Approximate Simulations for Task-Structured Probabilistic I/O Automata

Sayan Mitra [1] Nancy Lynch [2]

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

Abstract

A *Probabilistic I/O Automaton (PIOA)* is a countable-state automaton model that allows nondeterministic and probabilistic choices in state transitions. A *task-PIOA* adds a task structure on the locally controlled actions of a PIOA as a means for restricting the nondeterminism in the model. The task-PIOA framework defines exact implementation relations based on inclusion of sets of trace distributions. In this paper we develop the theory of approximate implementations and equivalences for task-PIOAs. We propose a new kind of approximate simulation between task-PIOAs and prove that it is sound with respect to approximate implementations. Our notion of similarity of traces is based on a metric on trace distributions and therefore, we do not require the state spaces nor the space of external actions (output alphabet) of the underlying automata to be metric spaces. We discuss applications of approximate implementations to probabilistic safety verification.

Key words: Approximate equivalence, Approximate simulation, Abstraction, Probabilistic I/O Automata.

1 Introduction

An automaton $\mathcal{A}$ is said to *implement* a second automaton $\mathcal{B}$ if every observable behavior or *trace* of $\mathcal{A}$ is also a trace of $\mathcal{B}$. $\mathcal{A}$ and $\mathcal{B}$ are said to be *equivalent* if they implement each other. Implementation and equivalence relations, also sometimes called simulation and bisimulation, play fundamental roles in the study of complex interacting systems. Many different kinds of implementation relations and their corresponding proof methods have been developed for timed [1], hybrid [14,22,21] and probabilistic automata [15,16,4,3,20,2]. All the above notions of implementation rely on exact equality of traces. It is well known from [12,6,11] that such strict equality based implementation relations are not robust. The problem is particularly acute where the traces contain information about real valued variables and probabilities. For example, consider the probabilistic automata in Figure 1. Clearly, $\mathcal{B}$ ought to be closer to

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

$\mathcal{A}$ than $\mathcal{C}$ is to $\mathcal{A}$. With exact notions of equivalence, all one could say is that no two of $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ are equivalent.

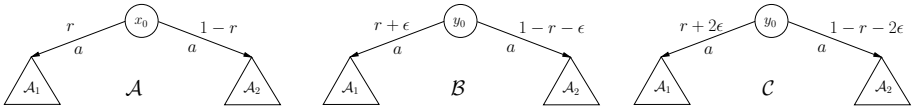

Fig. 1. Inequivalent automata with similar transition probabilities (adapted from [11]).

One way to fix this problem is to relax the notion of equivalence (or implementation) by taking into consideration the "similarity" of traces that are not exactly identical. Based on this idea there is a growing body of work on developing robust notions of "approximate" equivalences. In [12], "similarity" of traces is formalized using a metric and the corresponding notion of approximate equivalence is developed for probabilistic automata. More recently, approximate bisimilarity of hybrid systems [10,9], labelled Markov processes [7,23,24], generalized semi-Markov processes [11], and linear stochastic hybrid automata [13] have been studied.

In this paper we develop the theory of approximate implementations and equivalences for *task-structured Probabilistic I/O Automata (PIOA)*. A task-PIOA is a nondeterministic automaton with a countable set of states. 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. In [4] a task structure—an equivalence relation on the set of locally controlled actions—is used as a means for restricting the nondeterminism in the model. In order to obtain a probability distribution over the executions of a task-PIOA, the automaton is combined with a "scheduler" for resolving nondeterminism. Visible behavior of a task-PIOA combined with a scheduler is a *trace distribution* which is a probability distribution over its set of traces. Exact implementation relations and compositionality results for task-PIOAs are presented in [4]. A special kind of approximate implementation relation that tolerates small differences in the probability of occurrence of a particular action is used in [5] to verify a security protocol. In contrast, the notion of approximation introduced here is based on the differences in the probabilities of all the possible traces of the automata in question.

