Probabilistic Applicative Bisimulation and Call-by-Value Lambda Calculi

Joint work with Ugo Dal Lago

Raphaëlle Crubillé

ENS Lyon

February 9, 2014

Introduction

- Fundamental question: when can two programs be considered **equivalent**?

Introduction

- Fundamental question: when can two programs be considered **equivalent**?
- **Context equivalence** [Morris1968] :
  - Two terms M and N are context equivalent if their **observable behavior** is the same in **any** context.

Introduction

- Fundamental question: when can two programs be considered **equivalent**?
- **Context equivalence** [Morris1968] :
  - Two terms M and N are context equivalent if their **observable behavior** is the same in **any** context.
  - Proving that two programs are **not** equivalent is relatively easy: just find **a** context that separates them.
  - Proving that two program are indeed **equivalent**, on the other hand, can be quite complicated.
- Other equivalence notion : Bisimilarity

Our result

For a probabilistic λ-calculus ($\Lambda_\oplus$) :

$$\text{Context Equivalence} = \text{Bisimilarity}$$

1. $\Lambda_\oplus$
   - Syntax and Operational Semantics
   - Motivating Example : Perfect Security
2. Bisimulation
   - Probabilistic Bisimulation in the abstact
   - A Labelled Markov Chain for $\Lambda_\oplus$
   - Example
3. Context Equivalence vs. Bisimulation
   - $\sim \subseteq \equiv$
   - Full Abstraction
4. Conclusions

1. $\Lambda_\oplus$
   - Syntax and Operational Semantics
   - Motivating Example : Perfect Security
2. Bisimulation
   - Probabilistic Bisimulation in the abstact
   - A Labelled Markov Chain for $\Lambda_\oplus$
   - Example
3. Context Equivalence vs. Bisimulation
   - $\sim \subseteq \equiv$
   - Full Abstraction
4. Conclusions

Syntax and Operational Semantics of $\Lambda_\oplus$ [DLZorzi2012]

- **Terms**: $M, N ::= x \mid \lambda x.M \mid MM \mid M \oplus M;$

Syntax and Operational Semantics of $\Lambda_\oplus$ [DLZorzi2012]

- **Terms**: $M, N \;::=\; x \;\big|\; \lambda x.M \;\big|\; MM \;\big|\; M \oplus M;$
- **Values**: $V \;::=\; \lambda x.M;$

Syntax and Operational Semantics of $\Lambda_\oplus$ [DLZorzi2012]

- **Terms**: $M, N \ ::= \ x \ \big| \ \lambda x.M \ \big| \ MM \ \big| \ M \oplus M;$
- **Values**: $V \ ::= \ \lambda x.M;$
- **Approximation (Big-Step) Semantics**:
  - $M \Downarrow \mathscr{D}$, where $\mathscr{D}$: Values $\to [0,1]$ sub-probability distribution.
  - Approximation from below : only finite distributions

$$\frac{}{M \Downarrow \emptyset} \qquad \frac{}{V \Downarrow \{V^1\}} \qquad \frac{M \Downarrow \mathscr{D} \qquad N \Downarrow \mathscr{E}}{M \oplus N \Downarrow \frac{1}{2}\mathscr{D} + \frac{1}{2}\mathscr{E}}$$

$$\frac{M \Downarrow \mathscr{K} \qquad N \Downarrow \mathscr{F} \qquad \{P[V/x] \Downarrow \mathscr{E}_{P,V}\}_{\lambda x.P \in \mathsf{S}(\mathscr{K}),\, V \in \mathsf{S}(\mathscr{F})}}{MN \Downarrow \sum_{V \in \mathsf{S}(\mathscr{F})} \mathscr{F}(V) \left(\sum_{\lambda x.P \in \mathsf{S}(\mathscr{K})} \mathscr{K}(\lambda x.P)\mathscr{E}_{P,V}\right)}$$

Syntax and Operational Semantics of $\Lambda_\oplus$ [DLZorzi2012]

