Trace-based Semantics for Probabilistic Timed I/O Automata*

Sayan Mitra and Nancy Lynch

MIT Computer Science and AI Laboratory
{lynch, mitras}@csail.mit.edu

Abstract. We describe the main features of the Probabilistic Timed I/O Automata (PTIOA)—a framework for modeling and analyzing discretely communicating probabilistic hybrid systems. A PTIOA can choose the post-state of a discrete transition either nondeterministically or according to (possibly continuous) probability distributions. The framework supports modeling of large systems as compositions of concurrently executing PTIOAs, which interact through shared transition labels. We develop a trace-based semantics for PTIOAs and show that PTIOAs are compositional with respect a new notion of external behavior.

1. Introduction. Probabilistic automata with continuous state spaces provide a framework for studying computing systems that interact with unpredictable environments. In distributed systems, nondeterminism enables us to describe arbitrary interleaving of concurrently executing processes. For modeling and analyzing systems which have traits of both hybrid and distributed systems, such as sensor networks and mobile-robotic systems, we need frameworks that support continuous dynamics, probabilistic transitions, and nondeterminism. There are continuous state automaton frameworks which eschew *internal nondeterminism* in favor of fully probabilistic evolution such as [1,5,11], and those that support both nondeterministic and probabilistic transitions but restrict the state spaces and the probability distributions to be discrete [6,4,2,7]. In this note we describe the main features of the *Probabilistic Timed Input/Output Automaton (PTIOA)* framework (full version available as [9]) which generalizes both the *Timed I/O Automaton* [8] and the *Probabilistic I/O Automaton* [2] frameworks, and provides a basis for describing concurrent, continuous state-space systems, with both probabilistic and nondeterministic transitions.

Definition 1. *A* PTIOA *is a 6-tuple* $\mathcal{A} = ((X, \mathcal{F}_X), \bar{x}, A, \mathcal{R}, \mathcal{D}, \mathcal{T})$ *where: (1)* $(X, \mathcal{F}_X)$ *is a measurable space called the* state space*. (2)* $\bar{x} \in X$ *is the* start state*. (3)* A *is a countable set of* actions*, partitioned into* internal H*,* input I *and* output O *actions.* $L = O \cup H$ *is the set of* local actions *and* $E = O \cup I$ *is the set of* external actions*.* $\mathcal{A}$ *is said to be* closed *if* $I = \emptyset$*. (4)* $\mathcal{R}$ *is an equivalence relation on* L*; the equivalence classes of* $\mathcal{R}$ *are called* tasks*. A task* T *is called an* output task *if* $T \subseteq O$*. (5)* $\mathcal{D} \subseteq X \times A \times \mathcal{P}(X, \mathcal{F}_X)$ *is the set of* probabilistic transitions*.* If (x, a, μ) is an element of $\mathcal{D}$, we write $x \xrightarrow{a} \mu$ and action a is said to be *enabled* at x*. (6)* $\mathcal{T}$ *is a set of deterministic trajectories for* X*.* In addition, a PTIOA satisfies the following axioms:

* This work was supported by NSF's CSR program (Embedded and Hybrid Systems area) under grant NSF CNS-0614993.

M0 For all $B \subseteq A$, set of states in which at least one action from B is enabled is measurable. For measurable sets $R \subseteq \mathbb{R}_{\geq 0}$ and $Y \subseteq X$, the set of states from which some $r \in R$ amount of time can elapse and a state $y \in Y$ is reached (according to some trajectory in $\mathcal{T}$), is measurable.

D0 Input actions are enabled in all states.

D1 For any state x at most one of the following may exist: (1) a local action enabled at x (2) a non-point trajectory starting from x.

D2 For any state x, if there are actions a, b in the same task T and $x \xrightarrow{a} \mu_1$ and $x \xrightarrow{b} \mu_2$ then $a = b$ and $\mu_1 = \mu_2$.

D3 An execution of finite duration has at most finite number of internal actions.

The **M0** axiom ensures measurability of reasonable sets of executions. **D0** is a non-blocking axiom standard in I/O automata literature. Axiom **D1** allows resolution of nondeterminism in a structured manner; this axiom will be removed in Section **4. D1** prevents an action to remain enabled while time elapses. If time can elapse from x, then the state evolves according to the longest trajectory starting from x. If local actions are enabled at x then time cannot elapse and $\mathcal{A}$ nondeterministically chooses one action a from the set of enabled actions. This nondeterministic choice is resolved by a *task scheduler* (defined below). If a task T is specified then **D2** implies that at x there can be at most one enabled action in T, and at most one probabilistic transition corresponding to that action.