This work differs from all the approximate simulation related studies cited above in at least one of the following ways. (i) The implementation relation in the task-PIOA framework is based on trace distributions and not bisimilarity of states. Approximate implementation is derived from a metric over trace distributions. That is, we do not require the state spaces of the underlying automata nor the common space of external actions (output alphabet) to be metric spaces. (ii) The task-PIOA model allows both nondeterministic and probabilistic choices. In this setting, the external behavior of an automaton is the set of all possible trace distributions that may arise from combining the au-

tomaton with any scheduler. Thus, approximate simulations prove "nearness" of sets of trace distributions.

Our notion of approximate simulation is a natural extension of (exact) simulation relation for task-PIOAs. Let μ_1 and μ_2 be probability distributions over executions of task-PIOAs $\mathcal{A}$ and $\mathcal{B}$. An approximate simulation from $\mathcal{A}$ to $\mathcal{B}$ is a function ϕ mapping each μ_1, μ_2 pair to a nonnegative 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 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.

In the next Section, we give a condensed overview of the task-PIOA framework. In Section 3 we propose approximate implementations for closed task-PIOAs. We introduce approximate simulations and show that they are sound for proving approximate implementations. In Section 4, we discuss applications of approximate simulations to probabilistic safety verification and briefly outline how the results of Section 3 extend to general (not necessarily closed) task-PIOAs. In Section 5 we remark on some directions for future research. Proofs of auxiliary lemmas and formal statements of some relevant results from [4] 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 (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 [4]; here 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 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* action alphabet *of* $\mathcal{P}$*. If* $I = \emptyset$*, then* $\mathcal{A}$ *is* closed*. The set of* external *actions of* $\mathcal{A}$ *is* $I \cup O$ *and the set of* locally controlled *actions is* $O \cup H$*. (iv)* $D \subseteq (Q \times (I \cup O \cup H) \times \mathsf{Disc}(Q))$ *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 $\mathcal{P}$ 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 [4].

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 [5]. 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* $\alpha' aq$*, 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. 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 $\mathcal{A}\}$.

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 [4] 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 2.3. *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 2.4. *Let $\mathcal{A}_1, \mathcal{A}_2$ be task-PIOAs, and let $\mathsf{c} : ({R_1}^* \times R_1) \rightarrow {R_2}^*$ 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 $\mathsf{full}(\mathsf{c}) : {R_1}^* \rightarrow {R_2}^*$ 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 [5]).

Definition 2.5. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two comparable closed task-PIOAs. Let R be a relation from $\mathsf{Disc}(\mathsf{Execs}^*(\mathcal{A}_1))$ to $\mathsf{Disc}(\mathsf{Execs}^*(\mathcal{A}_2))$, such that, if $\mu_1\ R\ \mu_2$, then $\mathsf{tdist}(\mu_1) = \mathsf{tdist}(\mu_2)$. Then R is a simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$ if there exists $\mathsf{c} : ({R_1}^* \times R_1) \rightarrow {R_2}^*$ such that following properties hold:*

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

(ii) **Step condition:** *If $\mu_1\ R\ \mu_2$, $\sigma_1 \in {R_1}^*$, μ_1 is consistent with σ_1, μ_2 is consistent with $\mathsf{full}(\mathsf{c})(\sigma_1)$, and $T \in R_1$, then $\mu_1'\ \mathcal{E}(R)\ \mu_2'$ where $\mu_1' = \mathsf{apply}(\mu_1, T)$ and $\mu_2' = \mathsf{apply}(\mu_2, \mathsf{c}(\sigma_1, T))$.*

We close this section with the statement of the soundness theorem for the above simulation relation which has been proved in [4].

Theorem 2.6. *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 Approximate Implementations and Simulations

In this section we develop the theory of approximate implementations and equivalences for task-PIOAs. We define approximate implementation relations for closed task-PIOAs and propose a new kind of approximate simulation. In the next Section we discuss how these results carry over to general (not necessarily closed) task-PIOAs.

