new class of labelled transition systems - Labelled Markov Processes - and define bisimulation for them. Labelled Markov processes are probabilistic labelled transition systems where the state space is not necessarily discrete. We assume that the state space is a certain type of common metric space called an analytic space. We show that our definition of probabilistic bisimulation generalizes the Larsen-Skou definition given for discrete systems. The formalism and mathematics is substantially different from the usual treatment of probabilistic process algebra.

The main technical contribution of the present paper is a logical characterization of probabilistic bisimulation. This study revealed some unexpected results, even for discrete probabilistic systems.

- Bisimulation can be characterized by a very weak modal logic. The most striking feature is that one has no negation or any kind of negative proposition.
- We do not need any finite branching assumption, yet there is no need of infinitary conjunction.

We also show how to construct the maximal autobisimulation on a system. In the finite state case, this is just a state minimization construction. The proofs that we give are of an entirely different character than the typical proofs of these results. They use quite subtle facts about analytic spaces, and appear, at first sight, to be entirely nonconstructive. Yet one can give an algorithm for deciding bisimilarity of finite state systems which constructs a formula that witnesses the failure of bisimulation.

1 Introduction

The study of continuous systems is becoming a more common part of computer science. Since the sixties control theory, or systems theory as it is sometimes called, has led to an abstract notion of a system; see, for example, the excellent recent text by Sontag [Son90]. These notions were general enough to include the continuous physical systems normally studied by control engineers as well as systems such as automata which were inherently discrete in nature. General concepts such as "state", "transformation" and "observation" emerged and were used in studies of both types of systems. The present work is a foundational study - from the viewpoint of process algebra - of *interacting probabilistic systems* that may be either discrete or continuous.

[*]Research supported by NSERC, Canada

[†]Research supported by EPSRC, UK.

[‡]Research Supported by NSERC, Canada and by BRICS, Aarhus, Denmark

The research reported here has been motivated by the recent interest in hybrid systems among computer scientists. See volume 1066 or volume 1273 of Springer-Verlag Lecture Notes In Computer Science [AHS96, AKNS97] for examples of recent work on hybrid systems. What we have in mind is a purely physical system or a physical system interacting with a discrete system such as a controller. As such the domain of interest seems far from traditional concurrency theory. However the investigation has been influenced by and informed by two decades of activity in process algebra. Thus, for example, we are interested in formulating when two systems are "equivalent" according to some notion of "observable interaction." We are interested in eventually being able to apply verification technology to reason about such systems.

The main applications that we see for our formalism are:

- model verification, i.e. verifying that a model - perhaps discrete - is equivalent to an underlying system being analyzed,
- state reduction, i.e. taking a model and collapsing it to an equivalent reduced model; this may well turn a continuous model into a discrete model.
- verification of properties satisfied by a model.

The immediate motivation is to explore how the ideas extend to the continuous world, so we do not claim that practitioners would be able to read the present paper and immediately apply the results. We feel that the formalism needs to be developed further in a manner driven by our experiences with modelling real systems. Since the original presentation of this work, we have been working on such case studies involving industrial partners, particularly in the verification of flight control software. It is already clear however that whatever changes we might make to the notion of system the fundamental mathematical ideas that we invoked are going to be relevant.

The systems that we consider - continuous state space and discrete time - are of interest for two reasons. First, this is a reasonable middle ground before we get to a completely continuous model. Several authors have looked at discrete space, continuous time models. It is clear that fully continuous models will force us to make a significant deepening of the mathematics. One will have to consider stochastic differential equations rather than transition systems to describe the evolution of systems. Second, these systems occur in natural examples. A significant case study that we have become involved with is the treatment of avionics software mentioned in the last paragraph. Here the system is coded as a loop with a certain fixed periodicity. Every cycle some actions are taken *and some interactions occur*. The system itself is inherently continuous state but the temporal evolution has a discrete time step imposed by the main loop. Of course the physics one is modelling occurs in continuous time, hence an understanding of continuous time would be very useful and important in the future; but the discrete time, continuous space model is in fact how many of the engineers conceptualize the system.

The notion of bisimulation is central to the study of concurrent systems. While there are a bewildering variety of different equivalence relations between processes (two-way simulation, trace equivalence, failures equivalence and many more), bisimulation enjoys some fundamental mathematical properties, most notably its characterization as a fixed-point, which make it the most discussed process equivalence. Of course there are many different variants of bisimulation itself! In the present paper we are not so much concerned with adjudicating between the rival claims of all these relations, but rather, we are concerned with showing how to extend these ideas to the world of continuous state spaces. As we shall see below, new mathematical techniques (from the point of view of extant work in process algebra) have to be incorporated to do this. Once the model and the new mathematical ideas have been assimilated, the whole gamut of process equivalences can be studied and argued about.

This work is not an alternative to extant work on probabilistic process algebras. We will summarize our understanding of that work in a separate section. Here we note that almost all the existing work is intended for systems with discrete state spaces, thus it is inapplicable to the type of systems that we consider. Of course, we need to justify our interest in such systems and we hope to persuade the reader that such systems are indeed interesting through the examples in the next section. Ongoing work - to be reported in future papers - will deal with substantial case studies.

One might take the view that any automated analysis or logical reasoning must be inherently discrete in character. In particular, even if one is interested in reasoning about a physical system, one has to first discretize the system. In fact, this point of view actually provides a good argument for retaining the continuous view of the system. A given system may well be described in continuous terms. Without formalizing the continuous system - and having a notion of equivalence between discrete and continuous systems - how does one argue that the discretized system is a faithful model of the underlying continuous system? Even suppose one is willing to treat a discrete model as given, what if one needs to refine the model? For example, a given discretization may arise from some type of approximation based on a given *tolerance*; how does one refine the tolerance or discretize *adaptively*? Clearly the underlying continuous model has to be retained and used if we want to construct different discrete approximations. A systematic study of approximation is beyond the scope of the present paper, but it is the subject of present activity and will be reported in future work.

From an immediate practical point of view, bisimulation can be used to reason about probabilistic, continuous state space discrete-time systems (henceforth Markov processes) in the following simple way. One often “discretizes” a continuous system by partitioning the state space into a few equivalence classes. Usually one has some intuition that the resulting discrete system “behaves like” the original continuous system. This can be made precise by our notion of bisimulation. It is also the case that some systems cannot be discretized, and once again, one can formalize what this means via bisimulation.

The present paper develops a notion of *labelled Markov process* which is meant ultimately to be part of a theory of interacting dynamical systems. The key conceptual contribution is the development of a notion of bisimulation for processes which have continuous state spaces but make discrete temporal steps. These are called discrete-time Markov processes. If the state space is also discrete, the phrase “Markov chain” is used. The adjective “Markovian” signifies that the transitions are entirely governed by the present state rather than by the past history of the system. The interaction is governed by “labels” in the manner now familiar from process algebra [Hoa85, Mil80, Mil89].

In brief, a labelled Markov process can be described as follows. There is a set of states and a set of labels. The system is in a state at a point in time and moves between states. The state which it moves to is governed by which interaction with the environment is taking place and this is indicated by the labels. The system evolves according to a probabilistic law. If the system interacts with the environment by synchronizing on a label it makes a transition to a new state governed by a transition probability distribution. So far, this is essentially the model developed by Larsen and Skou [LS91] in their very important and influential work on probabilistic bisimulation. They specify the transitions by giving, for each label, a probability for going from one state to another. Bisimulation then amounts to matching the moves; this means that both the labels and the probabilities must be the same.

In the case of a continuous state space, however, one cannot simply specify transition probabilities from one state to another. In most interesting systems all such transition probabilities would be zero! Instead one must work with probability densities. In so doing, one has to confront the major issues that arose when probability theory was first formalized, such as the existence of subsets for which the notion of probability does not make sense. In the present case, we have to introduce a notion of set for which "probabilities make sense" (i.e. a σ-field) and instead of talking about probabilities of going from a state s to another state s', we have to talk about going from a state s to a *set of* states A.

The notion of bisimulation for these systems is a generalization of the definition of Larsen and Skou, which, as we shall argue in the next section, is a compelling, natural notion. However, one cannot adapt their definitions without making the changes needed to deal with continuous state spaces. We cannot even use their basic terminology for bisimulation because their definition is inextricably wound up in the notion of state-to-state transition probabilities. Furthermore, it is a formidable technical problem to even show that bisimulation as we have defined it is an equivalence relation. This is solved by a construction due to Edalat [Eda99]. The construction heavily relies on properties that are not true for measure spaces in general. We have assumed *analytic space* structure. In a brief appendix we recapitulate some of the basic mathematical facts about analytic spaces. In the classical study of Markov processes, metric ideas play a significant role [Par67]. In any example of physical interest, the spaces will have this analytic structure, indeed they will usually come as metric spaces. Any discrete space is analytic, as is any closed subspace of $\mathbb{R}^n$.

The main new mathematical contributions of the present paper are:

- the basic definition of continuous state space systems,
- the definition of bisimulation and the proof that it coincides with Larsen-Skou bisimulation in the discrete case,
- the logical characterization of bisimulation.

The first two items were first announced in [BDEP97] and the third item was announced in [DEP98]. The proofs for the logical characterization are complete in this paper; it is not necessary to read [Eda99] to follow the arguments of the present paper.

In the logical characterization we show that two states (or processes) are bisimilar if and only if they satisfy all the same formulas of a modal logic similar to Hennessy-Milner logic. The striking aspect of this result is that the logic is completely negation free. Even for purely discrete systems this result is new and quite unexpected. It shows that probabilistic systems are very close to determinate systems. The nature of the proof is quite different from proofs of other Hennessy-Milner type results. The key to the proof is a certain quotient construction which enjoys some quite strong (co)universal properties. Given the continuous nature of the systems, it is odd that such a simple discrete logic characterizes bisimulation. It shows that - in some intuitive sense - discrete and continuous ideas coexist in the realm of analytic spaces.

This paper evolved from two earlier conference papers which appeared in the 12th and 13th IEEE Symposia On Logic In Computer Science as [BDEP97] and [DEP98]. Richard Blute was involved at an early stage in the first of these. The present paper unifies the results of both of those papers and makes changes to correspond to the latest version of the paper by Edalat [Eda99].

2 Examples of Processes

In this section we present a number of examples of systems with continuous state spaces. We have not defined what bisimulation is as yet, but we will use the examples to motivate what the concept should be.

We begin with a simple example, more for introducing terminology and concepts than for any practical interest. Consider a system with two labels $\{a, b\}$. The state space is the real plane, $\mathbf{R}^2$. When the system 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 $K_\alpha \exp(-\alpha(x-x_0)^2)$, where $K_\alpha = \sqrt{\alpha/\pi}$ is the normalizing factor. 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 $K_\beta \exp(-\beta(y-y_0)^2)$. The meaning of these densities is as follows. 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 $\int_s^t K_\alpha \exp(-\alpha(x-x_0)^2)dx$. Note that the probability of jumping to any given point is, of course, 0. In this system 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.

Interestingly, this system is indistinguishable from a one-state system that can make a or b moves. Thus, from the point of view of an external observer, this system has an extremely simple behaviour. The more complex internal behaviour is not externally visible. All that an observer can see is that the process always accepts an a-transition or a b-transition with probability 1. The point of a theory of bisimulation that encompasses such systems is to say when systems are equivalent. Of course this example is already familiar from the nonprobabilistic setting; if there is a system in which all transitions are always enabled, it will be bisimilar (in the traditional sense) to a system with one state. Bisimulation, like most other process equivalences, abstracts from the structure of the internal state space and records only the interaction with the environment. This example shows that it is possible for a system presented as a continuum state system to be in fact reducible to a simple finite state system.

Now we consider a system which cannot be reduced to a discrete system. There are three labels $\{a, b, c\}$. Suppose that the state space is $\mathbf{R}$. The state gives the pressure of a gaseous mixture in a tank in a chemical plant. The environment can interact by (a) simply measuring the pressure, or (b) it can inject some gas into the tank, or (c) it can pump some gas from the tank. The pressure fluctuates according to some thermodynamic laws depending on the reactions taking place in the tank. With each interaction, the pressure changes according to three different probability density functions, say $f(p_0, p), g(p_0, p)$ and $h(p_0, p)$ respectively, with nontrivial dependence on p_0. In addition, there are two threshold values p_h and p_l. When the pressure rises above p_h the interaction labelled b is disabled, and when the pressure drops below p_l the interaction labelled c is disabled. It is tempting to model this as a three state system, with the continuous state space partitioned by the threshold values. Unfortunately one cannot assign unique transition probabilities to these sets of states for arbitrary choices of f, g and h; only if very implausible uniformity conditions are obeyed can one do this. These conditions require, for example, that for any pressure value, p say, between p_l and p_h the probability of jumping to a pressure value above p_h is independent of the actual value of p. This is very implausible given the intuition that if the value of p is close to p_h the pressure fluctuations in the system are much more likely to carry the value of the pressure above p_h than if the initial pressure p is far below p_h.

These two examples show that systems presented as continuua may or may not be "equivalent" to discrete systems. In particular if one wants to work with a discrete model one will need some precise definition of what it means for the discrete model to be equivalent to the original system.

We now consider a scenario motivated by an industrial example arising from a real-life example we are working on. The numerical values and details are all proprietary so the example will be discussed in qualitative terms. The example arises in the context of verifying flight-control software. One of the subsystems is responsible for maintaining and updating navigational data received from satellites in the Global Positioning System (GPS). The aircraft state is described by a number of real parameters: position, velocity, orientation, state of wing controllers and thrust provided by the motors. The movement of the plane is partly deterministic – governed by the thrust of the motors, the current velocity and the wind-speed – and partly stochastic, subject to turbulence and fluctuations in the thrust. The flight-control software is organized around a large repetitive loop (the so-called cyclic executive) which samples all the sensor data at a regular rate and then schedules other tasks. These tasks include programs that are to run at various rates; for example one program calculates the roll and yaw of the aircraft, another runs a Kalman filter to estimate the current state. Occasionally - in normal circumstances - corrective actions need to be taken.