2. Distributions over executions and traces. An *execution fragment* of an PTIOA $\mathcal{A}$ is a sequence $\alpha = \tau_0 a_1 \tau_1 a_2 \ldots$, where each $\tau_i \in \mathcal{T}$, $a_i \in A$ and a_i is enabled at the last state of τ_{i-1}. An execution fragment α is an *execution* of $\mathcal{A}$ if the first trajectory starts from $\bar{x}$. The *trace* of an execution α represents its externally visible part, namely the external actions and time passage. It is obtained by removing internal actions, concatenating consecutive trajectories, and replacing all the trajectories with their lengths. We denote the set of executions, and the set of traces of $\mathcal{A}$ by $\mathsf{Execs}_{\mathcal{A}}$ and $\mathsf{Traces}_{\mathcal{A}}$.

In order to construct a probability measure over $\mathsf{Execs}_{\mathcal{A}}$ we have to first define a σ-algebra over Execs_A. Adapting a construction given in [3], we proceed as follows: A *base* is a finite sequence $\Lambda = X_0 R_0 X_1 A_1 R_1 \ldots X_m A_m R_m X_{m+1}$, where for every $i \in \{0, \ldots, m+1\}$, $X_i \in \mathcal{F}_X$, R_i is a measurable set in $\mathbb{R}_{\geq 0}$ and for every $i \in \{1, \ldots, m\}$, $A_i \subseteq A$. The *basic set* C_Λ corresponding to this base Λ is the set of all executions which have a prefix that matches the pattern of actions and trajectories in Λ. We show that collection $\mathscr{C}$ of all basic sets of $\mathcal{A}$ generates a σ-algebra over $\mathsf{Execs}_{\mathcal{A}}$, which we denote by $\mathcal{F}_{\mathsf{Execs}_{\mathcal{A}}}$. We define the measurable space of executions of $\mathcal{A}$ to be $(\mathsf{Execs}_{\mathcal{A}}, \mathcal{F}_{\mathsf{Execs}_{\mathcal{A}}})$. In a similar manner, we construct a measurable space $(\mathsf{Traces}_{\mathcal{A}}, \mathcal{F}_{\mathsf{Traces}_{\mathcal{A}}})$ of traces of $\mathcal{A}$ from *trace bases*, which are defined to be finite alternating sequences of the measurable sets in $\mathbb{R}_{\geq 0}$ and subsets of E.

We combine $\mathcal{A}$ with a *task scheduler* for resolving nondeterministic choice over enabled actions, which is is simply a finite or infinite sequence $\rho = T_1 T_2 \ldots$ of tasks in $\mathcal{R}$. A task scheduler [2] chooses the next action deterministically and independently of the information produced during an execution. In [9] we inductively define a function apply such that given any task schedule ρ for $\mathcal{A}$, $\mathsf{apply}(\delta_{\bar{x}}, \rho)$ gives a probability measure over $(\mathsf{Execs}_{\mathcal{A}}, \mathcal{F}_{\mathsf{Execs}_{\mathcal{A}}})$ by "applying" ρ

to $\mathcal{A}$, one task at a time. This probability distribution over executions is called a *probabilistic execution* of $\mathcal{A}$.

For any probabilistic execution μ of a PTIOA we would like to have a single corresponding measure on $\mathsf{Traces}_{\mathcal{A}}$. This requires $trace : (\mathsf{Execs}_{\mathcal{A}}, \mathcal{F}_{\mathsf{Execs}_{\mathcal{A}}}) \rightarrow (\mathsf{Traces}_{\mathcal{A}}, \mathcal{F}_{\mathsf{Traces}_{\mathcal{A}}})$ to be a measurable function. A difficulty in proving this arises because the *trace* function concatenates trajectories that are separated by internal actions. Consider, for example, a simple trace base $[0, r]\{a\}$, where r is a positive real and a is an external action. Then, $\mathcal{E} = trace^{-1}(C_{[0,r]\{a\}})$ is the set of all finite executions of the form $\tau_1 h_1 \tau_2 h_2 \ldots h_{n-1}\tau_n a$, such that all the h_i's are internal actions and $\sum_{i=1}^{n} \tau_i.ltime \leq r$. For *trace* to be a measurable function $\mathcal{E}$ expressible as a countable union of basic sets. We prove the measurability of *trace* in [9] making use of several PTIOA properties including the **D3** axiom. With this result, we are able to prove a key theorem which asserts that each task schedule for a PTIOA $\mathcal{A}$ gives rise to a single distribution over the traces of $\mathcal{A}$. For each task schedule the corresponding distribution over traces is called a *trace distribution* and the set of all possible trace distributions of $\mathcal{A}$ is denoted by $tdists(\mathcal{A})$.