- **Terms**: $M, N ::= x \mid \lambda x.M \mid MM \mid M \oplus M$;
- **Values**: $V ::= \lambda x.M$;
- **Approximation (Big-Step) Semantics**:
  - $M \Downarrow \mathscr{D}$, where $\mathscr{D}$: Values $\to [0,1]$ sub-probability distribution.
  - Approximation from below : only finite distributions

$$\frac{}{M \Downarrow \emptyset} \qquad \frac{}{V \Downarrow \{V^1\}} \qquad \frac{M \Downarrow \mathscr{D} \qquad N \Downarrow \mathscr{E}}{M \oplus N \Downarrow \frac{1}{2}\mathscr{D} + \frac{1}{2}\mathscr{E}}$$

$$\frac{M \Downarrow \mathscr{K} \qquad N \Downarrow \mathscr{F} \qquad \{P[V/x] \Downarrow \mathscr{E}_{P,V}\}_{\lambda x.P \in \mathsf{S}(\mathscr{K}),\, V \in \mathsf{S}(\mathscr{F})}}{MN \Downarrow \sum_{V \in \mathsf{S}(\mathscr{F})} \mathscr{F}(V) \left(\sum_{\lambda x.P \in \mathsf{S}(\mathscr{K})} \mathscr{K}(\lambda x.P)\mathscr{E}_{P,V}\right)}$$

- **Semantics**: $[\![M]\!] = \sup_{M \Downarrow \mathscr{D}} \mathscr{D}$;

Syntax and Operational Semantics of $\Lambda_\oplus$ [DLZorzi2012]

- **Terms**: $M, N \;::=\; x \;\big|\; \lambda x.M \;\big|\; MM \;\big|\; M \oplus M;$
- **Values**: $V \;::=\; \lambda x.M;$
- **Approximation (Big-Step) Semantics**:
  - $M \Downarrow \mathscr{D}$, where $\mathscr{D}$: Values $\rightarrow [0,1]$ sub-probability distribution.
  - Approximation from below : only finite distributions

$$\frac{}{M \Downarrow \emptyset} \qquad \frac{}{V \Downarrow \{V^1\}} \qquad \frac{M \Downarrow \mathscr{D} \qquad N \Downarrow \mathscr{E}}{M \oplus N \Downarrow \frac{1}{2}\mathscr{D} + \frac{1}{2}\mathscr{E}}$$

$$\frac{M \Downarrow \mathscr{K} \qquad N \Downarrow \mathscr{F} \qquad \{P[V/x] \Downarrow \mathscr{E}_{P,V}\}_{\lambda x.P \in \mathsf{S}(\mathscr{K}),\, V \in \mathsf{S}(\mathscr{F})}}{MN \Downarrow \sum_{V \in \mathsf{S}(\mathscr{F})} \mathscr{F}(V) \left(\sum_{\lambda x.P \in \mathsf{S}(\mathscr{K})} \mathscr{K}(\lambda x.P)\mathscr{E}_{P,V}\right)}$$

- **Semantics**: $[\![M]\!] = \sup_{M \Downarrow \mathscr{D}} \mathscr{D};$
- Variations: **Small-Step** Semantics, **Call-by-name** Evaluation.

Why Probabilistic Computation?

An Example: Perfect Security

An Example: Perfect Security

Let $\Pi = (GEN, ENC, DEC)$ be a **cryptoscheme**.
Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an **adversary**.

An Example: Perfect Security

Let $\Pi = (GEN, ENC, DEC)$ be a **cryptoscheme**.
Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an **adversary**.

$\mathrm{PrivK}^{\Pi}_{\mathcal{A}}$
$m_0, m_1 \leftarrow \mathcal{A}_1;$

An Example: Perfect Security

Let $\Pi = (GEN, ENC, DEC)$ be a **cryptoscheme**.
Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an **adversary**.

$$
\begin{array}{l}
\mathrm{PrivK}^{\Pi}_{\mathcal{A}} \\
\quad m_0, m_1 \leftarrow \mathcal{A}_1; \\
\quad b \leftarrow \{0, 1\}; \\
\quad k \leftarrow GEN;
\end{array}
$$

An Example: Perfect Security

Let $\Pi = (GEN, ENC, DEC)$ be a **cryptoscheme**.
Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an **adversary**.

