# Probabilistic couplings for cryptography and privacy

Gilles Barthe

IMDEA Software Institute, Madrid, Spain

# Relational verification

- Two programs: relative correctness, program equivalence, translation validation…
- Two runs of the same program: stability, information flow security, truthfulness…

For security and privacy:

- Two programs: provable security
- Two runs of the same program: side-channel resistance, differential privacy

**Programs are probabilistic**

- S. Halevi: A plausible approach to computer-aided cryptographic proofs
- M. Bellare and P. Rogaway: Code-Based Game-Playing Proofs and the Security of Triple Encryption

**Call for Papers**
**CRYPTO 2011**

## General Information

Original papers on all technical aspects of cryptology are solicited for submission to CRYPTO 2011, the 31st Annual International Cryptology Conference. Besides the usual topics, submissions are also welcome on topics not routinely appearing at recent CRYPTOs, including cryptographic work in the style of the CHES workshop or CSF symposium. CRYPTO 2011 is sponsored by the International Association for Cryptologic Research (IACR), in cooperation with the Computer Science Department of the University of California, Santa Barbara.

# Computer-aided cryptography

Develop tool-assisted methodologies for design, analysis, and implementation of cryptographic constructions (primitives and protocols)

# Facets of computer-aided cryptography

- Symbolic security
- Provable security in computational model
- Side-channel resistance
- Verified implementations
- Automated synthesis of secure constructions
- Automated synthesis of physical attacks
- Automated analysis cryptographic of assumptions

# Benefits

Formal methods for cryptography

- higher assurance
- smaller gap between provable security and crypto engineering
- new proof techniques

Cryptography for formal methods

- Many new and challenging examples
- New theories

# Contents

- Couplings
- Probabilistic Relational Hoare Logic
- Provable security
- Approximate couplings
- Approximate Probabilistic Relational Hoare Logic
- Differential Privacy

# Probabilistic couplings (Doeblin, 1938)

- Given: two distributions $\mu_1$ over $A_1$ and $\mu_2$ over $A_2$
- Produce: distribution $\mu$ over $A_1 \times A_2$ that captures the behavior of $\mu_1$ and $\mu_2$ (via marginals)
- Such that relation $R$ is satisfied

**Coupling fair coins:** let $\mu_1, \mu_2$ be u.i.d. over $\{0, 1\}$.

- trivial coupling: $\mu(x, y) = \frac{1}{4}$
- equality coupling: $\mu(x, x) = \frac{1}{2}$ and $\mu(x, \neg x) = 0$
- inequality coupling: $\mu(x, \neg x) = \frac{1}{2}$ and $\mu(x, x) = 0$

# One-dimensional random walk

- Start at initial position $s$
- Each iteration, flip a fair coin
- Heads: $p \leftarrow p + 1$
- Tails: $p \leftarrow p - 1$

Goal: show memorylessness, i.e. two random walks starting at $s_1$ and $s_2$ converge at the limit

# Coupling the walks to meet

Assume $s_2 - s_1 = 2k$.

**Case $p_1 = p_2$: walks have met**

- Arrange samplings $x_1 = x_2$
- Continue to have $p_1 = p_2$

**Case $p_1 \neq p_2$: walks have not met**

- Arrange samplings $x_1 = \neg x_2$
- Walks make mirror moves

# Coupling the walks to meet

Assume $s_2 - s_1 = 2k$.

**Case $p_1 = p_2$: walks have met**

- Arrange samplings $x_1 = x_2$
- Continue to have $p_1 = p_2$

**Case $p_1 \neq p_2$: walks have not met**

- Arrange samplings $x_1 = \neg x_2$
- Walks make mirror moves

# Proving memorylessness

Invariant: $(\exists i \leq t.\ p_1(i) = p_2(i)) \implies p_1(t) = p_2(t)$

Consequence: for every number of steps $t$ and position $x$,

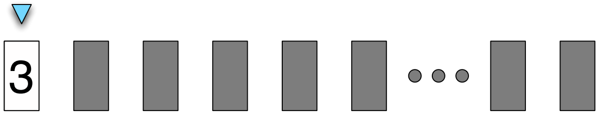
$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i \leq t.\ p_1(i) = s_1 + k]$$

(Question: why not $p_1(i) = p_2(i)$?)

