Proving Approximate Implementations for Probabilistic I/O Automata[??]

Sayan Mitra[2] Nancy Lynch[3]

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, USA

Abstract

In this paper we introduce the notion of approximate implementations for Probabilistic I/O Automata (PIOA) and develop methods for proving such relationships. We employ a *task structure* on the locally controlled actions and a *task scheduler* to resolve nondeterminism. The interaction between a scheduler and an automaton gives rise to a *trace distribution*—a probability distribution over the set of traces. We define a PIOA to be a (discounted) *approximate implementation* of another PIOA if the set of trace distributions produced by the first is close to that of the latter, where closeness is measured by the (resp. discounted) uniform metric over trace distributions. We propose simulation functions for proving approximate implementations corresponding to each of the above types of approximate implementation relations. Since our notion of similarity of traces is based on a metric on trace distributions, we do not require the state spaces nor the space of external actions of the automata to be metric spaces. We discuss applications of approximate implementations to verification of probabilistic safety and termination.

Keywords: Approximate implementation, equivalence, Approximate simulation, Abstraction, Probabilistic I/O Automata.

1 Introduction

Implementation relations play a fundamental role in the study of complex interacting systems because they allow us to prove that a given concrete system implements an abstract specification. Formally, an automaton is said to implement another automaton if the set of traces or the observable behavior of the first is subsumed by that of the latter. Many different kinds of implementation or abstraction relations and their corresponding proof methods have been developed for timed [1], hybrid [17,30,29] and probabilistic automata [19,20,5,2,28,4].

These traditional notions of implementation rely on equality of traces. That is, every trace of the concrete system must be exactly equal to some trace of the abstract specification. It is well known from [16,10,15] that such strict equality based implementation relations are not robust. Small perturbations to the parameters of the system produces traces with slightly different numbers (representing say, timing

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

[2] Email: mitras@theory.csail.mit.edu

[3] Email: lynch@theory.csail.mit.edu

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

or probability information), and thus breaks the equality between traces. One way to overcome this problem is to relax the notion of implementation by taking into consideration the "similarity" of traces that are not exactly equal. In [16] Jou and Smolka formalized "similarity" of traces using a metric and developed the corresponding notion of approximate equivalence for probabilistic automata. Based on similar ideas, there is now a growing body of work on developing robust notions of approximate implementations; in Section 1.1, we briefly describe previous contributions in this area that are related to our work. Apart from providing robust implementation relations, notions of approximate implementation also enable us to create abstract models without introducing extra nondeterminism.

In this paper we introduce the notion of approximate implementations for the *Probabilistic Input/Output Automaton (PIOA)* [27,6] and develop simulation based methods for proving such relationships. A PIOA is a nondeterministic automaton with a countable state space. Transitions are labelled by *actions*. Many transitions may be possible from a given state. Each transition gives a discrete probability distribution over the state space. We use a *task structure* [5]—an equivalence relation on the set of locally controlled actions—as a means for restricting the nondeterminism in a PIOA. The resulting automaton model is called *task-PIOA*. A task-PIOA interacts with a *task scheduler* to give rise to a probability distribution over its executions. For every such distribution there exists a corresponding distribution over its set of traces, which is called a *trace distribution*. Visible behavior of a task-PIOA is the set of *trace distributions* that it can produce. A task-PIOA is said to (exactly) implement another task-PIOA if the set of trace distributions of the first is a subset of the trace distributions of the latter. Implementations, simulation relations for proving implementations, and compositionality results for task-PIOAs are presented in [5]. A special kind of approximate implementation relation that tolerates small differences in the probability of occurrence of a particular action is used in [6] to verify a security protocol. In contrast, the notions of approximation introduced here are more general because they are based on metrics on trace distributions. We define two kinds of approximate implementations of task-PIOAs: (1) *uniform approximate implementation* is based on the uniform metric of trace distributions [23], and (2) *discounted approximate implementation* is based on the discounted uniform metric.

A PIOA $\mathcal{A}$ is a δ-approximate implementation of another PIOA $\mathcal{B}$, for a positive δ, if the for any trace distribution of $\mathcal{A}$, there exists a trace distribution of $\mathcal{B}$ such that their discrepancy over any measurable set of traces is at most δ. We present *Expanded Approximate Simulations (EAS)* for proving uniform approximate implementations. EAS is a natural generalization of the simulation relation presented in [6]. Let μ_1 and μ_2 be probability distributions over executions of task-PIOAs $\mathcal{A}$ and $\mathcal{B}$. An EAS from $\mathcal{A}$ to $\mathcal{B}$ is a function ϕ mapping each μ_1, μ_2 pair to a non-negative real. The number $\phi(\mu_1, \mu_2)$, is a measure of how similar μ_1 and μ_2 are in terms of producing similar trace distributions. Informally, if $\phi(\mu_1, \mu_2) \leq \epsilon$, for some $\epsilon \geq 0$, then it is possible to closely (with respect to the uniform metric on trace distributions) simulate from μ_2 anything that can happen from μ_1, and further, the resulting distributions, say μ_1' and μ_2', are also close in the following sense. There exists a joint distribution ψ supported on the set $\{(\eta_1, \eta_2) \mid \phi(\eta_1, \eta_2) \leq \epsilon\}$ such that

the marginals of ψ have means μ_1' and μ_2', respectively. Informally, this means that μ_1' and μ_2' can be decomposed into a set of measures that are close in the sense of ϕ.

Uniform approximate implementations are useful for deducing probabilistic safety properties. However, since they gives absolute bounds on the discrepancy over any set of traces, they do not give us useful information when the probability of the set itself is smaller than the approximation factor δ. To get useful bounds on the discrepancies over a sequence of sets of traces that have monotonically decreasing probabilities, we have to employ different approximation factors for each set. We address this problem by introducing a sequence $\{\delta_k\}_{k\in\mathbb{N}}$ of *discount factors*, and defining PTIOA $\mathcal{A}$ to be a δ_k-discounted approximate implementation of $\mathcal{B}$, if the for any trace distribution of $\mathcal{A}$, there exists a trace distribution of $\mathcal{B}$ such that their discrepancy *over any trace of length k* is at most δ_k. We define *Discounted Approximate Simulations (DAS)* in a similar way as we defined EAS and prove that they are sound for proving discounted approximate implementations. We demonstrate the utility of discounted approximate implementations and DASs by proving that the probability of termination of an ideal randomized consensus protocol (after a certain number of rounds) is close to the same probability for a protocol that uses biased coins.

1.1 Related Work

As we mentioned, Jou and Smolka [16] first introduced the idea of formalizing similarity of traces by using metrics. Approximation metrics for probabilistic systems in the context of *Labelled Markov Processes (LMP)* have been extensively investigated and many fundamental results have been obtained by Desharnais, Gupta, Jagadeesan and Panangaden [10,8,9] and by van Breugel, Mislove, Ouaknine, and Worrell [35,32,33,21,22]. The first set of authors introduced a Kantorovich-like metric for LMPs and presented the logical characterization of this metric. Van Breugel *et al.* have presented intrinsic characterizations of the topological space induced by the above metric. This characterization is based on a final coalgebra for a functor on the category of metric spaces and nonexpansive maps. Another interesting facet of this body of work is the polynomial time algorithm for computing the metric presented by van Breugel and Worrell in [34]. For *Generalized Semi-Markov Processes (GSMP)* [15], Gupta, Jagadeesan and Panangaden have developed pseudo-metric analogues of bisimulation and have shown that certain observable quantitative properties are continuous with respect to the introduced metric. Kwiatkowska and Norman have developed the denotational semantics for a divergence-free probabilistic process algebra based on a metric on probability distribution over executions [18].

In the non-probabilistic setting, Girard and Pappas [13,12] have developed the theory of approximate implementations for *Metric Transition Systems (MTS)*. The state space and the space of external actions of an MTS are metric spaces. Based on these metrics, the authors develop a hierarchy of approximation pseudo-metrics between MTSs measuring distance between reachable sets, sets of traces and bisimulations. The authors have also developed algorithms for exactly and approximately computing these metrics.

Our work differs from all of the above in at least one of the following ways: (a) the task-PIOA model allows both nondeterministic and probabilistic choices, and

(b) the implementation relation in our framework is based on trace distributions and not bisimilarity of states. Approximate implementation is derived from a metric over trace distributions, and thus, we do not require the state spaces of the underlying automata nor the common space of external actions to be metric spaces. Metrics on trace distributions of PIOAs are used by Cheung in [7]to show that sets of trace distributions form closed sets in a certain metric space. This result is then used to show that finite tests are sufficient to distinguish between a members of a certain class of PIOAs. The metric used in the above work is related to our uniform metric but it is defined on the set $[0,1]^{Traces}$ whereas our uniform metric is exclusively defined on the set of trace distributions.

1.2 Organization

In the next Section, we give the basic definitions and results from the task-PIOA framework. We refer the reader to [14] for a detailed treatment and for all the proofs. In Section 3 we introduce uniform approximate implementations for closed task-PIOAs and we propose expanded approximate simulations as a sound method for proving uniform implementations. In Section 4, we discuss the need for discounting when measuring discrepancies in trace distributions. This leads to the notion of discounted approximate implementations and we propose a second type of simulations for proving such implementation relationships. Finally, in Section 5 we outline how our results extend to general (not necessarily closed) task-PIOAs and conclude with a discussion on future research directions. Proofs of auxiliary lemmas and formal statements of some relevant results from [5] appear in the Appendices.

2 Task-PIOA Framework

Given a set X, we denote a σ-algebra over X by $\mathcal{F}_X$, the set of discrete (sub-) probability measures on X by $\mathsf{Disc}(X)$ (resp. $\mathsf{SubDisc}(X)$). If μ is a discrete probability or sub-probability measure on X, the *support* of μ, written as $\mathsf{supp}(\mu)$, is the set of elements of X that have non-zero measure. The task-PIOA model used in this paper is slightly more general than the one in [5] because we allow the starting configuration of an automaton to be any distribution over states and not just a Dirac mass.

Definition 2.1. *A* task-structured Probabilistic I/O Automaton $\mathcal{A}$ *is a 7-tuple* $(Q, \bar{\nu}, I, O, H, D, \mathcal{R})$ *where:*

(i) Q *is a countable set of* states*;*

(ii) $\bar{\nu} \in \mathsf{Disc}(Q)$ *is the* starting distribution *on states;*

(iii) I*,* O *and* H *are countable and pairwise disjoint sets of actions, referred to as* input, output and internal actions*, respectively. The set* $A := I \cup O \cup H$ *is called the set of* actions *of* $\mathcal{A}$*. If* $I = \emptyset$*, then* $\mathcal{A}$ *is* closed*. The set of* external *actions of* $\mathcal{A}$ *is* $E := I \cup O$ *and the set of* locally controlled *actions is* $L := O \cup H$*.*

(iv) $D \subseteq (Q \times A \times \mathsf{Disc}(Q))$ *is a* transition relation*. An action* a *is* enabled *in a state* q *if* $(q, a, \mu) \in D$ *for some* μ*.*

(v) $\mathcal{R}$ *is an equivalence relation on the locally controlled actions. The equivalence*

classes of $\mathcal{R}$ are called tasks. *A task T is* enabled *in a state q if some action $a \in T$ is enabled in q.*

In addition, $\mathcal{A}$ satisfies:

- Input enabling: *For every $q \in Q$ and $a \in I$, a is enabled in q.*
- Transition determinism: *For every $q \in Q$ and $a \in A$, there is at most one $\mu \in \mathsf{Disc}(Q)$ such that $(q, a, \mu) \in D$.*
- Action determinism: *For every $q \in Q$ and $T \in R$, at most one $a \in T$ is enabled in q.*

An *execution fragment* of $\mathcal{A}$ is a finite or infinite sequence $\alpha = q_0\, a_1\, q_1\, a_2\, \ldots$ of alternating states and actions, such that (i) if α is finite, then it ends with a state; and (ii) for every non-final i, there is a transition $(q_i, a_{i+1}, \mu) \in D$ with $q_{i+1} \in \mathsf{supp}(\mu)$. We write $\alpha.fstate$ for q_0, and, if α is finite, we write $\alpha.lstate$ for its last state. We use $\mathsf{Frags}_{\mathcal{A}}$ (resp., $\mathsf{Frags}^*_{\mathcal{A}}$) to denote the set of all (resp., all finite) execution fragments of $\mathcal{A}$. An *execution* of $\mathcal{A}$ is an execution fragment beginning from some state in $\mathsf{supp}(\bar{\nu})$. $\mathsf{Execs}_{\mathcal{A}}$ (resp., $\mathsf{Execs}^*_{\mathcal{A}}$) denotes the set of all (resp., finite) executions of $\mathcal{A}$. The *trace* of an execution fragment α, written $\mathsf{trace}(\alpha)$, is the restriction of α to the set of external actions of $\mathcal{A}$. We say that β is a *trace* of $\mathcal{A}$ if there is an execution α of $\mathcal{A}$ with $\mathsf{trace}(\alpha) = \beta$. $\mathsf{Traces}_{\mathcal{A}}$ (resp., $\mathsf{Traces}^*_{\mathcal{A}}$) denotes the set of all (resp., finite) traces of $\mathcal{A}$.

