proving approximate implementations

sayan mitra
mitras@ist.caltech.edu
postdoctoral scholar

CENTER FOR MATHEMATICS OF INFORMATION
CALIFORNIA INSTITUTE OF TECHNOLOGY

CMI retreat 2007
implementation

B

send

[3, 10]

done

fail

done

nack

ack

nack

tx
implementation

B: 

A: send

send

[3, 10]

fail
done

send

A: 

B: 

[3, 10]

fail
done

send
implementation

B

send → [3, 10] → done

A

send → tx → [3, 4] → nack → ack → done
implementation

- **B**
  - send
  - [3, 10]
  - done

- **A**
  - send
  - tx
  - [3, 4]
  - nack
  - tx
  - [3, 4]
  - nack
  - ack
  - done
implementation

\[ Traces_B = \{\text{send } [3, 10] (\text{done}|\text{fail})\} \]
T races_{B} = \{send [3, 10] (done|fail)\}

T races_{A} = \{send [3, 4] done\} \cup \{send [6, 8] (done|fail)\}
Traces_B = \{send [3, 10] (done|fail)\}

Traces_A = \{send [3, 4] \text{done}\} \cup \{send [6, 8] (done|fail)\}

does A implement B?
implementation

\[ \text{Traces}_B = \{ \text{send} [3, 10] \text{ (done|fail)} \} \]

\[ \text{Traces}_A = \{ \text{send} [3, 4] \text{ done} \} \cup \{ \text{send} [6, 8] \text{ (done|fail)} \} \]

does \( A \) implement \( B \)?

\[ \text{Traces}_A \subseteq \text{Traces}_B? \]
existence of simulation relation $\mathcal{R} \subseteq Q_A \times Q_B$ implies $\text{Traces}_A \subseteq \text{Traces}_B$
approximate implementations

exact implementation
  - based on equality of traces

\[ \delta \text{-implements } B \text{ if for every trace of } A \text{ there is a trace of } B \text{ within } \delta \]

forces nondeterminism in deterministic, quantified abstractions
approximate implementations

exact implementation

- based on equality of traces
- not robust with respect to perturbations
approximate implementations

exact implementation

- based on equality of traces
- not robust with respect to perturbations
- forces nondeterminism in abstractions
approximate implementations

exact implementation

► based on equality of traces
► not robust with respect to perturbations
► forces nondeterminism in abstractions

approximate implementation

► based on similarity of traces
approximate implementations

exact implementation
- based on equality of traces
- not robust with respect to perturbations
- forces non-determinism in abstractions

approximate implementation
- based on similarity of traces
- $A \delta$-implements $B$ if for every trace of $A$ there is a trace of $B$ within $\delta$
approximate implementations

**exact implementation**
- based on equality of traces
- not robust with respect to perturbations
- forces nondeterminism in abstractions

**approximate implementation**
- based on similarity of traces
- $A \delta$-implements $B$ if for every trace of $A$ there is a trace of $B$ within $\delta$
- deterministic, quantified abstractions
probabilistic automata

captures probabilistic and nondeterministic uncertainties
probabilistic automata
captures probabilistic and nondeterministic uncertainties

\[ A = (Q, \nu, Act, D) \]
- countable set of states \( Q \)
- initial distribution on states \( \nu \)
- countable set of actions \( Act = Ext \cup Int \)
- set of \( (q, a, \mu) \) probabilistic transitions \( D \)
probabilistic automata

captures probabilistic and nondeterministic uncertainties

\[ A = (Q, \nu, \text{Act}, D) \]

- countable set of states \( Q \)
- initial distribution on states \( \nu \)
- countable set of actions \( \text{Act} = \text{Ext} \cup \text{Int} \)

set of \((q, a, \mu)\) probabilistic transitions \( D \)

assumption
given a state \( q \) and an action \( a \) there is at most one transition \( q \xrightarrow{a} \mu \)
resolution of nondeterminism

- a schedule $\rho$ is a sequence of actions
- $\rho$ defines a probability distribution $\mu_\rho \in \mathbb{P}(\text{Paths}_A)$
trace distributions

for $\rho = \perp$

$\mu_{\rho}(1) = 1$
trace distributions

for $\rho = b$

$\mu_\rho(1) = 1$
trace distributions

for $\rho = b, a$

- $\mu_\rho(1 \ a \ 2) = \frac{1}{2}$
- $\mu_\rho(1 \ a \ 3) = \frac{1}{2}$
trace distributions

for $\rho = b, a, b$

- $\mu_\rho(1 \ a \ 2) = \frac{1}{2}$
- $\mu_\rho(1 \ a \ 3 \ b \ 2) = \frac{1}{6}$
- $\mu_\rho(1 \ a \ 3 \ b \ 3) = \frac{1}{3}$
trace distributions