# Shift coupling: Dynkin's trick

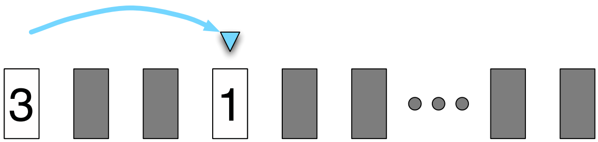
- Input: position in $\{1, \ldots, 9\}$
- Repeat:
  - Draw uniformly random card $\in \{1, \ldots, 9\}$
  - Go forward that many steps
- Output last position before crossing $100$

# In pictures



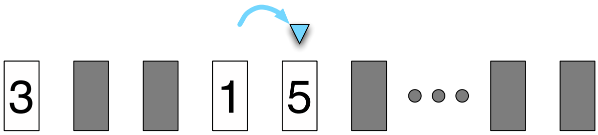

Output last position: 99

# In pictures



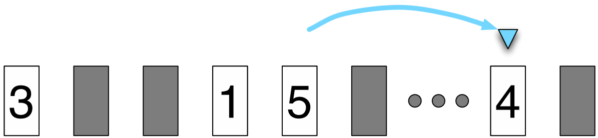

Output last position: 99

# In pictures



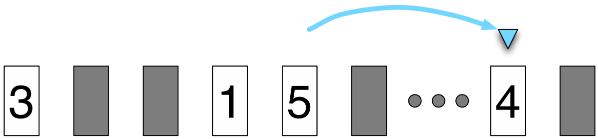

Output last position: 99

# In pictures



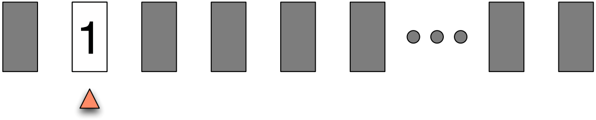

Output last position: 99

# In pictures



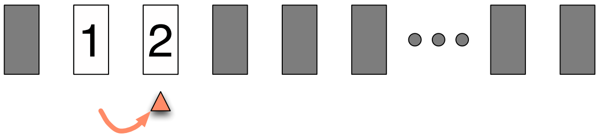

Output last position: 99

# Starting at a different position



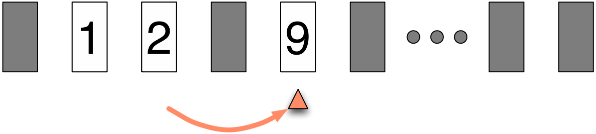

How close are the two output distributions?

# Starting at a different position



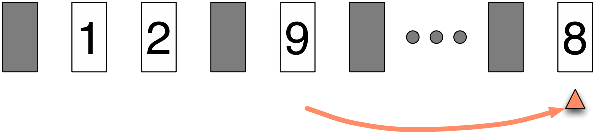

How close are the two output distributions?

# Starting at a different position



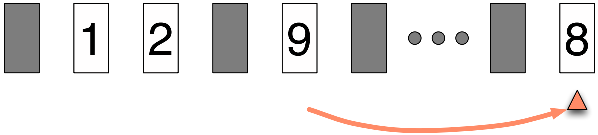

How close are the two output distributions?

# Starting at a different position



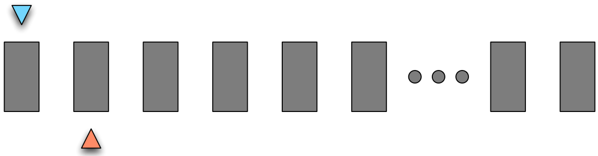

How close are the two output distributions?

# Starting at a different position




How close are the two output distributions?