Nondeterministic choices in $\mathcal{A}$ are resolved using a *scheduler*, which is a function $\sigma : \mathsf{Frags}^*_{\mathcal{A}} \longrightarrow \mathsf{SubDisc}(D)$ such that $(q, a, \mu) \in \mathsf{supp}(\sigma(\alpha))$ implies $q = \alpha.lstate$. Thus, σ decides (probabilistically) which transition (if any) to take after each finite execution fragment α. Since this decision is a discrete sub-probability measure, it may be the case that σ chooses to *halt* after α with non-zero probability: $1 - \sigma(\alpha)(D) > 0$. A scheduler σ and a finite execution fragment α *generate* a measure $\mu_{\sigma,\alpha}$ on the σ-field $\mathcal{F}_{\mathsf{Execs}_{\mathcal{A}}}$ generated by cones of execution fragments, where each cone $C_{\alpha'}$ is the set of execution fragments that have α' as a prefix. The theory of probabilistic executions of task-PIOAs with a general class of history dependent schedulers has been developed in [5].

In this paper we restrict our attention to *static* (or *oblivious*), schedulers that do not depend on dynamic information generated during execution. Although restrictive this class of schedulers arise naturally in many applications, including in analysis of security protocols [6]. A *task schedule* for $\mathcal{A}$ is any finite or infinite sequence $\sigma = T_1 T_2 \ldots$ of tasks in R. A task schedule can be used to generate a unique probabilistic execution of the task-PIOA $\mathcal{A}$. One can do this by repeatedly scheduling tasks, each of which determines at most one transition of $\mathcal{A}$. Formally, we define an operation that "applies" a task schedule to a task-PIOA:

Definition 2.2. *Let $\mathcal{A}$ be an action-deterministic task-PIOA. Given $\mu \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}})$ and a task schedule σ, $\mathsf{apply}(\mu, \sigma)$ is the probability measure on $\mathsf{Frags}_{\mathcal{A}}$ defined recursively by:*

(i) $\mathsf{apply}(\mu, \lambda) := \mu$. *($\lambda$ denotes the empty sequence.)*

(ii) *For $T \in R$, $\mathsf{apply}(\mu, T)$ is defined as follows. For every $\alpha \in \mathsf{Frags}^*_{\mathcal{A}}$, $\mathsf{apply}(\mu, T)(\alpha) := p_1 + p_2$, where:*
- *$p_1 = \mu(\alpha')\eta(q)$ if α is of the form $\alpha' a q$, where $a \in T$ and $(\alpha'.lstate, a, \eta) \in D$;*

$p_1 = 0$ *otherwise.*

- $p_2 = \mu(\alpha)$ *if* T *is not enabled in* $\alpha.lstate$*;* $p_2 = 0$ *otherwise.*

(iii) *For* σ *of the form* $\sigma' T$, $T \in R$, $\mathsf{apply}(\mu, \sigma) := \mathsf{apply}(\mathsf{apply}(\mu, \sigma'), T)$.

(iv) *For* σ *infinite,* $\mathsf{apply}(\mu, \sigma) := \lim_{i \to \infty}(\mathsf{apply}(\mu, \sigma_i))$*, where* σ_i *denotes the length-*i *prefix of* σ.

In Case (ii) above, p_1 represents the probability that α is executed when applying task T at the end of α'. Because of transition-determinism and action-determinism, the transition $(\alpha'.lstate, a, \eta)$ is unique, and so p_1 is well-defined. The term p_2 represents the original probability $\mu(\alpha)$, which is relevant if T is not enabled after α. It is routine to check that the limit in Case (iv) is well-defined. The other two cases are straightforward. Given any task schedule σ, $\mathsf{apply}(\bar{\nu}, \sigma)$ is a probability distribution over $\mathsf{Exec}_{\mathcal{A}}$. Several useful properties of the $\mathsf{apply}(,)$ function relating sequences of probability distributions on executions and traces are given in Appendix A.

We note that the trace function is a measurable function from $\mathcal{F}_{\mathsf{Execs}_{\mathcal{A}}}$ to the σ-field generated by cones of traces. Thus, given a probability measure μ on $\mathcal{F}_{\mathsf{Execs}_{\mathcal{A}}}$ we define the *trace distribution* of μ, denoted $\mathsf{tdist}(\mu)$, to be the image measure of μ under the trace function. We extend the $\mathsf{tdist}()$ notation to arbitrary measures on execution fragments of $\mathcal{A}$. We write $\mathsf{tdist}(\mu, \sigma)$ as shorthand for $\mathsf{tdist}(\mathsf{apply}(\mu, \sigma))$, the trace distribution obtained by applying task schedule σ starting from the measure μ on execution fragments. We write $\mathsf{tdist}(\sigma)$ for $\mathsf{tdist}(\mathsf{apply}(\bar{\nu}, \sigma))$. A *trace distribution* of $\mathcal{A}$ is any $\mathsf{tdist}(\sigma)$. We use $\mathsf{tdists}(\mathcal{A})$ to denote the set $\{\mathsf{tdist}(\sigma) : \sigma$ is a task schedule for $\mathcal{A}\}$ of all trace distributions of $\mathcal{A}$.

Composition of a pair of PIOAs is defined as follows:

Definition 2.3. *Two PIOAs* $\mathcal{A}_i = (Q_i, \bar{\nu}_i, I_i, O_i, H_i, D_i)$, $i \in \{1, 2\}$*, are said to be* compatible *if* $A_i \cap H_j = O_i \cap O_j = \emptyset$ *whenever* $i \neq j$*. In that case, we define their* composition $\mathcal{A}_1 || \mathcal{A}_2$ *to be the PIOA* $(Q_1 \times Q_2,\ (\bar{\nu}_1, \bar{\nu}_2),\ (I_1 \cup I_2) \setminus (O_1 \cup O_2),\ O_1 \cup O_2,\ H_1 \cup H_2, D)$*, where* D *is the set of triples* $((q_1, q_2), a, \mu_1 \times \mu_2)$ *such that*

(i) a *is enabled in some* q_i*, and*

(ii) *for every* i*, if* $a \in A_i$ *then* $(q_i, a, \mu_i) \in D_i$*, otherwise* $\mu_i = \delta_{q_i}$.

2.1 Exact implementations and Simulations

Two task-PIOAs $\mathcal{A}_1$ and $\mathcal{A}_2$ are *comparable* if they have the same set of external actions. Given comparable closed task-PIOAs $\mathcal{A}_1$ and $\mathcal{A}_2$, $\mathcal{A}_1$ is said to *implement* $\mathcal{A}_2$ if $\mathsf{tdists}(\mathcal{A}_1) \subseteq \mathsf{tdists}(\mathcal{A}_2)$. If $\mathcal{A}_1$ and $\mathcal{A}_2$ implement each other then they are said to be *equivalent*. In [5] a simulation relation for closed, task-PIOAs is defined and it is shown to be sound for proving the above implementation relation. This definition is based on three operations involving probability measures: flattening, lifting, and expansion.

Let X and Y be a sets. If $\eta \in \mathsf{Disc}(\mathsf{Disc}(X))$, then the *flattening* of η, denoted by $\mathsf{flatten}(\eta) \in \mathsf{Disc}(X)$, is defined by $\mathsf{flatten}(\eta) = \sum_{\mu \in \mathsf{Disc}(X)} \eta(\mu)\mu$. The *lifting* operation takes a relation $R \subseteq X \times Y$ and "lifts" it to a relation $\mathcal{L}(R) \subseteq \mathsf{Disc}(X) \times \mathsf{Disc}(Y)$ defined by: $\mu_1\ \mathcal{L}(R)\ \mu_2$ iff there exists a *weighting function* $w : X \times Y \to \mathbb{R}_{\geq 0}$ such that: (i) for each $x \in X$ and $y \in Y$, $w(x, y) > 0$ implies $x\ R\ y$, (ii) for each $x \in X$, $\sum_y w(x, y) = \mu_1(x)$, and (ii) for each $y \in Y$, $\sum_x w(x, y) = \mu_2(y)$. Finally,

the *expansion* operation takes a $R \subseteq \mathsf{Disc}(X) \times \mathsf{Disc}(Y)$, and returns a relation $\mathcal{E}(R) \subseteq \mathsf{Disc}(X) \times \mathsf{Disc}(Y)$ such that $\mu_1\ \mathcal{E}(R)\ \mu_2$ whenever they can be decomposed into two $\mathcal{L}(R)$-related measures. Formally, $\mathcal{E}(R)$, is defined by: $\mu_1\ \mathcal{E}(R)\ \mu_2$ iff there exist two discrete measures η_1 and η_2 on $\mathsf{Disc}(X)$ and $\mathsf{Disc}(Y)$, respectively, such that $\mu_1 = \mathsf{flatten}(\eta_1)$, $\mu_2 = \mathsf{flatten}(\eta_2)$, and $\eta_1\ \mathcal{L}(R)\ \eta_2$.

The next definition expresses consistency between a probability measure over finite executions and a task schedule. This condition is used to avoid useless proof obligations in the definition of both exact and approximate simulations.

Definition 2.4. *Suppose $\mathcal{A}$ is a closed, task-PIOA and σ is a finite task schedule for $\mathcal{T}$. $\mu \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}})$ is* consistent with *σ if $\mathsf{supp}(\mu) \subseteq \mathsf{supp}(\mathsf{apply}(\bar{\nu}, \sigma))$.*

Suppose we have a mapping c that, given a finite task schedule σ and a task T of a task-PIOA $\mathcal{A}_1$, yields a task schedule of another task-PIOA $\mathcal{A}_2$. The idea is that $\mathsf{c}(\sigma, T)$ describes how $\mathcal{A}_2$ matches task T, given that it has already matched the task schedule σ. Using c, we define a new function $\mathsf{full}(\mathsf{c})$ that, given a task schedule σ, iterates c on all the elements of σ, thus producing a "full" task schedule of $\mathcal{A}_2$ that matches all of σ.

Definition 2.5. *Let $\mathcal{A}_1, \mathcal{A}_2$ be task-PIOAs, and let $\mathsf{c} : ({R_1}^* \times R_1) \to {R_2}^*$ be a function that assigns a finite task schedule of $\mathcal{A}_2$ to each finite task schedule of $\mathcal{A}_1$ and task of $\mathcal{A}_1$. The function $\mathsf{full}(\mathsf{c}) : {R_1}^* \to {R_2}^*$ is recursively defined as: $\mathsf{full}(\mathsf{c})(\lambda) := \lambda$, and $\mathsf{full}(\mathsf{c})(\sigma T) := \mathsf{full}(\mathsf{c})(\sigma)^\frown \mathsf{c}(\sigma, T)$ (the concatenation of $\mathsf{full}(\mathsf{c})(\sigma)$ and $\mathsf{c}(\sigma, T)$).*

Now we give the definition of exact simulation relation for task-PIOAs. Note that the simulation relations do not just relate states to states, but rather, probability measures on executions to probability measures on executions. The use of measures on executions here rather than just executions is motivated by certain cases that arise in proofs where related random choices are made at different points in the low-level and high-level models (see, e.g., proof of OT protocol in [6]).