$$
\begin{array}{l}
\mathrm{PrivK}^{\Pi}_{\mathcal{A}} \\
\quad m_0, m_1 \leftarrow \mathcal{A}_1; \\
\quad b \leftarrow \{0,1\}; \\
\quad k \leftarrow GEN; \\
\quad c \leftarrow ENC(m_b, k);
\end{array}
$$

An Example: Perfect Security

Let $\Pi = (GEN, ENC, DEC)$ be a **cryptoscheme**.
Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an **adversary**.

$$
\begin{array}{l}
\mathrm{PrivK}^{\Pi}_{\mathcal{A}} \\
\quad m_0, m_1 \leftarrow \mathcal{A}_1; \\
\quad b \leftarrow \{0, 1\}; \\
\quad k \leftarrow GEN; \\
\quad c \leftarrow ENC(m_b, k); \\
\quad b' \leftarrow \mathcal{A}_2(c);
\end{array}
$$

An Example: Perfect Security

Let $\Pi = (GEN, ENC, DEC)$ be a **cryptoscheme**.
Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an **adversary**.

$\mathrm{PrivK}_{\mathcal{A}}^{\Pi}$
$m_0, m_1 \leftarrow \mathcal{A}_1;$
$b \leftarrow \{0, 1\};$
$k \leftarrow GEN;$
$c \leftarrow ENC(m_b, k);$
$b' \leftarrow \mathcal{A}_2(c);$
return $b = b'$.

An Example: Perfect Security

For every adversary $\mathcal{A}$,

$$Pr(\mathsf{PrivK}^{\Pi}_{\mathcal{A}} = \mathtt{true}) = \frac{1}{2}$$

An Example: Perfect Security

One-Time-Pad

$$GEN = \underline{\text{true}} \oplus \underline{\text{false}} : \textbf{bool};$$
$$ENC = \lambda x.\lambda y.\text{if } x \text{ then } (NOT\ \ y) \text{ else } y : \textbf{bool} \to \textbf{bool} \to \textbf{bool};$$
$$DEC = ENC.$$

An Example: Perfect Security

One-Time-Pad

$$GEN = \underline{\text{true}} \oplus \underline{\text{false}} : \textbf{bool};$$
$$ENC = \lambda x.\lambda y.\text{if } x \text{ then } (NOT\ y) \text{ else } y : \textbf{bool} \to \textbf{bool} \to \textbf{bool};$$
$$DEC = ENC.$$

The Experiment as a Pair of Terms

$$EXP_{FST} = \lambda x.\lambda y.ENC\ x\ GEN : \textbf{bool} \to \textbf{bool} \to \textbf{bool};$$
$$EXP_{SND} = \lambda x.\lambda y.ENC\ y\ GEN : \textbf{bool} \to \textbf{bool} \to \textbf{bool}.$$

An Example: Perfect Security

One-Time-Pad

$$GEN = \underline{\text{true}} \oplus \underline{\text{false}} : \textbf{bool};$$
$$ENC = \lambda x.\lambda y.\text{if } x \text{ then } (NOT\ y) \text{ else } y : \textbf{bool} \to \textbf{bool} \to \textbf{bool};$$
$$DEC = ENC.$$

The Experiment as a Pair of Terms

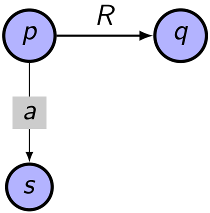
$$EXP_{FST} = \lambda x.\lambda y.ENC\ x\ GEN : \textbf{bool} \to \textbf{bool} \to \textbf{bool};$$
$$EXP_{SND} = \lambda x.\lambda y.ENC\ y\ GEN : \textbf{bool} \to \textbf{bool} \to \textbf{bool}.$$

$$\forall \mathcal{A}.Pr(\text{PrivK}_{\mathcal{A}}^{\text{OTP}} = \texttt{true}) = \frac{1}{2} \quad \Leftrightarrow \quad EXP_{FST} \equiv EXP_{SND}$$

Bisimilarity (deterministic case)

Let $(S, Act, \rightarrow)$ be a LTS (Labelled Transition System).

- A Simulation is a relation R on S such that : If $p\,R\,q$, and $p \xrightarrow{a} s$, there exists t such that $q \xrightarrow{a} t$ and $s\,R\,t$.