for $\rho = b, a, b, c$

- $\mu_\rho(1 \ a \ 2 \ c \ 1) = \frac{1}{2}$
- $\mu_\rho(1 \ a \ 3 \ b \ 2 \ c \ 1) = \frac{1}{6}$
- $\mu_\rho(1 \ a \ 3 \ b \ 3) = \frac{1}{3}$
trace distributions

for $\rho = b, a, b, c$

- $\mu_\rho(1 a 2 c 1) = \frac{1}{2}$
- $\mu_\rho(1 a 3 b 2 c 1) = \frac{1}{6}$
- $\mu_\rho(1 a 3 b 3) = \frac{1}{3}$

trace distribution for $\rho$

- $\eta_\rho(a c) = \frac{2}{3}$
- $\eta_\rho(a) = \frac{1}{3}$
exact implementation for probabilistic automata

- a schedule $\rho$ is a sequence of actions
- $\rho$ defines a probability distribution $\mu_\rho \in \mathbb{P}(\text{Paths}_A)$
- $\rho$ defines a probability distribution $\eta_\rho \in \mathbb{P}(\text{Ext}^*)$
- $\text{Tdists}_A$: set of all trace distributions for $A$
exact implementation for probabilistic automata

- a schedule $\rho$ is a sequence of actions
- $\rho$ defines a probability distribution $\mu_\rho \in \mathbb{P}(\text{Paths}_A)$
- $\rho$ defines a probability distribution $\eta_\rho \in \mathbb{P}(\text{Ext}^*)$
- $\text{Tdists}_A$: set of all trace distributions for $A$

Definition

A implements $B$ is $\text{Tdists}_A \subseteq \text{Tdists}_B$. 
fragility of exact implementation
fragility of exact implementation

no schedule $\rho'$ for $B$ gives $\eta'_{\rho}(a \ c) = \frac{2}{3}$ and $\eta'_{\rho}(a) = \frac{1}{3}$
fragility of exact implementation

no schedule $\rho'$ for $B$ gives $\eta'_\rho(a\ c) = \frac{2}{3}$ and $\eta'_\rho(a) = \frac{1}{3}$

$T_{\text{dists}}_B \not\subseteq T_{\text{dists}}_A$
fragility of exact implementation

no schedule $\rho'$ for $B$ gives $\eta'_\rho(a, c) = \frac{2}{3}$ and $\eta'_\rho(a) = \frac{1}{3}$

$T_{dists}_B \not\subseteq T_{dists}_A$

we cannot say anything relating the behavior of $A$ and $B$
metric on trace distributions

Definition
uniform metric on trace distributions, \( \eta_1, \eta_2 \in \mathbb{P}(\text{Ext}^*) \)

\[
d(\eta_1, \eta_2) = \sup_{\beta \in \text{Ext}^*} |\eta_1(\beta) - \eta_2(\beta)|.
\]
metric on trace distributions

Definition
uniform metric on trace distributions, \( \eta_1, \eta_2 \in \mathbb{P}(\text{Ext}^*) \)

\[
d(\eta_1, \eta_2) = \sup_{\beta \in \text{Ext}^*} |\eta_1(\beta) - \eta_2(\beta)|.
\]

Definition
A \( \delta \)-implements \( B \) if for every \( \eta_1 \in \text{Tdists}_A \) there exists \( \eta_2 \in \text{Tdists}_B \) such that \( d(\eta_1, \eta_2) \leq \delta \).
metric on trace distributions

Definition
uniform metric on trace distributions, $\eta_1, \eta_2 \in \mathbb{P}(Ext^*)$

$$d(\eta_1, \eta_2) = \sup_{\beta \in Ext^*} |\eta_1(\beta) - \eta_2(\beta)|.$$ 

Definition
A $\delta$-implements $B$ if for every $\eta_1 \in Tdists_A$ there exists $\eta_2 \in Tdists_B$ such that $d(\eta_1, \eta_2) \leq \delta$.

how to prove $\delta$-implementations?
strong approximate simulations

\[ \phi : \mathbb{P}(Paths_A) \times \mathbb{P}(Paths_B) \rightarrow \mathbb{R}^+ \text{ is an } (\epsilon, \delta)\text{-strong approximate simulation if} \]

\[ \phi (\mu_A^0, \mu_B^0) \leq \epsilon \Rightarrow \phi (\mu_A', \mu_B') \leq \epsilon \]

\[ \text{implies } d(\text{tdist}(\mu_A), \text{tdist}(\mu_B)) \leq \delta. \]
strong approximate simulations