Informally, a task-PIOA $\mathcal{A}_1$ 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 some metric on trace distributions. Metrics for probability distributions have been a subject of intense research in probability theory (see, for example, the books [18] and [8]). There are different choices of metrics over trace distributions. It turns out that our definition of approximate simulations and the proof of its soundness weakly relies on the choice of this metric. In fact, any metric satisfying Proposition 3.2 is suitable for our purpose and in practice this choice would be guided by the automata under consideration. In this paper we work with the following (uniform) metric.

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* $\mathbf{d}_u : \mathsf{Disc}(\mathsf{Traces}_{\mathcal{A}}) \times \mathsf{Disc}(\mathsf{Traces}_{\mathcal{A}}) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ *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, however, this distinction is not significant in our discourse and with some abuse of terminology we will refer to $\mathbf{d}_u$ as a metric.

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 $\lim_{j\to\infty} \mathbf{d}_u(\mu_{1j}, \mu_{2j}) = \mathbf{d}_u(\mu_1, \mu_2)$.*

Approximate implementation for task-PIOAs is defined based on the metric $\mathbf{d}_u$ on trace distributions.

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 $\mathcal{A}_2 \leq_\delta \mathcal{A}_1$.*

Thus, $\mathcal{A}_1$ δ-implements $\mathcal{A}_2$ if the one-sided Hausdorff distance from $\mathsf{tdists}(\mathcal{A}_1)$ to $\mathsf{tdists}(\mathcal{A}_2)$ is less than or equal to δ.

3.1 Definition of Approximate Simulation

In this section we define approximate simulations for task-PIOAs and we prove its soundness with respect to δ-implementations.

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.

Analogous to expansion of relations as defined in Section 2.1, our definition of approximate simulation uses on the following notion of *expansion* of a function.

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

(i) ψ *minimizes* $\max_{x,y\in\mathsf{supp}(\psi)}\phi(x,y)$ *and* $\max_{x,y\in\mathsf{supp}(\psi)}\phi(x,y) = \epsilon$,

(ii) $x_1 = \sum_{x,y\in\mathsf{supp}(\psi)}\psi(x,y)x$, *and*

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

The consistency requirements *(ii)* and *(iii)* 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 *(ii)* and *(iii)*. If ψ is feasible for x_1, y_1 and it satisfies *(i)* with $\max_{x,y\in\mathsf{supp}(\psi)}\phi(x,y) = \epsilon$, 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 3.5.

Proposition 3.6. *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 2 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 $\phi : \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_1})\times\mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_2}) \to \mathbb{R}_{\geq 0}\cup\{\infty\}$ 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 $\hat{\phi}(\mu_1', \mu_2') \leq \epsilon$.

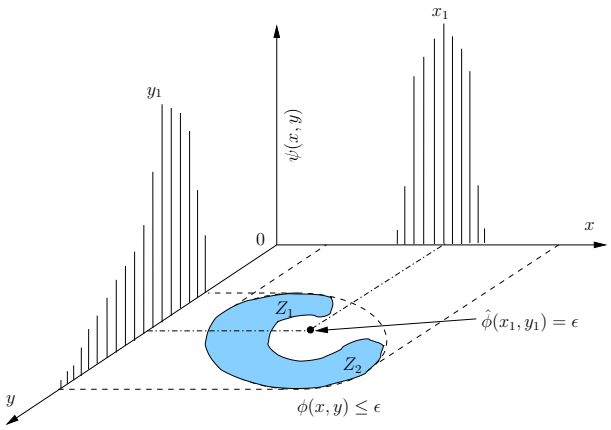

Fig. 2. Marginal distributions of the optimal joint distribution ψ for $\hat{\phi}(x_1, y_1) = \epsilon$. Support of ψ is contained within the elliptical region. In particular, ψ is concentrated in the regions Z_1 and Z_2 each carrying half of the total mass.