# Combine first process and second process

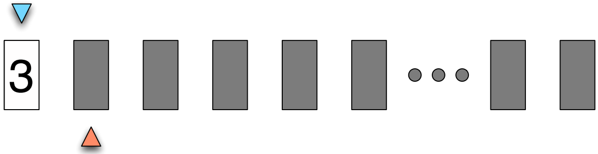


Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

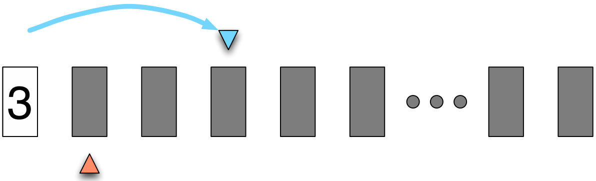



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

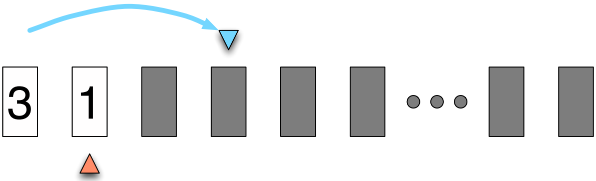



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

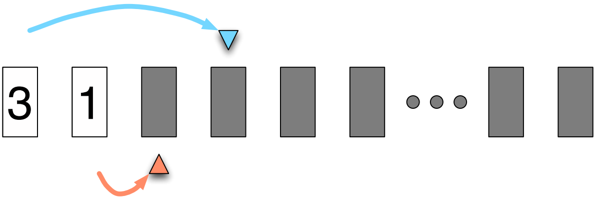



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

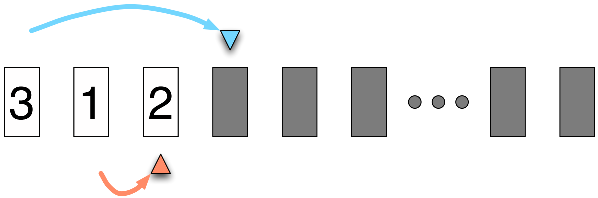



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

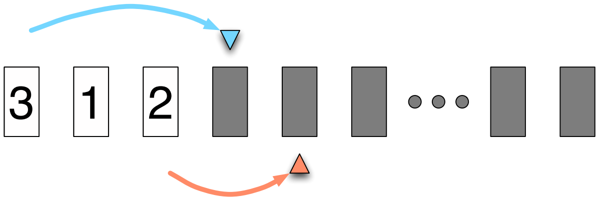



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

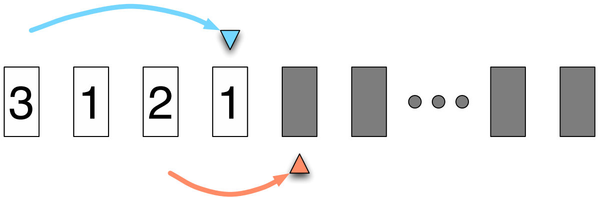



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

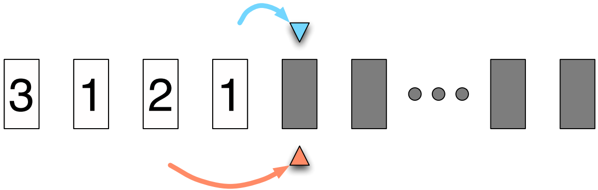



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

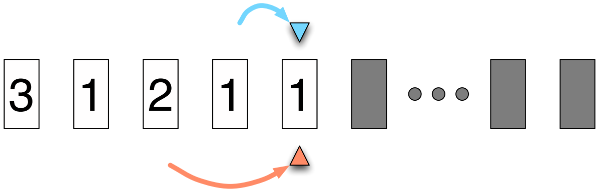



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

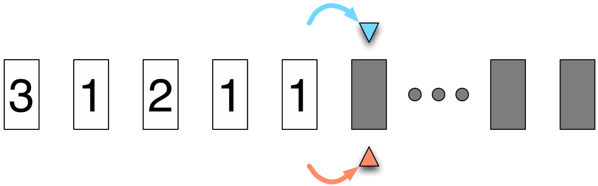



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

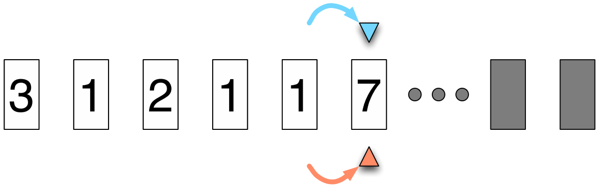



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

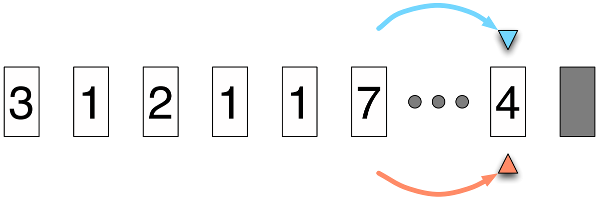



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process

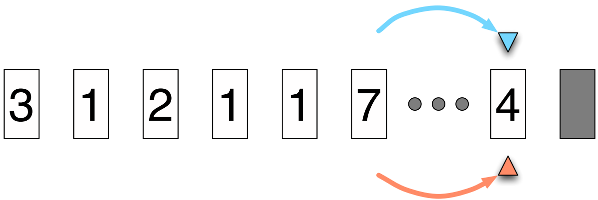



Consequence: for every number of steps $t$ and position $x$,

$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

# Combine first process and second process




Consequence: for every number of steps $t$ and position $x$,

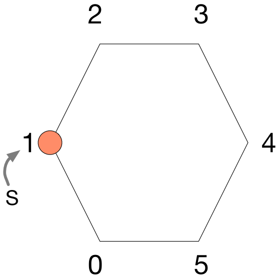
$$|\Pr[p_1(t) = x] - \Pr[p_2(t) = x]| \leq \Pr[\exists i, j \leq t.\ p_1(i) = p_2(j)]$$