\( \phi : \mathbb{P}(\text{Paths}_A) \times \mathbb{P}(\text{Paths}_B) \to \mathbb{R}^+ \) is an \((\epsilon, \delta)\)-strong approximate simulation if

\[
\text{START } \phi(\mu_A, \mu_B) \leq \epsilon
\]
strong approximate simulations

\[ \phi : \mathbb{P}(Paths_A) \times \mathbb{P}(Paths_B) \rightarrow \mathbb{R}^+ \] is an \((\epsilon, \delta)\)-strong approximate simulation if

\begin{align*}
\text{START} & \quad \phi(\mu_{A0}, \mu_{B0}) \leq \epsilon \\
\text{STEP} & \quad \phi(\mu_A, \mu_B) \leq \epsilon \quad \text{implies} \quad \phi(\mu'_A, \mu'_B) \leq \epsilon
\end{align*}
strong approximate simulations

\[ \phi : \mathbb{P}(Paths_A) \times \mathbb{P}(Paths_B) \to \mathbb{R}^+ \text{ is an } (\epsilon, \delta)\text{-strong approximate simulation if} \]

START \[ \phi(\mu_{A_0}, \mu_{B_0}) \leq \epsilon \]

STEP \[ \phi(\mu_A, \mu_B) \leq \epsilon \text{ implies } \phi(\mu'_A, \mu'_B) \leq \epsilon \]

TRACE \[ \phi(\mu_A, \mu_B) \leq \epsilon \text{ implies } d(tdist(\mu_A), tdist(\mu_B)) \leq \delta. \]
strong approximate simulations

\[ \phi : \mathbb{P}(\text{Paths}_A) \times \mathbb{P}(\text{Paths}_B) \to \mathbb{R}^+ \] is an \((\epsilon, \delta)\)-strong approximate simulation if

- **START** \( \phi(\mu_{A0}, \mu_{B0}) \leq \epsilon \)
- **STEP** \( \phi(\mu_A, \mu_B) \leq \epsilon \) implies \( \phi(\mu'_A, \mu'_B) \leq \epsilon \)
- **TRACE** \( \phi(\mu_A, \mu_B) \leq \epsilon \) implies \( d(t\text{dist}(\mu_A), t\text{dist}(\mu_B)) \leq \delta \).

\[
\begin{array}{ccc}
\mu_{B0} & \mu_B & \mu'_B \\
\blacktriangle & \rho' & \blacktriangle \\
\blacktriangle & \blacktriangle \\
\mu_{A0} & \mu_A & \mu'_A
\end{array}
\]

**Theorem**
existence of \((\epsilon, \delta)\)-strong approximate simulation implies \( A \delta \)-implements \( B \)
automata with unaligned branching structure

\[
\begin{align*}
B &= r_0 \\
    &\quad \text{choose: } \frac{1}{n} \text{ or } \frac{n-1}{n} \\
    &\quad \text{choose: } \frac{1}{n} \text{ or } \frac{n+1}{n} \\
    &\quad \text{choose: } \frac{1}{n} \text{ or } \frac{n-1}{n}
\end{align*}
\]

\[
\begin{align*}
z &= 1 \\
    &\quad \text{out(1)}
\end{align*}
\]

\[
\begin{align*}
z &= 2 \\
    &\quad \text{out(2)}
\end{align*}
\]

\[
\begin{align*}
z &= n \\
    &\quad \text{out(n)}
\end{align*}
\]

\[
\begin{align*}
\theta(\mu_A, \mu_B) &= \begin{cases} 
\max & \sum_{sA \cdot z = x} \mu_1(sA) - \sum_{sB \cdot z = x} \mu_2(sB) \\
\max & sA \cdot z \neq sB \cdot z \\
\mu_1(sA) + \mu_2(sB) & \text{no mismatched } z \text{'s}
\end{cases}
\end{align*}
\]
automata with unaligned branching structure

\[\begin{align*}
A & : \text{choose } \frac{1}{n} - \epsilon \\
& \quad \text{choose } \frac{1}{n} + \epsilon \\
y = 1 & \quad y = 2 & \quad \ldots & \quad y = n
\end{align*}\]

\[\begin{align*}
B & : \text{choose } \frac{1}{n} \\
& \quad \text{choose } \frac{1}{n} \\
z = 1 & \quad z = 2 & \quad \ldots & \quad z = n
\end{align*}\]
automata with unaligned branching structure

A

\[ A \]

\[ \begin{align*}
  & y = 1 \\
  & \downarrow \text{comp} \\
  & z = 2 \\
\end{align*} \]

\[ \begin{align*}
  & y = 2 \\
  & \downarrow \text{comp} \\
  & z = 3 \\
\end{align*} \]

\[ \begin{align*}
  & y = n \\
  & \downarrow \text{comp} \\
  & z = 1 \\
\end{align*} \]

\[ \begin{align*}
  & \text{choose, } \frac{1}{n} - \epsilon \\
  & \text{choose, } \frac{1}{n} + \epsilon \\
\end{align*} \]