There are a number of verification tasks that need to be addressed. Overall one wants to know that the system is safe with high probability. It cannot be expected to be absolutely safe because the exact dynamics is unknown. An important verification task is ensuring that corrective actions are taken often enough. Though this is a continuous-time system the periodicity of the cyclic executive imposes a discrete-time character on the entire problem. However the spatial parameters are continuous and the stochastic evolution of the system is inherently continuous. For example, the filtering and control algorithms are fundamentally based on continuous dynamics. This system is an example of the type of system that we are interested in modelling. It has a continuous state space subject to stochastic (Markovian) evolution, it has a discrete temporal character. The system (the aircraft) is subject to synchronized interactions with its control software. The basic actions include sensing actions and control actions. At every time step the sensor data will report new values for the state parameters. It is reasonable to model this by assuming that every sensing interaction causes the transition of the system to the new values. The control goals are to keep the system parameters within specified safe ranges.

3 Discrete Probabilistic Systems

In this section, we recapitulate the Larsen-Skou definition of probabilistic bisimulation [LS91]. The systems that they consider will be referred to as *labelled Markov chains* in the present paper.

Definition 3.1 *A* ***labelled Markov chain*** *is a quadruple $(S, \mathsf{L}, C_l, P_l)$, where S is a countable set of states, L is a set of labels, and for each $l \in \mathsf{L}$, we have a subset C_l of S, a function, P_l, called a* **transition probability matrix***,*

$$P_l : C_l \times S \longrightarrow [0,1]$$

satisfying the normalization condition

$$\forall l \in \mathsf{L}, s \in C_l. \Sigma_{s' \in S} P_l(s, s') = 1.$$

If we have the weaker property,

$$\forall l \in \mathsf{L}, s \in C_l. \Sigma_{s' \in S} P_l(s, s') \leq 1$$

we call the system a ***partial*** *labelled Markov chain.*

The sets C_l are the sets of states that *can* do an l-action. If we have partial labelled Markov chains then we can just dispense with the C_l sets. In what follows we suppress the label set, i.e. we assume a fixed label set given once and for all.

Definition 3.2 *Let $T = (S, P_l)$ be a labelled Markov chain. Then a **probabilistic bisimulation** $\equiv_p$, is an equivalence on S such that, whenever $s \equiv_p t$, the following holds:*

$$\forall l \in \mathsf{L}.\forall A \in S/\equiv_p,\ \Sigma_{s' \in A} P_l(s, s') = \Sigma_{s' \in A} P_l(t, s').$$

*Two states s and t are said to be **probabilistically bisimilar** ($s \sim_{LS} t$) in case (s, t) is contained in some probabilistic bisimulation.*

Intuitively, we can read this as saying that two states are bisimilar if we get the same probability when we add up the transition probabilities to all the states in an equivalence class of bisimilar states. The addition is crucial – the probabilities are not just another label. The subtlety in the definition is that one has to somehow know what states are probabilistically bisimilar in order to know what the equivalence classes are, which in turn one needs in order to compute the probabilities to match them appropriately. In fact a very natural notion of probabilistic synchronization trees yields a model of a probabilistic version of CCS with both probabilistic branching and non-deterministic branching. If, in that model, one looks only at probabilistic branching, equality is precisely the Larsen-Skou notion of bisimulation [BK96].

The paper by Larsen and Skou does much more than just define bisimulation. They introduce the notion of testing a probabilistic process and associating probabilities with the possible outcomes. They then introduce a notion of testable properties. The link with probabilistic bisimulation is that two processes are probabilistically bisimilar precisely when they produce the same results for all possible tests. They also introduce a probabilistic modal logic and show that bisimulation holds precisely when two processes satisfy the same formulas. In fact these results are true for processes satisfying the "minimal deviation assumption" which is a slightly stronger condition than finite branching.

One slight awkwardness in the Larsen-Skou definition is that when one compares two processes, one has to combine the state sets and define probabilistic bisimulation on the combined states. This is a minor point but the reader should keep this in mind when reading the proofs below.

4 A Category of Markov Processes

A Markov process is a transition system with the property that the transition probabilities depend only on the current state and not on the past history of the process. We will consider systems where there is an interaction with the environment described by a set of labels as in process algebra. For each fixed label, the system may undergo a transition governed by a transition probability. One could have a new set of possible states at every instant but, for simplicity, we restrict to a single state space.

We will organize the theory in categorical terms with objects being transition systems and morphisms being simulations. Bisimulation is most easily thought of in these terms. This presentation will also allow us to compare the theory with the more traditional theory of nonprobabilistic processes; see, for example, the handbook article by Winskel and Nielsen [WN95].

In formulating the notion of Markov processes, we need to refine two concepts that were used in the discrete case. First we cannot simply define transition probabilities between states; except in rare cases, such transition probabilities are zero. We have to define transition probabilities between a state and a set of states. Second, we cannot define transition probabilities to any arbitrary set of states; we need to identify a family of sets for which transition probabilities can be sensibly defined. These are the *measurable sets*. Thus, in addition to specifying a set of states, we need to specify a σ-field on the set of states [Ash72, Bil95, Hal74, KT66, Rud66]. A review of the pertinent definitions appears in the appendix.

A key ingredient in the theory is the transition probability function.

Definition 4.1 *A* ***transition probability function*** *on a measurable space* (X, Σ) *is a function* $T : X \times \Sigma \longrightarrow [0, 1]$ *such that for each fixed* $x \in X$*, the set function* $T(x, \cdot)$ *is a (sub)probability measure, and for each fixed* $A \in \Sigma$ *the function* $T(\cdot, A)$ *is a measurable function.*

One interprets $T(x, A)$ as the probability of the system starting in state x making a transition into one of the states in A. The transition probability is really a *conditional probability*; it gives the probability of the system being in one of the states of the set A after the transition, *given* that it was in the state x before the transition. In general the transition probabilities could depend on time, in the sense that the transition probability could be different at every step (but still independent of past history); we are looking at the time-independent case.

We will work with *sub-probability* functions; i.e. with functions where $T(x, X) \leq 1$ rather than $T(x, X) = 1$. The mathematical results go through in this extended case and the resulting categories often are nicer, but the stochastic systems studied in the literature are usually only the very special version where $T(x, X)$ is either 1 or 0. In fact what is often done is that a state x with no possibility of making a transition is modeled by having a transition back to itself. For questions concerning which states will eventually be reached (the bulk of the analysis in the traditional literature) this is convenient. If, however, we are modeling the interactions that the system has with its environment, it is essential that we make a distinction between a state which can make a transition and one which cannot. It is taken as a fundamental observable in process algebra that a process rejects or accepts an action.

The key mathematical construction requires an analytic space structure on the set of states. Thus instead of imposing an arbitrary σ-field structure on the set of states, we will require that the set of states be an analytic space and the σ-field be the Borel algebra generated by the topology.

Definition 4.2 *A* ***partial labelled Markov process*** *with label set* L *is a structure* $(S, \Sigma, \{k_l \mid l \in \mathsf{L}\})$*, where* S *is the set of states, which is assumed to be an analytic space, and* Σ *is the Borel* σ*-field on* S*, and*

$$\forall l \in \mathsf{L}, k_l : S \times \Sigma \longrightarrow [0, 1]$$

is a transition sub-probability function.

We will fix the label set to be some L once and for all. The resulting theory is not seriously restricted by this. We will write (S, Σ, k_l) for partial labelled Markov processes, instead of the more precise $(S, \Sigma, \{k_l \mid l \in \mathsf{L}\})$ and often refer to a process by its set of states. In case we are talking about discrete systems, we will use the phrase "labelled Markov chain" rather than "discrete, labelled, Markov process". We are sometimes interested in the following special case. We have a partial labelled Markov process as above and a predicate **Can** on $S \times \mathsf{L}$ such that for every $(x, l) \in \mathbf{Can}$ we have $k_l(x, S) = 1$ and for every $(x, l) \notin \mathbf{Can}$ we have $k_l(x, S) = 0$. In other words if the action is enabled then we have transition probabilities normalized to 1.

In a (partial) labelled, Markov *chain* the set of states is countable, and one can easily define a metric so that each point is an open set and the space is complete and obviously separable, the Borel σ-field is then the entire powerset and the transition probabilities are given by a L-indexed family of functions $\forall l \in \mathsf{L}.P_l : S \times S \longrightarrow [0,1]$ satisfying the conditions required of a (sub)probability distribution. From this presentation we can construct the k_l in the following way $k_l(s,A) \stackrel{\text{def}}{=} \sum_{s' \in A} P_l(s,s')$. We use the phrase "transition function" for an object of type $S \times \Sigma \longrightarrow [0,1]$ and "transition matrix" for an object of type $S \times S \longrightarrow [0,1]$. A probabilistic transition system as defined by Larsen and Skou is precisely a labelled Markov chain. The partial notion has been used by Cheng and Nielsen [CN95] to give an open maps [JNW96] presentation of bisimulation.

In order to define a category of Markov processes, we define simulation morphisms between processes.

Definition 4.3 *A* ***simulation morphism*** *f between two partial labelled Markov processes, (S,Σ,k_l) and (S',Σ',k'_l) is a measurable function $f : (S,\Sigma) \longrightarrow (S',\Sigma')$ such that*

$$\forall l \in \mathsf{L}.\forall s \in S.\forall A' \in \Sigma'.k_l(s, f^{-1}(A')) \leq k'_l(f(s), A').$$

Suppose that the system being simulated can make a transition from the state s to the set of states A with a certain probability, say p. Then the simulating system will simulate it in the following sense: the transition probability from $f(s)$ to any measurable set B such that $f(A) \subseteq B$ is greater than p. This is equivalent to the statement in the definition. We cannot directly talk about the transition probability from $f(s)$ to $f(A)$ since the latter may not be measurable. We require simulation to be measurable[1] for the definition to make sense. If f were not measurable we would not be guaranteed that $f^{-1}(A')$ is measurable.

This notion extends the standard notion of simulation of labelled transition systems [WN95] in the following way. Given a partial labelled Markov chain (S, P_l), we can define a labelled transition system (lts) with the same label set as follows. We take the same set of states S and we define a labelled transition relation $\rightarrow \subseteq (S \times \mathsf{L} \times S)$ by $(s,l,s') \in \rightarrow \iff P_l(s,s') > 0$. Given two labelled transition systems, $(S_1, \rightarrow_1)$ and $(S_2, \rightarrow_2)$, a function $f : S_1 \longrightarrow S_2$ is a simulation morphism if $\forall s \in S_1.s \xrightarrow{l} s' \Rightarrow f(s) \xrightarrow{l} f(s')$. We cannot do this for Markov processes because we can easily have systems where all the point-to-point transition probabilities are zero but the Markov process is nontrivial because the transition probabilities are nonzero to "larger" sets.

Proposition 4.4 *Given two partial labelled Markov chains, a simulation morphism between them is also a simulation morphism between the associated labelled transition systems.*

Proof (sketch). Suppose that we have two partial labelled Markov chains (S,Σ,k_l) and (S',Σ',k'_l) with f a simulation morphism from S to S'. Now suppose that in the associated lts the transition $s_1 \xrightarrow{l} s_2$ is possible. This means that $k_l(s_1,\{s_2\}) > 0$. Since f is a morphism we must have that $k'_l(f(s_1),\{f(s_2)\}) \geq k_l(s_1, f^{-1}(f(s_2))) \geq k_l(s_1,\{s_2\}) > 0$; hence in the lts $f(s_1) \xrightarrow{l} f(s_2)$ is possible. ∎

From now on, we assume that all systems are partial and we will stop writing the adjective "partial" explicitly.

[1] In older texts, such as Halmos [Hal74] or Rudin [Rud66] measurable is defined to mean that the inverse image of an *open* set is measurable. This means that the composite of two measurable functions need not be measurable. Our definitions are the current standard and, of course with this definition the composite of two measurable functions is measurable.

Definition 4.5 *The objects of the category* ***LMP*** *are labelled Markov processes, having a fixed set* L *as the set of labels, with simulations as the morphisms. The category of labelled, Markov chains is written* ***LMC*** *and is the full subcategory of* ***LMP*** *that includes only the labelled, Markov chains as objects.*

In order to use the results of [Eda99] we need to introduce another - closely related - category defined below. Unless the reader is interested in going into all details of the proofs and in reading [Eda99] he can safely skip those parts of the discussion pertaining to what we call *generalized labelled Markov processes*. We use the concept of *universally measurable* sets and functions; which are defined in the appendix.

Definition 4.6 *A* ***generalized labelled Markov process*** *is a labelled Markov process except that the transition probability function need only be* universally measurable, *that is, for all* $l \in \mathsf{L}$, *we have that* $k_l(\cdot, A)$ *is a universally measurable function for* $A \in \Sigma$, *and for* $s \in S$, $k_l(s, \cdot)$ *is a sub-probability measure on* (S, Σ).

Most of our definitions for labelled Markov processes will be used unchanged and without special comments for generalized labelled Markov processes. For simplicity, we will not define "generalized simulation morphisms" explicitly. Simulation morphisms for generalized labelled Markov processes are defined in exactly the same way as already defined for labelled Markov processes.

Definition 4.7 *The objects of the category* ***LMP**** *are generalized labelled Markov processes, having a fixed set* L *as the set of labels, with simulations as the morphisms.*

Every labelled Markov process is a generalized labelled Markov process and hence **LMP** is a full subcategory of **LMP***.

5 Bisimulation for Markov Processes

The definition of bisimulation is heavily influenced by the ideas of Joyal, Nielsen and Winskel [JNW96]. The idea is to identify a class of special systems called "observations" or "observable paths" or better still "observable path shapes", and to define bisimulation as a relation satisfying a kind of path-lifting property, the so-called "open maps".

What one can prove for ordinary labelled transition systems is that if we take paths to be labelled paths in the usual sense, then the open maps are morphisms that satisfy a condition called the "zigzag" condition[2]. In our case, the zigzag condition is easy to state and it is easy to see that it corresponds to Larsen-Skou bisimulation in the case of labelled Markov chains.

For bisimulation between labelled Markov processes, we essentially want to say that there is a "zigzag relation". One can talk about relations by talking about **spans**. A span in any category between an object S_1 and another object S_2 is a third object T together with morphisms from T to both S_1 and S_2. One can think of this in the category of **Sets** as viewing a relation as a set of ordered pairs with the morphisms being the projections. Bisimulation is then defined to hold between two systems if they are connected by a span of zigzags. We will use this idea to define bisimulation for labelled Markov processes.