3. Composition and External Behavior. The composition operation allows a PTIOA representing a complex system to be constructed by composing PTIOAs representing smaller subsystems. In [9] we state the conditions under which two PTIOAs $\mathcal{A}_1$ and $\mathcal{A}_2$ can be composed, we define the *parallel composition* operator, and we prove a theorem asserting that a composition $\mathcal{A}_1||\mathcal{A}_2$ is a valid PTIOA.

An *environment* for PTIOA $\mathcal{A}$ is a PTIOA $\mathcal{E}$ such that $\mathcal{A}$ and $\mathcal{E}$ can be composed and their composition $\mathcal{A}||\mathcal{E}$ is closed. The *external behavior* of a PTIOA $\mathcal{A}$, written as $extbeh_{\mathcal{A}}$, is defined as a function that maps each environment $\mathcal{E}$ for $\mathcal{A}$ to the set $tdists(\mathcal{A}||\mathcal{E})$. Two PTIOAs $\mathcal{A}_1$ and $\mathcal{A}_2$ are *comparable* if $E_1 = E_2$. If $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable then $\mathcal{A}_1$ is said to *implement* $\mathcal{A}_2$, written as $A_1 \leq A_2$ if, for every environment PTIOA $\mathcal{E}$ for both $\mathcal{A}_1$ and $\mathcal{A}_2$, $extbeh_{\mathcal{A}_1}(\mathcal{E}) \subseteq extbeh_{\mathcal{A}_2}(\mathcal{E})$. Viewing external behavior as a mapping from environments as opposed to a set of trace distributions is natural in many applications, including analysis of security protocols [2], and indeed, it lets us circumvent some of the difficulties that underlie compositionality in the probabilistic setting.

Theorem 1. *Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable PTIOAs and $\mathcal{A}_1 \leq \mathcal{A}_2$. If PTIOA $\mathcal{B}$ can be composed with both $\mathcal{A}_1$ and $\mathcal{A}_2$ then $\mathcal{A}_1||\mathcal{B} \leq \mathcal{A}_2||\mathcal{B}$.*

4. Generalized PTIOAs. In this section, we relax the deterministic assumptions (axiom **D1**) on PTIOAs. A *Generalized PTIOA* is a tuple $\mathcal{A} = ((X, \mathcal{F}_X), \bar{x}, A, \mathcal{R}, \mathcal{D}, \mathcal{T})$ as in Definition 1, but $\mathcal{A}$ does not necessarily satisfy **D1** and $\mathcal{T}$ is not necessarily deterministic. Thus, from a given state $x \in X$ of a generalized PTIOA $\mathcal{A}$ there may be nondeterministic choice of actions and also choice of distinct trajectories. A *local scheduler* for generalized PTIOA $\mathcal{A}$, is a PTIOA $S = ((X, \mathcal{F}_X), \bar{x}, A, \mathcal{R}, \mathcal{D}', \mathcal{T}')$ that is identical to $\mathcal{A}$ except that $\mathcal{D}' \subseteq D$ and $\mathcal{T}' \subseteq \mathcal{T}$. A *local scheduler* S satisfies **D1** and has deterministic trajectories.