B

\[ B \]

\[ \begin{align*}
  & z = 1 \\
  & \downarrow \text{out(1)} \\
\end{align*} \]

\[ \begin{align*}
  & z = 2 \\
  & \downarrow \text{out(2)} \\
\end{align*} \]

\[ \begin{align*}
  & \ldots \\
  & \downarrow \text{out(n)} \\
\end{align*} \]

\[ \begin{align*}
  & z = n \\
  & \downarrow \text{choose, } \frac{1}{n} \\
  & \downarrow \text{choose, } \frac{1}{n} \\
  & \text{choose, } \frac{1}{n} \\
\end{align*} \]

\[ \begin{align*}
  & \text{no mismatched } z \text{'s} \\
  & \max \ s_A \cdot z = x \mu_1(s_A) - \sum s_B \cdot z = x \mu_2(s_B) \\
  & \text{otherwise.} \\
\end{align*} \]

\[ \theta(\mu_A, \mu_B) = \begin{cases} 
  \max \ x \in [n] \cup \{\perp\} \left| \sum s_A \cdot z = x \mu_1(s_A) - \sum s_B \cdot z = x \mu_2(s_B) \right| 
  & \text{no mismatched } z \text{'s} \\
  \max s_A \cdot z \neq s_B \cdot z \left[ \mu_1(s_A) + \mu_2(s_B) \right] 
  & \text{otherwise.} 
\end{cases} \]
automata with unaligned branching structure

A

\[
\begin{align*}
\text{\textcolor{blue}{comp}} & \quad \text{\textcolor{blue}{comp}} & \quad \text{\textcolor{blue}{comp}} & \quad \text{\textcolor{blue}{comp}} \\
y = 1 & \quad y = 2 & \quad \ldots & \quad y = n \\
z = 2 & \quad z = 3 & \quad \ldots & \quad z = 1 \\
\text{\textcolor{blue}{out}(2)} & \quad \text{\textcolor{blue}{out}(3)} & \quad \ldots & \quad \text{\textcolor{blue}{out}(1)} \\
\end{align*}
\]

B

\[
\begin{align*}
\text{\textcolor{blue}{choose}, } \frac{1}{n} - \epsilon & \quad \text{\textcolor{blue}{choose}, } \frac{1}{n} + \epsilon \\
y = 1 & \quad y = 2 & \quad \ldots & \quad y = n \\
z = 1 & \quad z = 2 & \quad \ldots & \quad z = n \\
\text{\textcolor{blue}{out}(1)} & \quad \text{\textcolor{blue}{out}(2)} & \quad \ldots & \quad \text{\textcolor{blue}{out}(n)} \\
\end{align*}
\]
automata with unaligned branching structure

\[ \theta(\mu_A, \mu_B) = \begin{cases} \max_{x \in \{\text{comp}, \text{out}(1), \ldots, \text{out}(n)\} \cup \{\perp\}} \left| \sum_{s_A \cdot z = x} \mu_1(s_A) - \sum_{s_B \cdot z = x} \mu_2(s_B) \right| & \text{no mismatched } z \text{'s} \\ \mu_1(s_A) + \mu_2(s_B) & \text{otherwise.} \end{cases} \]
automata with unaligned branching structure

\[ \theta(\mu_A, \mu_B) = \begin{cases} 
\max_{x \in [n] \cup \{\perp\}} \left( \sum_{s_A.z = x} \mu_1(s_A) - \sum_{s_B.z = x} \mu_2(s_B) \right) & \text{no mismatched } z's 
\end{cases} \]
automata with unaligned branching structure

candidate simulation function

\[
\theta(\mu_A, \mu_B) = \begin{cases} 
\max_{x \in [n] \cup \{\bot\}} \sum_{s_A \cdot z = x} \mu_1(s_A) - \sum_{s_B \cdot z = x} \mu_2(s_B) & \text{no mismatched } z's \\
\max_{s_A \cdot z \neq s_B \cdot z} [\mu_1(s_A) + \mu_2(s_B)] & \text{otherwise.}
\end{cases}
\]
automata with unaligned branching structure

\[ A \quad t_0 \quad B \quad r_0 \]

\[
\theta(\mu_A, \mu_B) = \begin{cases} 
\max_{x \in [n] \cup \{\bot\}} \left| \sum_{s_A : z = x} \mu_1(s_A) - \sum_{s_B : z = x} \mu_2(s_B) \right| & \text{no mismatched } z \text{'s} \\
\max_{s_A : z \neq s_B : z} \left[ \mu_1(s_A) + \mu_2(s_B) \right] & \text{otherwise.}
\end{cases}
\]

\[
\theta(\mu_{A0}, \mu_{B0}) = 1 - 1 \leq \epsilon
\]
automata with unaligned branching structure