Definition 2.6. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two comparable closed task-PIOAs. A relation R from $\mathsf{Disc}(\mathsf{Execs}^*(\mathcal{A}_1))$ to $\mathsf{Disc}(\mathsf{Execs}^*(\mathcal{A}_2))$ is a* simulation *from $\mathcal{A}_1$ to $\mathcal{A}_2$ if there exists $\mathsf{c} : ({R_1}^* \times R_1) \to {R_2}^*$ such that following properties hold:*

(i) **Start condition:** $\bar{\nu}_1\ R\ \bar{\nu}_2$.

(ii) **Step condition:** *If $\mu_1\ R\ \mu_2$, $\sigma \in {R_1}^*$, μ_1 is consistent with σ, μ_2 is consistent with $\mathsf{full}(\mathsf{c})(\sigma)$, and $T \in R_1$, then $\mu_1'\ \mathcal{E}(R)\ \mu_2'$ where $\mu_1' = \mathsf{apply}(\mu_1, T)$ and $\mu_2' = \mathsf{apply}(\mu_2, \mathsf{c}(\sigma, T))$.*

(iii) **Trace condition:** *If $\mu_1\ R\ \mu_2$, then $\mathsf{tdist}(\mu_1) = \mathsf{tdist}(\mu_2)$.*

We close this section with the statement of the soundness theorem for the above simulation relation which has been proved in [5].

Theorem 2.7. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be comparable closed action-deterministic task-PIOAs. If there exists a simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$, then $\mathsf{tdists}(\mathcal{A}_1) \subseteq \mathsf{tdists}(\mathcal{A}_2)$.*

3 Uniform Approximate Implementation

In this section we define approximate implementations for task-PIOAs based on the uniform metric on trace distributions and propose *Expanded Approximate Simulations (EAS)* as a sound method for proving uniform implementations. Informally, a task-PIOA $\mathcal{A}_1$ uniformly approximately implements a task-PIOA $\mathcal{A}_2$, if every trace distribution of $\mathcal{A}_1$ is "close" to some trace distribution of $\mathcal{A}_2$, where "closeness" is defined by the uniform metric on trace distributions.

Definition 3.1. *Let $\mathcal{A}$ be a closed task-PIOA. The* uniform metric (pseudo-metric) *over trace distributions of $\mathcal{A}$ is the function $\mathbf{d}_u : \mathsf{Disc}(\mathsf{Traces}_{\mathcal{A}}) \times \mathsf{Disc}(\mathsf{Traces}_{\mathcal{A}}) \to \mathbb{R}_{\geq 0}$ defined by:*

$$\mathbf{d}_u(\mu_1, \mu_2) := \sup_{C \in \mathcal{F}_{\mathsf{Traces}_{\mathcal{A}}}} |\mu_1(C) - \mu_2(C)| .$$

In general, the above definition makes $\mathbf{d}_u$ a pseudo-metric over trace distributions; some abuse of terminology we will refer to $\mathbf{d}_u$ as a metric. We define $\mathcal{A}_1$ to be an δ-implementation of $\mathcal{A}_2$ if the one-sided Hausdorff distance from $\mathsf{tdists}(\mathcal{A}_1)$ to $\mathsf{tdists}(\mathcal{A}_2)$ is at most δ.

Proposition 3.2. *Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are closed task-PIOAs. For $i \in \{1, 2\}$, let $\{\mu_{ij}\}_{j \in J}$ be a chain of discrete probability distributions on the traces of $\mathcal{A}_i$ and let $\lim_{j\to\infty} \mu_{ij} = \mu_i$. Then $\lim_{j\to\infty} \mathbf{d}_u(\mu_{1j}, \mu_{2j}) = \mathbf{d}_u(\mu_1, \mu_2)$.*

Proof. We have to show that for every $\epsilon > 0$, there exists $N \in \mathbb{N}$, such that for all $k > N$, $\mathbf{d}_u(\mu_{1k}, \mu_{2k}) - \mathbf{d}_u(\mu_1, \mu_2) < \epsilon$. From triangle inequality, we get that for any k, $\mathbf{d}_u(\mu_{1k}, \mu_{2k}) \leq \mathbf{d}_u(\mu_{1k}, \mu_1) + \mathbf{d}_u(\mu_1, \mu_2) + \mathbf{d}_u(\mu_2, \mu_{2k})$. Therefore, it suffices to show that exists $N \in \mathbb{N}$, such that for all $k > N$, $\mathbf{d}_u(\mu_{1k}, \mu_1) + \mathbf{d}_u(\mu_2, \mu_{2k}) \leq \epsilon$. Now since $\lim_{j\to\infty} \mu_{1j} = \mu_1$, $\lim_{j\to\infty} \mu_{2j} = \mu_2$, we know that there exists $N' \in \mathbb{N}$, such that for all $k > N'$, for every $C \in \mathcal{F}_{\mathsf{Traces}_{\mathcal{A}_i}}$, $|\mu_{ij}(C) - \mu_i(C)| \leq \frac{\epsilon}{2}$. If we choose $N = N'$, we have for all $k > N$, $\mathbf{d}_u(\mu_{1k}, \mu_1) + \mathbf{d}_u(\mu_2, \mu_{2k}) \leq \epsilon$, are required. □

Definition 3.3. *Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable, closed task-PIOAs. For $\delta > 0$, $\mathcal{A}_1$ is said to δ-implement $\mathcal{A}_2$, written as $\mathcal{A}_1 \leq_\delta \mathcal{A}_2$, if for every $\mu_1 \in \mathsf{tdists}(\mathcal{A}_1)$ there exists $\mu_2 \in \mathsf{tdists}(\mathcal{A}_2)$ such that $\mathbf{d}_u(\mu_1, \mu_2) \leq \delta$. Closed task-PIOAs $\mathcal{A}_1$ and $\mathcal{A}_2$ are said to be* δ-equivalent*, written as $\mathcal{A}_1 \cong_\delta \mathcal{A}_2$, if $\mathcal{A}_1 \leq_\delta \mathcal{A}_2$ and $\mathcal{A}_2 \leq_\delta \mathcal{A}_1$.*

Metrics over probability distributions have been a subject of intense research in probability theory (see, for example, the books [26] and [11]). Because of their applicability to probabilistic safety and termination proofs, in this paper we use the uniform metric and the discounted version of the uniform metric (see Section 4), to define approximate implementations for task-PIOAs. As we shall see in the next section, the soundness of expanded approximate simulations rely only weakly on the choice of the metric. In fact, with the appropriate changes in the definition of EAS, it is sound for proving approximate implementations with respect to any metric satisfying Proposition 3.2.

3.1 Expanded Approximate Simulations

Our definition of EAS relies on an *expansion* operation on real valued functions. This operation generalizes the notion of expansion of a relation used in Defini-

tion 2.6.

Definition 3.4. *Let x be an element of the set $\mathcal{X}$ and $\{\lambda_i\}_{i\in I}$ be a countable sequence of numbers such that $\sum_{i\in I}\lambda_i = 1$. If there exists a sequence $\{x_i\}$ in $\mathcal{X}$ such that $x = \sum_{i\in I}\lambda_i x_i$, then x is a* convex combination *of the $\{x_i\}'s$. A function $\phi : \mathcal{X} \to \mathbb{R}_{\geq 0}\cup\{\infty\}$ is* convex *if for every $x = \sum_{i\in I}\lambda_i x_i$, $\phi(x) \leq \sum_{i\in I}\lambda_i\phi(x_i)$. If equality holds then the function is said to be* distributive.

Definition 3.5. *Given a function $\phi : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}_{\geq 0}\cup\{\infty\}$, the* expansion *of ϕ, written as $\hat{\phi}$, is a function $\hat{\phi} : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}_{\geq 0}\cup\{\infty\}$ defined as:*

$$\hat{\phi}(x_1, y_1) := \min_{\substack{\psi\in\mathsf{Disc}(\mathcal{X}\times\mathcal{Y}) \\ x_1=\sum_x \psi(x,y)x \\ y_1=\sum_x \psi(x,y)y}} \left[\max_{(x,y)\in\mathsf{supp}(\psi)} \phi(x,y)\right] \tag{1}$$

The value of $\hat{\phi}$ is defined in terms of a minimization problem over all joint distributions over $\mathsf{Disc}(\mathcal{X}\times\mathcal{Y})$ that have first and second marginals with means equal to x_1 and y_1, respectively. The function that is minimized is the maximum value of ϕ over all points in the support of ψ. When stated in this form the definition of the expansion of ϕ is indeed reminiscent of the p^{th} Wasserstein metric for $p = \infty$. Given a function $\phi : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}_{\geq 0}\cup\{\infty\}$, an alternative but equivalent way of defining the *expansion* $\hat{\phi}$, is as follows:

Definition 3.6. *For any $\epsilon \geq 0$, $\hat{\phi}(x_1, y_1) \leq \epsilon$ if and only if there exists a joint distribution $\psi \in \mathsf{Disc}(\mathcal{X}\times\mathcal{Y})$ such that:*

$$\max_{x,y\in\mathsf{supp}(\psi)} \phi(x,y) \leq \epsilon \tag{2}$$

$$x_1 = \sum_{x,y\in\mathsf{supp}(\psi)} \psi(x,y)x \tag{3}$$

$$y_1 = \sum_{x,y\in\mathsf{supp}(\psi)} \psi(x,y)y \tag{4}$$

The *consistency requirements* imposed by Equations (3) and (4) constrain the choice of ψ to those joint distributions over $\mathcal{X}\times\mathcal{Y}$, for which the expected values of x and y coincide with x_1 and y_1. Given ϕ, we say that joint distribution ψ is a *feasible* for x_1 and y_1 if it satisfies the consistency requirements. If ϵ is the smallest nonnegative real for which there exists a feasible ψ that also satisfies Equation (2), that is, $\max_{x,y\in\mathsf{supp}(\psi)} \phi(x,y) \leq \epsilon$, then we say that ψ is an *optimal distribution* for $\hat{\phi}(x_1, y_1) = \epsilon$. The next proposition is a straightforward consequence of Definition 3.6.

Proposition 3.7. *For any $\phi : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}_{\geq 0}\cup\{\infty\}$ and $\epsilon > 0$, if $\phi(x_1, y_1) \leq \epsilon$ for some $x_1 \in \mathcal{X}$, $y_1 \in \mathcal{Y}$, then $\hat{\phi}(x_1, y_1) \leq \epsilon$.*

Proof. Suppose $\phi(x_1, y_1) = \epsilon_1$ for some $0 < \epsilon_1 \leq \epsilon$. The joint distribution δ_{x_1,y_1} is a feasible distribution for x_1 and y_1. Since $\phi(x_1, y_1) = \epsilon_1 \leq \epsilon$, $\hat{\phi}(x_1, y_1) \leq \epsilon$. □

Figure 1 shows a point (x_1, y_1) outside the set $\{(x, y) \mid \phi(x, y) \leq \epsilon\}$, where $\hat{\phi}(x_1, y_1) = \epsilon$. The marginal distributions for the optimal joint distribution ψ are shown on the x and the y axes.

Our new notion of approximate simulation for task-PIOAs is a function ϕ :

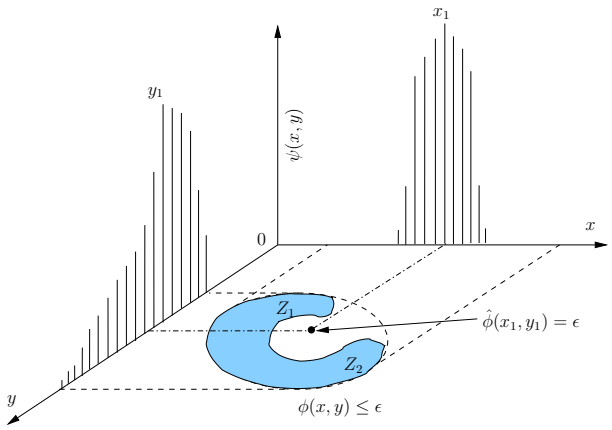


Fig. 1. Marginal distributions of the optimal joint distribution ψ for $\hat{\phi}(x_1, y_1) = \epsilon$. Support of ψ is contained within the elliptical region. In particular, ψ is concentrated in the regions Z_1 and Z_2 each carrying half of the total mass.