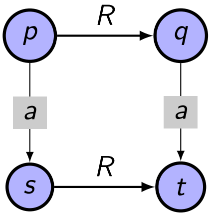


- Bisimilarity : p and q are bisimilar if : $p\,R\,q$, and R is a bisimulation.

Bisimilarity (deterministic case)

Let $(S, Act, \rightarrow)$ be a LTS (Labelled Transition System).

- A Simulation is a relation R on S such that : If $p\,R\,q$, and $p \xrightarrow{a} s$, there exists t such that $q \xrightarrow{a} t$ and $s\,R\,t$.



- Bisimilarity : p and q are bisimilar if : $p\,R\,q$, and R is a bisimulation.

Applicative Bisimulation [Abramsky93]

Terms

Applicative Bisimulation [Abramsky93]

Terms **Values**

Applicative Bisimulation [Abramsky93]

Terms	**Values**
M	
N	
L	
$\vdots$	

Applicative Bisimulation [Abramsky93]

Terms	**Values**
M	V
N	W
L	Z
$\vdots$	$\vdots$

Applicative Bisimulation [Abramsky93]

Terms	**Values**
M	

Applicative Bisimulation [Abramsky93]

Terms **Values**

$$M \xrightarrow{\quad eval \quad} V$$

Applicative Bisimulation [Abramsky93]

Terms **Values**

$$M \xrightarrow{eval} V$$

$$\lambda x.N$$

Applicative Bisimulation [Abramsky93]

Terms **Values**

$$M \xrightarrow{eval} V$$

$$N\{L/x\} \xleftarrow{L} \lambda x.N$$

Applicative Bisimulation [Abramsky93]

- **Simulation**

$$
\begin{tikzcd}
 & & \lambda x.L \arrow[dr, bend left, "R"] & & \\
M \arrow[urr, bend left, "eval"] & N \arrow[dr, bend right, "eval"'] & & L\{R/x\} & P\{R/x\} \\
 & & \lambda x.P \arrow[urr, bend right, "R"'] & &
\end{tikzcd}
$$

Applicative Bisimulation [Abramsky93]

- **Simulation**

$$\begin{tikzcd}
 & & & \lambda x.L \arrow[drr, bend left, "R"] & & & \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & & & L\{R/x\} & P\{R/x\} \\
 & & & \lambda x.P \arrow[urrr, bend right, "R"'] & & &
\end{tikzcd}$$

Applicative Bisimulation [Abramsky93]

- **Simulation**

$$
\begin{tikzcd}
 & \lambda x.L \arrow[rr, bend left, "R"] & & \\
M \; \mathcal{R} \; N \arrow[ur, bend left, "eval"] \arrow[dr, bend right, "eval"'] & & & L\{R/x\} \quad \mathcal{R} \quad P\{R/x\} \\
 & \lambda x.P \arrow[urr, bend right, "R"'] & &
\end{tikzcd}
$$

Applicative Bisimulation [Abramsky93]

- **Simulation**

$$
\begin{tikzcd}
 & & & \lambda x.L \arrow[dr, bend left, "R"] & & & \\
M \arrow[urrr, bend left, "eval"] & \mathcal{R} & N \arrow[dr, bend right, "eval"'] & & L\{R/x\} & \mathcal{R} & P\{R/x\} \\
 & & & \lambda x.P \arrow[urrr, bend right, "R"'] & & &
\end{tikzcd}
$$

- **Similarity**: union of all simulations, denoted $\precsim$;
- **Bisimilarity**: union of all bisimulations, denoted $\sim$.

Theorem

$M \equiv N$ *iff* $M \sim N$.