(where $p_1$ and $p_2$ are taken from the coupled process)

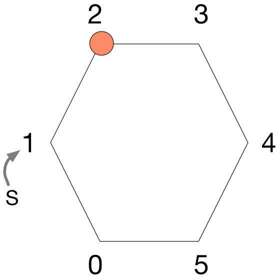
# Random walk over a circle

- Start at position $s \in \{0, 1, \ldots, n-1\}$
- Each iteration, flip a fair coin
  - Heads: increment position (modulo $n$)
  - Tails decrement position (modulo $n$)
- Return: last edge $(r, r+1)$ to be traversed

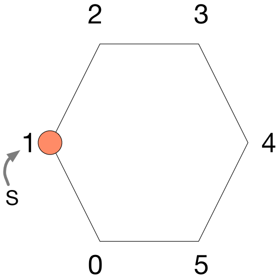
# Random walk over a cycle



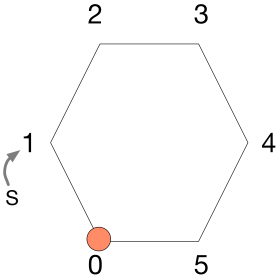
# Random walk over a cycle



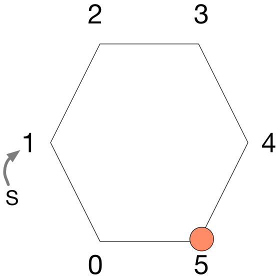
# Random walk over a cycle



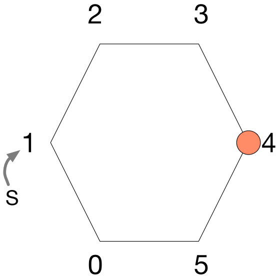
# Random walk over a cycle



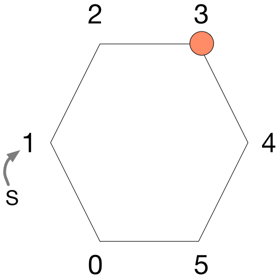
# Random walk over a cycle



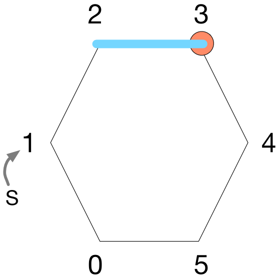
# Random walk over a cycle



# Random walk over a cycle



# Random walk over a cycle



# Random walk over a cycle

How is the returned edge distributed relative to starting position $s$?

# Preliminaries

**Discrete sub-distributions:** $\mathsf{Distr}(A)$ is the set of functions $\mu : A \to [0,1]$ s.t.

- $\mathsf{supp}(\mu) = \{a \in A \mid \mu(a) > 0\}$ of $\mu$ is discrete;
- $|\mu| = \sum_{a \in A} \mu(a)$ of $\mu$ is defined and verifies $|\mu| \leq 1$.

**Marginals:** given $\mu \in \mathsf{Distr}(A_1 \times A_2)$ define $\pi_1(\mu) \in \mathsf{Distr}(A_1)$ and $\pi_2(\mu) \in \mathsf{Distr}(A_2)$ by

$$\pi_1(\mu)(a_1) = \sum_{a_2 \in A_2} \mu(a_1, a_2) \qquad \pi_2(\mu)(a_2) = \sum_{a_1 \in A_1} \mu(a_1, a_2)$$

# $R$-couplings

Let $R \subseteq A_1 \times A_2$, and $\mu_1 \in \mathrm{Distr}(A_1)$ and $\mu_2 \in \mathrm{Distr}(A_2)$. Then $\mu \in \mathrm{Distr}(A_1 \times A_2)$ is a $R$-coupling for $(\mu_1, \mu_2)$ iff:

- **marginals:** $\pi_1(\mu) = \mu_1$ and $\pi_2(\mu) = \mu_2$
- **support:** $\mathrm{supp}(\mu) \subseteq R$

Notation: $\mu \blacktriangleleft_R \langle \mu_1 \,\&\, \mu_2 \rangle$, or $\blacktriangleleft_R \langle \mu_1 \,\&\, \mu_2 \rangle$

**Original definition**

- does not include support condition ($\top$-coupling)
- restricted to full distributions

# (In)equality couplings

Let $A_1 = A_2 = A$.

**Stochastic dominance:** Assume $(A, \leq)$ is a partial order. Then the following are equivalent:

- $\blacktriangleleft_{\leq} \langle \mu_1 \ \& \ \mu_2 \rangle$
- for every $a$, $\mu_1(\{x \in \mathbb{Z} \mid x \geq a\}) \leq \mu_2(\{x \in \mathbb{Z} \mid x \geq a\})$

**Equality couplings:** Assume $|\mu_1| = |\mu_2| = 1$. Then the following are equivalent:

- $\mu_1 = \mu_2$
- $\blacktriangleleft_{=} \langle \mu_1 \ \& \ \mu_2 \rangle$
- for every $a \in A$, $\blacktriangleleft_{x_1 = a \implies x_2 = a} \langle \mu_1 \ \& \ \mu_2 \rangle$

# Fundamental theorem of $R$-couplings

Let $E_1 \subseteq A_1$ and $E_2 \subseteq A_2$. Let $R \subseteq A_1 \times A_2$ s.t. every $(a_1, a_2) \in A_1 \times A_2$,

$$(a_1, a_2) \in R \wedge a_1 \in E_1 \implies a_2 \in E_2$$

If $\blacktriangleleft_R \langle \mu_1 \ \& \ \mu_2 \rangle$ then $\Pr_{\mu_1}[E_1] \leq \Pr_{\mu_2}[E_2]$.

- Bridging step: if $(a_1, a_2) \in R \Rightarrow (a_1 \in E_1 \Leftrightarrow a_2 \in E_2)$, then

$$\Pr_{\mu_1}[E_1] = \Pr_{\mu_2}[E_2]$$

- Failure event: if $(a_1, a_2) \in R \wedge a_1 \in E_1 \Rightarrow a_2 \in E_2 \cup F$, then

$$\Pr_{\mu_1}[E_1] - \Pr_{\mu_2}[E_2] \leq \Pr_{\mu_2}[F]$$

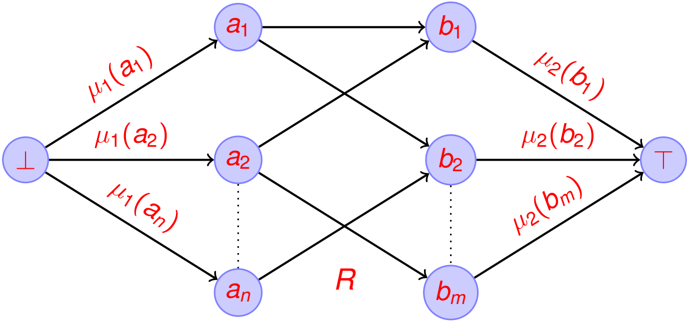
# Existence of $R$-couplings (Strassen, 1965)

For every $\mu_1 \in \mathrm{Distr}(A_1)$ and $\mu_2 \in \mathrm{Distr}(A_2)$ s.t. $|\mu_1| = |\mu_2| = 1$, the following are equivalent:

- $\blacktriangleleft_R \langle \mu_1 \ \& \ \mu_2 \rangle$
- for every $X \subseteq A_1$, $\mu_1(X) \leq \mu_2(R(X))$

# $R$-couplings and optimal transport

$\blacktriangleleft_R \langle \mu_1 \,\&\, \mu_2 \rangle$ iff the maximum flow in the following network is 1:



# Sequential composition of $R$-couplings

**Composition of probabilistic mappings**

Let $\mu \in \mathsf{Distr}(A)$ and $M : A \to \mathsf{Distr}(B)$; set $\mathbb{E}_\mu[M] \in \mathsf{Distr}(B)$

$$\mathbb{E}_\mu[M](b) \triangleq \sum_{a \in \mathsf{supp}(\mu)} \mu(a) M(a)(b)$$