Definition 5.1 *A morphism* f *from* (S, Σ, k_l) *to* (S', Σ', k'_l) *is a* ***zigzag morphism*** *if it satisfies the properties:*

[2]The name arises from modal logic, see, for example, [Pop94].

- *f is surjective;*
- *For all $l \in \mathsf{L}$ and $s \in S$*

$$A' \in \Sigma', \;\; k_l(s, f^{-1}(A')) = k'_l(f(s), A').$$

Note that this definition makes sense in both categories **LMP** and **LMP***. Asking f to be surjective allows us to avoid introducing initial states and worrying about reachable states. One can immediately check that the identity morphism is a zigzag.

Following Joyal, Nielsen and Winskel ([JNW96]), we define bisimulation as the existence of a span of zigzag morphisms. However, in order to prove that bisimulation is an equivalence relation, we need to go beyond the category **LMP**. One cannot prove transitivity just working with the category **LMP**, we need to move to **LMP***. We will use the expression "generalized span" between **LMP** objects to mean that we have a span in **LMP*** between the objects. The precise definition is:

Definition 5.2 *Let T and T' be two labelled Markov processes. T is **probabilistically bisimilar to** T' (written $T \sim T'$) if there is a generalized span of zigzag morphisms between them, i.e. there exists a generalized labelled Markov process S in **LMP*** and zigzag morphisms f and g such that*

$$\begin{tikzcd} & S \arrow[dl, "f"'] \arrow[dr, "g"] & \\ T & & T' \end{tikzcd}$$

Notice that if there is a zigzag morphism between two systems, they are bisimilar since the identity is a zigzag morphism and because **LMP** is a subcategory of **LMP***. The last fact also implies that if there is a span in **LMP** between two processes, they are bisimilar.

It is interesting to note that we can take a coalgebraic view of bisimulation [AM89, Rut95, Rut96] as well. We can view a labelled Markov process as a coalgebra of a suitable functor; in fact it is a functor introduced by Giry [Gir81] in order to define a monad on **Mes** analogous to the powerset monad. From this point of view, bisimulation is a span of coalgebra homomorphisms. But if one checks what this means, these are precisely our zigzag morphisms in **LMP**.

We want bisimulation to be an equivalence, so we need to prove transitivity of the existence of span, since it is obviously reflexive and symmetric. Proving transitivity presents formidable difficulties. In particular, it probably isn't true for probabilistic transition systems without the assumption that the state space is analytic. Transitivity relies on the following theorem.

Theorem 5.3 ([Eda99]) *Consider the following diagram in **LMP****

$$\begin{tikzcd} S_1 \arrow[dr, "f_1"] & & S_2 \arrow[dl, "f_2"'] \\ & S & \end{tikzcd}$$

where f_1 and f_2 are zigzags. Then we can find an object U in $\boldsymbol{LMP}^*$ and zigzag morphisms g_1 and g_2 such that the following diagram commutes:

$$\begin{tikzcd} & U \arrow[dl, "g_1"'] \arrow[dr, "g_2"] & \\ S_1 \arrow[dr, "f_1"'] & & S_2 \arrow[dl, "f_2"] \\ & S & \end{tikzcd}$$

This is not a pullback because it does not have the universal property, it is not even a weak pullback. We refer to it as the *semi-pullback* construction. This proof rests on a technical result which involves quite intricate computations in measure theory. Edalat's paper contains the proof.

The important consequence of this theorem is the following.

Corollary 5.4 *Probabilistic bisimulation is an equivalence relation.*

6 More Examples

In this section we give some examples using the formalism. The first example is just a repetition of the first example of 2 expressed in our formalism.

Example 6.1 We let the label set be the one element set. Consider a system (S, Σ, k) with S an arbitrarily complicated state space and Σ a σ-field generated by some analytic space structure on S. For example, S could be $\mathbf{R}$, the reals with the Borel algebra. We define the transition function, $k(s, A)$ in any manner we please subject only to the conditions of the definition of a transition function and to the condition that $\forall s \in S.k(s, S) = 1$; i.e. for every s, the distribution $k(s, \cdot)$ is a probability measure. Consider the trivial labelled Markov chain with just one label, one state and one transition from the state to itself with probability 1. These two systems are bisimilar!

This example allows us to clarify the discussion in the introduction. All of conventional stochastic process theory is described by systems like the first system above. From our point of view they are trivial. This is to be expected, as we are modeling *interaction* and all such systems are indeed trivial from the point of view of interaction. In order to get nontrivial examples, one has to consider systems with richer label sets, and which are not always capable of making transitions with every label. Note also that many continuous systems are bisimilar to discrete ones. If we know this we can use the discrete system when reasoning about composite systems.

The next example shows that bisimulation really must be defined in terms of spans, i.e. it genuinely is a relation.

Example 6.2 Consider the labelled Markov chain over a one-element label set $(\{a, b, c\}, P)$ where the transition matrix, P is given by $P(a, b) = P(a, c) = 1/2$ with all other entries being zero. Consider another system over the same label set with 4 states $\{w, x, y, z\}$ and with transition matrix Q given by $Q(w, x) = Q(w, y) = Q(w, z) = 1/3$. These two systems are bisimilar. One can easily verify this by constructing the obvious "product" system and checking that the projections are zigzags. On the other hand there is no zigzag morphism between the two systems. These systems are small enough to check this by hand.

Example 6.3 This example illustrates a continuous system. Consider the labelled Markov process, over the trivial label set, defined as follows $S = (\mathbf{R}, \mathcal{B}, k)$, i.e. the states are real numbers, the measurable sets are Borel sets and the transition function is defined on intervals (and then extended to arbitrary Borel sets) as follows:

$$k(x,[r,s]) = \begin{cases} \lambda/2 \int_r^s e^{-\lambda|x-y|} dy & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

where the constant factor of $\lambda/2$ chosen to make k be 1 on the whole space. Intuitively this is a system where a particle makes random jumps with probability exponentially distributed with the length. However, there is an "absorbing wall" at the point $x = 0$ so that if the system jumps to the left of this point it gets stuck there. Note that every positive state has a different probability density for jumping to a negative state. Now consider the system $U = (\mathbf{R}^2, \mathcal{B}^2, h)$ defined as

$$h((x,y),[r,s] \times [p,q]) = k(x,[r,s])P([p,q]),$$

where P is some arbitrary probability measure over $\mathbf{R}$. This system should behave "observably" just like the first system because, roughly speaking, the first coordinate behaves just like the first system and the second has trivial dynamics, i.e. it is bisimilar to the one-state, one-transition system. Indeed these two systems are bisimilar with the projection from the second to the first being a zigzag.

The next example illustrates a possible objection to our definition.

Example 6.4 Suppose that we have two systems, $(\mathbf{R}, \mathcal{B}, \{a,b\}, k_l)$ and $(\mathbf{R}, \mathcal{B}, \{a,b\}, h_l)$. In the first system we have the following transitions

$$k_a(x,S) = \mu(S \cap [x-0.5, x+0.5])$$

and

$$k_b(x,S) = \mu(S \cap [x+0.5, x+1.5])$$

where S is a Borel set and μ is Lebesgue measure. For the other system we have

$$h_a(x,S) = \begin{cases} \mu(S \cap [x-0.5, x+0.5]) & \text{if } x \text{ is irrational} \\ 0 & \text{if } x \text{ is rational.} \end{cases}$$

the b transitions are the same as those for the first system. These two systems are not bisimilar by our definition. The first one is bisimilar to the trivial one-state system with both a and b enabled all the time, while the second one has states in which a gets disabled. However, the probability of landing in one of these states is 0. Thus, in some sense, the difference is visible only on a set of probability 0. Should they be distinguished?

Example 6.5 We abstract from the aircraft control situation mentioned in section 2 above to provide an example of a continuous state space probabilistic system. We consider the problem of controlling a moving object (abstracted as a point) on a 2-dimensional plane. The state of the system is given by 4 real variables x, y, u, v. The (x, y) pair specifies the position and the (u, v) pair specifies the velocity. Every time step the state is measured. Given a state (x_0, y_0, u_0, v_0) the new position at the next time step is not determined because the object is subject to random forces as well as the main driving force. Thus the new x-coordinate is given by the following density function:

$$f(x, x_0, y_0, u_0, v_0) = \phi(x - (x_0 + K * v_0))$$

where the function ϕ represents the random effects which are distributed symmetrically about its argument. If there were no random effects the new x-coordinate would be $x - (x_0 + K * v_0)$. The density function is interpreted as above; i.e. the probability is given by integrating the density. We have similar functions g for the y-coordinate, p and q for the velocities. Suppose that we call the basic time step a we have

$$k_a((x_0, y_0, u_0, v_0), [x_1, x_2] \times [y_1, y_2] \times [u_1, u_2] \times [v_1, v_2]) =$$

$$\int_{x_1}^{x_2} \int_{y_1}^{y_2} \int_{u_1}^{u_2} \int_{v_1}^{v_2} f(x, x_0, y_0, u_0, v_0) \ldots q(v, x_0, y_0, u_0, v_0 dx\ dy\ du\ dv.$$

Now the estimates that there might be sudden turbulence or other random effects are encoded into the density functions. In general the density functions need not separate into a nice product as we have shown. The particle is also subject to control forces that are applied by the controller to keep the particle state within prescribed boundaries. This interaction b say will give rise to a similar expression describing the distribution of possible states. In order to reason about the correctness of a given controller we can model a controller which reacts to the state measurements and makes decisions to apply or not apply corrective forces.

$$cont = a.(if\ (\ldots)\ then\ cont\ else\ b.cont).$$

We obtain the composite system of object and controller by simply composing the two systems. The correctness of the controller and object combination is a standard calculation in control systems theory[3].

So far the process algebraic aspects have been trivial, but note that the presentation of the model has been inherently continuous and the reasoning is typically carried out using the mathematics of continuous systems. Now we can modify the controller to react to changes in the goal. Thus the controller can have a new action which allows it to accept a new goal and the control decisions are modified accordingly. We can also endow the object with more actions to allow it to have interactions with other systems, for example with communication systems with other objects. As the model gets more complex it is very advantageous to have the ability to build up the system compositionally. We thus obtain a system which may have some communication protocol (roughly as complicated as the sliding window protocols extensively analyzed in the literature) and some inherently continuous probabilistic subsystem. The communication may affect the control goals so the overall system has the complexity one sees in many discrete models as well as the complexity associated with having continuous components. To use a discrete model would involve justifying that the discrete model correctly approximates the continuous model. Of course if one is lucky, a continuous model may turn out to be bisimilar to a discrete model and then one can use all the apparatus developed for reasoning about discrete systems. However, in this case one needs to have a notion of equivalence that encompasses discrete and continuous systems.

[3]In fact one can even assume that the observations are noisy and use filtering algorithms to estimate the state.

7 Discrete Systems Revisited

In this section, we reconsider discrete systems (Markov chains) from the point of view of the bisimulation notion that we have defined for labelled Markov processes. We show that the Larsen-Skou definition coincides with the "span of zigzags" definition. First, we have to say what it means for two Markov chains to be Larsen-Skou bisimilar, since the Larsen-Skou definition involves states of a single process rather than states of two different processes, and so doesn't apply without an appropriate interpretation. Roughly speaking, if there were an initial state in every Markov chain, then we would say that two Markov chains are Larsen-Skou bisimilar if and only if in a system "combining" them, their initial states are Larsen-Skou bisimilar. We have chosen not to equip Markov chains with initial states, instead we will use the following as a definition of Larsen-Skou bisimulation between two Markov processes.

Definition 7.1 *Let (S_1, P_l^1) and (S_2, P_l^2) be two Markov chains and (T, H_l) their disjoint union, that is, T is the disjoint union of the two sets S_1 and S_2, and for $s, t \in T$, $H_l(s,t) = P_l^i(s,t)$ if s, t are both in S_i $(i = 1, 2)$ and zero otherwise. We will say that $S_1 \sim_{LS} S_2$ if in T there is a probabilistic bisimulation $\equiv_p$ such that for every state s_i of S_i there is a state s_j of S_j such that $s_i \equiv_p s_j$ $(i, j \in \{1, 2\}, i \neq j)$.*

Recall that we can always define a topology in which all the sets are open and hence all functions are continuous. Since the space is countable, the topology is clearly analytic. Thus, we can forget about the topology in this section but we keep in mind that discrete systems are a special case of the formalism of the previous section.

Proposition 7.2 *Let (S, P_l) and (S', P'_l) be two labelled Markov chains. $(S, P_l) \sim_{LS} (S', P'_l)$ if and only if there exists a span of zigzag morphisms f and f':*

$$\begin{tikzcd} & U \arrow[dl, "f"'] \arrow[dr, "f'"] & \\ S & & S' \end{tikzcd}$$

Proof . We write k_l and k'_l for the two transition functions induced by the transition matrices P_l and P'_l respectively.

$\Leftarrow$: We first show that if $S \xrightarrow{f} S'$, where f is a zigzag morphism, then $S \sim_{LS} S'$. Let (t, H_l) be the disjoint union of (S, P_l) and (S', P'_l), and h_l be the transition function induced by the transition matrix H_l. Now f defines the following equivalence relation, R, on T:

$$s_1 R s_2 \iff (s_1 = s_2) \vee (f(s_1) = s_2) \vee (f(s_1) = f(s_2)).$$

The equivalence classes are of the form $\{s'\} \cup f^{-1}(s')$ for each $s' \in S'$; thus each equivalence class can be represented uniquely by an element of S'. Let $l \in \mathsf{L}, s_1, s_2 \in T$ such that $s_1 R s_2$, and choose any $t' \in S'$ that represents the equivalence class $\{t'\} \cup f^{-1}(t')$. We want to show that $h_l(s_1, -)$ and $h_l(s_2, -)$ agree on $\{t'\} \cup f^{-1}(t')$.