$\mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_1}) \times \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_2}) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ and the expansion of this function plays a key role in the definition of simulation. Informally, the simulation function ϕ gives a measure of similarity between two distributions over the execution fragments of two automata. If $\phi(\mu_1, \mu_2) \leq \epsilon$, then, first of all, it is possible to closely simulate from μ_2 anything that can happen from μ_1. Here closeness of simulation is measured with the $\mathbf{d}_u$ metric on the trace distributions. Secondly, if μ_1' and μ_2' are the distributions obtained by taking a step from μ_1 and μ_2, then μ_1' and μ_2' are also close in the sense that $\hat{\phi}(\mu_1', \mu_2') \leq \epsilon$.

Definition 3.8. *Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are two comparable closed task-PIOAs, ϵ is a nonnegative constant, and ϕ is a function $\mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_1}) \times \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_2}) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$. The function ϕ is an (ϵ, δ)-*expanded approximate simulation *from $\mathcal{A}_1$ to $\mathcal{A}_2$ if exists a function $\mathsf{c} : R_1^* \times R_1 \to R_2^*$ such that the following properties hold:*

(i) **Start condition:** $\phi(\bar{\nu}_1, \bar{\nu}_2) \leq \epsilon$.

(ii) **Step condition:** *If $\phi(\mu_1, \mu_2) \leq \epsilon$, $T \in R_1, \sigma \in R_1^*$ and μ_1 is consistent with σ, and μ_2 is consistent with $\mathsf{full}(\mathsf{c})(\sigma)$, then $\hat{\phi}(\mu_1', \mu_2') \leq \epsilon$, where $\mu_1' = \mathsf{apply}(\mu_1, T)$ and $\mu_2' = \mathsf{apply}(\mu_2, c(\sigma, T))$.*

(iii) **Trace condition:** *There exists $\delta > 0$ such that if $\phi(\mu_1, \mu_2) \leq \epsilon$ then $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) \leq \delta$.*

3.2 Soundness of Expanded Approximate Simulations

This section culminates in Theorem 3.11 which states that (ϵ, δ)-expanded approximate simulations are sound with respect to δ-approximate implementations. First we prove two key lemmas used in the proof of the theorem.

Lemma 3.9. *Suppose ϕ is a (ϵ, δ)-expanded approximate simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$. For any $\mu_1 \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_1})$ and $\mu_2 \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_2})$, if $\hat{\phi}(\mu_1, \mu_2) \leq \epsilon$ then $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) \leq \delta$.*

Proof. Since $\hat{\phi}(\mu_1, \mu_2) \leq \epsilon$ we know that there exists a joint distribution ψ which is feasible for μ_1, μ_2, and for every $\eta_1, \eta_2 \in \mathsf{supp}(\psi)$, $\phi(\eta_1, \eta_2) \leq \epsilon$. So, for $i \in \{1, 2\}$, $\mu_i = \sum_{\eta_1, \eta_2 \in \mathsf{supp}(\psi)} \psi(\eta_1, \eta_2)\eta_i$ and from the trace condition it follows that

$$\mathsf{tdist}(\mu_i) = \sum_{\eta_1,\eta_2 \in \mathsf{supp}(\psi)} \psi(\eta_1,\eta_2)\,\mathsf{tdist}(\eta_i).$$

We can then express $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2))$ as follows:

$$\begin{aligned}
&\sup_{C \in \mathcal{F}_{\mathsf{Traces}^*_{\mathcal{A}}}} |\,\mathsf{tdist}(\mu_1)(C) - \mathsf{tdist}(\mu_2)(C)| \\
&= \sup_{C \in \mathcal{F}_{\mathsf{Traces}^*_{\mathcal{A}}}} |\sum_{\eta_1,\eta_2} \psi(\eta_1,\eta_2)\,\mathsf{tdist}(\eta_1)(C) - \sum_{\eta_1,\eta_2} \psi(\eta_1,\eta_2)\,\mathsf{tdist}(\eta_2)(C)| \\
&\leq \sup_{C \in \mathcal{F}_{\mathsf{Traces}^*_{\mathcal{A}}}} \sum_{\eta_1,\eta_2} \psi(\eta_1,\eta_2)|(\mathsf{tdist}(\eta_1)(C) - \mathsf{tdist}(\eta_2)(C))|.
\end{aligned}$$

For any $\eta_1, \eta_2 \in \mathsf{supp}(\psi)$, $\phi(\eta_1,\eta_2) \leq \epsilon$ and since ϕ is an (ϵ,δ)-expanded approximate simulation, $\mathbf{d}_u(\mathsf{tdist}(\eta_1), \mathsf{tdist}(\eta_2)) \leq \delta$. From Definition 3.1, it follows that $|\,\mathsf{tdist}(\eta_1)(C) - \mathsf{tdist}(\eta_2)(C)| \leq \delta$. Therefore, we have $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) \leq \sum_{\eta_1,\eta_2} \psi(\eta_1,\eta_2)\delta \leq \delta$. □

Lemma 3.10. *Suppose $\phi : \mathsf{Disc}(X_1) \times \mathsf{Disc}(X_2) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a function, $\mu_i \in \mathsf{Disc}(X_i)$ for $i \in \{1,2\}$, $\hat{\phi}(\mu_1,\mu_2) \leq \epsilon$ with optimal distribution ψ. Let $f_i : \mathsf{Disc}(X_i) \to \mathsf{Disc}(X_i)$ be distributive functions, for $i \in \{1,2\}$. If for each $\rho_1,\rho_2 \in \mathsf{supp}(\psi)$, $\hat{\phi}(f_1(\rho_1), f_2(\rho_2)) \leq \epsilon$, then $\hat{\phi}(f_1(\mu_1), f_2(\mu_2)) \leq \epsilon$.*

Proof: For each $\rho_1,\rho_2 \in \mathsf{supp}(\psi)$, let $\psi_{\rho_1\rho_2}$ be the optimal distribution for $\hat{\phi}(f_1(\rho_1), f_2(\rho_2)) = \epsilon$. We define a joint distribution ψ' on $\mathsf{Disc}(X_1) \times \mathsf{Disc}(X_2)$ as follows:

$$\psi' := \sum_{(\rho_1,\rho_2) \in \mathsf{supp}(\psi)} \psi(\rho_1,\rho_2)\psi_{\rho_1,\rho_2} \tag{5}$$

and show that ψ' is a feasible distribution for $f_1(\mu_1)$ and $f_2(\mu_2)$ and for any $\eta_1,\eta_2 \in \mathsf{supp}(\psi')$, $\phi(\eta_1,\eta_2) \leq \epsilon$.

(i) For feasibility of ψ' we have to show that for $i \in \{1,2\}$, $f_i(\mu_i)$ equals:

$$\begin{aligned}
&\sum_{\eta_1 \in \mathsf{Disc}(X_1), \eta_2 \in \mathsf{Disc}(X_2)} \psi'(\eta_1,\eta_2)\eta_i \\
&= \sum_{\eta_1 \in \mathsf{Disc}(X_1), \eta_2 \in \mathsf{Disc}(X_2)} \left[\sum_{(\rho_1,\rho_2) \in \mathsf{supp}(\psi)} \psi(\rho_1,\rho_2)\psi_{\rho_1,\rho_2}(\eta_1,\eta_2) \right] \eta_i \\
&= \sum_{(\rho_1,\rho_2) \in \mathsf{supp}(\psi)} \psi(\rho_1,\rho_2) \left[\sum_{\eta_1 \in \mathsf{Disc}(X_1), \eta_2 \in \mathsf{Disc}(X_2)} \psi_{\rho_1,\rho_2}(\eta_1,\eta_2)\eta_i \right] \\
&= \sum_{(\rho_1,\rho_2) \in \mathsf{supp}(\psi)} \psi(\rho_1,\rho_2) f_i(\rho_i) \quad \text{[from feasibility of } \psi_{\rho_1,\rho_2}\text{]} \\
&= f_i\left(\sum_{(\rho_1,\rho_2) \in \mathsf{supp}(\psi)} \psi(\rho_1,\rho_2)\rho_i \right) \quad \text{[from distributivity of } f_i\text{]} \\
&= f_i(\mu_i) \quad \text{[from feasibility of } \psi\text{]}.
\end{aligned}$$

(ii) For optimality of ψ' it suffices to show that for all $\eta_1,\eta_2 \in \mathsf{supp}(\psi')$, $\phi(\eta_1,\eta_2) \leq$

ϵ. If $\psi'(\eta_1, \eta_2) > 0$ then from Equation (5) it follows that there exists $\rho_1, \rho_2 \in \mathsf{supp}(\psi)$ such that $\psi_{\rho_1,\rho_2}(\eta_1, \eta_2) > 0$. Since ψ_{ρ_1,ρ_2} is a optimal distribution for $\hat{\phi}(f_1(\rho_1), f_2(\rho_2)) = \epsilon$, from its optimality we know that for any $\nu_1, \nu_2 \in \mathsf{supp}(\psi_{\rho_1,\rho_2})$, $\phi(\nu_1, \nu_2) \leq \epsilon$. In particular, $\eta_1, \eta_2 \in \mathsf{supp}(\psi_{\rho_1,\rho_2})$ and so we have $\phi(\eta_1, \eta_2) \leq \epsilon$.

Theorem 3.11. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two closed comparable task-PIOAs. If there exists a (ϵ, δ)-expanded approximate simulation function from $\mathcal{A}_1$ to $\mathcal{A}_2$ then $\mathcal{A}_1 \leq_\delta \mathcal{A}_2$.*

Proof. Let ϕ be the assumed (ϵ, δ)-expanded approximate simulation function from $\mathcal{A}_1$ to $\mathcal{A}_2$. Let μ_1 be the probabilistic execution of $\mathcal{A}_1$ generated by the starting distribution $\bar{\nu}_1$ and a (finite or infinite) task schedule $T_1, T_2, \ldots$. For each $i > 0$, we define σ_i to be $\mathsf{c}(T_1 \ldots T_{i-1}, T_i)$. Let μ_2 be the probabilistic execution of $\mathcal{A}_2$ generated by $\bar{\nu}_2$ and the concatenation $\sigma_1, \sigma_2, \ldots$. It suffices to show that: $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) \leq \delta$.

For each $j \geq 0$, let us define $\mu_{1,j} := \mathsf{apply}(\bar{\nu}_1, T_1, \ldots, T_j)$ and $\mu_{2,j} := \mathsf{apply}(\bar{\nu}_2, \sigma_1, \ldots, \sigma_j)$. For $i \in \{1, 2\}$ and for each $j \geq 0$, $\mu_{i,j} \leq \mu_{i,j+1}$ and $\lim_{j\to\infty} \mu_{i,j} = \mu_i$. (the above uses Lemma A.7 of Appendix A). Observe that for every $j \geq 0$, $\mu_{1,j+1} = \mathsf{apply}(\mu_{1,j}, T_{j+1})$ and also that $\mu_{2,j+1} = \mathsf{apply}(\mu_{2,j}, \sigma_{j+1})$.

Step 1a. We prove by induction that for all $j \geq 0$, $\hat{\phi}(\mu_{1,j}, \mu_{2,j}) \leq \epsilon$. For $j = 0$, $\mu_{1,0} = \bar{\nu}_1$ and $\mu_{2,0} = \bar{\nu}_2$. By the start condition of the simulation function, $\phi(\mu_{1,0}, \mu_{2,0}) \leq \epsilon$ and therefore by Proposition 3.7 $\hat{\phi}(\mu_{1,0}, \mu_{2,0}) \leq \epsilon$.

Step 1b. For the inductive step, we assume that $\hat{\phi}(\mu_{1,j}, \mu_{1,j}) \leq \epsilon$ and show that $\hat{\phi}(\mu_{1,j+1}, \mu_{1,j+1}) \leq \epsilon$. First of all, note that $\mu_{1,j+1} = \mathsf{apply}(\mu_{1,j}, T_{j+1})$ and $\mu_{2,j+1} = \mathsf{apply}(\mu_{2,j}, \mathsf{c}(\sigma_j T_{j+1}))$. For $i \in \{1, 2\}$, let us define $f_i : \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_i}) \to \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_i})$ as $f_1(\eta) := \mathsf{apply}(\eta, T_{j+1})$ and $f_2(\eta) := \mathsf{apply}(\eta, \mathsf{c}(\sigma_j T_{j+1}))$. If we can apply Lemma 3.10, to the functions f_1 and f_2 then it follows that $\hat{\phi}(f_1(\mu_{1,j}), f_2(\mu_{2,j})) \leq \epsilon$ as required.