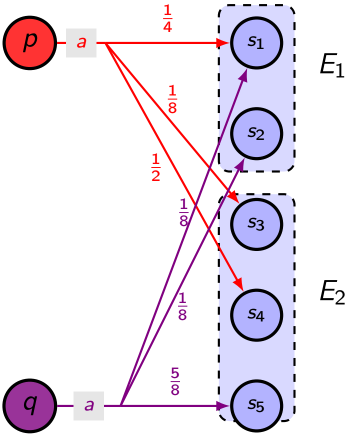
Probabilistic Bisimulation in the Abstract [LS1992]

Labelled Markov Chain (LMC): a triple $\mathcal{M} = (\mathcal{S}, \mathcal{L}, \mathcal{P})$, where

- $\mathcal{S}$ is a countable set of *states*;
- $\mathcal{L}$ is a set of *labels*;
- $\mathcal{P}$ is a *transition probability matrix*, i.e., a function $\mathcal{P} : \mathcal{S} \times \mathcal{L} \times \mathcal{S} \to \mathbb{R}$ such that for every state s and for every label l, $\mathcal{P}(\mathcal{S}, l, t) = \sum_{t \in \mathcal{S}} \mathcal{P}(s, l, t) \leq 1$;

Bisimilarity (probabilistic case)

Let $(\mathcal{S}, \mathcal{L}, \mathcal{P})$ be a LMC (Labelled Markov Chain).



Bisimulation : R such that

- R equivalence relation on $\mathcal{S}$.
- $(p, q) \in R \Rightarrow$ for every equivalence class E, $a \in \mathcal{L}$,

$$\sum_{s \in E} \mathcal{P}(p, a, s) = \sum_{s \in E} \mathcal{P}(q, a, s)$$

.

A Labelled Markov Chain for $\Lambda_\oplus$

Terms

M

Values

A Labelled Markov Chain for $\Lambda_\oplus$

Terms **Values**

$$\begin{tikzcd}
M \arrow[r, "{eval,\ [\![M]\!](V)}"] \arrow[dr, "{eval,\ [\![M]\!](W)}"] \arrow[ddr, "{eval,\ [\![M]\!](Z)}"'] & V \\
& W \\
& Z \\
& \vdots
\end{tikzcd}$$

A Labelled Markov Chain for $\Lambda_\oplus$

Terms	Values
	$\lambda x.N$

A Labelled Markov Chain for $\Lambda_\oplus$

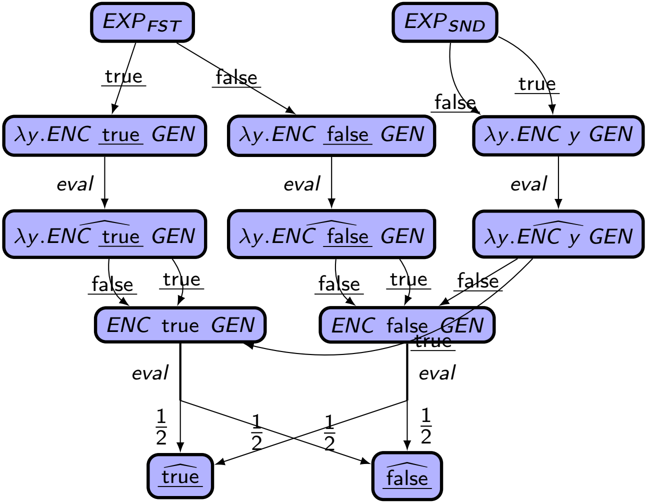
Terms **Values**

$$N\{W/x\} \xleftarrow{W,\ 1} \lambda x.N$$

Back to Our Example

$$EXP_{FST} = \lambda x.\lambda y.ENC\ x\ GEN : \mathbf{bool} \to \mathbf{bool} \to \mathbf{bool};$$
$$EXP_{SND} = \lambda x.\lambda y.ENC\ y\ GEN : \mathbf{bool} \to \mathbf{bool} \to \mathbf{bool}.$$

Back to Our Example



Back to Our Example

$$
\begin{aligned}
\mathcal{R}_\sigma &= X_\sigma \cup ID_\sigma; \\
X_{\mathbf{bool}} &= \{(ENC\ \underline{\text{true}}\ GEN), (ENC\ \underline{\text{false}}\ GEN)\}; \\
X_{\mathbf{bool}\to\mathbf{bool}} &= \{(\lambda y.ENC\ y\ GEN), (\lambda y.ENC\ \underline{\text{true}}\ GEN), \\
&\qquad (\lambda y.ENC\ \underline{\text{false}}\ GEN)\}; \\
X_{\mathbf{bool}\to\mathbf{bool}\to\mathbf{bool}} &= \{EXP_{FST}, EXP_{SND}\};
\end{aligned}
$$

Context Equivalence vs. Bisimulation

- **Contexts**:
  $C \ ::= \ [\cdot] \ \big| \ \lambda x.C \ \big| \ CM \ \big| \ MC \ \big| \ M \oplus C \ \big| \ C \oplus M.$
- **Context Equivalence**: $M \equiv N$ iff for every context C it holds that $\sum[\![C[M]]\!] = \sum[\![C[N]]\!]$.