First assume $s_1 \in S$ and $s_2 \in S'$, meaning that $f(s_1) = s_2$. Then

$$\begin{aligned} h_l(s_1, \{t'\} \cup f^{-1}(t')) &= k_l(s_1, f^{-1}(t')) \quad \text{since } s_1 \in S \\ &= k'_l(f(s_1), \{t'\}) \quad \text{since } f \text{ is a zigzag morphism} \\ &= h_l(s_2, \{t'\} \cup f^{-1}(t')), \end{aligned}$$

which is precisely the condition for Larsen-Skou bisimulation. Now if s_1 and s_2 are both in S (and still R-related), they have the same image, so $s_1 \, R \, f(s_1)$ and $f(s_1) = f(s_2)$ and $f(s_2) \, R \, s_2$ and we can simply apply the above calculation. Finally we have the trivial case where s_1 and s_2 are both in S'. They are then equal and we are done since, f being a surjective function, every state of S is R-equivalent to a state of S' and vice versa. Since we know that Larsen-Skou bisimulation is an equivalence relation it follows that whenever we have a span of zigzags connecting two labelled Markov chains they are Larsen-Skou bisimilar.

$\Rightarrow$: Assume $(S, P_l) \sim_{LS} (S', P_l')$, with $\equiv_p$ the probabilistic bisimulation over (T, H_l) the disjoint union of (S, P_l) and (S', P_l'). We need to construct a span of zigzag morphisms

$$\begin{tikzcd} & U \arrow[dl, "f"'] \arrow[dr, "f'"] & \\ S & & S' \end{tikzcd}$$

To do this, let $U = (U, (Q_l)_{l \in \mathsf{L}})$ where $U = \{(s, s') \in S \times S' : s \sim_{LS} s' \text{ in } T\}$ and where the transition matrix Q is given by, for $l \in \mathsf{L}$,

$$Q_l((s, s'), (t, t')) = \frac{P_l(s, t) \, P_l'(s', t')}{h_l(s, [t]_{\sim_{LS}})}$$

where $[t]_{\sim_{LS}}$ denotes the equivalence class containing t in T, and h_l is the transition function induced by the transition matrix H_l. Since $s \sim_{LS} s'$ and $t \sim_{LS} t'$, we have by definition of $\sim_{LS}$ that $h_l(s, [t]_{\sim_{LS}}) = h_l(s', [t]_{\sim_{LS}}) = h_l(s', [t']_{\sim_{LS}})$.

To prove that U is a labelled Markov chain, we need that for any $(s, s') \in U$,

$$\sum_{(t, t') \in U} Q_l((s, s'), (t, t')) \leq 1.$$

This will follow from the proof that we have zigzag morphisms from U to S and S'. As morphisms $f : U \to S$ and $f' : U \to S'$, we simply take the left and right projections which are surjective by definition of Larsen-Skou probabilistic bisimulation. We prove that they are zigzag morphisms. We write q_l for the transition function derived from the transition matrix Q_l. First note that

$$\forall t \in S. f^{-1}(t) = \{t\} \times (S' \cap [t]_{\sim_{LS}}).$$

For any $l \in \mathsf{L}, (s, s') \in U, t \in S$, we have

$$\begin{aligned} q_l((s, s'), f^{-1}(t)) &= \sum_{t' \in S' \cap [t]_{\sim_{LS}}} Q_l((s, s'), (t, t')) \\ &= \sum_{t' \in S' \cap [t]_{\sim_{LS}}} \frac{P_l(s, t) \, P_l'(s', t')}{h_l(s, [t]_{\sim_{LS}})} \\ &= \frac{P_l(s, t)}{h_l(s', [t]_{\sim_{LS}})} \sum_{t' \in S' \cap [t]_{\sim_{LS}}} P_l'(s', t') \\ &= \frac{P_l(s, t)}{k_l'(s', [t]_{\sim_{LS}} \cap S')} k_l(s, [t]_{\sim_{LS}} \cap S) \\ &= P_l(s, t) = k_l(f(s, s'), \{t\}). \end{aligned}$$

and f is thus a zigzag morphism. The same argument applies to f'. ∎

8 Hennessy-Milner Logics for Labelled Markov Processes

We now describe five logics that will each be proven to characterize bisimulation in the next section. Thus *all* these logics play the role of Hennessy-Milner logic for nonprobabilistic bisimulation.

We assume that there is a fixed set of "labels" or "actions", we usually use letters like a or b for actions. The simplest logic will be called $\mathcal{L}_0$ and has as syntax the following formulas:

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

where a is an action from the fixed (countable) set of actions L and q is a rational number. Given a labelled Markov process (S, Σ, k_a) 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 and conjunction. We say $s \models \langle a \rangle_q \phi$ if and only if $\exists A \in \Sigma.(\forall s' \in A.s' \models \phi) \wedge (k_a(s, A) \geq q)$. In other words, the system in state s can make an a-move to a state, that satisfies ϕ, with probability greater than q. We write $[\![\phi]\!]_S$ for the set $\{s \in S | s \models \phi\}$. We often omit the subscript when no confusion can arise.

In the following table we define four additional logics. They are all syntactic extensions of $\mathcal{L}_0$.

$$\begin{array}{rcl} \mathcal{L}_{\mathrm{Can}} & := & \mathcal{L}_0 \mid \mathrm{Can}(a) \\ \mathcal{L}_{\Delta} & := & \mathcal{L}_0 \mid \Delta_a \\ \mathcal{L}_{\neg} & := & \mathcal{L}_0 \mid \neg\phi \\ \mathcal{L}_{\wedge} & := & \mathcal{L}_{\neg} \mid \bigwedge_{i \in \mathbf{N}} \phi_i \end{array}$$

Given a labelled Markov process (S, Σ, k_a) we write:

$$\begin{array}{ll} s \models \mathrm{Can}(a) & \text{to mean that } k_a(s, S) > 0; \\ s \models \Delta_a & \text{to mean that } k_a(s, S) = 0; \\ s \models \neg\phi & \text{to mean that } s \not\models \phi; \\ s \models \bigwedge_{i \in \mathbf{N}} \phi_i & \text{to mean that } s \models \phi_i \text{ for all } i \in N. \end{array}$$

Although they all characterize bisimulation, they don't have the same expressive power. Clearly all of them are at least as expressive as $\mathcal{L}_0$, and $\mathcal{L}_\wedge$ is more expressive than all the others. $\mathcal{L}_{\mathrm{Can}}$, $\mathcal{L}_\Delta$ and $\mathcal{L}_\neg$ are incomparable. It is interesting to note that none of these differences will have any impact on the characterization of bisimulation, as we have already said.

The logic that Larsen and Skou used in [LS91] is $\mathcal{L}_\Delta$ with the additional formula $\phi_1 \vee \phi_2$. They show that for finitely branching systems[4], two *states* of the same system are bisimilar if and only if they satisfy the same formulas of their logic.

In the next section we prove the main result namely that two states of a system are bisimilar if and only if they satisfy all the same formulas of the logic $\mathcal{L}_0$, and that this also extends to all the other logics on our list. The fact that a logic without negation and without infinitary conjunction is sufficient for systems with infinite branching is somewhat of a surprise based on what we expect from the nonprobabilistic case. The point is that the probabilistic systems we are considering, without explicit nondeterminism, resemble deterministic systems quite closely, rather than nondeterministic

[4] They actually use a stronger property, the "minimum deviation condition" which uniformly bounds the degree of branching everywhere.

systems. In the latter case - as is well-known [Mil90] - without a finite-branching assumption the result is false, unless the logic has infinitary conjunctions.

Consider the processes shown in figure 1. They are both nonprobabilistic processes. The usual formula distinguishing them is $\langle a\rangle\neg\langle b\rangle\mathsf{T}$, which says that the process can perform an a action and then be in a state where it cannot perform a b-action. The process on the left satisfies this formula while the process on the right does not. However, it is well-known that they cannot be distinguished by a negation-free formula of Hennessy-Milner logic. If we now consider probabilistic versions of these processes we find that the situation is different. For no assignment of probabilities are the two processes going to be bisimilar. Suppose that the two a-labelled branches of the left hand process are given probabilities p and q, assume that the b-labelled transitions have probability 1. Now if the right hand process has its a-labelled transition given a probability anything other than $p+q$, say $r > p+q$ we can immediately distinguish the two processes by the formula $\langle a\rangle_r\mathsf{T}$ which will not be satisfied by the left hand process. If $r = p+q$ then we can use the formula $\langle a\rangle_r\langle b\rangle_1\mathsf{T}$. The left hand process cannot satisfy this formula unless $p = 0$, in which case the two processes are in fact equal.

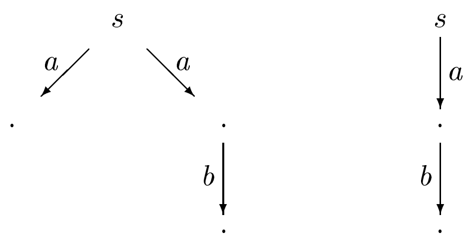

Figure 1: Two processes that cannot be distinguished without negation.

This simple example shows that one can use the probabilities to finesse the need for negation but one cannot actually encode negation with just $\mathcal{L}_0$. Of course this example does not constitute a proof but it makes it more plausible that indeed negation is not needed.

The next example shows why we do not need infinite conjunctions even if we have infinite branching. Consider a process with infinitely many a-labelled branches. The first branch ends in a state that can perform no further actions, call it a "dead state." The second branch ends in a state that can perform a single a to a dead state. Similarly the nth branch can perform a sequence of n a-actions and then reach a dead state. Call this process P. Now define a process Q which is just like P except that there is an additional transition to a state which then has an a-labelled transition back to itself. Now consider the formula

$$\langle a\rangle(\bigwedge_n \langle a\rangle^{(n)}\mathsf{T})$$

where the notation $\langle a\rangle^{(n)}$ means n nested $\langle a\rangle$ modalities. The conjunction is over all $n \geq 1$. This formula says that the process can jump to a state from which arbitrarily many a-labelled transitions are possible. The process P does not satisfy this formula but Q does. Now if we associate probabilities with these transitions we find that we can find distinguishing formulas that do not involve infinite conjunction. To see this assume that both processes satisfy all the same $\mathcal{L}_0$ formulas. We will show that the probability associated with the extra branch in Q has to be 0, i.e. it really cannot be present. Now the sum of the initial probabilities have to match since they both satisfy all the same formulas of the form $\langle a\rangle_p\mathsf{T}$. Now in both processes the branch that takes the initial state to a dead state has to have the same probability because they both satisfy all the same formulas of the form $\langle a\rangle_p \langle a\rangle_1 \mathsf{T}$. By induction it follows that each branch in P must have the same probability as the equal length branch in Q. Thus the branch to the looping state in Q must have probability 0. Conversely if this probability is not 0, in which case the two systems are not bisimilar, they cannot satisfy all the same formulas of $\mathcal{L}_0$ and hence a distinguishing formula can be constructed which does not involve infinitary conjunction.

It is tempting to think that the ability to distinguish processes comes from the power to encode negation and infinitary conjunction by manipulations of the probability subscripts in the modal formulas of the form $\langle a\rangle\phi$. In fact this is not the case. With negation we can write a formula which is only satisfied by dead states assuming that there are only finitely many distinct labels, namely

$$\bigwedge_{a\in \mathsf{L}} \neg\langle a\rangle \mathsf{T}.$$

It is not possible to write a formula that is only satisfied by dead states using just $\mathcal{L}_0$. There is no paradox of course. Given two states one can write a $\mathcal{L}_0$ formula which distinguishes them but this formula may depend on both states and cannot be constructed just by looking at one of them. For example, suppose that there is a family of states s_n, where n is a positive integer, such that the only transition is an a-labelled transition to a dead state with probability $\frac{1}{n}$. Now no *single* formula of $\mathcal{L}_0$ can distinguish all these states from the dead state but given any s_n the formula $\langle a\rangle_{\frac{1}{n}} \mathsf{T}$ will work.

9 Logical Characterization for Bisimulation

In this section we prove the main theorem of the paper. The proof relies on various properties of analytic spaces. We give an overview of the proof first. To show that two bisimilar states satisfy all the same formulas of $\mathcal{L}_0$ is relatively easy. To show the converse, the general plan is to construct a cospan using logical equivalence and then to use the semi-pullback construction [Eda97] to obtain a span. To obtain the cospan one defines an equivalence relation on states - two states are equivalent if they satisfy the same formulas - and then form the quotient. We need a general theorem to assure us that the result is analytic. We then define a transition probability on this quotient system in such a way as to ensure that the morphisms are zigzags. This is the part of the construction where we need most of the measure-theoretic machinery. We use a *unique structure theorem* to show that the measurable sets defined by the formulas of the logic generate the σ-field. We use a theorem on *unique extension of measure* in order to show that the transition probability is well-defined.

Recall that **LMP** is a *full* subcategory of **LMP***. Thus whenever we talk about morphisms between labelled Markov processes it makes no difference which category we are talking about. The only place to be careful is when we have an object that isn't a labelled Markov process but is a generalized labelled Markov process.

The first proposition below says that sets of states definable by formulas in a labelled Markov process are always measurable.

Proposition 9.1 *Let (S,Σ,k_a) be an object of **LMP**. Then for all formulas ϕ, we have $[\![\phi]\!]\in\Sigma$.*

Proof . We proceed by structural induction on ϕ. The base case corresponding to T is trivial since $S\in\Sigma$. Conjunction is trivial because, by definition, a σ-field is closed under intersection. Finally, we have $[\![\langle a\rangle_q\phi]\!] = k_a(\cdot,[\![\phi]\!])^{-1}([q,1])\in\Sigma$. To justify this first note that, by hypothesis, $[\![\phi]\!]\in\Sigma$ so $k_a(s,[\![\phi]\!])$ is meaningful. Secondly, k_a is a measurable function in its first argument and finally intervals are Borel. ∎

The next proposition links zigzag morphisms with formulas in the logic.