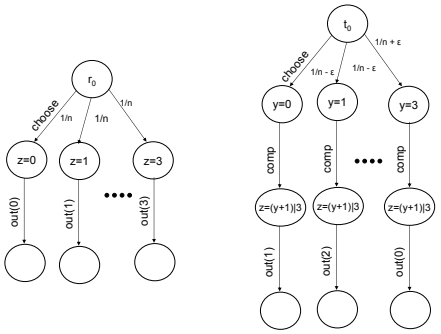
Step 1c. It remains to check that these two functions satisfy all the conditions in the hypothesis of Lemma 3.10. Distributivity of f_1 and f_2 follow from Proposition B.2 (see Appendix B). Suppose $\hat{\phi}(\mu_{1,j}, \mu_{1,j}) \leq \epsilon$ with optimal distribution ψ, and suppose $\eta_1, \eta_2 \in \mathsf{supp}(\psi)$, we have to show that $\hat{\phi}(f_1(\eta_1), f_2(\eta_2)) \leq \epsilon$. Since $\eta_1, \eta_2 \in \mathsf{supp}(\psi)$, from optimality of ψ, we know that $\phi(\eta_1, \eta_2) \leq \epsilon$. Observe that for $i \in \{1, 2\}$, $\mathsf{supp}(\eta_i) \subseteq \mathsf{supp}(\mu_{i,j})$, and thus η_1 is consistent with T_{j+1} and η_2 is consistent with $\mathsf{c}(\sigma_j T_{j+1})$. Therefore, by the step condition on ϕ, $\hat{\phi}(\mathsf{apply}(\eta_1, T_{j+1}), \mathsf{apply}(\eta_2, \mathsf{c}(\sigma_j T_{j+1}))) \leq \epsilon$. Since $f_1(\eta_1) = \mathsf{apply}(\eta_1, T_{j+1})$ and $f_2(\eta_2) = \mathsf{apply}(\eta_2, \mathsf{c}(\sigma_j T_{j+1}))$, we have $\hat{\phi}(f_1(\mu_{1,j}), f_2(\mu_{2,j})) \leq \epsilon$, as required in the hypothesis of Lemma 3.10.

Step 2. From Lemma 3.9, for each $j \geq 0$, $\mathbf{d}_u(\mathsf{tdist}\,\mu_{1,j}, \mathsf{tdist}\,\mu_{2,j}) \leq \delta$. From Lemma A.5 of Appendix A we know that for $i \in \{1, 2\}$, $\lim_{j\to\infty} \mathsf{tdist}(\mu_{i,j}) = \mathsf{tdist}(\mu_i)$. From Proposition 3.2 we conclude that $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) = \lim_{j\to\infty} \mathbf{d}_u(\mathsf{tdist}(\mu_{1,j}), \mathsf{tdist}(\mu_{2,j})) \leq \delta$. □

3.3 Need for Expansion

In the step condition in the definition of EAS (Definition 3.8) it is required that if $\phi(\mu_1, \mu_2) \leq \epsilon$ then $\hat{\phi}(\mu'_1, \mu'_2) \leq \epsilon$. Indeed, if we replace this condition with the weaker condition—if $\phi(\mu_1, \mu_2) \leq \epsilon$ then $\phi(\mu'_1, \mu'_2) \leq \epsilon$—the resulting approximate

simulation functions that we would obtain would be sound for proving approximate implementations. However, such *non-expanded* approximate simulation functions are be considerably less powerful than EASs. The key motivation for generalizing simulation relations to their current expanded form, first came from the verification of the Oblivious transfer protocol in [6]. In this section, we present a version of this example adapted to our setting of approximate implementations.



	$\delta_{y=0z=1}$	$\delta_{y=1z=2}$	$\delta_{y=2z=3}$	$\delta_{y=3\ z=0}$
$\delta_{z=0}$				¼
$\delta_{z=1}$	¼ -ε			ε
$\delta_{z=2}$		¼ - ε	ε	
$\delta_{z=3}$			¼	

Fig. 2. Left: *Rand* and *Trapdoor* automata. Right: Witnessing joing distribution.

Example 1 *(Trapdoor and Rand)* Consider an abstract automaton *Rand* that randomly chooses a number z between 0 and n and outputs it. We assume that n is odd. *Trapdoor*, on the other hand, first chooses a random number y with slightly different probabilities. The first $\frac{n-1}{2}$ numbers are chosen with probability $\frac{1}{n+1} - \epsilon$ and the remaining are chosen with probability $\frac{1}{n+1} + \epsilon$) *Trapdoor* then applies a known permutation (e.g., z = (y+1) mod n) to the chosen number, and outputs the result. The *Rand* and the *Trapdoor* for $n = 3$ automata are shown in Figure 2. Suppose the out actions producing the final value of z are external actions. Then, we would like these actions (tasks) to correspond which means that the choose step of *Trapdoor* should map to no step of *Rand*. We present an approximate simulation function that "ought to work" for this example. Instead of using a simulation function on distributions of finite execution fragments, we use a simpler ϕ that is a function on distribution of states.

$$\phi(\mu_1, \mu_2) := \begin{cases} \max_{s,u}[\mu_1(s) + \mu_2(u)] & \forall\ s \in \mathsf{supp}(\mu_1), u \in \mathsf{supp}(\mu_2), s.z \neq u.z \\ 0 & \forall\ s \in \mathsf{supp}(\mu_1), u \in \mathsf{supp}(\mu_2), s.z = s.y = \perp \\ \max_s |\frac{1}{n+1} - \mu_1(s)| & \text{otherwise.} \end{cases}$$

Informally, states corresponding to different values to z produce completely different outputs, and thus they should be relatively unrelated. The first condition in the definition of ϕ assigns a large value $(max_{s,u}\mu_1(s) + \mu_2(u))$ to distributions that contain such mismatched states. The second condition is satisfied only for the Dirac masses δ_{r_0} and δ_{t_0}, and therefore ϕ is 0. Finally, the third condition is satisfied for distributions supported on states that have the same value of z, and where the variable y has been assigned a value in the *Trapdoor* automaton.

Let $\mu_{11} = \mathsf{apply}(\delta_{t_0}, \mathsf{choose})$ and $\mu_{21} = \mathsf{apply}(\delta_{r_0}, \lambda) = \delta_{r_0}$. Then, for all $s \in \mathsf{supp}(\mu_{11})$, $s.z = r_0.z = \perp$, and hence by the third condition in the definition of ϕ, $\phi(\mu_{11}, \mu_{21}) = \epsilon$. Next, let $\mu_{12} = \mathsf{apply}(\mu_{11}, \mathsf{comp})$ and $\mu_{21} = \mathsf{apply}(\mu_{22}, \mathsf{comp})$. Then, there exists $s \in \mathsf{supp}(\mu_{21})$ and $u \in \mathsf{supp}(\mu_{22})$, such that $s.z \neq u.z$, and by

the first condition, $\phi(\mu_{12}, \mu_{22}) \geq \frac{2}{n+1}$, which is much larger than ϵ. Therefore, we cannot use ϕ as an approximate simulation function to prove that *Trapdoor* is a good approximate implementation for *Rand*.

We show that ϕ *can* be used as an approximate simulation function if we use ϕ as an EAS. It suffices to prove that $\hat{\phi}(\mu_{12}, \mu_{22}) \leq 2\epsilon$, and we will use the witnessing joint distribution shown in the table of Figure 2. Indeed, the marginal distributions of ψ match with μ_{21} and μ_{22}. Further, for any η, ν in the support of ψ, η and ν have the following properties: (1) either they are Dirac masses at states that have the same value of z, in which case $\phi(\eta, \nu) = \epsilon$ from the third condition in the definition of ϕ, otherwise (2) for any $s \in X_1$ and $u \in X_2$, $\eta(s) \leq \epsilon$ and $\nu(u) \leq \epsilon$, and therefore by the first condition $\phi(\eta, \nu) \leq 2\epsilon$. From the above it follows that $\hat{\phi}(\mu_{12}, \mu_{21}) \leq 2\epsilon$, which is what we set out to prove.

3.4 Probabilistic Safety

Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable closed task-PIOAs such that $\mathcal{A}_2 \leq_\delta A_1$. Suppose further that $\mathcal{A}_1$ violates some safety property S with probability at most p then we can conclude that $\mathcal{A}_2$ violates S with probability at most $p + \delta$. We first prove the following more general result. Let $(\mathsf{Traces}, \mathcal{F}_{\mathsf{Traces}})$ be the measurable space of traces containing the traces of both $\mathcal{A}_1$ and $\mathcal{A}_2$. Let $(X, \mathcal{F}_X)$ be another measurable space. A random variable is a measurable function $\mathbf{X} : (\mathsf{Traces}, \mathcal{F}_{\mathsf{Traces}}) \to (X, \mathcal{F}_X)$. We use the standard notation $\mu[\mathbf{X} = x] := \mu(\{\beta \in \mathsf{Traces} \mid \mathbf{X}(\beta) = x\}$, for $x \in X$.

Proposition 3.12. *Let* $\mathbf{X}$ *be random variable on* $(\mathsf{Traces}, \mathcal{F}_{\mathsf{Traces}})$. *Suppose* $\mathcal{A}_2 \leq_\delta \mathcal{A}_1$ *and there exists* $0 \leq p \leq 1$ *such that for all* $\mu_1 \in \mathsf{tdists}(\mathcal{A}_1)$, $\mu_1([\mathbf{X} = x]) \leq p$. *Then, for all* $\mu_2 \in tdist(\mathcal{A}_2)$, $\mu_2[\mathbf{X} = x] \leq \delta + p$.

Proof. Fix $\mu_2 \in \mathsf{tdists}\,\mathcal{A}_2$. Since $\mathcal{A}_2 \leq_\delta \mathcal{A}_1$ from Definition 3.3 there exists $\mu_1 \in \mathsf{tdists}(\mathcal{A}_1)$, such that $\mathbf{d}_u(\mu_1, \mu_2) \leq \delta$. We know that $\sup_C |\mu_2(C) - \mu_1(C)| \leq \delta$. In particular, $|\mu_2([\mathbf{X} = x]) - \mu_1([\mathbf{X} = x])| \leq \delta$. As $\mu_1([\mathbf{X} = x]) \leq p$, we have $\mu_2([\mathbf{X} = x]) \leq p + \delta$ as required. □

We denote the common set of external actions of $\mathcal{A}_1$ and $\mathcal{A}_2$ by E. Let us assume that violation of some safety property S is indicated by the occurrence of one of the external actions from the set $U \subseteq E$. We define the function $\mathbf{X}_U : \mathsf{Traces} \to \{0, 1\}$ as $\mathbf{X}_U(\beta) := 1$ if some action from U occurs in the trace β, otherwise $\mathbf{X}_U(\beta) := 0$. It can be easily checked that $\mathbf{X}_U$ is a measurable function and therefore is a boolean valued random variable. Then, the event $[\mathbf{X}_U = 1]$ corresponds to the set of traces in which S is violated. Now, if we know that in any trace distribution of $\mathcal{A}_1$ the probability of any U occurring is at most p and that $\mathcal{A}_2 \leq_\delta \mathcal{A}_1$, then from Proposition 3.12 we can conclude that in any trace distribution of $\mathcal{A}_2$ the probability of occurrence of U is at most $\delta + p$.

4 Discounted Uniform Metric

In the preceding section we defined uniform approximate implementation for PIOAs and proved that EASs are sound for proving this implementation relationship. We

also demonstrated that uniform approximate implementations are suitable for reasoning about certain classes of properties, like safety properties, where it is sufficient to quantify the absolute discrepancy in the trace distributions over all sets of traces. For certain other classes of properties the uniform metric is not suitable, because the worst case discrepancy over all sets of traces does not convey useful information. We illustrate this with the following example.