Assume that:

- $\blacktriangleleft_R \langle \mu_1 \ \& \ \mu_2 \rangle$
- $\blacktriangleleft_S \langle M_1(a_1) \ \& \ M_2(a_2) \rangle$ for every $(a_1, a_2) \in R$

Then $\blacktriangleleft_S \langle \mathbb{E}_{\mu_1}[M_1] \ \& \ \mathbb{E}_{\mu_2}[M_2] \rangle$ where:

- $R \subseteq A_1 \times A_2$ and $S \subseteq B_1 \times B_2$
- $\mu_1 \in \mathsf{Distr}(A_1)$ and $M_1 : A_1 \to \mathsf{Distr}(B_1)$
- $\mu_2 \in \mathsf{Distr}(A_2)$, and $M_2 : A_2 \to \mathsf{Distr}(B_2)$

# Other properties of $R$-couplings

- Trivial couplings:
  $\blacktriangleleft_{\top} \langle \mu_1 \mathbin{\&} \mu_2 \rangle$ iff $|\mu_1| = |\mu_2|$
- Monotonicity:
  if $\mu \blacktriangleleft_R \langle \mu_1 \mathbin{\&} \mu_2 \rangle$ and $R \subseteq S$ then $\mu \blacktriangleleft_S \langle \mu_1 \mathbin{\&} \mu_2 \rangle$
- Closed under relation composition:
  if $\blacktriangleleft_R \langle \mu_1 \mathbin{\&} \mu_2 \rangle$ and $\blacktriangleleft_S \langle \mu_2 \mathbin{\&} \mu_3 \rangle$ then $\blacktriangleleft_{R \circ S} \langle \mu_1 \mathbin{\&} \mu_3 \rangle$
- Closed under convex combinations:
  if $\blacktriangleleft_R \langle \mu_{1,i} \mathbin{\&} \mu_{2,i} \rangle$ for every $i \in I$ and $\sum_{i \in I} p_i \leq 1$ then
  $\blacktriangleleft_R \langle \sum_{i \in I} p_i \mu_{1,i} \mathbin{\&} \sum_{i \in I} p_i \mu_{2,i} \rangle$

# Summary and outlook

- Relational verification matters
- Couplings naturally support relational reasoning
- Probabilities are hidden
- Some examples need more general notions of couplings

# Programming language

$$
\begin{array}{rll}
c ::= & \mathsf{abort} & \text{abort} \\
| & \mathsf{skip} & \text{noop} \\
| & x \leftarrow e & \text{deterministic assignment} \\
| & x \xleftarrow{\$} d & \text{probabilistic assignment} \\
| & c; c & \text{sequencing} \\
| & \mathsf{if}\ e\ \mathsf{then}\ c\ \mathsf{else}\ c & \text{conditional} \\
| & \mathsf{while}\ e\ \mathsf{do}\ c & \text{while loop} \\
| & x \leftarrow \mathcal{F}(e) & \text{procedure call}
\end{array}
$$

**Semantics:** for every initial memory $m \in \mathbf{Mem}$, compute output sub-distribution of memories $[\![s]\!]_m \in \mathsf{Distr}(\mathbf{Mem})$

# Denotational semantics

$$
\begin{aligned}
[\![\text{abort}]\!]_m &= \mathbf{0} \\
[\![\text{skip}]\!]_m &= \mathbb{1}_m \\
[\![x \leftarrow e]\!]_m &= \mathbb{1}_{m[x \leftarrow [\![e]\!]_m]} \\
[\![x \xleftarrow{\$} d]\!]_m &= \mathbf{E}_{v \sim [\![d]\!]_m}[\mathbb{1}_{m[x \leftarrow v]}] \\
[\![c_1 ; c_2]\!]_m &= \mathbf{E}_{\xi \sim [\![c_1]\!](m)}[[\![c_2]\!](\xi)] \\
[\![\text{if } e \text{ then } c_1 \text{ else } c_2]\!]_m &= \begin{cases} [\![c_1]\!]_m & \text{if } [\![e]\!]_m = \top \\ [\![c_2]\!]_m & \text{if } [\![e]\!]_m = \bot \end{cases} \\
[\![\text{while } e \text{ do } c]\!]_m &= \lim_{i \in \mathbb{N}} \left( \mathbf{E}_{\xi \sim [\![(\text{if } e \text{ then } c)^i]\!]_m}[[\![\text{if } e \text{ then abort}]\!]_\xi] \right)
\end{aligned}
$$