Proposition 9.2 *If f is a zigzag morphism from S in **LMP*** to S' in **LMP**, then for all state $s \in S$ and all formulas ϕ,*

$$s \models \phi \iff f(s) \models \phi.$$

Proof . We show that $f^{-1}([\![\phi]\!]_{S'}) = [\![\phi]\!]_S$ by structural induction on ϕ, which implies the result. Notice that by Proposition 9.1, $[\![\phi]\!]_{S'} \in \Sigma'$ since S' is in **LMP**, and hence $f^{-1}([\![\phi]\!]_{S'}) \in \Sigma$. (This will imply in particular that $[\![\phi]\!]_S$ will be measurable in S.) The only nontrivial case corresponds to the modal formula. We proceed as follows; $s \models \langle a \rangle_q \phi$ means $k_a(s, [\![\phi]\!]_S) \geq q$. Thus we have, by the induction hypothesis and because f is a zigzag morphism,

$$q \leq k_a(s, [\![\phi]\!]_S) = k_a(s, f^{-1}([\![\phi]\!]_{S'})) = k'_a(f(s), [\![\phi]\!]_{S'}).$$

But the last equality means $f(s) \models \langle a \rangle_q \phi$. ∎

From this we get the immediate corollary below, but first we need to say what it means for two systems to satisfy the same formulas. Suppose that (S, Σ, k_a) and (S', Σ', k'_a) are two systems. We say that they satisfy *all the same formulas* if $\forall s \in S\ \exists s' \in S'$ such that s and s' satisfy all the same formulas, written $s \approx s'$, and the same with s and s' interchanged. Clearly, $\approx$ is an equivalence relation.

Corollary 9.3 *If S and S' are bisimilar, then they satisfy the same formulas.*

In order to show the logic gives a complete characterization of bisimulation, we also want to show the converse. We first show that there is a zigzag morphism from any labelled Markov process S to its quotient under $\approx$. If (S, Σ) is a Borel space, the quotient $(S/_{\approx}, \Sigma_{\approx})$ is defined as follows. $S/_{\approx}$ is the set of all equivalence classes. Then the function $q : S \to S/_{\approx}$ which assigns to each point of S the equivalence class containing it maps onto $S/_{\approx}$, and thus determines a Borel structure on $S/_{\approx}$: by definition a subset E of $S/_{\approx}$ is a Borel set if $q^{-1}(E)$ is a Borel set in S.

Proposition 9.4 *Let (S, Σ, k_a) be an object of **LMP**. Then $(S/_{\approx}, \Sigma_{\approx})$ is an analytic space and we can define h_a so that the canonical projection q from (S, Σ, k_a) to $(S/_{\approx}, \Sigma_{\approx}, h_a)$ is a zigzag morphism.*

In order to prove this proposition we need a few lemmas. The first allows us to work with direct images of q. The second is elementary, while the next two are known results about analytic spaces. The final lemma is a standard uniqueness theorem. The first lemma just says that the transition probabilities to definable sets are completely determined by the formulas, independently of the system.

Proposition 9.5 *Let (S, Σ, k_a) be a labelled Markov process.*

(i) Each equivalence class in S is a Borel subset.

(ii) The equivalence classes in S refine $[\![\phi]\!]$ for each formula ϕ of the logic.

(iii) $q^{-1}q[\![\phi]\!] = [\![\phi]\!]$ for each formula ϕ of the logic.

Proof . (i): let $t \in S$. Then it is easy to see that the equivalence class containing t is equal to $\bigcap_{t \models \phi} [\![\phi]\!] \setminus \bigcup_{t \not\models \phi} [\![\phi]\!]^c$ which is obviously a Borel subset of S. (ii): Clearly, $[\![\phi]\!] = \bigcup_{t \models \phi} [t]$ where $[t]$ is the equivalence class containing t. (iii): The reversed inclusion is obvious and direct inclusion follows from the fact that if s, t are mapped to the same state, they must satisfy the same formulas, so if $s \in [\![\phi]\!]$ and $t \in q^{-1}q[\![\phi]\!]$, then t must be in $[\![\phi]\!]$ as well. ∎

Lemma 9.6 *Let (S, Σ, k_a) and (S', Σ', k'_a) be two labelled Markov processes. Then for all formulas ϕ and all pairs (s, s') such that $s \approx s'$, we have $k_a(s, [\![\phi]\!]_S) = k'_a(s', [\![\phi]\!]_{S'})$.*

Proof . Suppose that the equation does not hold. Then, say, for some ϕ, $k_a(s, [\![\phi]\!]_S) < k'_a(s', [\![\phi]\!]_{S'})$. We choose a rational number q between these values. Now it follows that $s' \models \langle a \rangle_q \phi$ but $s \not\models \langle a \rangle_q \phi$, which contradicts the assumption that s and s' satisfy all the same formulas. ∎

The next lemmas are Theorem 3.3.5 of [Arv76] and one of its corollaries. We omit the proofs.

Lemma 9.7 *Let X be an analytic Borel space and let $\sim$ be an equivalence relation in X. Assume there is a sequence $f_1, f_2, \ldots$ of real valued Borel functions on X such that for any pair of points x, y in X one has $x \sim y$ if and only if $f_n(x) = f_n(y)$ for all n. Then $X/_\sim$ is an analytic Borel space.*

Lemma 9.8 *Let $(X, \mathcal{B})$ be an analytic Borel space and let $\mathcal{B}_0$ be a countably generated sub-σ-field of $\mathcal{B}$ which separates points in X. Then $\mathcal{B}_0 = \mathcal{B}$.*

The final lemma that we need is a result which gives a condition under which two measures are equal. It is Theorem 10.4 of Billingsley [Bil95].

Lemma 9.9 *Let X be a set and $\mathcal{A}$ a family of subsets of X, closed under finite intersections, and such that X is a countable union of sets in $\mathcal{A}$. Let $\sigma(\mathcal{A})$ be the σ-field generated by $\mathcal{A}$. Suppose that μ_1, μ_2 are finite measures on $\sigma(\mathcal{A})$. If they agree on $\mathcal{A}$ then they agree on $\sigma(\mathcal{A})$.*

Proof of Proposition 9.4: We first show that $S/_\approx$ is an analytic space. Let $\{\phi_i | i \in \mathbf{N}\}$ be the set of all formulas. We know that $[\![\phi_i]\!]_S$ is a Borel set for each i. Therefore the characteristic functions $\chi_{\phi_i} : S \to \{0, 1\}$ are Borel measurable functions. Moreover we have

$$x \approx y \text{ iff } (\forall i \in \mathbf{N}.\ x \in [\![\phi_i]\!]_S \iff y \in [\![\phi_i]\!]_S) \text{ iff } (\forall i \in \mathbf{N}.\ \chi_{\phi_i}(x) = \chi_{\phi_i}(y)).$$

It now follows by Lemma 9.7 that $S/_\approx$ is an analytic space.

Let $\mathcal{B} = \{q([\![\phi_i]\!]_S) : i \in \mathbf{N}\}$. We show that $\sigma(\mathcal{B}) = \Sigma_\approx$. We have $\mathcal{B} \subseteq \Sigma_\approx$, since, by Proposition 9.5 (iii), for any $q([\![\phi_i]\!]_S) \in \mathcal{B}$, $q^{-1}q([\![\phi_i]\!]_S) = [\![\phi_i]\!]_S$ which is in Σ by Lemma 9.1. Now $\sigma(\mathcal{B})$ separates points in $S/_\approx$, for if x and y are different states of $S/_\approx$, take states $x_0 \in q^{-1}(x)$ and $y_0 \in q^{-1}(y)$. Then since $x_0 \not\approx y_0$, there is a formula ϕ such that x_0 is in $[\![\phi]\!]_S$ and y_0 is not. By Proposition 9.5 (iii), it follows that x is in $q[\![\phi]\!]_S$, whereas y is not. Since $\sigma(\mathcal{B})$ is countably generated, it follows by Lemma 9.8, that $\sigma(\mathcal{B}) = \Sigma_\approx$.

We are now ready to define $h_a(t, \cdot)$ over $\Sigma_\approx$ for $t \in S/_\approx$. We define it so that $q : S \to S/_\approx$ is a zigzag morphism (recall that q is measurable and surjective by definition), i.e., for any $B \in \Sigma_\approx$ we put

$$h_a(t, B) = k_a(s, q^{-1}(B)),$$

where $s \in q^{-1}(t)$. Clearly, for a fixed state s, $k_a(s, q^{-1}(\cdot))$ is a sub-probability measure on $\Sigma_\approx$. We now show that the definition does not depend on the choice of s in $q^{-1}(t)$ for if $s, s' \in q^{-1}(t)$, we know that $k_a(s, q^{-1}(\cdot))$ and $k_a(s', q^{-1}(\cdot))$ agree over $\mathcal{B}$ again by the fact that $q^{-1}q([\![\phi_i]\!]_S) = [\![\phi_i]\!]_S$ and by Lemma 9.6. So, since $\mathcal{B}$ is closed under the formation of finite intersections we have, from Lemma 9.9, that $k_a(s, q^{-1}(\cdot))$ and $k_a(s', q^{-1}(\cdot))$ agree on $\sigma(\mathcal{B}) = \Sigma_\approx$.

It remains to prove that for a fixed Borel set B of $\Sigma_\approx$, $h_a(\cdot, B) : S/_\approx \to [0, 1]$ is a Borel measurable function. Let A be a Borel set of $[0, 1]$. Then $h_a(\cdot, B)^{-1}(A) = q[k_a(\cdot, q^{-1}(B))^{-1}(A)]$;

we know that $C = k_a(\cdot, q^{-1}(B))^{-1}(A)$ is Borel since it is the inverse image of A under a Borel measurable function. Now we have that $q(C) \in \Sigma_{\approx}$, since $q^{-1}q(C) = C$: indeed, if $s_1 \in q^{-1}q(C)$, there exists $s_2 \in C$ such that $q(s_1) = q(s_2)$, and we have just proved above that then the $k_a(s_i, q^{-1}(\cdot))$'s must agree, so if $k_a(s_i, q^{-1}(B)) \in A$ for $i = 2$, then it is also true for $i = 1$, so $s_1 \in C$ as wanted. So $h_a(\cdot, B)$ is Borel measurable. This concludes the proof that $S/_{\approx}$ is a **LMP** and q a zigzag morphism. ∎

We now state the main result on logical characterization of bisimulation.

Theorem 9.10 *Two labelled Markov processes are bisimilar if and only if they obey the same formulas of our logic.*

Proof . One direction has already been shown; what remains is to prove that two systems obeying all the same formulas are bisimilar. Suppose that (S, Σ, k_a) and (S', Σ', k'_a) satisfy the same formulas. Instead of defining a span of zigzags directly, we can define a cospan and use the semi-pullback property to infer that S and S' are bisimilar. We first construct a system, (T, Σ_T, j_a), called the *direct sum* of S and S', as follows. We set $T = S \uplus S'$ with the evident σ-field. We define the transition probabilities as follows: $j_a(s, A \uplus A') = k_a(s, A)$ if $s \in S$ and $j_a(s', A \uplus A') = k'_a(s', A')$ if $s' \in S'$ where $A \in \Sigma$ and $A' \in \Sigma'$. There are the evident canonical injections ι, ι' which are *not* zigzags because they are not surjective. We know that the quotient system $(T/_{\approx}, \Sigma_{\approx}, h_a)$ is a **LMP** and that the canonical projection r from T to $T/_{\approx}$ is a zigzag morphism. Thus we have the diagram of figure 2. The composites $r \circ \iota$ and $r \circ \iota'$ are measurable and surjective, henceforth we

$$\begin{tikzcd}
S \arrow[dr, "\iota"] \arrow[ddr, "q"'] & & S' \arrow[dl, "\iota'"'] \arrow[ddl, "q'"] \\
& T \arrow[d, "r"'] & \\
& T/_{\approx} &
\end{tikzcd}$$

Figure 2: Constructing a cospan.

call them q and q' respectively. To see that q, q' are surjective we recall that an equivalence class must, by hypothesis, include members of both S and S'. It remains to prove the zigzag property for q and q'. So take a set B in $\Sigma_{\approx}$ and $s \in S$. Then

$$\begin{aligned}
h_a(q(s), B)) &= j_a(\iota(s), r^{-1}(B)) \quad \text{by Proposition 9.4} \\
&= k_a(s, r^{-1}(B) \cap S) \quad \text{by definition of } j_a \text{ and because } s \in S \\
&= k_a(s, q^{-1}(B))
\end{aligned}$$

This proves that q and similarly q' are zigzag morphisms. Thus we have defined a cospan of zigzag morphisms and using the semi-pullback theorem there is a corresponding span, hence S and S' are bisimilar. ∎

Note that this implies the result for discrete systems without using the minimum deviation assumption used by Larsen and Skou.

Definition 9.11 *Given two labelled Markov processes S and S' that are bisimilar, we say that two states $s \in S$ and $s' \in S'$ are* ***bisimilar****, denoted $s \sim s'$, if there is a generalized span $f : U \longrightarrow S, g : U \longrightarrow S'$ such that for some $u \in U$ we have $f(u) = s$ and $g(u) = s'$.*

Note that in the above U is in $\mathbf{LMP}^*$. It follows from the existence of semi-pullbacks that $\sim$ is an equivalence relation and we have:

Corollary 9.12 *Let S and S' be two labelled Markov processes that are bisimilar and let $s \in S$ and $s' \in S'$, then $s \sim s$ if and only if $s \approx s'$.*

We also have the following corollary.

Corollary 9.13 *Let S_1 and S_2 be two labelled Markov processes, and let U be defined as in the semi-pullback construction in the diagram of figure 3.*

$$\begin{tikzcd}
& U \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
S_1 \arrow[dr, "f_1"'] & & S_2 \arrow[dl, "f_2"] \\
& (S_1 \uplus S_2)/_{\approx} &
\end{tikzcd}$$

Figure 3: The span construction

Then for $s_i \in S_i$, $i = 1, 2$, $s_1 \sim s_2 \iff \exists u \in U$ such that $\pi_i(u) = s_i$, $i = 1, 2$.

The quotient construction has the following couniversal property. In the case of finite state systems this says that the quotienting construction gives the minimal finite state system bisimilar to the given one.