Theorem

$\sim$ is included in $\equiv$.

Lemma

$\sim$ is a congruence.

- $M \sim N \implies C[M] \sim C[N]$
- Howe's technique.

Full Abstraction?

- $\sim$ is a **sound** methodology for program equivalence.
- Is it also **complete**?
- **CBN** : **No** [DLSA2014]
  - Counterexample:

$$M = \lambda x.\lambda y.(\Omega \oplus I); \qquad N = \lambda x.(\lambda y.\Omega) \oplus (\lambda y.I).$$

Full Abstraction?

- $\sim$ is a **sound** methodology for program equivalence.
- Is it also **complete**?
- **CBN** : **No** [DLSA2014]
  - Counterexample:

$$M = \lambda x.\lambda y.(\Omega \oplus I); \qquad N = \lambda x.(\lambda y.\Omega) \oplus (\lambda y.I).$$

  - Of course, $I \not\sim \Omega$ and as a consequence

$$\lambda y.\Omega \not\sim \lambda y.I \not\sim \lambda y.(\Omega \oplus I) \implies M \not\sim N.$$

Full Abstraction?

- $\sim$ is a **sound** methodology for program equivalence.
- Is it also **complete**?
- **CBN** : **No** [DLSA2014]
  - Counterexample:

$$M = \lambda x.\lambda y.(\Omega \oplus I); \qquad N = \lambda x.(\lambda y.\Omega) \oplus (\lambda y.I).$$

  - Of course, $I \not\sim \Omega$ and as a consequence

$$\lambda y.\Omega \not\sim \lambda y.I \not\sim \lambda y.(\Omega \oplus I) \implies M \not\sim N.$$

  - On the other hand, $M \equiv N$.
    - We need a CIU-Theorem for that.

Full Abstraction?

- $\sim$ is a **sound** methodology for program equivalence.
- Is it also **complete**?
- **CBN** : **No** [DLSA2014]
  - Counterexample:

$$M = \lambda x.\lambda y.(\Omega \oplus I); \qquad N = \lambda x.(\lambda y.\Omega) \oplus (\lambda y.I).$$

  - Of course, $I \not\sim \Omega$ and as a consequence

$$\lambda y.\Omega \not\sim \lambda y.I \not\sim \lambda y.(\Omega \oplus I) \implies M \not\sim N.$$

  - On the other hand, $M \equiv N$.
    - We need a CIU-Theorem for that.
- **CBV**
  - The counterexample above cannot be easily adapted.
  - Contexts seem to be more powerful.

Full Abstraction in CBV

- **Tests**: $t \ ::= \ \omega \ \Big| \ a \cdot t \ \Big| \ \langle t, t \rangle.$
- **Semantics of Tests**

$$P_{\mathcal{M}}(x, \omega) = 1; \qquad P_{\mathcal{M}}(x, a \cdot t) = \sum_{s \in \mathcal{S}} \mathcal{P}(x, a, s) \cdot P_{\mathcal{M}}(s, t)$$

$$P_{\mathcal{M}}(x, \langle t, s \rangle) = P_{\mathcal{M}}(x, t) \cdot P_{\mathcal{M}}(x, s).$$

Theorem (vBMMW2004)

$x \sim y$ iff for every test t it holds that $P_{\mathcal{M}}(x, t) = P_{\mathcal{M}}(y, t)$.