# pRHL judgments and validity

**Judgment:**

$$\models c_1 \sim c_2 : \Phi \Rightarrow \Psi$$

where $\Phi, \Psi \subseteq \mathbf{Mem} \times \mathbf{Mem}$

**Validity:** for every $(m_1, m_2)$ s.t. $(m_1, m_2) \in \Phi$,

$$\blacktriangleleft_\Psi \langle [\![c_1]\!]_{m_1} \ \& \ [\![c_2]\!]_{m_2}) \rangle$$

**Proof rules:**

- *structural rules*: apply to all programs
- *2-sided rules*: both programs have the same specific shape
- *1-sided rules*: one program has a specific shape

# Structural rules

$$\frac{\models c_1 \sim c_2 : \Phi' \Rightarrow \Psi' \qquad \Phi \Longrightarrow \Phi' \qquad \Psi' \Longrightarrow \Psi}{\models c_1 \sim c_2 : \Phi \Rightarrow \Psi} \text{ [Conseq]}$$

$$\frac{\begin{array}{c}\models c_1 \sim c_2 : \Phi \Rightarrow \Psi \\ \mathrm{vars}(\Theta) \cap (\mathrm{mod}(c_1)\langle 1\rangle \cup \mathrm{mod}(c_2)\langle 2\rangle) = \emptyset\end{array}}{\models c_1 \sim c_2 : \Phi \wedge \Theta \Rightarrow \Psi \wedge \Theta} \text{ [Frame]}$$

$$\frac{\models c_1 \sim c_2 : \Phi_1 \Rightarrow \Psi \qquad \models c_1 \sim c_2 : \Phi_2 \Rightarrow \Psi}{\models c_1 \sim c_2 : \Phi_1 \vee \Phi_2 \Rightarrow \Psi} \text{ [Case]}$$

# Two-sided rules

$$\frac{\models c_1 \sim c_2 : \Phi \Rightarrow \Theta \qquad \models c_1' \sim c_2' : \Theta \Rightarrow \Psi}{\models c_1; c_1' \sim c_2; c_2' : \Phi \Rightarrow \Psi} \text{ [SEQ]}$$

$$\frac{}{\models x_1 \leftarrow e_1 \sim x_2 \leftarrow e_2 : \Psi[e_1\langle 1\rangle / x_1\langle 1\rangle][e_2\langle 2\rangle / x_2\langle 2\rangle] \Rightarrow \Psi} \text{ [ASSN]}$$

$$\frac{\begin{array}{c}\Phi \implies e_1\langle 1\rangle = e_2\langle 2\rangle \\ \models c_1 \sim c_2 : \Phi \wedge e_1\langle 1\rangle \Rightarrow \Psi \\ \models c_1' \sim c_2' : \Phi \wedge \neg e_1\langle 1\rangle \Rightarrow \Psi\end{array}}{\models \text{if } e_1 \text{ then } c_1 \text{ else } c_1' \sim \text{if } e_2 \text{ then } c_2 \text{ else } c_2' : \Phi \Rightarrow \Psi} \text{ [COND]}$$

$$\frac{\Theta \implies e_1\langle 1\rangle = e_2\langle 2\rangle \qquad \models c_1 \sim c_2 : \Theta \wedge e_1\langle 1\rangle \Rightarrow \Theta}{\models \text{while } e_1 \text{ do } c_1 \sim \text{while } e_2 \text{ do } c_2 : \Theta \Rightarrow \Theta \wedge \neg e_1\langle 1\rangle} \text{ [WHILE]}$$

# One-sided rules

$$\frac{}{\models x_1 \leftarrow e_1 \sim \mathsf{skip} : \Psi[e_1\langle 1\rangle / x_1\langle 1\rangle] \Rightarrow \Psi} \text{ [Assg-L]}$$

$$\frac{\models c_1 \sim c_2 : \Phi \wedge e_1\langle 1\rangle \Rightarrow \Psi \qquad \models c_1' \sim c_2 : \Phi \wedge \neg e_1\langle 1\rangle \Rightarrow \Psi}{\models \mathsf{if}\ e_1\ \mathsf{then}\ c_1\ \mathsf{else}\ c_1' \sim c_2 : \Phi \Rightarrow \Psi} \text{ [Cond-L]}$$

$$\frac{\models c_1 \sim \mathsf{skip} : \Theta \wedge e_1\langle 1\rangle \Rightarrow \Theta \qquad \mathsf{ast}(\mathsf{while}\ e_1\ \mathsf{do}\ c_1)}{\models \mathsf{while}\ e_1\ \mathsf{do}\ c_1 \sim \mathsf{skip} : \Theta \Rightarrow \Theta \wedge \neg e_1\langle 1\rangle} \text{ [While-L]}$$