Proposition 9.14 *Let S and T be two labelled Markov processes. If $S \sim T$, then there exists a unique zigzag morphism r from S to $T/_{\approx}$.*

Proof . If $(S, \Sigma, k_a) \sim (T, \Sigma_T, l_a)$, there is a span (U, Σ_U, j_a) with zigzag morphisms $f : U \to S$ and $g : U \to T$. Let $s \in S$. Since f, g and $q : T \to T/_{\approx}$ are zigzag morphisms, for every formula ϕ we have,

$$s \models \phi \iff \forall u \in f^{-1}(s).u \models \phi \iff \forall u \in f^{-1}(s).g(u) \models \phi \iff \forall u \in f^{-1}(s).qg(u) \models \phi$$

This implies that all $u \in f^{-1}(s)$ are mapped by qg to the same state $t \in T/_{\approx}$ and that we can set $r(s) = t$. This makes the diagram commute. Surjectivity of r is obvious; to see it is also Borel measurable, let $A \in \Sigma_{\approx}$. Then $r^{-1}(A) = f(g^{-1}q^{-1}A)$, $B_1 := g^{-1}q^{-1}A$ is obviously Borel in U and so is $B_2 := g^{-1}q^{-1}A^c$. Now we have not only that B_1 and B_2 are disjoint, but their images under f are also disjoint. To see this, suppose the contrary. Then there exist $u_i \in B_i$ such that $fu_1 = fu_2$; but since the diagram commutes, it implies that $qg(u_1) = qg(u_2)$, which is a contradiction to the definition of the B_i's. Thus we have that fB_1 and fB_2 are disjoint analytic sets of S; since analytic sets are separable by Borel sets and since $fB_1 \cup fB_2 = S$, fB_1 and fB_2 must be Borel sets of S, concluding the proof that r is Borel measurable. We now show that r is zigzag; let $s \in S, A \in \Sigma_\approx$ and $u \in f^{-1}(s)$. Then

$$\begin{aligned} h_a(r(s), A) &= l_a(g(u), q^{-1}A) \\ &= j_a(u, g^{-1}q^{-1}A) \\ &= j_a(u, f^{-1}r^{-1}A) \\ &= k_a(s, r^{-1}A) \end{aligned}$$

Finally, r is unique because every state s is mapped in $T/_\approx$ to the only state that satisfies the same formulas as it does (since in $T/_\approx$ there is no pair of distinct states satisfying the same formulas). ∎

The next corollary says that the quotient construction gives canonical representatives for labelled Markov processes.

Corollary 9.15 *Let S and T be two labelled Markov processes. $S \sim T$ if and only if $S/_\approx$ and $T/_\approx$ are isomorphic in **LMP**.*

Now we consider the other logics. The proof of the following proposition is very easy and is only sketched here.

Proposition 9.16 *All the logics defined in the previous section characterize bisimulation.*

Proof . First we note that there is no need to prove that if two systems satisfy all the same formulas they are bisimilar because all these logics extend $\mathcal{L}_0$.

For the other direction we have to show two things just as in Propositions 9.1 and 9.2. The first is that the sets definable by formulas are measurable. We show that for all formulas ϕ of all our logics, we have $[\![\phi]\!] \in \Sigma$. $[\![\mathrm{Can}(a)]\!] = k_a(\cdot, S)^{-1}((0,1])$ which is in Σ and $[\![\Delta_a]\!]$ is its complement and hence is in Σ. Now for $\mathcal{L}_\wedge$ and $\mathcal{L}_\neg$ we only have to show that if $[\![\phi]\!] \in \Sigma$, then so is $[\![\neg\phi]\!]$ which is straightforward, and if $\forall i \in \mathbf{N}$, $[\![\phi_i]\!] \in \Sigma$, then so is $[\![\bigwedge_{i\in\mathbf{N}} \phi_i]\!]$ which is also straightforward since Σ is a σ-field. The results follow by structural induction.

The second is that a state and its image under a zigzag satisfy the same formulas. More precisely, we show that for every zigzag morphism $f : S \to S'$ and every state $s \in S$, s and $f(s)$ satisfy all the same formulas. For $\mathcal{L}_{\mathrm{Can}}$ and $\mathcal{L}_\Delta$, since f is zigzag, $k_a(s, S) = k'_a(f(s), S')$, so $s \models \mathrm{Can}(a)$ if and only if $s' \models \mathrm{Can}(a)$, and $s \models \Delta_a$ if and only if $s' \models \Delta_a$. Now for $\mathcal{L}_\wedge$ and $\mathcal{L}_\neg$ it is obvious by structural induction. ∎

Although these logics all characterize bisimulation, they don't all characterize equivalence classes, in the sense that there does not necessarily exist a formula for each equivalence class which is satisfied only by states in that class. The most powerful logic does characterize equivalence classes.

Proposition 9.17 *The logic $\mathcal{L}_\wedge$ characterizes equivalence classes of arbitrary Markov processes.*

Proof . Let Θ be an equivalence class of processes with respect to $\mathcal{L}_\wedge$, and $F(\Theta)$ the set of *finite* formulas (i.e. formulas of $\mathcal{L}_0$) which are satisfied by one member t (hence by all members) of Θ; clearly, this set is countable. Then $\Theta = \bigcap_{\phi \in F(\Theta)} [\![\phi]\!] = [\![\bigwedge_{\phi \in F(\Theta)} \phi]\!]$: indeed, $s \approx t$ if and only if s satisfies all the same formulas of $\mathcal{L}_0$ as t i.e. if and only if s satisfies $\bigwedge_{\phi \in F(\Theta)} \phi$. ∎

For finite-state systems negation by itself is enough to characterize equivalence classes.

Proposition 9.18 *The logic $\mathcal{L}_\neg$ doesn't characterize equivalence classes of Markov processes, but given a finite Markov chain, for every bisimulation equivalence class, there exists a formula of $\mathcal{L}_\neg$ such that a state is in the equivalence class if and only if it satisfies this formula.*

Proof . The last fact is well known [Arn94]: using the same proof as for the last proposition, we see that given a finite Markov chain, there exists a finite set $A \subseteq F(\Theta)$ such that $\Theta = \bigcap_{\phi \in A} [\![\phi]\!] = [\![\wedge_{\phi \in A} \phi]\!]$, where $\wedge_{\phi \in A} \phi$ is a (finite) formula, as wanted.

This argument does not work if we consider the problem of writing a formula characterizing equivalence classes of arbitrary finite Markov chains and not just the equivalence classes of states within a fixed Markov chain. This happens because there are infinitely many finite Markov chains. We now prove that $\mathcal{L}_\neg$ does not characterize equivalence classes of even finite Markov chains. To do so, consider the equivalence class of the single-state process that can do action a with probability 1 (and then ends up in the same state); this process can do infinitely many a's, call it S_∞. Now let S_n be the process having $n+1$ states that can do the action a n times and then nothing. There is no finite formula that distinguishes S_∞ from all the S_n's at the same time. We prove this by showing that

if $S_\infty \models \phi$, then $\exists N. \forall k \geq N$, $S_n \models \phi$, and
if $S_\infty \not\models \phi$, then $\exists N. \forall k \geq N$, $S_k \not\models \phi$.

The base case corresponding to T is trivial. So assume the statement is true for formulas ϕ, ϕ_1 and ϕ_2. Now assume $S_\infty \models \phi_1 \wedge \phi_2$. Then there exist N_1 and N_2 such that $\forall k \geq N_i, S_k \models \phi_i$, $i = 1, 2$. For $N = \max(N_1, N_2)$ we have $\forall k \geq N, S_k \models \phi_1 \wedge \phi_2$. If $S_\infty \not\models \phi_1 \wedge \phi_2$. Then there exists $i \in \{1, 2\}$ such that $S_\infty \not\models \phi_i$. So there is N such that $\forall k \geq N_i, S_k \not\models \phi_i$ so $\forall k \geq N, S_k \not\models \phi_1 \wedge \phi_2$. The induction step corresponding to negation is obvious. Finally let $S_\infty \models \langle b \rangle_q \phi$. Then $b = a$ and $S_\infty \models \phi$. By induction hypothesis, there exists an N such that $\forall k \geq N, S_k \models \phi$, so $S_{k+1} \models \langle a \rangle_q \phi$, i.e. $\forall k \geq N+1, S_k \models \langle a \rangle_q \phi$ as wanted. For $S_\infty \not\models \langle b \rangle_q \phi$, there are two cases. Either $b \neq a$ or $b = a$. In the first case, no S_k satisfies $\langle b \rangle_q \phi$, so take $N = 0$; in the second case, we have $S_\infty \not\models \phi$, so there is an N such that $\forall k \geq N, S_k \not\models \phi$. Since for all $k \geq 0$ we have $k_a(S_k, S_{k+1}) = 1$, then for all $k \geq N$ $S_{k+1} \not\models \langle a \rangle_q \phi$, so the statement is true with $N+1$ in place of N, and the proof is complete. ∎

Note that the example given in the previous proof cannot be applied to states inside a finite Markov chain, since it involves infinitely many states. Nevertheless, it can be used to show that neither $\mathcal{L}_\neg$ nor $\mathcal{L}_\Delta$ can characterize equivalence classes inside a discrete system satisfying the minimal deviation assumption defined by Larsen and Skou, since the processes defined can all be glued into one system with only 1 or 2-branching states, each with uniform distribution.

We summarize the results of this section as follows. The logic $\mathcal{L}_0$ characterizes bisimulation of probabilistic processes, without any hypothesis of finite branching and for systems that may have continuous state spaces. The various stronger logics also have this property. In the weak logic $\mathcal{L}_0$ one cannot write a formula such that any bisimulation equivalence class of states is described by this formula. This holds even for simple finite state systems. On the other hand with just negation added to $\mathcal{L}_0$ one can characterize the bisimulation equivalence classes of states in a fixed finite state Markov chain but not in countable discrete Markov chains. One cannot characterize equivalence classes of Markov chains with $\mathcal{L}_\neg$. One can characterize bisimulation equivalence classes of states in an arbitrary Markov process using infinitary conjunction.

10 Finite-State Systems

The previous proof uses machinery that is unconventional in concurrency theory. The result gives a characterization of bisimulation in terms of a logic without negation, an unexpectedly weak logic. It is natural to question the constructive content of such a proof. In this section we show that for finite-state systems we can prove a completely constructive version of the characterization result. This does also use a measure theoretic fact that one could conceivably have thought of without knowing any measure theory, but it is unlikely.

The following theorem is a special case of the general theorem but is proved without the powerful tools invoked for the general theorem. Of course it already follows from the general theorem but it is of interest to see what types of arguments are needed to prove the purely finite state case. In the proof below a result from measure theory *is* used but not the full apparatus needed for the general theorem. This gives some indication that one is unlikely to have thought of the finite-state version using only ideas from concurrency theory; it is unlikely that one would have thought of theorem 3.2 of Billingsley's book [Bil95] on purely combinatorial grounds.

Theorem 10.1 *Two finite Markov chains satisfying all the same formulas are bisimilar.*

Proof . Let

$$\mathcal{S} = (S, P_a : S \times S \to [0,1]),$$

$$\mathcal{S}' = (S', P'_a : S' \times S' \to [0,1]),$$

be two Markov chains satisfying all the same formulas. We will show that we can construct a new Markov chain $\mathcal{U}$ and zigzag morphisms $\pi : \mathcal{U} \to \mathcal{S}$ and $\pi' : \mathcal{U} \to \mathcal{S}'$.

Let $U = \{(s, s') \in S \times S' \mid s \approx s'\}$ and $\pi : U \to S$ and $\pi' : U \to S'$ be the projection maps. We define $h_a : U \times U \to [0,1]$ as follows:

$$h_a((s,s'),(t,t')) = \frac{P_a(s,t)P'_a(s',t')}{P_a(s,[t]_S)},$$

where $[t]_S$ is the $\approx$-equivalence class in S containing t.
Claim: *for all $a \in \mathsf{L}$, for all $(s,s'),(t,t') \in U$, we have $P_a(s,[t]_S) = P'_a(s',[t']_{S'})$.*
To prove the claim, let (T, j_a) be the *direct sum* of $\mathcal{S}$ and $\mathcal{S}'$. Of course we have that for all formulas ϕ $P_a(s,[\![\phi]\!]_S) = P'_a(s',[\![\phi]\!]_{S'})$, which implies that $j_a(s,[\![\phi]\!]_T) = j_a(s',[\![\phi]\!]_T)$. Now we know that the set $\mathcal{F}$ of formulas is closed under intersections, so it is a π-system as defined by Billingsley in [Bil95]. Let $\mathcal{C}$ be the class of subsets C of T satisfying

$$j_a(s,C) = j_a(s',C).$$

Then $\mathcal{C}$ contains $\mathcal{F}$ and is a λ-system, that is

1. $T \in \mathcal{C}$;

2. $C \in \mathcal{C}$ implies $C^c \in \mathcal{C}$; and

3. $C_1, C_2, \ldots, \in \mathcal{C}$ and $C_n \cap C_m = \emptyset$ for $n \neq m$ imply $\cup_n C_n \in \mathcal{C}$.

So by Theorem 3.2 of [Bil95], $\mathcal{C}$ contains the σ-field generated by $\mathcal{F}$. This means that, since the equivalence classes $[t]_T$ are in this σ-field, we have

$$P_a(s, [t]_S) = j_a(s, [t]_T) = j_a(s', [t']_T) = P'_a(s', [t']_{S'})$$

and the claim is proved.

It remains to prove that the projections are zigzag morphisms. So let $a \in \mathsf{L}$, $(s, s') \in U$, and $t' \in S'$. We have

$$\begin{aligned}
h_a((s,s'), {\pi'}^{-1}(t')) &= h_a((s,s'), [t']_S \times \{t'\}) \\
&= \sum_{t \in [t']_S} \frac{P_a(s,t) P'_a(s',t')}{P_a(s,[t]_S)} \\
&= \sum_{t \in [t']_S} \frac{P_a(s,t) P'_a(s',t')}{P'_a(s',[t']_{S'})} \\
&= \frac{P'_a(s',t')}{P'_a(s',[t']_{S'})} \sum_{t \in [t']_S} P_a(s,t) \\
&= \frac{P'_a(s',t')}{P'_a(s',[t']_{S'})} P_a(s,[t']_S) \\
&= P'_a(s',t').
\end{aligned}$$

To show that π is a zigzag morphism is even more straightforward. ∎

Note that this gives the result for discrete systems without the minimum deviation assumption used by Larsen and Skou.