\[ \theta(\mu_A, \mu_B) = \begin{cases} 
\max_{x \in [n] \cup \{\bot\}} \left| \sum_{s_A : z = x} \mu_1(s_A) - \sum_{s_B : z = x} \mu_2(s_B) \right| & \text{no mismatched } z \text{'s} \\
\max_{s_A : z \neq s_B : z} \left[ \mu_1(s_A) + \mu_2(s_B) \right] & \text{otherwise.} 
\end{cases} \]

\[ \theta(\mu_{A1}, \mu_{B1}) = \left( \frac{1}{n} + \epsilon + \ldots + \frac{1}{n} - \epsilon \right) - 1 \leq \epsilon \]
automata with unaligned branching structure

\[ \theta(\mu_A, \mu_B) = \begin{cases} 
\max_{x \in [n] \cup \{\bot\}} \left| \sum_{s_A \cdot z = x} \mu_1(s_A) - \sum_{s_B \cdot z = x} \mu_2(s_B) \right| & \text{no mismatched } z's \\
\max_{s_A \cdot z \neq s_B \cdot z} \left[ \mu_1(s_A) + \mu_2(s_B) \right] & \text{otherwise.}
\end{cases} \]

\[ \theta(\mu_{A2}, \mu_{B2}) = \frac{2}{n} + \epsilon \]
automata with unaligned branching structure

\[ \theta(\mu_A, \mu_B) = \begin{cases} 
\max_{x \in [n] \cup \{\perp\}} \left| \sum_{s_A \cdot z = x} \mu_1(s_A) - \sum_{s_B \cdot z = x} \mu_2(s_B) \right| & \text{no mismatched } z's \\
\max_{s_A \cdot z \neq s_B \cdot z} [\mu_1(s_A) + \mu_2(s_B)] & \text{otherwise.} 
\end{cases} \]

\( \theta \) is not a strong approximate simulation
automata with unaligned branching structure

\[ \theta(\mu_A, \mu_B) = \begin{cases} 
\max_{x \in [n] \cup \{\perp\}} \left| \sum_{s_A : z = x} \mu_1(s_A) - \sum_{s_B : z = x} \mu_2(s_B) \right| & \text{no mismatched } z \text{'s} \\
\max_{s_A : z \neq s_B : z} [\mu_1(s_A) + \mu_2(s_B)] & \text{otherwise.}
\end{cases} \]

strong approximate simulations cannot relate automata with “unaligned” internal branching structure
expansion of a function

given $\phi : X \times Y \rightarrow \mathbb{R}^+$,

$$\hat{\phi}(x_1, y_1) = \min_{\psi \in \Gamma(x_1, y_1)} \left[ \max_{(x, y) \in \psi} \phi(x, y) \right]$$
expansion of a function

given $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$,

$$\hat{\phi}(x_1, y_1) = \min_{\psi \in \Gamma(x_1, y_1)} \left[ \max_{(x, y) \in \psi} \phi(x, y) \right]$$

Lemma

for any $\epsilon \geq 0$, $\epsilon$-sublevel set of $\hat{\phi}$ is the convex hull of the $\epsilon$-sublevel set of $\phi$. 
expansion of a function

given $\phi : X \times Y \rightarrow \mathbb{R}^+$,

$$\hat{\phi}(x_1, y_1) = \min_{\psi \in \Gamma(x_1, y_1)} \left[ \max_{(x, y) \in \psi} \phi(x, y) \right]$$

**Lemma**

for any $\epsilon \geq 0$, $\epsilon$-sublevel set of $\hat{\phi}$ is the convex hull of the $\epsilon$-sublevel set of $\phi$. 
expanded approximate simulations

\[ \phi : \mathbb{P}(Paths_A) \times \mathbb{P}(Paths_B) \rightarrow \mathbb{R}^+ \] is an \((\epsilon, \delta)\)-approximate simulation if:

**START** \( \phi(\mu_{A0}, \mu_{B0}) \leq \epsilon \)

Theorem: existence of \((\epsilon, \delta)\)-approximate simulation implies \(A^{\delta}\)-implements \(B\)
expanded approximate simulations

\[ \phi : \mathbb{P}(\text{Paths}_A) \times \mathbb{P}(\text{Paths}_B) \rightarrow \mathbb{R}^+ \] is an \((\epsilon, \delta)\)-approximate simulation if:

- **START** \[ \phi(\mu_{A0}, \mu_{B0}) \leq \epsilon \]
- **STEP** \[ \phi(\mu_A, \mu_B) \leq \epsilon \text{ implies } \hat{\phi}(\mu'_A, \mu'_B) \leq \epsilon \]
expanded approximate simulations

$\phi : \mathbb{P}(Paths_A) \times \mathbb{P}(Paths_B) \to \mathbb{R}^+$ is an $\epsilon, \delta$-approximate simulation if:

- **START** $\phi(\mu_{A0}, \mu_{B0}) \leq \epsilon$
- **STEP** $\phi(\mu_A, \mu_B) \leq \epsilon$ implies $\hat{\phi}(\mu'_A, \mu'_B) \leq \epsilon$
- **TRACE** $\phi(\mu_A, \mu_B) \leq \epsilon$ implies $d(tdist(\mu_A), tdist(\mu_B)) \leq \delta$.