Example 2. *(Randomized Consensus)* The Ben-Or consensus protocol [3] is a randomized algorithm for n fault-prone processors to agree on a valid value by communicating over an asynchronous network. The algorithm proceeds in a sequence of stages in each of which nonfaulty processes send and receive messages based on coin-flips and comparison of values. If the processes have access to perfectly random coins, then with probability $\frac{1}{2^n}$, a stage ends successfully and all nonfaulty processes agree on a value, and after one communication round of a successful stage the consensus value is disseminated. An unsuccessful stage is followed by the beginning of the next stage.

The automaton in Figure 3 captures the termination behavior of the algorithm. The protocol starts is state s_{10}, the starting state for each of the successive stages are the states $s_{20}, s_{30}, \ldots$. The successful completion of the i^{th} stage is represented by state s_{i1}. The action a models the computation and communication within a stage. From stage s_{i0}, with probability p it leads to $s_{(i+1)0}$, the next stage, and with probability $1-p$ it leads to s_{i1}. The action d marks the termination of the protocol and it takes s_{i1} to s_{i2} with probability 1.

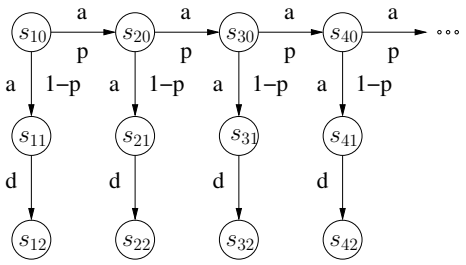


Fig. 3. Automata representing Ben-Or consensus protocol.

Suppose PIOA $\mathcal{A}_1$ is an instance of the automaton in Figure 3 with perfect random coins, that is, $p = 1 - \frac{1}{2^n}$ and $1 - p = \frac{1}{2^n}$. And let $\mathcal{A}_2$ be a PTIOA instance of the same automaton with slightly biased coins. We model the transition probabilities for $\mathcal{A}_2$ by $p + \epsilon$ and $1 - p - \epsilon$, for a small positive ϵ. We would like to compare the probabilities of termination of $\mathcal{A}_1$ and $\mathcal{A}_2$ after a certain number of rounds, say k. With the uniform approximate implementation, we can show that the difference in the probabilities is less than δ, for a fixed $\delta > 0$, however if individual probabilities of termination are themselves less than δ then this δ-approximation is too coarse and does not give us any useful information. In the remainder of this section, we show how a discounted version of the uniform metric can be used to make more fine grained comparison of probabilities of traces.

4.1 Discounted Approximate Simulations

Definition 4.1. *A probability distribution μ on execution fragments of $\mathcal{A}$ is said to be* finite *if $\mathsf{Frags}^*_{\mathcal{A}}$ is a support for μ. A trace distribution μ of $\mathcal{A}$ is finite if $\mathsf{Traces}^*_{\mathcal{A}}$ is a support for μ.*

Since any set of finite execution fragments is measurable, any finite probability distribution on execution fragments of $\mathcal{A}$ can also be viewed as a discrete probability measure on $\mathsf{Frags}^*_{\mathcal{A}}$. Likewise, a finite trace distribution can be viewed as a discrete distribution over $\mathsf{Traces}^*_{\mathcal{A}}$. In this section, we consider task-PIOAs with finite (trace) distributions and will treat these distributions as discrete distributions on execution fragments or traces.

Definition 4.2. *For any $k \in \mathbb{N}$, the k^{th} uniform metric is a function $\mathbf{d}_k : \mathsf{Disc}(\mathsf{Traces}_{\mathcal{A}}) \times \mathsf{Disc}(\mathsf{Traces}_{\mathcal{A}}) \to \mathbb{R}_{\geq 0}$ defined as:*

$$\mathbf{d}_k(\mu_1, \mu_2) := \max_{\beta \in E^*, |\beta| = n} |\mu_1(\beta) - \mu_2(\beta)|.$$

Definition 4.3. *Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable, closed task-PIOAs and $\{\delta_k\}_{k\in\mathbb{N}}$ is a collection of positive real numbers, called* discount factors. *If for every trace distribution μ_1 in $\mathsf{tdist}(\mathcal{A}_1)$ there exists a trace distribution $\mu_2 \in \mathsf{tdist}(\mathcal{A}_2)$ such that for every $k \in \mathbb{N}$, $\mathbf{d}_k(\mu_1, \mu_2) \leq \delta_k$, then we say that $\mathcal{A}_1$ δ_k-*implements *$\mathcal{A}_2$ and write this as $\mathcal{A}_1 \leq_{\delta_k} \mathcal{A}_2$. $\mathcal{A}_1$ and $\mathcal{A}_2$ are said to be δ_k-*equivalent*, written as $\mathcal{A}_1 \cong_{\delta_k} \mathcal{A}_2$, if $\mathcal{A}_1 \leq_{\delta_k} \mathcal{A}_2$ and $\mathcal{A}_2 \leq_{\delta_k} \mathcal{A}_1$.*

Proposition 4.4. *For all $k \in \mathbb{N}$, d_k is a pseudometric.*

Proof. The symmetry property is easy to check. We prove that $\mathbf{d}_k$ satisfies the triangle inequality. Let μ_1, μ_2, μ_3 be distributions on E^*. $\mathbf{d}_k(\mu_1, \mu_3) = \max_{\beta \in E^*, |\beta| = k} |\mu_1(\beta) - \mu_3(\beta)|$. Suppose β_3 is a trace that realizes the supremum.

$$\begin{aligned} |\mu_1(\beta_3) - \mu_3(\beta_3)| &\leq |\mu_1(\beta_3) - \mu_2(\beta_3)| + |\mu_2(\beta_3) - \mu_3(\beta_3)| \\ \mathbf{d}_k(\mu_1, \mu_3) &\leq \max_{\beta, |\beta| = k} |\mu_1(\beta) - \mu_2(\beta)| + \max_{\beta, |\beta| = k} |\mu_2(\beta) - \mu_3(\beta)|, \\ &\leq \mathbf{d}_k(\mu_1, \mu_2) + \mathbf{d}_k(\mu_2, \mu_3). \end{aligned}$$

□

We define a new kind of approximate simulation called *Discounted Approximate Simulation (DAS)* for proving discounted approximate implementations for task-PIOAs. Given a distribution μ over executions (or traces) we denote the longest execution (respectively trace) in the support of μ by $\mathsf{L}(\mu)$. We extend this notation to a pair of distributions by defining $\mathsf{L}(\mu_1, \mu_2) = \max(\mathsf{L}(\mu_1), \mathsf{L}(\mu_2))$.

Definition 4.5. *Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are two comparable closed task-PIOAs, and $\{\phi_k\}_{k\in\mathbb{N}}$ is a collection of functions, where each $\phi_k : \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_1}) \times \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_2}) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$. Given a collection of real number pairs $\{\epsilon_k, \delta_k\}_{k\in\mathbb{N}}$, the collection $\{\phi_k\}$ is an (ϵ_k, δ_k)-*discounted approximate simulation *from $\mathcal{A}_1$ to $\mathcal{A}_2$ if there exists a function $\mathsf{c} : R_1^* \times R_1 \to R_2^*$ such that the following properties hold:*

(i) **Start condition:** $\phi_0(\bar{\nu}_1, \bar{\nu}_2) \leq \epsilon_0$.

(ii) **Step condition:** *If for all $k \leq \mathsf{L}(\mu_1, \mu_2)$, $\phi_k(\mu_1, \mu_2) \leq \epsilon_k$, $T \in R_1, \sigma \in R_1^*$, μ_1 is consistent with σ, and μ_2 is consistent with $\mathsf{full}(\mathsf{c})(\sigma)$, then for all $k \leq$*

$\mathsf{L}(\mu_1', \mu_2')$, $\phi_k(\mu_1', \mu_2') \leq \epsilon_k$, *where* $\mu_1' = \mathsf{apply}(\mu_1, T)$ *and* $\mu_2' = \mathsf{apply}(\mu_2, c(\sigma, T))$.

(iii) **Trace condition:** *If for all* $k \leq \mathsf{L}(\mu_1, \mu_2)$, $\phi_k(\mu_1, \mu_2) \leq \epsilon_k$ *then for all* $k \leq \mathsf{L}(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2))$ $\mathbf{d}_k(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) \leq \delta_k$.

We prove Theorem 4.6 which states that (ϵ_k, δ_k)- approximate simulations are sound with respect to δ_k-approximate implementations.

Theorem 4.6. *Let* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *be two closed isomorphic comparable task-PIOAs. If there exists a* (ϵ_k, δ_k)*-discounted approximate simulation function from* $\mathcal{A}_1$ *to* $\mathcal{A}_2$ *then* $\mathcal{A}_1 \leq_{\delta_k} \mathcal{A}_2$.

Proof. Let ϕ be the assumed (ϵ_k, δ_k)-discounted approximate simulation function from $\mathcal{A}_1$ to $\mathcal{A}_2$. Let μ_1 be the probabilistic execution of $\mathcal{A}_1$ generated by the starting distribution $\bar{\nu}_1$ and a finite task schedule $T_1, T_2, \ldots, T_n$. For each $i > 0$, we define σ_i to be $\mathsf{c}(T_1 \ldots T_{i-1}, T_i)$. Let μ_2 be the probabilistic execution of $\mathcal{A}_2$ generated by $\bar{\nu}_2$ and the concatenation $\sigma_1, \sigma_2, \ldots, \sigma_n$. It suffices to show that $\mathbf{d}_w(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) \leq \delta$.

For each $j \geq 0$, let us define $\mu_{1,j} := \mathsf{apply}(\bar{\nu}_1, T_1, \ldots, T_j)$ and $\mu_{2,j} := \mathsf{apply}(\bar{\nu}_2, \sigma_1, \ldots, \sigma_j)$. For $i \in \{1, 2\}$ and for each $j \geq 0$, $\mu_{i,j} \leq \mu_{i,j+1}$ and $\mu_{i,n} = \mu_i$. Observe that for every $j \geq 0$, $\mu_{1,j+1} = \mathsf{apply}(\mu_{1,j}, T_{j+1})$ and $\mu_{2,j+1} = \mathsf{apply}(\mu_{2,j}, \sigma_{j+1})$.

We prove by induction that for all $j \geq 0$, for all $k \leq \mathsf{L}(\mu_{1,j}, \mu_{2,j})$, $\phi_k(\mu_{1,j}, \mu_{2,j}) \leq \epsilon_k$. For $j = 0$, $\mu_{1,0} = \bar{\nu}_1$ and $\mu_{2,0} = \bar{\nu}_2$. By the start condition of the simulation function, $\phi_0(\mu_{1,0}, \mu_{2,0}) \leq \epsilon$. For the inductive step, we assume that for all $k \leq \mathsf{L}(\mu_{1,j}, \mu_{2,j})$, $\phi_k(\mu_{1,j}, \mu_{2,j}) \leq \epsilon_k$. Then, from Part (ii) of Definition 3.8 it follows that for all $k \leq \mathsf{L}(\mu_{1,j+1}, \mu_{2,j+1})$, $\phi_k(\mu_{1,j+1}, \mu_{2,j+1}) \leq \epsilon_k$. In particular, for all $k \leq \mathsf{L}(\mu_1, \mu_2)$, $\phi_k(\mu_1, \mu_2) \leq \epsilon_k$, from which, using condition (iii) it follows that for all $k \leq \mathsf{L}(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2))$, $d_k(tdist(\mu_1), \mathsf{tdist}(\mu_2)) \leq \delta_k$. □

Example 2. *(Continued)* Let $\epsilon_k = \delta_k = (p + \epsilon)^k - p^k$, for each $k \in \mathbb{N}$. We will show that $\mathcal{A}_1$ and $\mathcal{A}_2$ are δ_k-equivalent using the following discounted approximate simulation:

$$\text{for each } k, \quad \phi_k(\mu_1, \mu_2) = max_{\alpha, \mathsf{anum}(\alpha)=k} |\mu_1(\alpha) - \mu_2(\alpha)|, \tag{6}$$

where $\mu_1 \in \mathsf{Disc}(\mathsf{Execs}^*_{\mathcal{A}_1})$, $\mu_2 \in \mathsf{Disc}(\mathsf{Execs}^*_{\mathcal{A}_2})$, and $\mathsf{anum}(\alpha)$ is the number of occurrence of the action a in the execution α.