10.1 The algorithm

The proof of the last section uses machinery that is unconventional in concurrency theory. The result gives a characterization of bisimulation in terms of a logic without negation, an unexpectedly weak logic. It is natural to question the constructive content of such a proof. Now a careful analysis of constructivity in measure theory is beyond the scope of this work, but we will show that in the case of finite state systems there is an algorithm which decides bisimilarity of finite Markov chains and also produces a witnessing formula *from the logic* $\mathcal{L}_0$ in case the systems are not bisimilar. The algorithm is a modification of Cleaveland's algorithm [Cle90].

The algorithm shown in figure 4 allows us to distinguish states that do not satisfy the same formulas. Beginning with D_1 containing only the set S of all states, bisim operates as follows: for each $a \in \mathsf{L}$ and B' in D_1, it "splits" every set B of D_1 into subsets having the same probability of jumping to B' with action a; this is done until it does not modify D_1. At the end, all states satisfying the same formulas will belong to exactly the same sets in D_1.

The function split first lists in L all possible values of $P_a(b, B')$ for $b \in B$. Then, for every possible value in the list L, the states in B that can jump into B' with probability at least this value are added to the set D which contains B at the beginning. Before adding a set to D, split checks if it is already a member of $D \cup D_1 \cup D_2$; if not, it defines the formula represented by the set

```
bisim(S, L, D)
begin
  F(S) := T
  D1 := {S}
  D2 := ∅
  while D1 ≠ D2 do
    for each a ∈ L and B' ∈ D1 do begin
      D2 := D1
      D1 := ∅
      for each B ∈ D2 do D1 := D1 ∪ split(B, a, B')
    end
end
```

```
split(B, a, B')
begin
  L := ∅
  D := {B}
  for each b ∈ B do
    L := L ∪ {P_a(b, B')}
  L := L \ {0}
  while L ≠ ∅ do begin
    l := min L
    L := L \ {l}
    B1 := {b ∈ B : P_a(b, B') ≥ l}
    if B1 ∉ D ∪ D1 ∪ D2 then
    begin
      F(B1) := F(B) ∧ ⟨a⟩_l F(B')
      D := D ∪ B1
    end
  end
  return D
end
```

Figure 4: The bisimulation algorithm

and adds the set to D; otherwise it does not add the set to D. This ensures that we don't compute a new formula for a set that already has a shorter one.

As shown in the following proposition, given two states, we can determine if they satisfy all the same formulas by checking if they belong to the same sets of D_1. If they do not, by finding the "first" set B in D_1 that distinguishes them, we get a formula $F(B)$ that also distinguishes them. The algorithm itself doesn't record an order of creation on the sets B, but it could be easily modified to do so.

Proposition 10.2 *Two states satisfy the same formulas iff they belong to exactly the same sets in D_1 at the end of executing the algorithm above.*

Proof . First note that the algorithm must terminate since $2^{|S|}$ is finite and since the while-loop in bisim is executed at most once without increasing the set D_1.

Now we prove necessity by showing that in bisim, every element of D_1 or D_2 represents a formula, i.e., for every $B \in D_1 \cup D_2$, there exists a formula ϕ such that $B = [\![\phi]\!]$. This will be proved by induction. The whole set S represents the formula T. The set $D_1 \cup D_2$ is transformed inside the nested for-loops, so suppose that after n iterations of the last for-loop inside bisim, every element of D_1 or D_2 represents a formula. Then we must show that the set returned by $\mathsf{split}(B, a, B')$ also contains sets which represent formulas. It is easy to see that if B and B' represent respectively formulas ϕ and ϕ', the set $\{b \in B : P_a(b, B') \geq l\}$ represents the formula $\phi \wedge \langle a \rangle_l \phi'$, so that the function split returns the set

$$D = \bigcup_{l \in L} \{[\![\phi \wedge \langle a \rangle_l \phi']\!]\},$$

so each set in D_1 at the end of executing this algorithm represents the set of states that satisfy some formula. This implies that if two states satisfy the same formulas, they must be in the same sets of D_1 (and D_2).

For sufficiency, we want to show that if two states s, s' don't satisfy the same formulas they are not in the same sets of D_1. To do so, we will show by structural induction on formulas that every formula is represented by a set in D_1 when the algorithm is finished. So assume the algorithm is finished and hence that D_1 is constructed. The formula T is represented by $S \in D_1$. Now assume ϕ and ψ are represented by sets in D_1. If $\langle a_0 \rangle_l \phi$ is not represented for some label a_0 and some rational q (and hence is satisfied for some $s \in S$), then use $a := a_0$ and $B' := [\![\phi]\!]$ in the for-loop. Then D_1 will be modified by the call to $\mathsf{split}(B, a, B')$, with $l := \min\{k_a(s, [\![\phi]\!]) \geq q | s \in B\}$; l exists since $[\![\langle a_0 \rangle_l \phi]\!]$ is not empty. This contradicts the fact that the algorithm is finished, so $\langle a_0 \rangle_l \phi$ is represented in D_1. Now consider $\phi \wedge \psi$. By construction of the algorithm, ϕ and ψ can be written as

$$\phi := \langle a_1 \rangle_{q_1} \phi_1 \wedge \cdots \wedge \langle a_n \rangle_{q_n} \phi_n$$

$$\psi := \langle a_1 \rangle_{p_1} \phi_1 \wedge \cdots \wedge \langle a_n \rangle_{p_n} \phi_n$$

where n is the total number of for-loops executed during the algorithm, $\phi_1 = \mathsf{T}$, $(S, a_1, [\![\phi_1]\!])$ are the parameters given in the first call to split, and $([\![\wedge_{1 \leq i < k} \langle a_i \rangle_{q_i} \phi_i]\!], a_k, [\![\phi_k]\!])$ are the parameters given in the k^{th} call to split. Hence

$$\phi \wedge \psi = \langle a_1 \rangle_{r_1} \phi_1 \wedge \cdots \wedge \langle a_n \rangle_{r_n} \phi_n$$

where $r := \max\{p_i, q_i\}$. We show by induction on n that if $[\![\phi]\!]$, $[\![\psi]\!] \in D_1$, then $[\![\phi \wedge \psi]\!] \in D_1$. If $n = 1$, then $\phi \wedge \psi = \phi$ or ψ, so $[\![\phi \wedge \psi]\!] \in D_1$. Now assume $B := [\![\wedge_{1 \leq i < n} \langle a_i \rangle_{r_i} \phi_i]\!] \in D_1$. Then if for all $b \in B$, $k_{a_n}(b, [\![\phi_n]\!]) < r_n$, then $[\![\phi \wedge \psi]\!]$ is the empty set and hence is represented in D_1. Else as above consider $l := \min\{k_{a_n}(s, [\![\phi_n]\!]) \geq r_n | s \in B\}$; then $\mathsf{split}(B, a_n, [\![\phi_n]\!])$ will create the set $B \cap [\![\langle a_n \rangle_{r_n} \phi_n]\!] = [\![\phi \wedge \psi]\!]$ since $l \in L$, so this set is in D_1. This concludes the proof of sufficiency. ∎

11 Related Work

There has been a substantial amount of work on probabilistic transition systems and their associated equivalences. As far as we are aware, none of them have looked at bisimulation for continuous state spaces, except for the work of deVink and Rutten [RdV97] discussed below.

The starting point of work in the area of probabilistic semantics are the fundamental papers of Saheb-Djahromi [SD78, SD80] and of Kozen [Koz81, Koz85]. These are concerned with domain theory and programming languages rather than with process equivalences, but they both introduced nontrivial measure-theoretic ideas. Kozen[5] also noticed a very interesting duality between state-transformer semantics, as described by stochastic kernels, and a probabilistic predicate-transformer semantics in which programs are seen as inducing linear continuous maps on the Banach algebra of bounded measurable functions. This duality influenced our search for a logical characterization, though the logic we actually used owes more to Larsen and Skou [LS91].

The first paper with an abstract categorical approach to stochastic processes is by Giry [Gir81]. She studies categorical constructions rather than process equivalences. Her work - inspired originally by Lawvere - does provide some of the mathematical underpinnings of our ideas. In particular she shows that the stochastic kernels (conditional probability distributions) that we use to define transition probabilities arise as the Kleisli category of a monad, which is a natural generalization of the powerset monad to the probabilistic case. If we recall that the category **Rel** of sets and relations

[5]Kozen credits Plotkin for suggesting this possibility.

is the Kleisli category of the powerset monad, we see that the stochastic kernels can reasonably be viewed as the probabilistic analogues of relations [Pan98]. This makes the analogy between labelled Markov processes and ordinary transitions quite striking.

The fundamental work on this topic is the paper by Larsen and Skou [LS91] which analyzes not just bisimulation but also testing. Our work extends theirs in two fundamental ways. It applies to continuous state space systems. The mathematics is therefore entirely different and even the structure of the arguments is different. If we specialize to the discrete case, we have extended their work in two ways, we characterize bisimulation with a logic that is weaker - it has no negative formulas or disjunction - and we do not have any finite branching assumption or minimum deviation assumption. Of course the minimum deviation assumption is a reasonable one in the context of discrete systems, but it is interesting to note that it is not needed. We know of no proof of the results of this paper *even if we are only interested in discrete systems* that does not use some of the apparatus of measure theory, at the very least the $\lambda - \pi$ theorem of Dynkin. Thus it is not surprising that these results were not discovered before.

The work of Joyal, Nielsen and Winskel [JNW96] and of Cheng and Nielsen [CN95] provided vital clues. By analogy with Joyal et al., we define probabilistic bisimulation as spans of zigzags. However we never succeeded in casting our work into their open map framework, nor did we succeed in giving a presheaf presentation of our work.

From the point of view of applications there have been a number of very interesting results. The most interesting work, in our opinion, is the work of Jane Hillston [Hil94] on developing a process algebra for performance evaluation. Her work is not comparable to ours, because she works with temporal delay in discrete space Markov chains. The main point of her work is a compositional approach to performance evaluation. In her framework, she address continuous time in the following way. The systems being modelled are described by a probabilistic process algebra called PEPA. The semantics of PEPA are given in terms of labelled transition systems. Associated with the transitions is a continuous time random process with an exponentially distributed delay. Associated with each type of action is a different rate. Indeterminacy is resolved by races between events executing at different rates. In her work a crucial role is played by a congruence called strong equivalence. Strong equivalence is in fact closely analogous to Larsen-Skou bisimulation and - as she points out - to an old idea in queuing theory called *lumpability*. It is defined in a way very similar to the way that Larsen and Skou proceed, with the difference being that instead of using the probabilities associated with the actions she uses the *rates* associated with the actions. In this sense it is related to our probabilistic bisimulation but, of course, her treatment of time is totally different.

The other area where bisimulation has appeared in a continuous context is the theory of timed automata [AD94]. Here the basic framework is ordinary automata theory augmented with clocks. These concepts have proved to be very useful in practice and lead to fundamental theoretical questions. The main technical result is the so called region construction. This is a quotienting of the state space of the timed automaton - which is a continuum because of the clocks - by an equivalence relation, which, like bisimulation, has a coinductive definition. By imposing certain conditions on the way clocks can be read they guarantee that the region construction leads to a finite-state system. The bisimulation relation that they use is not like ours; theirs is concerned with how actions are enabled as time passes. The point is that time "evolves" in a simple uniform way; the resulting equivalence is more like a trace equivalence.

In our systems the state space is a continuum not because time is a continuum, but because the state space is continuous. The time steps can lead to almost arbitrary jumps and hence lead to more complicated evolutions. It would be much more complicated to obtain something analogous to the region construction in our setting.

The region construction is also used by Henzinger [HHWT97] for his linear hybrid automata. This is a setup even closer in spirit to ours since there are explicitly continuous state spaces. Here also the bisimulation is used to collapse the continuum state space to a finite state space.

The group at Oxford has been developing an extensive theory of probabilistic systems; see the collection of reports available from the web [Pro]. The focus has been on equational laws satisfied by processes. From the semantic point of view, they have extensively developed and enriched Kozen's [Koz85] predicate-transformer view. They have considered continuous state space systems and have incorporated nondeterminism in their framework. As with the earlier work of Kozen [Koz81, Koz85], stochastic kernels play an important role.

There are several papers now on probabilistic analysis, modeling and verification. There are even several papers on probabilistic process algebra analyzing notions of testing and simulation, investigating model checking and exploring various other ideas [SL94, vGSST90, JL91, JY95, JS90, CSZ92, BK96, HK96]. There are several interesting practical developments, other than PEPA, which are worthy of attention. In particular telecommunication [AJKvO97], real-time systems [BLFG95] and modeling physical systems [GSS95] are areas where probabilistic systems are very important. It is particularly for the last type of application that we expect that the continuous space formalism developed here will be useful. In a recent paper [GJP99] a programming language with probabilistic choice and recursion was developed. This immediately puts the work in the realm of continuous spaces. The semantics of such systems involved the basic ideas of measure theory that we found useful in the present work.

The other related paper is the work of de Vink and Rutten [dVR97]. This is also an investigation into the realm of continuous state spaces. They define a probabilistic transition system as a coalgebra of a suitable kind using the Giry monad mentioned above. The definition of coalgebra homomorphism is easily seen to be exactly the same as our definition of zigzag morphism. Indeed we knew this at an early stage of the work and it constituted, for us, evidence that the definition of zigzag was correct. Their coalgebraic approach is definitely attractive and it should be interesting to explore whether there is any way of extending their results to metric spaces such as the reals. In particular there should be interesting links between logic and coalgebras. However they work with ultrametric spaces, not with the kind of metric spaces that actually arise in physical examples. Thus, for example, the real numbers do not form an ultrametric space and their results do not apply there; nor is any easy extension likely to work. Furthermore they only are able to show that bisimulation is an equivalence relation in the case that the space is discrete. We feel however that this work is definitely of interest.

The algorithm for bisimulation of finite-state systems owes a lot to the treatment of Cleaveland [Cle90]. His algorithm is based on working with partitions and if the logic has negation this is an obvious strategy. Our algorithm for $\mathcal{L}_0$ is based on using a nested family of sets rather than a partition. We have not explored systematically whether any of the other logics leads to formulas of smaller size. Medium sized examples based on our implementation do not suggest that there is a significant penalty in using the simplest logic but we have not tried "pathological" examples.