A *probabilistic-system* is a pair $\mathcal{M} = (\mathcal{A}, \mathcal{S})$, where $\mathcal{A}$ is a generalized PTIOA and $\mathcal{S}$ is a set of local schedulers for $\mathcal{A}$. An *environment* for $\mathcal{M}$ is any PTIOA $\mathcal{E}$ such that $\mathcal{A}||\mathcal{E}$ is closed. A probabilistic execution for $\mathcal{M}$ is defined to be any

probabilistic execution of S, for any $S \in \mathcal{S}$. The notion of trace distribution carries over naturally to generalized PTIOAs. For probabilistic system $\mathcal{M} = (\mathcal{A}, \mathcal{S})$, we define the *external behavior* of $\mathcal{M}$ to be the total function $extbeh_{\mathcal{M}}$ that maps each environment PTIOA $\mathcal{E}$ for $\mathcal{M}$ to the set $\cup_{S' \in \mathcal{S}} tdists(S'||\mathcal{E})$. Thus for each environment, we consider the set of trace distributions that arise from the choices of the local scheduler of $\mathcal{M}$ and the task scheduler ρ. This leads to a notion of implementation of probabilistic systems, similar to that of PTIOAs. Let $\mathcal{M}_1 = (\mathcal{A}_1, \mathcal{S}_1)$ and $\mathcal{M}_2 = (\mathcal{A}_2, \mathcal{S}_2)$ be probabilistic systems such that $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable generalized PTIOAs. Then, $\mathcal{M}_1$ *implements* $\mathcal{M}_2$ if for every environment $\mathcal{E}$ of $\mathcal{M}_1$ and $\mathcal{M}_2$, $extbeh_{\mathcal{M}_1}(\mathcal{E}) \subseteq extbeh_{\mathcal{M}_2}(\mathcal{E})$. Theorem 2 gives a sufficient condition for implementation of probabilistic systems:

Theorem 2. *If $\mathcal{M}_1 = (\mathcal{A}_1, \mathcal{S}_1)$, $\mathcal{M}_2 = (\mathcal{A}_2, \mathcal{S}_2)$ are comparable and there exists $f : \mathcal{S}_1 \rightarrow \mathcal{S}_2$, such that for all $S_1 \in \mathcal{S}_1$, S_1 implements $f(S_1)$, then $\mathcal{M}_1$ implements M_2.*

In the future we would like to extend PTIOAs to support shared variables and develop a suite of analysis techniques for proving probabilistic safety, stability and approximate implementation relations [10].

We thank Sanjoy Mitter for many valuable comments on this work.

References

1. M. Bujorianu and J. Lygeros. General stochastic hybrid systems: Modelling and optimal control. In *IEEE Conference on Decision and Control*, December 2004.
2. R. Canetti, L. Cheung, D. Kaynar, M. Liskov, N. Lynch, O. Pereira, and R. Segala. Task-structured probabilistic I/O automata. Tech. Report MIT-CSAIL-TR-2006-060, MIT, Cambridge, 2006.
3. S. Cattani, R. Segala, M. Z. Kwiatkowska, and G. Norman. Stochastic transition systems for continuous state spaces and non-determinism. In *FoSSaCS'05*, LNCS 3441, pages 125–139, 2005.
4. L. Cheung. *Reconciling nondeterministic and probabilistic choices.* PhD thesis, ICIS, Radboud University Nijmegen, The Netherlands, 2006.
5. V. Danos, J. Desharnais, F. Laviolette, and P. Panangaden. Bisimulation and cocongruence for probabilistic systems. *Information and Computation, Special issue for selected papers from CMCS04*, 2005.
6. L. de Alfaro. *Formal Verification of Probabilistic Systems.* PhD thesis, Stanford University, CA, 1997. Technical Report STAN-CS-TR-98-1601.
7. S. Smolka E. Stark, R. Cleaveland. A process-algebraic language for probabilistic I/O automata. In *Proc. CONCUR 03*, *LNCS* 2761:189–203, 2003.

H200. H. Hermanns. *Interactive Markov Chains : The Quest for Quantified Quality.* Springer, 2002.

8. D. Kaynar, N. Lynch, R. Segala, and F. Vaandrager. *The Theory of Timed I/O Automata.* Synthesis Lectures on Computer Science. Morgan Claypool, 2005.
9. S. Mitra and N. Lynch. Trace-based Semantics for Probabilistic Timed I/O Automata Full version `http://theory.lcs.mit.edu/~mitras/research/PTIOA06-full.pdf`
10. S. Mitra and N. Lynch Approximate simulations for task-structured probabilistic I/O automata. In LICS Workshop on Probabilistic Automata and Logics, 2006.
11. F. van Breugel, M. W. Mislove, J. Ouaknine, and J. Worrell. Domain theory, testing and simulation for labelled markov processes. *Theoretical Computer Science*, 2005.