Proposition 4.7. *The collection of functions* $\{\phi_k\}$ *defined above is an* (ϵ_k, δ_k)*-discounted approximate simulation from* $\mathcal{A}_1$ *to* $\mathcal{A}_2$.

Proof sketch. We check that the collection $\{\phi_k\}$ satisfies the three conditions in Definition 4.5.

(i) **Start condition:** $\nu_1 = \nu_2 = \delta_{s_{10}}$, and therefore $\phi_0(\nu_1, \nu_2) = 0$.

(ii) **Step condition:** We define the task correspondence function in the obvious way, $\mathsf{c}(\sigma, T) := T$, where σ is a task schedule and T is a task for $\mathcal{A}_1$. Thus for any $\mu_1 \in \mathsf{Disc}(\mathsf{Execs}^*_{\mathcal{A}_1})$ and $\mu_2 \mathsf{Disc}(\mathsf{Execs}^*_{\mathcal{A}_2})$ that are obtained from ν_1 and ν_2 by applying a sequence of tasks, $\mathsf{L}(\mu_1, \mu_2) = \mathsf{L}(\mu_1) = \mathsf{L}(\mu_2)$. Consider any $\mu_1 \in \mathsf{Disc}(\mathsf{Execs}^*_{\mathcal{A}_1})$,$\mu_2 \in \mathsf{Disc}(\mathsf{Execs}^*_{\mathcal{A}_2})$, and suppose $\mu_1 = \mathsf{apply}(\nu_1, \sigma)$ and $\mu_2 = \mathsf{apply}(\nu_2, \mathsf{full}(\mathsf{c})(\sigma))$. Let us denote $\mu_1' = \mathsf{apply}(\mu_1, T)$, and $\mu_2' = \mathsf{apply}(\mu_2, \mathsf{c}(\sigma, T)) = \mathsf{apply}(\mu_2, T)$. Then, it suffices to show that for all $k \leq \mathsf{L}(\mu_1, \mu_2)$, $\phi_k(\mu_1', \mu_2') \leq (p+\epsilon)^k - \epsilon^k$. This part of the proof is by a case analysis

on the types of tasks, $T = \{\mathsf{a}\}, \{\mathsf{d}\}$ and the types of executions.

The interesting cases are for $T = \{\mathsf{a}\}$ and executions of the form $\alpha = \alpha'\mathsf{a}s_{k0}$ or $\alpha = \alpha'\mathsf{a}s_{(k-1)1}$, for some $k \leq \mathsf{L}(\mu_1)$. For the first case, $\mu_1'(\alpha) = \mu_1(\alpha')p$ and $\mu_2'(\alpha) = \mu_2(\alpha')(p+\epsilon)$, and therefore $\phi_{k+1}(\mu_1', \mu_2') = p|\mu_2(\alpha') - \mu_1(\alpha')| + \epsilon\mu_2(\alpha')$. From the inductive hypothesis, $|\mu_2(\alpha') - \mu_1(\alpha')| \leq \epsilon_k$. It follows that, $\phi_{k+1}(\mu_1', \mu_2') \leq p|(p+\epsilon)^k - p^k| + \epsilon(p+\epsilon)^k \leq \epsilon^{k+1}$. Likewise in the second case, $\mu_1'(\alpha) = \mu_1(\alpha')(1-p)$ and $\mu_2'(\alpha) = \mu_2(\alpha')(1-p-\epsilon)$, and performing a similar calculation as above, we can show that $\phi_{k+1}(\mu_1', \mu_2') \leq \epsilon_k$.

(iii) **Trace condition:** First of all, for any $\mu_1 \in \mathsf{Disc}(\mathsf{Execs}^*_{\mathcal{A}_1})$ that are obtained from ν_1 by applying a sequence of tasks, $\mathsf{L}(\mu_1) = \mathsf{L}(\mathsf{tdist}(\mu_1))$. If β is a trace of the form $a^k d$, for some $k \geq 0$. Then, for $i \in \{1,2\}$, $\mathsf{tdist}(\mu_i)(\beta) = \mu_i(\alpha)$, where $\alpha = s_{10}\mathsf{a}s_{20}\ldots s_{k0}\mathsf{a}s_{k1}\mathsf{d}s_{k2}$. From which it follows that $|\,\mathsf{tdist}(\mu_1)(\beta) - \mathsf{tdist}(\mu_2)(\beta)| = |\mu_1(\alpha) - \mu_2(\alpha)| \leq \phi_{k+1}(\mu_1, \mu_2) \leq \epsilon_{k+1}$. On the other hand, if β is a trace of the form a^{k+1}, for some $k \geq 0$. Then, for $i \in \{1,2\}$, $\mathsf{tdist}(\mu_i)(\beta) = \mu_i(\alpha_1) + \mu_i(\alpha_2)$, where $\alpha_1 = s_{10}\mathsf{a}s_{20}\ldots s_{(k+1)0}\mathsf{a}s_{(k+1)1}$ and where $\alpha_2 = s_{10}\mathsf{a}s_{20}\ldots s_{(k+2)0}$. Thus, $|\,\mathsf{tdist}(\mu_1)(\beta) - \mathsf{tdist}(\mu_2)(\beta)| = |\mu_1(\alpha_1) + \mu_1(\alpha_2) - \mu_1(\alpha_1) - \mu_2(\alpha_1)| = |\mu_1(\alpha) - \mu_1(\alpha)|$, where $\alpha = s_{10}\mathsf{a}s_{20}\ldots s_{(k+1)0}$. Therefore, $|\,\mathsf{tdist}(\mu_1)(\beta) - \mathsf{tdist}(\mu_2)(\beta)| \leq \phi_{k+1}(\mu_1, \mu_2) \leq \epsilon_{k+1}$ as required.

5 Approximations for Task-PIOAs

In this section, we discuss how the notion of uniform approximate implementations and the soundness of EASs extendeds to general (not necessarily closed) task-PIOAs. In an analogous manner, discounted approximate implementation and DAS can also be extended.

The basic idea is to define a new notion of implementation following the approach of [5]. We formulate the external behavior of a $\mathcal{A}$ as a mapping from possible "environments" for $\mathcal{A}$ to sets of trace distributions that can arise when $\mathcal{A}$ is composed with the given environment.

Definition 5.1. *An* environment *for task-PIOA $\mathcal{A}$ is a task-PIOA $\mathcal{E}$ such that the* composition *of $\mathcal{A}$ and $\mathcal{E}$ is closed.*

Definition 5.2. *The* external behavior *of a task-PIOA $\mathcal{A}$, written as $extbeh_{\mathcal{A}}$, is a function that maps each environment task-PIOA $\mathcal{E}$ for $\mathcal{A}$ to the set of trace distributions of the composition of $\mathcal{A}$ and $\mathcal{E}$.*

Approximate implementation for general task-PIOAs can then be defined to be inclusion of external behavior for all environments.

Definition 5.3. *If $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable then $\mathcal{A}_1$ is said to* δ-implement *$\mathcal{A}_2$, for some $\delta \geq 0$, if for every environment task-PIOA $\mathcal{E}$ for both $\mathcal{A}_1$ and $\mathcal{A}_2$, for every $\mu_1 \in extbeh_{\mathcal{A}_1}(\mathcal{E})$ there exists $\mu_2 \in extbeh_{\mathcal{A}_2}(\mathcal{E})$ such that $\mathbf{d}_u(\mu_1, \mu) \leq \delta$.*

Based on this modified definition of approximate implementation the soundness of expanded approximate simulations for general task-PIOAs follow as a Corollary to Theorem 3.11.

Corollary 5.4. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two comparable task-PIOAs. Suppose that for every environment $\mathcal{E}$ for both $\mathcal{A}_1$ and $\mathcal{A}_2$, there exists a $(\epsilon_{\mathcal{E}}, \delta)$-approximate simu-*

lation function from the composition of $\mathcal{A}_1$ and $\mathcal{E}$ to the composition of $\mathcal{A}_2$ and $\mathcal{E}$. Then $\mathcal{A}_1 \leq_\delta \mathcal{A}_2$.

5.1 Conclusions

In this paper we have introduced approximate implementations for probabilistic I/O automata. We have employed the task mechanism of [6] to obtain the trace distributions of a PIOA, and then we have defined two different kinds of approximate implementations, based on the uniform metric and the discounted uniform metric on trace distributions. We proposed expanded approximate simulations and discounted approximate simulations for proving, the two proposed implementation relations, respectively. EAS and DAS can be used to approximately reason about probabilistic safety and termination properties. PIOAs can be nondeterministic and our construction does not require the underlying state spaces of the automata or the space of external actions to be metric spaces.

In our formulation of expanded approximate simulations, a simulation proof reduces to finding an optimal joint distribution satisfying certain constrains on the marginals. This is closely related to the well known Kantorovich optimal transportation problem. For well-behaved classes of simulation functions, therefore, we would like to explore the possibility of proving approximate simulations by solving optimization problems.

In the future, we want develop a new kind of *Discounted Expanded Approximate Simulations* that combines the features of EAS and DAS. We would also like to develop simulation based proof techniques where the simulation functions are functions of distributions over states and not functions of distributions over execution fragments. Finally, we would like to extend the notion approximate implementations to the Probabilistic Timed I/O Automaton framework [24].

Acknowledgment
We thank Professor Sanjoy Mitter for many useful comments on this work.

References

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

[2] C. Baier. Polynomial-time algorithms for testing probabilistic bisimulation and simulation. In R. Alur and T. A. Henzinger, editors, *Proceedings of the Eighth International Conference on Computer Aided Verification CAV*, volume 1102 of *LNCS*, pages 50–61, New Brunswick, NJ, USA, 1996.

[3] M. Ben-Or. Another advantage of free choice: Completely asynchronous agreement protocols. In *PODC*, pages 27–30, Montreal, Canada, August 1983.

[4] M. L. Bujorianu, J. Lygeros, and M. C. Bujorianu. Bisimulation for general stochastic hybrid systems. In Morari and Thiele [25], pages 198–214.

[5] R. Canetti, L. Cheung, D. Kaynar, M. Liskov, N. Lynch, O. Pereira, and R. Segala. Task-structured probabilistic I/O automata. Technical Report MIT-CSAIL-TR-2006-060, Massachusetts Institute of Technology, Cambridge, MA, September 2006.

[6] R. Canetti, L. Cheung, D. Kaynar, M. Liskov, N. Lynch, O. Pereira, and R. Segala. Using task-structured probabilistic I/O automata to analyze an oblivious transfer protocol. Technical Report MIT-CSAIL-TR-2006-019, Massachusetts Institute of Technology, Cambridge, MA, March 2006. Available from http://theory.csail.mit.edu/tds/papers/Kirli/TR-2006-019.pdf.

[7] L. Cheung. *Reconciling nondeterministic and probabilistic choices.* PhD thesis, ICIS, Radboud University Nijmegen, The Netherlands, 2006.

[8] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating labelled markov processes. *Inf. Comput.*, 184(1):160–200, 2003.

[9] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled markov processes. *Theor. Comput. Sci.*, 318(3):323–354, 2004.

[10] J. Desharnais, R. Jagadeesan, V. Gupta, and P. Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS), Copenhagen, Denmark, 22-25 July 2002*, pages 413–422. IEEE Computer Society, 2002.

[11] R. M. Dudley. *Probabilities and Metrics:Convergence of laws on metric spaces, with a view to statistical testing*. Number 45 in Lecture Notes Series. Aarhus Universitet, June 1976.

[12] A. Girard, A. A. Julius, and G. J. Pappas. Approximate simulation relations for hybrid systems. In *IFAC Analysis and Design of Hybrid Systems*, Alghero, Italy, June 2006.

[13] A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. In *IEEE Transactions on Automatic Control*, 2005.

[14] D. K. Goldenberg, J. Lin, and A. S. Morse. Towards mobility as a network control primitive. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 163–174. ACM Press, 2004.

[15] V. Gupta, R. Jagadeesan, and P. Panangaden. Approximate reasoning for real-time probabilistic processes. *The Quantitative Evaluation of Systems, First International Conference on (QEST'04)*, 00:304–313, 2004.

[16] C.-C. Jou and S. A. Smolka. Equivalences, congruences and complete approximations for probabilistic processes. In *CONCUR 90*, number 458 in LNCS. Springer-Verlag, 1990.