12 Conclusions

The main contribution of the present paper is an exploration of the concept of probabilistic bisimulation when the state space is continuous. The most significant result was the fact that bisimulation can be characterized by a simple negation-free logic. This opens the way to a logical treatment of simulation. It means for example that two-way simulation is bisimulation, very close to the situation with deterministic processes (in a nonprobabilistic setting).

We have begun using our framework to look at the problem of modelling implemented systems and reasoning about them. The particular example we are working on is an industrial example from the avionics industry. Our modelling experiences with this extensive example are:

- continuous-state discrete-time models occur naturally in practical examples,
- these examples often feature a computational setting, possibly a controller or communication software,
- these systems are inherently probabilistic and the reasoning has to be probabilistic.

We are far from ready to verify such systems automatically with tools available today. We hope to develop a full-scale automated verification within the next two to three years.

A new perspective on system theory offered by computer science is *compositionality*. Thus the traditional theory of stochastic processes [CM65] is concerned with a detailed analysis of the time evolution of systems behaving according to probabilistic laws, but very little is ever done to analyze the behaviour of coupled systems in a systematic way. Computer scientists, on the other hand, have stressed compositionality as a way to attack the formidable intricacy of the systems they have dealt with. For example, in Hillston's work, compositionality is the key contribution of her approach to performance modelling. Of course our work is just a first step; one needs a calculus to describe systems. We have developed such a calculus in the context of concurrent constraint programming [GJP99]. The calculus described there is asynchronous, unlike the process algebra paradigm of synchronous communication that we have addressed here, so we do not have a perfect match with the synchronous formalism of the present work, but it is now clear to us how to go about bridging that gap.

To conclude, the topics that we are working on now are

- the development of calculi for continuous state systems
- the extension to continuous time
- the theory of approximation for continuous state space systems
- metrics between processes and
- the development of a substantial case study.

Acknowledgments

Abbas Edalat would like to thank Terry Lyons for helpful discussions and EPSRC (UK) for support. Prakash Panangaden would like to thank Samson Abramsky, Richard Blute, Vineet Gupta, Jørgen Hoffman-Jørgensen, Radha Jagadeesan, and Glynn Winskel for helpful discussions. We are all grateful to the referees for their careful reading of the manuscript. Prakash Panangaden would like to thank BRICS for its hospitality during the course of this work. Josée Desharnais and Prakash Panangaden thank NSERC (Canada) for its support.

A Relevant Concepts from Measure Theory

For completeness, we give the relevant definitions from measure theory in this section. We assume that the reader knows the basic ideas of measure theory and probability as expounded in, for example "Probability and Measure" by Billingsley [Bil95] or "Real Analysis and Probability" by Ash [Ash72] or the book with the same title by Dudley [Dud89] or "Introduction to Measure and Probability" by Kingman and Taylor [KT66].

Definition A.1 *A* **σ-field** *on a set X is a family of subsets of X which includes X itself and which is closed under complementation and countable unions.*

A set equipped with a σ-field is called a *measurable space*. Given a topological space $(X, \mathcal{T})$, we can define the σ-field, often written $\mathcal{B}$, generated by the open sets (or, equivalently, by the closed sets). This is usually called the *Borel algebra*.

Definition A.2 *Given a σ-field (X, Σ), a* **subprobability measure** *on X is a $[0,1]$-valued set function, μ, defined on Σ such that*

- $\mu(\emptyset) = 0$,
- *for a pairwise disjoint, countable collection of sets, $\{A_i | i \in I\}$, in Σ, we require*

$$\mu(\bigcup_{i \in I} A_i) = \sum_{i \in I} \mu(A_i).$$

In addition, for probability measures we require $\mu(X) = 1$.

It is worth clarifying how the word "measurable" is used in the literature. Given a σ-field Σ on a set X one says "measurable set" for a member of Σ. Suppose that one has a measure μ. One can have the following situation. There can be sets of measure zero which contain non-measurable subsets. Because these sets are not measurable one cannot say that they have measure zero. This happens with Lebesgue measure on the Borel sets in the real line, for example. There is a "completion" procedure[6] which produces a larger σ-field and an extension of the original measure in such a way that all subsets of sets of measure zero are measurable and have measure zero. The completion works by adding to the σ-field all sets X such that there exist measurable sets Y, Z having the same measure, with $Y \subseteq X \subseteq Z$. When applied to the Borel subsets of the real line we get the so called Lebesgue measurable sets. One often uses the phrase "measurable set" to mean a set which belongs to the completed σ-field rather than the original σ-field.

Definition A.3 *A set X in a measurable space (S, Σ) is said to be* ***universally measurable*** *if for every finite measure μ there exist Y and Z in Σ such that $Y \subseteq X \subseteq Z$ and $\mu(Y) = \mu(Z)$.*

What this means is that if we take any measure on the original measurable space and complete it, the set X will always be measurable in the resulting completed σ-field. This definition can be generalized to σ-finite measures but for applications to probability theory finite measures are enough. Note that any measurable set is obviously universally measurable.

Definition A.4 *A function $f : (X, \Sigma_X) \longrightarrow (Y, \Sigma_Y)$ between measurable spaces is said to be* **measurable** *if $\forall B \in \Sigma_Y . f^{-1}(B) \in \Sigma_X$.*

[6] This is an unfortunate name because it gives the mistaken impression that the result cannot be further extended.

Definition A.5 *A function $f : (X, \Sigma_X) \longrightarrow (Y, \Sigma_Y)$ between measurable spaces is said to be* ***universally measurable*** *if $\forall B \in \Sigma_Y . f^{-1}(B)$ is universally measurable.*

The next several definitions and results pertain to analytic spaces.

Definition A.6 *A* ***Polish*** *space is the topological space underlying a complete, separable metric space; i.e. it has a countable dense subset.*

Definition A.7 *An* ***analytic*** *space is the image of a Polish space under a continuous function from one Polish space to another.*

The following proposition [Dud89] gives equivalent definitions of analytic set.

Proposition A.8 *Suppose that X and Y are Polish spaces and f is a function from X to Y. The following are equivalent:*

- *f is continuous and A is the image of X under f,*
- *f is measurable and A is the image of X under f,*
- *f is continuous and A is the image of a Borel subset B of X,*
- *f is measurable and A is the image of a Borel subset B of X,*
- *$g : \mathbf{N}^\infty \longrightarrow Y$ is continuous and A is the image of $\mathbf{N}^\infty$ and*
- *$g : \mathbf{N}^\infty \longrightarrow Y$ is measurable and A is the image of $\mathbf{N}^\infty$.*

Thus in this definition it turns out to be equivalent to say "measurable" image and it makes no difference if we take the image of the whole Polish space or of a Borel subset of the Polish space.

Analytic spaces are more general than Polish spaces but they also have the basic property that regular conditional probability distributions can be defined on them. These regular conditional probability distributions are the basic building blocks for the whole theory, for example the stochastic kernels that need to be constructed in Edalat's work [Eda99] are based on rcpds.

References

[AD94] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183:235, 1994.

[AHS96] R. Alur, T. Henzinger, and E. Sontag, editors. *Hybrid Systems III*, number 1066 in Lecture Notes in Computer Science. Springer-Verlag, 1996.

[AJKvO97] R. Alur, L. 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*, 1997.

[AKNS97] P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors. *Hybrid Systems IV*, volume 1273 of *Lecture Notes In Computer Science*. Springer-Verlag, 1997.

[AM89] P. Aczel and N. Mendler. A final-coalgebra theorem. In *Category Theory and Computer Science*, Lecture Notes In Computer Science, pages 357–365, 1989.

[Arn94] A. Arnold. *Finite Transition Systems.* Prentice-Hall, 1994.

[Arv76] W. Arveson. *An Invitation to C^*-Algebra.* Springer-Verlag, 1976.

[Ash72] R. B. Ash. *Real Analysis and Probability.* Academic Press, 1972.

[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.

[Bil95] P. Billingsley. *Probability and Measure.* Wiley-Interscience, 1995.

[BK96] C. Baier and M. Kwiatkowska. Domain equations for probabilistic processes. Available from URL http://www.cs.bham.ac.uk/ mzk/, March 1996.

[BLFG95] A. Benveniste, B. C. Levy, E. Fabre, and P. Le Guernic. A calculus of stochastic systems for the specification, simulation and hidden state estimation of mixed stochastic/nonstochastic systems. *Theoretical Computer Science*, 152(2):171–217, 1995.

[Cle90] R. Cleaveland. On automatically explaining bisimulation inequivalence. In E.M. Clarke and R.P. Kurshan, editors, *Computer-Aided Verification CAV 90*, number 531 in Lecture Notes in Computer Science, pages 364–372, 1990.

[CM65] D. R. Cox and H. D. Miller. *The Theory of Stochastic Processes.* Chapman and Hall, 1965.

[CN95] A. Cheng and M. Nielsen. Open maps at work. In P. S. Thiagarajan, editor, *Proceedings of the 15th Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 1026 of *Lecture Notes In Computer Science.* Springer-Verlag, 1995.

[CSZ92] 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.

[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*, pages 478–489. IEEE Press, June 1998.

[Dud89] R. M. Dudley. *Real Analysis and Probability.* Wadsworth and Brookes/Cole, 1989.

[dVR97] 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.

[Eda97] A. Edalat. Semi-pullbacks in categories of markov procosses. Submitted to Mathematical Structures in Computer Science. Available on the WWW from http://theory.doc.ic.ac.uk:80/ ae, 1997.

[Eda99] A. Edalat. Semi-pullbacks and bisimulation in categories of markov processes. *Mathematical Structures in Computer Science*, 1999.

[Gir81] 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.

[GJP99] 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.

[GSS95] V. Gupta, V. Saraswat, and P. Struss. A model of a photocopier paper path. In *Proceedings of the 2nd IJCAI Workshop on Engineering Problems for Qualitative Reasoning*, 1995.

[Hal74] P. Halmos. *Measure Theory.* Number 18 in Graduate Texts in Mathematics. Springer-Verlag, 1974. Originally published in 1950.

[HHWT97] T. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech: a model checker for hybrid systems. *Software Tools for Technology Transfer*, 1(1), 1997.

[Hil94] 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.

[HK96] M. Huth and M. Kwiatkowska. On probabilistic model checking. Technical Report CSR-96-15, University of Birmingham, 1996. Available from http://www.cs.bham.ac.uk/ mzk/.

[Hoa85] C. A. R. Hoare. *Communicating Sequential Processes.* Series in Computer Science. Prentice-Hall International, London, 1985.

[JL91] 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.

[JNW96] A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996.

[JS90] C.-C. Jou and S. A. Smolka. Equivalences, congruences, and complete axiomatizations for probabilistic processes. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR 90 First International Conference on Concurrency Theory*, number 458 in Lecture Notes In Computer Science. Springer-Verlag, 1990.

[JY95] 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.

[Koz81] D. Kozen. Semantics of probabilistic programs. *Journal of Computer and Systems Sciences*, 22:328–350, 1981.

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

[KT66] J. F. C. Kingman and S. J. Taylor. *Introduction to Measure and Probability.* Cambridge University Press, 1966.

[LS91] K. G. Larsen and A. Skou. Bisimulation through probablistic testing. *Information and Computation*, 94:1–28, 1991.

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

[Mil89] R. Milner. *Communication and Concurrency.* Prentice-Hall, 1989.

[Mil90] R. Milner. *Handbook of Theoretical Computer Science: Volume B*, chapter Operational and Algebraic Senmantics of Concurrent Processes, pages 1201–1242. MIT Press, 1990.

[Pan98] P. Panangaden. Probabilistic relations. In C. Baier, M. Huth, M. Kwiatkowska, and M. Ryan, editors, *PROBMIV98*, pages 59–74, 1998.

[Par67] K. R. Parthasarathy. *Probability Measures on Metric Spaces.* Academic Press, 1967.

[Pop94] Sally Popkorn. *First Steps in Modal Logic.* Cambridge University Press, 1994.

[Pro] Probabilistic systems group, collected reports. Available from `www.comlab.ox.ac.uk` in the directory `/oucl/groups/probs/bibliography.html`.

[RdV97] J. J. M. M. Rutten and E. de Vink. Bisimulation for probabilistic transition systems: a coalgebraic approach. In P. Degano, editor, *Proceedings of ICALP 97*, number 1256 in Lecture Notes In Computer Science, pages 460–470. Springer-Verlag, 1997.

[Rud66] W. Rudin. *Real and Complex Analysis.* McGraw-Hill, 1966.

[Rut95] J. J. M. M. Rutten. A calculus of transition systems (towards universal coalgebra). In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra, a bisimulation perspective*, number 53 in CSLI Lecture Notes, 1995. Available electronically from `www.cwi.nl/~janr`.

[Rut96] J. J. M. M. Rutten. Universal coalgebra: a theory of systems. Technical Report CS-R9652, CWI AMsterdam, 1996. Available from URL `www.cwi.nl/~janr/papers/`.

[SD78] N. Saheb-Djahromi. Probabilistic LCF. In *Mathematical Foundations Of Computer Science*, number 64 in Lecture Notes In Computer Science. Springer-Verlag, 1978.

[SD80] N. Saheb-Djahromi. Cpos of measures for nondeterminism. *Theoretical Computer Science*, 12(1):19–37, 1980.

[SL94] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In B. Jonsson and J. Parrow, editors, *Proceedings of CONCUR94*, number 836 in Lecture Notes In Computer Science, pages 481–496. Springer-Verlag, 1994.

[Son90] E. Sontag. *Mathematical Control Theory.* Number 6 in Texts in Applied Mathematics. Springer-Verlag, 1990.

[vGSST90] R. van Glabbeek, S. Smolka, B. Steffen, and C. Tofts. Reactive generative and stratified models for probabilistic processes. In *Proceedings of the 5th Annual IEEE Symposium On Logic In Computer Science*, 1990.

[WN95] G. Winskel and M. Nielsen. *Handbook of Logic in Computer Science*, volume 4, chapter Models for Concurrency. Oxford University Press, 1995.