**Theorem**

existence of $(\epsilon, \delta)$-approximate simulation implies $A \delta$-implements $B$
example (continued)

\[ \theta(\mu_{A0}, \mu_{B0}) \leq \epsilon \quad \theta(\mu_{A1}, \mu_{B1}) \leq \epsilon \quad \theta(\mu_{A2}, \mu_{B2}) = \frac{2}{n} + \epsilon \]

\[ \hat{\theta}(\mu_{A2}, \mu_{B2}) \leq 2\epsilon \]
\[
\theta(\mu_{A0}, \mu_{B0}) \leq \epsilon \quad \theta(\mu_{A1}, \mu_{B1}) \leq \epsilon \quad \theta(\mu_{A2}, \mu_{B2}) = \frac{2}{n} + \epsilon
\]

<table>
<thead>
<tr>
<th>(\psi)</th>
<th>(\delta_{y=1,z=2})</th>
<th>(\delta_{y=2,z=3})</th>
<th>(\delta_{y=3,z=4})</th>
<th>(\delta_{y=4,z=1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\delta_{z=1})</td>
<td></td>
<td></td>
<td></td>
<td>(\frac{1}{4})</td>
</tr>
<tr>
<td>(\delta_{z=2})</td>
<td>(\frac{1}{4} - \epsilon)</td>
<td></td>
<td></td>
<td>(\epsilon)</td>
</tr>
<tr>
<td>(\delta_{z=3})</td>
<td></td>
<td>(\frac{1}{4} - \epsilon)</td>
<td></td>
<td>(\epsilon)</td>
</tr>
<tr>
<td>(\delta_{z=4})</td>
<td></td>
<td></td>
<td>(\frac{1}{4})</td>
<td></td>
</tr>
</tbody>
</table>

\(\psi\) consistent with marginals \(\mu_{A2}\) and \(\mu_{B2}\)
\[ \theta(\mu_{A0}, \mu_{B0}) \leq \epsilon \quad \theta(\mu_{A1}, \mu_{B1}) \leq \epsilon \quad \theta(\mu_{A2}, \mu_{B2}) = \frac{2}{n} + \epsilon \]

<table>
<thead>
<tr>
<th>\psi</th>
<th>\delta_{y=1,z=2}</th>
<th>\delta_{y=2,z=3}</th>
<th>\delta_{y=3,z=4}</th>
<th>\delta_{y=4,z=1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\delta_{z=1}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\delta_{z=2}</td>
<td>\frac{1}{4} - \epsilon</td>
<td></td>
<td></td>
<td>\frac{1}{4}</td>
</tr>
<tr>
<td>\delta_{z=3}</td>
<td></td>
<td>\frac{1}{4} - \epsilon</td>
<td>\epsilon</td>
<td></td>
</tr>
<tr>
<td>\delta_{z=4}</td>
<td></td>
<td></td>
<td>\frac{1}{4}</td>
<td></td>
</tr>
</tbody>
</table>

if \( \nu_1, \nu_2 \) have matched values of \( z \), \( \theta(\nu_1, \nu_2) = \epsilon \)
\[ \theta(\mu_{A0}, \mu_{B0}) \leq \epsilon \quad \theta(\mu_{A1}, \mu_{B1}) \leq \epsilon \quad \theta(\mu_{A2}, \mu_{B2}) = \frac{2}{n} + \epsilon \]

<table>
<thead>
<tr>
<th>( \psi )</th>
<th>( \delta_{y=1, z=2} )</th>
<th>( \delta_{y=2, z=3} )</th>
<th>( \delta_{y=3, z=4} )</th>
<th>( \delta_{y=4, z=1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta_{z=1} )</td>
<td>1</td>
<td>1</td>
<td>( \frac{1}{4} )</td>
<td>1</td>
</tr>
<tr>
<td>( \delta_{z=2} )</td>
<td>( \frac{1}{4} - \epsilon )</td>
<td>( \frac{1}{4} - \epsilon )</td>
<td>( \epsilon )</td>
<td>( \epsilon )</td>
</tr>
<tr>
<td>( \delta_{z=3} )</td>
<td>( \frac{1}{4} - \epsilon )</td>
<td>( \epsilon )</td>
<td>( \epsilon )</td>
<td>( \epsilon )</td>
</tr>
<tr>
<td>( \delta_{z=4} )</td>
<td>1</td>
<td>1</td>
<td>( \frac{1}{4} )</td>
<td>( \frac{1}{4} )</td>
</tr>
</tbody>
</table>