[17] D. K. Kaynar, N. Lynch, R. Segala, and F. Vaandrager. *The Theory of Timed I/O Automata*. Synthesis Lectures on Computer Science. Morgan Claypool, November 2005. Also available as Technical Report MIT-LCS-TR-917.

[18] M. Z. Kwiatkowska and G. Norman. Probabilistic metric semantics for a simple language with recursion. In *MFCS '96: Proceedings of the 21st International Symposium on Mathematical Foundations of Computer Science*, pages 419–430, London, UK, 1996. Springer-Verlag.

[19] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991.

[20] P. Lincoln, J. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic poly-time framework for protocol analysis. In *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*, pages 112–121, New York, NY, USA, 1998. ACM Press.

[21] M. W. Mislove, J. Ouaknine, D. Pavlovic, and J. Worrell. Duality for labelled markov processes. In *Proceedings of FOSSACS 04*, volume 2987 of *LNCS*. Springer, 2004.

[22] M. W. Mislove, J. Ouaknine, and J. Worrell. Axioms for probability and nondeterminism. *ENTCS*, 2004.

[23] S. Mitra and N. Lynch. Approximate simulations for task-structured probabilistic I/O automata. In *LICS workshop on Probabilistic Automata and Logics (PAul06)*, Seattle, WA, August 2006.

[24] S. Mitra and N. Lynch. Probabilistic timed I/O automata, October 2006. Submitted for review.

[25] M. Morari and L. Thiele, editors. *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, volume 3414 of *Lecture Notes in Computer Science*. Springer, 2005.

[26] S. T. Rachev. *Probability metrics and the stability of stochastic models*. John Wiley & Sons, 1991.

[27] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Technology, June 1995.

[28] S. Strubbe and A. J. van der Schaft. Bisimulation for communicating piecewise deterministic markov processes (cpdps). In Morari and Thiele [25], pages 623–639.

[29] P. Tabuada, G. J. Pappas, and P. U. Lima. Composing abstractions of hybrid systems. In Tomlin and Greenstreet [31], pages 436–450.

[30] A. Tiwari and G. Khanna. Series of abstractions for hybrid automata. In Tomlin and Greenstreet [31], pages 465–478.

[31] C. Tomlin and M. R. Greenstreet, editors. *Hybrid Systems: Computation and Control, 5th International Workshop, HSCC 2002, Stanford, CA, USA, March 25-27, 2002, Proceedings*, volume 2289 of *Lecture Notes in Computer Science*. Springer, 2002.

[32] F. van Breugel, M. Mislove, J. Ouaknine, and J. B. Worrell. An intrinsic characterization of approximate probabilistic bisimilarity. In *Proceedings of FOSSACS 03*, LNCS. Springer, 2003.

[33] F. van Breugel, M. W. Mislove, J. Ouaknine, and J. Worrell. Domain theory, testing and simulation for labelled markov processes. *Theoretical Computer Science*, 2005.

[34] F. van Breugel and J. Worrell. An algorithm for quantitative verification of probabilistic transition systems. In *CONCUR '01: Proceedings of the 12th International Conference on Concurrency Theory*, pages 336–350, London, UK, 2001. Springer-Verlag.

[35] F. van Breugel and J. Worrell. Towards quantitative verification of probabilistic transition systems. In *ICALP '01: Proceedings of the 28th International Colloquium on Automata, Languages and Programming,*, pages 421–432, London, UK, 2001. Springer-Verlag.

A Appendix: Limits of Chains of Distributions

All the definitions and lemmas in this Appendix are from [5]. In this Appendix $\mathcal{A}$ will be a task-PIOA. Given a finite execution fragment α of $\mathcal{A}$, the cone of executions generated by this fragment C_α is the set of all execution fragments that extend α. Given a finite trace β of $\mathcal{A}$, C_α is the set of all traces that extend β.

Definition A.1. *If $\mu_1, \mu_2 \in \mathsf{Disc}(\mathsf{Frags}_\mathcal{A})$, such that for every $\alpha \in \mathsf{Frags}^*_\mathcal{A}$, $\mu_1(C_\alpha) \leq \mu_2(C_\alpha)$, then we write $\mu_1 \leq \mu_2$.*

Definition A.2. *A* chain *of probability measures on execution fragments of $\mathcal{A}$ is an infinite sequence $\mu_1, \mu_2, \ldots$ of probability measures on execution fragments of $\mathcal{A}$ such that $\mu_1 \leq \mu_2 \ldots$. Given a chain, the* limit *of the chain is defined as a function μ on the σ-algebra generated by the cones of execution fragments of $\mathcal{A}$, as follows: for each $\alpha \in \mathsf{Frags}^*_\mathcal{A}$, $\mu(C_\alpha) := \lim_{i\to\infty} \mu_i(C_\alpha)$.*

Standard measure theoretic arguments guarantee that μ can be extended uniquely to a probability measure on the σ-field generated by the cones of finite execution fragments.

Definition A.3. *If μ_1, μ_2 are probability measures on traces of $\mathcal{A}$, such that for every finite trace β of $\mathcal{A}$ $\mu_1(C_\beta) \leq \mu_2(C_\beta)$, then we write $\mu_1 \leq \mu_2$.*

Definition A.4. *A* chain *of probability measures on traces of $\mathcal{A}$ is an infinite sequence $\mu_1, \mu_2, \ldots$ of probability measures on traces of $\mathcal{A}$ such that $\mu_1 \leq \mu_2 \ldots$. Given a chain of probability measure on traces, the* limit *of the chain is defined as a function μ on the σ-algebra generated by the cones of traces of $\mathcal{A}$, as follows: for each finite trace β of $\mathcal{A}$, $\mu(C_\beta) := \lim_{i\to\infty} \mu_i(C_\beta)$.*

Again, μ can be extended uniquely to a probability measure on the σ-field generated by the cones of finite traces.

Lemma A.5 (4 of [5])**.** *Let $\mu_1, \mu_2, \ldots$ be a chain of measures on $\mathsf{Frags}_\mathcal{A}$ and let $\mu = \lim_{i\to\infty} \mu_i$, then $\lim_{i\to\infty} \mathsf{tdist}(\mu_i) = \mathsf{tdist}(\mu)$.*

Lemma A.6 (11 of [5])**.** *Let $\mu \in \mathsf{Disc}(\mathsf{Frags}^*_\mathcal{A})$ and σ be a finite task schedule for $\mathcal{A}$. Then $\mathsf{apply}(\mu, \sigma) \in \mathsf{Disc}(\mathsf{Frags}^*_\mathcal{A})$.*

Lemma A.7 (20 of [5])**.** *Let $\mu \in \mathsf{Disc}(\mathsf{Frags}^*_\mathcal{A})$ and $\sigma_1, \sigma_2, \ldots$ be a finite or infinite sequence of task schedulers for $\mathcal{A}$. For each $i > 0$ let $\eta_i = \mathsf{apply}(\mu, \sigma_1\sigma_2 \ldots \sigma_i)$. Let $\sigma = \sigma_1\sigma_2 \ldots$ be the concatenation of the all the task schedulers, and let $\eta = \mathsf{apply}(\mu, \sigma)$. Then the η_i's form a chain and $\eta = \lim_{i\to\infty} \eta_i$.*

B Appendix: Lemmas for Approximate Simulations

This Appendix provides proofs of several propositions stated in the paper and also some auxiliary lemmas used for proving the soundness theorem.

The following is a proof of Proposition 3.2.

Lemma B.1. *Let $\{\mu_i\}_{i\in I}$ be a countable family of discrete probability measures $\mu_i \in \mathsf{Disc}(\mathsf{Frags}^*_\mathcal{A})$ and let $\mu = \sum_{i\in I} \lambda_i\mu_i$ be a convex combination of $\{\mu_i\}$, where $\sum_{i\in I} \lambda_i = 1$. Let T be task of $\mathcal{A}$. Then $\mathsf{apply}(\mu, T) = \sum_{i\in I} \lambda_i \mathsf{apply}(\mu_i, T)$.*

Proof. Suppose p_1 and p_2 are the functions used in the definition of $\mathsf{apply}(\mu, T)$,

and suppose for each $i \in I$, p_1^i and p_2^i be the functions used in the definition of $\mathsf{apply}(\mu_i, T)$. Fix a finite execution fragment α. We show that $p_1(\alpha) = \sum_i \lambda_i p_1^i(\alpha)$ and $p_2(\alpha) = \sum_i \lambda_i p_2^i(\alpha)$, from which it follows that $\mathsf{apply}(\mu, T)(\alpha) = p_1(\alpha) + p_2(\alpha) = \sum_i \lambda_i (p_1^i(\alpha) + p_3^i(\alpha)) = \sum_i \lambda_i \mathsf{apply}(\mu_i, T)$.

To prove that $p_1(\alpha) = \sum_i \lambda_i p_1^i(\alpha)$, we consider two cases. If $\alpha = \alpha' a q$ where $\alpha' \in \mathsf{supp}(\mu)$, $a \in T$, and $(\alpha'.lstate, a, \eta) \in \mathcal{D}$, then, by Definition 2.2 $p_1(\alpha) = \mu(\alpha')\eta(q)$ and for each $i \in I$, $p_1^i(\alpha) = \mu_i(\alpha')\eta(q)$. Thus, $p_1(\alpha) = \sum_i \lambda_i p_1^i(\alpha)$. Otherwise, again by Definition 2.2 $p_1(\alpha) = 0$ and for each $i \in I$, $p_1^i(\alpha) = 0$, and the result holds trivially.

To prove that $p_2(\alpha) = \sum_i \lambda_i p_2^i(\alpha)$, we consider two cases. If T is not enabled in $\alpha.lstate$ then, by Definition 2.2, $p_2(\alpha) = \mu(\alpha)$, and for each $i \in I$, $p_2^i(\alpha) = \mu_i(\alpha)$. Thus, $p_2(\alpha) = \sum_i \lambda_i p_2^i(\alpha)$. Otherwise, again by Definition 2.2 $p_2(\alpha) = 0$ and for each $i \in I$, $p_2^i(\alpha) = 0$, and the result holds trivially. □

Proposition B.2. *Let $\{\mu_i\}_{i \in I}$ be a countable family of discrete probability measures $\mu_i \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}})$ and let $\mu = \sum_{i \in I} \lambda_i \mu_i$ be a convex combination of $\{\mu_i\}$, where $\sum_{i \in I} \lambda_i = 1$. Let σ be a finite sequence of tasks. Then $\mathsf{apply}(\mu, \sigma) = \sum_{i \in I} \lambda_i \mathsf{apply}(\mu_i, \sigma)$.*

Proof. The proof is by induction on the length of σ. If σ is the empty sequence, then for any $\eta \in \mathsf{Disc}(\mathsf{Frags}^*_A)$, $\mathsf{apply}(\eta, \sigma) = \eta$ and it follows that $\mu = \sum_{i \in I} \lambda_i \mu_i = \sum_{i \in I} \lambda_i \mathsf{apply}(\mu_i, \sigma)$. For the induction step, let $\sigma = \sigma' T$. By Definition 2.2, $\mathsf{apply}(\mu, \sigma' T) = \mathsf{apply}(\mathsf{apply}(\mu, \sigma'), T)$. By the induction hypothesis, $\mathsf{apply}(\mu, \sigma') = \sum_i \lambda_i \mathsf{apply}(\mu_i, \sigma')$ and thus, $\mathsf{apply}(\mu, \sigma' T) = \mathsf{apply}(\sum_i \lambda_i \mathsf{apply}(\mu_i, \sigma'), T)$. For each $i \in I$, $\mathsf{apply}(\mu_i, \sigma')$ is a discrete probability measure in $\mathsf{Disc}(\mathsf{Frags}^*_A)$. By Lemma B.1, $\mathsf{apply}(\sum_i \lambda_i \mathsf{apply}(\mu_i, \sigma'), T) = \sum_i \lambda_i \mathsf{apply}(\mathsf{apply}(\mu_i, \sigma'), T)$. Using Definition 2.2 it follows that $\mathsf{apply}(\mu, \sigma' T) = \sum_i \lambda_i \mathsf{apply}(\mu_i, \sigma' T)$ as required. □