Full Abstraction in CBV

- **Tests**: $t \ ::= \ \omega \ \Big| \ a \cdot t \ \Big| \ \langle t, t \rangle.$
- **Semantics of Tests**

$$P_{\mathcal{M}}(x, \omega) = 1; \qquad P_{\mathcal{M}}(x, a \cdot t) = \sum_{s \in \mathcal{S}} \mathcal{P}(x, a, s) \cdot P_{\mathcal{M}}(s, t)$$

$$P_{\mathcal{M}}(x, \langle t, s \rangle) = P_{\mathcal{M}}(x, t) \cdot P_{\mathcal{M}}(x, s).$$

Theorem (vBMMW2004)

$x \sim y$ iff for every test t it holds that $P_{\mathcal{M}}(x, t) = P_{\mathcal{M}}(y, t)$.

- But the question now is: are contexts powerful enough to implement every possible test?

Full Abstraction in CBV

- Contexts do **not** have the necessary discriminating power in **CBN**.
  - Conjecture: only tests in the form $\langle t_1, \ldots, t_n \rangle$ where each t_i is a *trace* can be captured.
- In **CBV** evaluation, terms can be copied *after* being evaluated!

Full Abstraction in CBV

- Contexts do **not** have the necessary discriminating power in **CBN**.
  - Conjecture: only tests in the form $\langle t_1, \ldots, t_n \rangle$ where each t_i is a *trace* can be captured.
- In **CBV** evaluation, terms can be copied *after* being evaluated!
- **Lemma**. For every test t there is a context C_t which is equivalent to t in CBV.

Full Abstraction in CBV

- Contexts do **not** have the necessary discriminating power in **CBN**.
  - Conjecture: only tests in the form $\langle t_1, \ldots, t_n \rangle$ where each t_i is a *trace* can be captured.
- In **CBV** evaluation, terms can be copied *after* being evaluated!
- **Lemma**. For every test t there is a context C_t which is equivalent to t in CBV.
- **Theorem**. In CBV, $\sim$ and $\equiv$ coincide.

How About Simulation (in CBV)?

- Similarity can itself be characterized by a notion of testing, but for a **stronger** notion of test.
  - General boolean tests are allowed, including **disjunctive** tests.

How About Simulation (in CBV)?

- Similarity can itself be characterized by a notion of testing, but for a **stronger** notion of test.
  - General boolean tests are allowed, including **disjunctive** tests.
  - The grammar of test needs to be enriched:
    $t ::= \omega \;\big|\; a \cdot t \;\big|\; \langle t, t \rangle \;\big|\; t \vee t \;\big|\; \ldots.$

How About Simulation (in CBV)?

- Similarity can itself be characterized by a notion of testing, but for a **stronger** notion of test.
  - General boolean tests are allowed, including **disjunctive** tests.
  - The grammar of test needs to be enriched:
    $t ::= \omega \mid a \cdot t \mid \langle t, t \rangle \mid t \vee t \mid \ldots.$
- Let us look at the counterexample for CBN:

$$M = \lambda x.\lambda y.(\Omega \oplus I); \qquad N = \lambda x.(\lambda y.\Omega) \oplus (\lambda y.I).$$

  - The two terms are incomparable by $\precsim$.
  - But how about context equivalence?

How About Simulation (in CBV)?

- Similarity can itself be characterized by a notion of testing, but for a **stronger** notion of test.
  - General boolean tests are allowed, including **disjunctive** tests.
  - The grammar of test needs to be enriched:
    $t \ ::= \ \omega \ \big| \ a \cdot t \ \big| \ \langle t, t \rangle \ \big| \ t \vee t \ \big| \ \ldots.$
- Let us look at the counterexample for CBN:

$$M = \lambda x.\lambda y.(\Omega \oplus I); \qquad N = \lambda x.(\lambda y.\Omega) \oplus (\lambda y.I).$$

  - The two terms are incomparable by $\precsim$.
  - But how about context equivalence?
  - **Lemma**. $M \leq N$.
    Proof. Purely operational.

Our Neighborhood

- Λ, where we observe **convergence**

	$\sim \subseteq \equiv$	$\equiv \subseteq \sim$	$\precsim \subseteq \leq$	$\leq \subseteq \precsim$
CBN	✓	✓	✓	✓
CBV	✓	✓	✓	✓

[Abramsky1990,Howe1993]

- $\Lambda_\oplus$ with nondeterministic semantics, where we observe **convergence**, in its **may** or **must** flavors.