Definition 3.7. *Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are two comparable closed task-PIOAs, ϵ is a nonnegative constant, and ϕ is a function* $\mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_1}) \times \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}_2}) \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$. *Suppose further, that there exists $\delta > 0$ such that if $\phi(\mu_1, \mu_2) \leq \epsilon$ then $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) \leq \delta$. The function ϕ is an (ϵ, δ)-*approximate simulation *from $\mathcal{A}_1$ to $\mathcal{A}_2$ if it satisfies the following conditions:*

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

(ii) **Step condition:** *There exists a function* $\mathsf{c} : R_1^* \times R_1 \rightarrow R_2^*$ *such that, if $\phi(\mu_1, \mu_2) \leq \epsilon$, $T_1 \in R_1, \sigma_1 \in R_1^*$ and μ_1 is consistent with σ_1, and μ_2 is consistent with* $\mathsf{full}(\mathsf{c})(\sigma_1)$, *then* $\hat{\phi}(\mathsf{apply}(\mu_1, T), \mathsf{apply}(\mu_1, c(\sigma, T))) \leq \epsilon$.

The above definition of (ϵ, δ)-approximate simulation generalizes [3] (exact) simulation relation of Section 2.1.

3.2 Soundness of Approximate Simulation

In this section we prove Theorem 3.10 which is the main result of this paper and it states that (ϵ, δ)- approximate simulations are sound with respect to δ-approximate implementations. First we prove two key lemmas used in the proof of the theorem.

Lemma 3.8. *Suppose ϕ is a (ϵ, δ)-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

[3] We claim that the following relation between approximate simulations and exact simulation relations exists. Let ϕ be an $(\epsilon, 0)$-approximate simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$. Let us define R:= $\{(\mu_1, \mu_2) \mid \phi(\mu_1, \mu_2) \leq \epsilon\}$. R is a simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$. A proof of this claim will be given in the full version of the paper.

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 it follows that $\mathsf{tdist}(\mu_i) = \sum_{\eta_1,\eta_2 \in \mathsf{supp}(\psi)} \psi(\eta_1, \eta_2)\, \mathsf{tdist}(\eta_i)$. $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) = \sup_{C \in \mathcal{F}^*_{\mathsf{Traces}_{\mathcal{A}}}} |\, \mathsf{tdist}(\mu_1)(C) - \mathsf{tdist}(\mu_2)(C)|$

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

For any $\eta_1, \eta_2 \in \mathsf{supp}(\psi)$, $\phi(\eta_1, \eta_2) \leq \epsilon$ and since ϕ is an (ϵ, δ)-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 3.9. *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:

$$
\psi' := \sum_{(\rho_1,\rho_2) \in \mathsf{supp}(\psi)} \psi(\rho_1, \rho_2)\psi_{\rho_1,\rho_2} \tag{1}
$$

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:

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

(ii) For optimality of ψ' it suffices to show that for all $\eta_1, \eta_2 \in \mathsf{supp}(\psi')$, $\phi(\eta_1, \eta_2) \leq \epsilon$. If $\psi'(\eta_1, \eta_2) > 0$ then from Equation (1) 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 3.10. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two closed comparable task-PIOAs. If there exists a (ϵ, δ)-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)-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 3.6 $\hat{\phi}(\mu_{1,0}, \mu_{2,0}) \leq \epsilon$.

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 3.9, 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 3.9. 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 3.9.

Step 2. From Lemma 3.8, 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 $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) = \lim_{j\to\infty} \mathbf{d}_u(\mathsf{tdist}(\mu_{1,j}), \mathsf{tdist}(\mu_{2,j})) \leq \delta$. □