If \( \nu_1, \nu_2 \) have mismatched values of \( z \), \( \nu_1(s_A), \nu_2(s_B) \leq \epsilon \), \( \theta(\nu_1, \nu_2) = 2\epsilon \)
\[\theta(\mu_{A_0}, \mu_{B_0}) \leq \epsilon \quad \theta(\mu_{A_1}, \mu_{B_1}) \leq \epsilon \quad \theta(\mu_{A_2}, \mu_{B_2}) = \frac{2}{n} + \epsilon\]

\[
\begin{array}{|c|c|c|c|c|}
\hline
\psi & \delta_{y=1,z=2} & \delta_{y=2,z=3} & \delta_{y=3,z=4} & \delta_{y=4,z=1} \\
\hline
\delta_{z=1} & & & & \frac{1}{4} \\
\hline
\delta_{z=2} & \frac{1}{4} - \epsilon & & \epsilon & \\
\hline
\tilde{\delta}_{z=3} & \frac{1}{4} - \epsilon & \epsilon & & \\
\hline
\delta_{z=4} & \frac{1}{4} & & & \\
\hline
\end{array}
\]

\[\hat{\theta}(\mu_{A_2}, \mu_{B_2}) \leq 2\epsilon\]
example (continued)

\[ \theta(\mu_{A0}, \mu_{B0}) \leq \epsilon \quad \theta(\mu_{A1}, \mu_{B1}) \leq \epsilon \quad \theta(\mu_{A2}, \mu_{B2}) = \frac{2}{n} + \epsilon \]

<table>
<thead>
<tr>
<th>\psi</th>
<th>\delta_{y=1,z=2}</th>
<th>\delta_{y=2,z=3}</th>
<th>\delta_{y=3,z=4}</th>
<th>\delta_{y=4,z=1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\delta_{z=1}</td>
<td></td>
<td></td>
<td>\frac{1}{4}</td>
<td>\epsilon</td>
</tr>
<tr>
<td>\delta_{z=2}</td>
<td>\frac{1}{4} - \epsilon</td>
<td></td>
<td>\epsilon</td>
<td>\epsilon</td>
</tr>
<tr>
<td>\delta_{z=3}</td>
<td></td>
<td>\frac{1}{4} - \epsilon</td>
<td>\epsilon</td>
<td>\epsilon</td>
</tr>
<tr>
<td>\delta_{z=4}</td>
<td></td>
<td></td>
<td>\frac{1}{4}</td>
<td>\epsilon</td>
</tr>
</tbody>
</table>

\[ \hat{\theta}(\mu_{A2}, \mu_{B2}) \leq 2\epsilon \]

\( \theta \) is a \((2\epsilon, 2\epsilon)\)-approximate simulation from \( A \) to \( B \)
\[ \theta(\mu_{A0}, \mu_{B0}) \leq \epsilon \quad \theta(\mu_{A1}, \mu_{B1}) \leq \epsilon \quad \theta(\mu_{A2}, \mu_{B2}) = \frac{2}{n} + \epsilon \]

<table>
<thead>
<tr>
<th>\psi</th>
<th>\delta_{y=1,z=2}</th>
<th>\delta_{y=2,z=3}</th>
<th>\delta_{y=3,z=4}</th>
<th>\delta_{y=4,z=1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\delta_{z=1}</td>
<td></td>
<td></td>
<td>\frac{1}{4}</td>
<td></td>
</tr>
<tr>
<td>\delta_{z=2}</td>
<td>\frac{1}{4} - \epsilon</td>
<td></td>
<td>\epsilon</td>
<td></td>
</tr>
<tr>
<td>\delta_{z=3}</td>
<td>\frac{1}{4} - \epsilon</td>
<td>\frac{1}{4} - \epsilon</td>
<td>\epsilon</td>
<td></td>
</tr>
<tr>
<td>\delta_{z=4}</td>
<td></td>
<td>\frac{1}{4} - \epsilon</td>
<td>\epsilon</td>
<td>\frac{1}{4}</td>
</tr>
</tbody>
</table>

\[ \hat{\theta}(\mu_{A2}, \mu_{B2}) \leq 2\epsilon \]

\theta \text{ is a } (2\epsilon, 2\epsilon)-\text{approximate simulation from } A \text{ to } B