	$\sim \subseteq \equiv$	$\equiv \subseteq \sim$	$\precsim \subseteq \leq$	$\leq \subseteq \precsim$
CBN	✓	✗	✓	✗
CBV	✓	✗	✓	✗

[Ong1993,Lassen1998]

Conclusions

- Summing up:

	$\sim \subseteq \equiv$	$\equiv \subseteq \sim$	$\precsim \subseteq \leq$	$\leq \subseteq \precsim$
CBN	✓	✗	✓	✗
CBV	✓	✓	✓	✗

- **Further work:**
  - What if we add **sequencing** to CBN?
  - What if we add **parallel or** to CBN?
  - How about **approximate** notions of bisimulation?
  - How about λ-calculi for probabilistic polynomial time?

Questions?

Howe's Technique

$\mathcal{R}$ $\mathcal{R}^H$

Howe's Technique

$$\begin{tikzcd} \mathcal{R} \arrow[rr, bend left, "\subseteq"] & & \mathcal{R}^H \end{tikzcd}$$

Howe's Technique

$\mathcal{R}^H$ is a
Congruence
whenever $\mathcal{R}$ is
an equivalence

$$\begin{tikzcd} \mathcal{R} \arrow[rr, bend left, "\subseteq"] & & \mathcal{R}^H \end{tikzcd}$$

Howe's Technique

$\sim^H$ is a
Congruence

$$\sim \xrightarrow{\subseteq} \sim^H$$

Howe's Technique

$\sim^H$ is a **Congruence**

$$\begin{tikzcd}
\sim \arrow[rr, bend left, "\subseteq"] & & \sim^H \arrow[ll, bend left, "\supseteq"]
\end{tikzcd}$$

?

Howe's Technique

$$\frac{\overline{x} \vdash x \,\mathcal{R}\, M}{\overline{x} \vdash x \,\mathcal{R}^H\, M} \qquad \frac{\overline{x} \cup \{x\} \vdash M \,\mathcal{R}^H\, L \qquad \overline{x} \vdash \lambda x.L \,\mathcal{R}\, N \qquad x \notin \overline{x}}{\overline{x} \vdash \lambda x.M \,\mathcal{R}^H\, N}$$

$$\frac{\overline{x} \vdash M \,\mathcal{R}^H\, P \qquad \overline{x} \vdash N \,\mathcal{R}^H\, T \qquad \overline{x} \vdash (PT) \,\mathcal{R}\, L}{\overline{x} \vdash MN \,\mathcal{R}^H\, L}$$

$$\frac{\overline{x} \vdash M \,\mathcal{R}^H\, P \qquad \overline{x} \vdash N \,\mathcal{R}^H\, T \qquad \overline{x} \vdash (P \oplus T) \,\mathcal{R}\, L}{\overline{x} \vdash M \oplus N \,\mathcal{R}^H\, L}$$

The Key Lemma

- Proving that $\precsim^H$ is indeed a precongruence is a convenient way to proceed.
- **Statement**: If $M \precsim^H N$, then for every $X \subseteq \Lambda_\oplus(x)$ it holds that $[\![M]\!](\lambda x.X) \leq [\![N]\!](\lambda x.(\precsim^H (X)))$.
- **Proof**.
  - We prove that $\mathscr{D}(\lambda x.X) \leq [\![N]\!](\lambda x.(\precsim^H (X)))$ for every $\mathscr{D}$ such that $M \Downarrow \mathscr{D}$.
  - By induction on the structure of any derivation of $M \Downarrow \mathscr{D}$ (which is finite).
  - Everything goes through smoothly, except…the application case.
  - We need to prove that probability assignments can always be *disentangled*. This is the case, though.

- So we have :

$$
\begin{aligned}
\precsim^H \subseteq \precsim \quad & \Longrightarrow \quad \precsim^H = \precsim \\
& \Longrightarrow \quad \precsim \text{ is a precongruence} \\
& \Longrightarrow \quad \sim \text{ is a congruence} \\
& \Longrightarrow \quad \sim \subseteq \equiv .
\end{aligned}
$$

Theorem

$\sim \subseteq \equiv$