4 Applications of Approximate Implementations

4.1 A Simple Example

As a quick review of all the concepts introduced in this paper we return to a simple version of the example of Section 1. We assume that: $\mathcal{A}_1$ is replaced by single states x_1 and y_1 in the automata $\mathcal{A}$ and $\mathcal{B}$, and likewise $\mathcal{A}_2$ is replaced by x_2 and y_2. Of course, for this simple example we can directly prove

exact equivalence of $\mathcal{A}$ and $\mathcal{B}$. As an exercise, we will show that $\mathcal{A}$ is indeed approximately equivalent to $\mathcal{B}$. Formally, we will first present a simple $(\epsilon, 2\epsilon)$-approximate simulation from $\mathcal{A}$ to $\mathcal{B}$, which would imply, by Theorem 3.10, that $\mathcal{A} \leq_{2\epsilon} \mathcal{B}$. Likewise, we can show that $\mathcal{B} \leq_{2\epsilon} \mathcal{A}$, from which it follows that $\mathcal{A} \cong_{2\epsilon} \mathcal{B}$.

The set of execution fragments [4] of $\mathcal{A}$ is $\mathsf{Frags}^*_{\mathcal{A}} = \{x_0, x_0ax_1, x_0ax_2\}$. Likewise, $\mathsf{Frags}^*_{\mathcal{B}} = \{y_0, y_0ay_1, y_0ay_2\}$. A distribution μ over $\mathsf{Frags}^*_{\mathcal{A}}$ (resp. $\mathsf{Frags}^*_{\mathcal{B}}$) is a triple in which the i^{th} term $\mu[i]$ gives the probability of the i^{th} fragment in the set $\mathsf{Frags}^*_{\mathcal{A}}$ (resp. $\mathsf{Frags}^*_{\mathcal{B}}$). Let the starting distribution $\bar{\nu}_1$ of $\mathcal{A}$ be defined by $\bar{\nu}_1(x_i) = p_i$, where $\sum_{i=0}^{2} p_i = 1$. Let the starting distribution $\bar{\nu}_2$ of $\mathcal{B}$ be defined by $\bar{\nu}_2(y_0) = p_i + \epsilon_1$, $\bar{\nu}_2(y_1) = p_1$, and $\bar{\nu}_2(y_2) = p_2 - \epsilon_1$, where $0 \leq \epsilon_1 \leq \frac{\epsilon}{2}$. Our choice of the approximate simulation function ϕ is the $\mathcal{L}^1$ distance between distributions, that is, $\phi(\mu_1, \mu_2) := \sum_{i=0}^{2} |\mu_1[i] - \mu_2[i]|$, where $\mu_1 \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}})$ and $\mu_2 \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{B}})$.

Claim. *If ϵ and r satisfy Equations (B.1)-(B.4), then ϕ is an $(\epsilon, 2\epsilon)$-approximate simulation from $\mathcal{A}$ to $\mathcal{B}$.*

Proof. Let $\mu_1 \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{A}})$ and $\mu_2 \in \mathsf{Disc}(\mathsf{Frags}^*_{\mathcal{B}})$. It is easy to check that if $\phi(\mu_1, \mu_2) \leq \epsilon$, then $\mathbf{d}_u(\mathsf{tdist}(\mu_1), \mathsf{tdist}(\mu_2)) \leq 2\epsilon$. The function ϕ satisfies the start condition by our assumption that $\epsilon_1 \leq \frac{\epsilon}{2}$. Now we check the step condition. Note that the only action for $\mathcal{A}$ is a and the corresponding action for $\mathcal{B}$ is also a. Let $\mu_1' = \mathsf{apply}(\bar{\nu}_1, a)$ and $\mu_2' = \mathsf{apply}(\bar{\nu}_2, a)$. From Definition 2.2, $\mu_1' = \langle 0, p_1 r + p_2, p_1(1-r) + p_3 \rangle$, and $\mu_2' = \langle 0, (p_1+\epsilon_1)(r+\epsilon) + p_2, (p_1+\epsilon_1)(1-r-\epsilon) + (p_3 - \epsilon_1) \rangle$. To prove that $\hat{\phi}(\mu_1', \mu_2') \leq \epsilon$, we construct a feasible joint distribution ψ for μ_1' and μ_2'. Our strategy is to equally distribute ϵ in each of the dimensions of the domain of the distribution ψ. We define:

$$\psi(0, \mu_1'[2] \pm \frac{\epsilon}{2}, \mu_1'[3], 0, \mu_2'[2], \mu_2'[3]) = \psi(0, \mu_1'[2], \mu_1'[3] \pm \frac{\epsilon}{2}, 0, \mu_2'[2], \mu_2'[3]) = 1/8$$

$$\psi(0, \mu_1'[2], \mu_1'[3], 0, \mu_2'[2] \pm \frac{\epsilon}{2}, \mu_2'[3]) = \psi(0, \mu_1'[2], \mu_1'[3], 0, \mu_2'[2], \mu_2'[3] \pm \frac{\epsilon}{2}) = 1/8.$$

It can be checked easily that for any two points η_1, η_2 in the support of ψ, $\phi(\eta_1, \eta_2) \leq \epsilon$. The feasibility of ψ follows from Equations (B.1)-(B.4). □

4.2 Probabilistic Safety

Given two δ-equivalent closed task-PIOAs $\mathcal{A}_1$ and $\mathcal{A}_2$, if we know that $\mathcal{A}_1$ violates some safety S property 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.

As the two automata are comparable their trace spaces are identical. Let $(\mathsf{Traces}, \mathcal{F}_{\mathsf{Traces}})$ be the measurable space of traces for both $\mathcal{A}_1$ and $\mathcal{A}_2$. Let

[4] The execution fragments x_1 and x_2 are not included because they play no role in the analysis.

$(X, \mathcal{F}_X)$ be another measurable space. A random variable is a measurable function $\mathbf{X} : (\mathsf{Traces}, \mathcal{F}_{\mathsf{Traces}}) \rightarrow (X, \mathcal{F}_X)$. We use the standard notation $\mu[\mathbf{X} = x] := \mu(\{\beta \in \mathsf{Traces} \mid \mathbf{X}(\beta) = x\}$, for $x \in X$.

Proposition 4.1. *Let* $\mathbf{X}$ *be random variable on* $(\mathsf{Traces}, \mathcal{F}_{\mathsf{Traces}})$. *Suppose* $\mathcal{A}_1 \cong_\delta \mathcal{A}_1$ *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 $\mathcal{A}_1 \cong_\delta \mathcal{A}_2$ 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 variables 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 $\mathbf{X}_U : \mathsf{Traces} \rightarrow \{0, 1\}$ 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 $\mathcal{A}_1 \cong_\delta \mathcal{A}_2$, then from Proposition 4.1 we can conclude that in any trace distribution of $\mathcal{A}_2$ the probability of occurrence of U is at most $\delta + p$.

4.3 Approximations for Task-PIOAs

We briefly discuss how the results presented in this paper can be extended to general (not necessarily closed) task-PIOAs. The basic idea is to define a new notion of implementation following the approach of [4].

An "environment" for task-PIOA $\mathcal{A}$ is a task-PIOA $\mathcal{E}$ such that the *composition* [5] of $\mathcal{A}$ and $\mathcal{E}$ is closed. 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 as follows:

Definition 4.2. *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 approximate simulations for general task-PIOAs follow as a Corollary

[5] The composition operation for task-PIOAs is formally defined in [4]. Informally, the composition of $\mathcal{A}$ and $\mathcal{E}$ is another task-PIOA that combines the transitions corresponding to the common external actions of $\mathcal{A}$ and $\mathcal{E}$ in a consistent manner.

to Theorem 3.10.

Corollary 4.3. *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 simulation 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 Conclusions

In this paper we have proposed approximate simulations for task-structured probabilistic I/O automata and we have proved that approximate simulations are sound for proving approximate implementation relations. Approximate implementation relations are defined based on a metric over trace distributions. As a result, we do not require the underlying state spaces of the automata or the space of external actions of the automata to be metric spaces.