A \text{ 2\epsilon-implements } B
discussion

approximate implementations for probabilistic automata

- metric on \( \mathbb{P}(Ext) \), \( Ext \) itself not required to be a metric space
discussion

approximate implementations for probabilistic automata

- metric on $\mathbb{P}(\text{Ext})$, $\text{Ext}$ itself not required be a metric space
- discounted uniform metric $d_\delta(\eta_1, \eta_2) = \sup_{\beta \in \text{Ext}^*} \delta^{\mid \beta \mid} |\eta_1(\beta) - \eta_1(\beta)|$
discussion

approximate implementations for probabilistic automata

- metric on \( \mathbb{P}(Ext) \), \( Ext \) itself not required be a metric space
- discounted uniform metric \( d_\delta(\eta_1, \eta_2) = \sup_{\beta \in Ext^*} \delta^{|\beta|} |\eta_1(\beta) - \eta_1(\beta)| \)

inductive proof technique based on real-valued simulation function \( \phi \)
discussion

approximate implementations for probabilistic automata

- metric on $\mathbb{P}(Ext)$, Ext itself not required be a metric space
- discounted uniform metric $d_\delta(\eta_1, \eta_2) = \sup_{\beta \in Ext^*} \delta^{\lvert \beta \rvert} \lvert \eta_1(\beta) - \eta_1(\beta) \rvert$

inductive proof technique based on real-valued simulation function $\phi$

- expansion of $\phi$ is same as Wasserstein distance
discussion

approximate implementations for probabilistic automata

- metric on \( \mathbb{P}(\text{Ext}) \), Ext itself not required be a metric space
- discounted uniform metric
  \[
  d_\delta(\eta_1, \eta_2) = \sup_{\beta \in \text{Ext}^*} \delta|\beta| |\eta_1(\beta) - \eta_1(\beta)|
  \]

inductive proof technique based on real-valued simulation function \( \phi \)

- expansion of \( \phi \) is same as Wasserstein distance

probabilistic reasoning about systems

- safety: if \( B \) is unsafe with probability at most \( p \) and \( A \infty\)-implements \( B \)
  then \( A \) is unsafe with probability at most \( p + \delta \)
discussion

approximate implementations for probabilistic automata

- metric on $\mathbb{P}(Ext)$, $Ext$ itself not required be a metric space
- discounted uniform metric $d_\delta(\eta_1, \eta_2) = \sup_{\beta \in Ext^*} \delta |\beta| |\eta_1(\beta) - \eta_2(\beta)|$

inductive proof technique based on real-valued simulation function $\phi$

- expansion of $\phi$ is same as Wasserstein distance

probabilistic reasoning about systems

- safety: if $B$ is unsafe with probability at most $p$ and $A \delta$-implements $B$ then $A$ is unsafe with probability at most $p + \delta$
- termination: consider a consensus protocol $A$ which uses perfectly random coins. if the coins of $A$ are slightly biased then the probability of termination after $r$ rounds degrades, but not too rapidly.
discussion

approximate implementations for probabilistic automata

- metric on \( \mathbb{P}(Ext) \), \( Ext \) itself not required be a metric space
- discounted uniform metric
  \[
  d_\delta(\eta_1, \eta_2) = \sup_{\beta \in Ext^*} \delta|\beta| |\eta_1(\beta) - \eta_1(\beta)|
  \]

inductive proof technique based on real-valued simulation function \( \phi \)

- expansion of \( \phi \) is same as Wasserstein distance

probabilistic reasoning about systems

- safety: if \( B \) is unsafe with probability at most \( p \) and \( A \) \( \delta \)-implements \( B \) then \( A \) is unsafe with probability at most \( p + \delta \)
- termination: consider a consensus protocol \( A \) which uses perfectly random coins. if the coins of \( A \) are slightly biased then the probability of termination after \( r \) rounds degrades, but not too rapidly.
- SZK: indistinguishability of traces is captured as \( \delta \)-implementations
future research directions

approximate simulations of the form $\phi : \mathcal{P}(Q_A) \times \mathcal{P}(Q_B) \rightarrow \mathbb{R}^+$
future research directions

approximate simulations of the form \( \phi : P(Q_A) \times P(Q_B) \rightarrow \mathbb{R}^+ \)

compute approximate simulations for probabilistic automata
future research directions

approximate simulations of the form $\phi : \mathcal{P}(Q_A) \times \mathcal{P}(Q_B) \rightarrow \mathbb{R}^+$

compute approximate simulations for probabilistic automata

- Worrell and van Breugel gave poly-time algorithm for computing approximate bisimulations of finite state probabilistic automata (without nondeterminism) in 2006.
future research directions

approximate simulations of the form $\phi : \mathbb{P}(Q_A) \times \mathbb{P}(Q_B) \rightarrow \mathbb{R}^+$

compute approximate simulations for probabilistic automata

- Worrell and van Breugel gave poly-time algorithm for computing approximate bisimulations of finite state probabilistic automata (without nondeterminism) in 2006.

approximate implementations and simulations for hybrid systems—automata with discrete and continuous evolution