# Random samplings

$$\frac{\begin{array}{c}\blacktriangleleft_{\Theta} \langle [\![\mu_1]\!] \ \& \ [\![\mu_2]\!] \rangle \\ \Phi \triangleq \forall v_1 : T_1,\ v_2 : T_2,\ \Theta \implies \Psi[v_1/x_1\langle 1\rangle][v_2/x_2\langle 2\rangle]\end{array}}{\models x_1 \xleftarrow{\$} \mu_1 \sim x_2 \xleftarrow{\$} \mu_2 : \Phi \Rightarrow \Psi} \ [\textsc{Rand}]$$

$$\frac{}{\models x_1 \xleftarrow{\$} d_1 \sim \mathsf{skip} : \forall v_1 \in \mathsf{supp}(d_1), \Psi[v_1/x_1\langle 1\rangle] \Rightarrow \Psi} \ [\textsc{Rand-L}]$$

# Examples

**Optimistic sampling**

$$\models x_1 \xleftarrow{\$} \mathbb{Z}_p; x_1 = x_1 \oplus k \sim x_2 \xleftarrow{\$} \mathbb{Z}_p : \top \Rightarrow x_1 = x_2$$

Proof: by [Ass-L], must show

$$\models x_1 \xleftarrow{\$} \mathbb{Z}_p \sim x_2 \xleftarrow{\$} \mathbb{Z}_p : \top \Rightarrow x_1 \oplus k = x_2$$

By [Rand] with $\mu(x_1, x_2) = \frac{\mathbb{1}_{x_1 \oplus k = x_2}}{p}$, must show

$$\forall x_1\ x_2,\ x_1 \oplus k = x_2 \implies x_1 \oplus k = x_2$$

**Eager sampling** $\models c_1 \sim c_2 : z_1 = z_2 \Rightarrow x_1 = x_2$ where

$$\begin{aligned} c_1 &\triangleq x_1 \xleftarrow{\$} \mathbb{Z}_p; \text{if } z_1 = 0 \text{ then } z_1 \leftarrow z_1 + x_1 \text{ else } x_1 \leftarrow z_1 \\ c_2 &\triangleq \text{if } z_2 = 0 \text{ then } x_2 \xleftarrow{\$} \mathbb{Z}_p; z_2 \leftarrow z_2 + x_2 \text{ else } x_2 \leftarrow z_2 \end{aligned}$$

# Adversaries

$$[\textsc{Adv}]\ \frac{\forall \mathcal{F}, y, z.\ \models y \leftarrow \mathcal{F}(z) \sim y \leftarrow \mathcal{F}(z) : =_z \wedge \Psi \Rightarrow =_y \wedge \Psi}{\models x_1 \leftarrow \mathcal{A}(e_1) \sim x_2 \leftarrow \mathcal{A}(e_2) : =_e \wedge \Theta \Rightarrow =_x \wedge \Theta}$$

where $\Theta \triangleq \Psi \wedge \mathrm{eqmem}_{\mathcal{A}}$ and $=_e \triangleq z\langle 1\rangle = z\langle 2\rangle$.

# Product programs

- Every proof in pRHL builds a product program
- Product programs can be maded explicit

$$\models c_1 \sim c_2 : \Phi \Rightarrow \Psi \rightsquigarrow c$$

Example:

$$\frac{\models c_1 \sim c_2 : \Phi \wedge \Phi' \Rightarrow \Psi \rightsquigarrow c \qquad \models c_1 \sim c_2 : \Phi \wedge \neg\Phi' \Rightarrow \Psi \rightsquigarrow c_\neg}{\models c_1 \sim c_2 : \Phi \Rightarrow \Psi \rightsquigarrow \text{if } \Phi' \text{ then } c^\times \text{ else } c_\neg^\times}$$

Application: Dynkin's trick

- product program simulates two programs
- bound probability of coinciding in product program