In the future, we want to extend the notion of approximate simulations to the task-PIOA model with continuous state spaces [17]. We also want to extend simulations to be defined as functions of distributions over states as opposed to distributions over execution fragments. In our formulation of approximate simulations, a simulation proof boils down to finding an optimal joint distribution satisfying certain constrains. For well-behaved classes of simulation functions this opens up the possibility of proving approximate simulations by solving optimization problems.

Acknowledgments
We thank Dilsun Kaynar for taking the time to explain some of the results in [4] and for commenting on the results in this paper.

References

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

[2] M. L. Bujorianu, J. Lygeros, and M. C. Bujorianu. Bisimulation for general stochastic hybrid systems. In *HSCC 2005*, pages 198–214, volume 3414 of LNCS, Springer-Verlag, 2005.

[3] C. Baier. Polynomial-time algorithms for testing probabilistic bisimulation and simulation. In R. Alur and T. A. Henzinger, editors, *CAV'96*, volume 1102, pages 50–61, New Brunswick, NJ, USA, / 1996. Springer Verlag.

[4] 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-2006-023, MIT, Cambridge, MA, March 2006.

[5] 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. Tech Report MIT-CSAIL-TR-2006-019, MIT, Cambridge, MA, March 2006.

[6] J. Desharnais, R. Jagadeesan, V. Gupta, and P. Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *LICS 2002*, pages 413–422. IEEE Computer Society, 2002.

[7] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004.

[8] 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.

[9] 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.

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

[11] 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.

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

[13] A. A. Julius. Approximate abstraction of stochastic hybrid automata. In João P. Hespanha and Ashish Tiwari, editors, *HSCC06*, pages 318–332, volume 3927 of LNCS, Springer-Verlag, 2006.

[14] 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.

[15] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information Computation*, 94(1):1–28, 1991.

[16] 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.

[17] S. Mitra and N. Lynch. Probabilistic timed I/O automata with continuous state spaces, April 2006. Submitted for review. Available from `http://theory.lcs.mit.edu/~mitras/research/CONCUR06_06.pdf`.

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

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

[20] S. Strubbe and A. J. van der Schaft. Bisimulation for communicating piecewise deterministic markov processes (cpdps). In *HSCC 2005*, pages 623–639, volume 3414 of LNCS, Springer-Verlag, 2005.

[21] P. Tabuada, G. J. Pappas, and P. U. Lima. Composing abstractions of hybrid systems. In *HSCC 2002*, volume 2289 of LNCS, pages 436–450, Springer-Verlag.

[22] A. Tiwari and G. Khanna. Series of abstractions for hybrid automata. In *HSCC 2002*, volume 2289 LNCS, pages 465–478, Springer-Verlag.

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

[24] F. van Breugel and J. Worrell. Towards quantitative verification of probabilistic transition systems. In *ICALP '01*, 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 [4]. 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 [4]). *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 [4]). *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 [4]). *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.

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. □

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. □

B.1 Simple Example of Section 4.1

The following conditions are used in the hypothesis of the Claim in Section 4.1.

$$\frac{1}{p_1}\left[\frac{\epsilon_1}{2} - p_2\right] \leq \quad r \quad \leq \frac{1}{p_1}\left[\frac{1-\epsilon_1}{2} - p_2\right] \tag{B.1}$$

$$\frac{1}{p_1}\left[\frac{\epsilon_1}{2} - p_3\right] \leq \quad 1-r \quad \leq \frac{1}{p_1}\left[\frac{1-\epsilon_1}{2} - p_3\right] \tag{B.2}$$

$$\frac{1}{p_1+\epsilon_1}\left[\frac{\epsilon_1}{2} - p_2\right] \leq \quad r+\epsilon \quad \leq \frac{1}{p_1+\epsilon_1}\left[\frac{1-\epsilon}{2} - p_2\right] \tag{B.3}$$

$$\frac{1}{p_1+\epsilon_1}\left[\frac{3\epsilon_1}{2} - p_3\right] \leq 1-(r+\epsilon) \leq \frac{1}{p_1+\epsilon_1}\left[\frac{1+\epsilon}{2} - p_3\right] \tag{B.4}$$