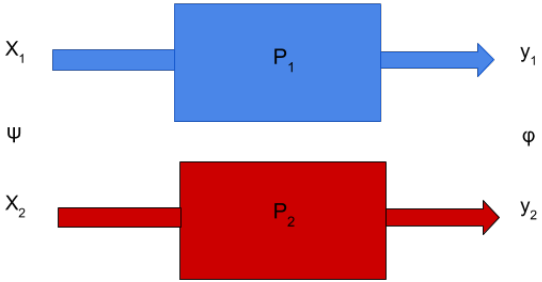
Relational Verification of Probabilistic Programs

Gilles Barthe
IMDEA Software Institute, Madrid, Spain

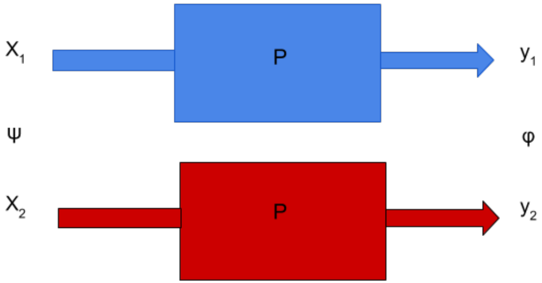
July 27, 2018

Relational properties



- programs P_1 and P_2
- ψ-related inputs yield ϕ-related outputs

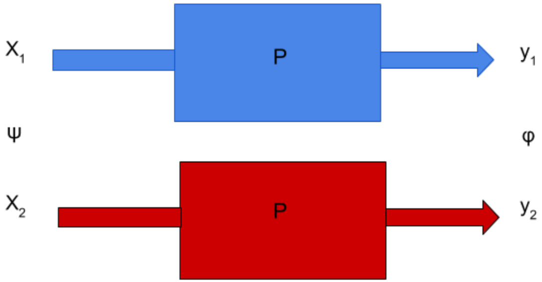
Relational properties



Non-interference

Precondition: $x_1 =_L x_2$
Postcondition: $y_1 =_L y_2$

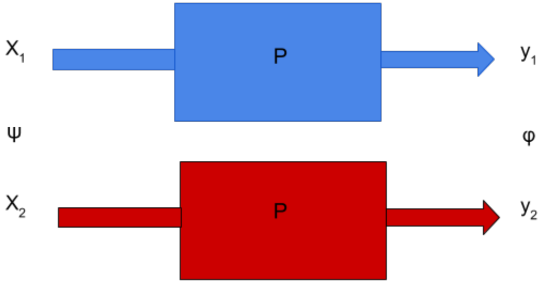
Relational properties



Side-channel leakage

Precondition: $x_1 =_L x_2$
Postcondition: $\ell_1 = \ell_2$

Relational properties

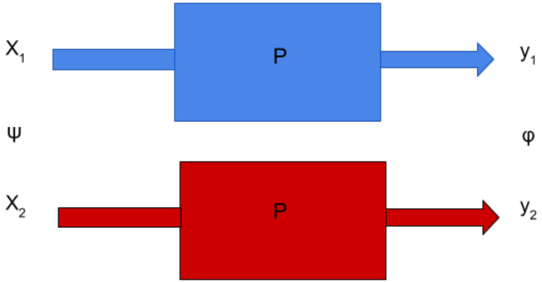


Cryptographic proofs

Postcondition:

$\Pr[\mathcal{A} \text{ breaks scheme}] \leq \lambda \Pr[\mathcal{S} \text{ solves hard problem}] + \epsilon$

Relational properties

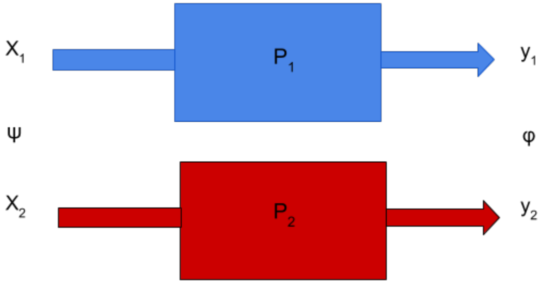


ϵ-differential privacy

Precondition: x_1 and x_2 are adjacent (at distance ≤ 1)

Precondition: $\Pr[y_1 = v] \leq \exp(\epsilon) \Pr[y_2 = v]$

Relational properties

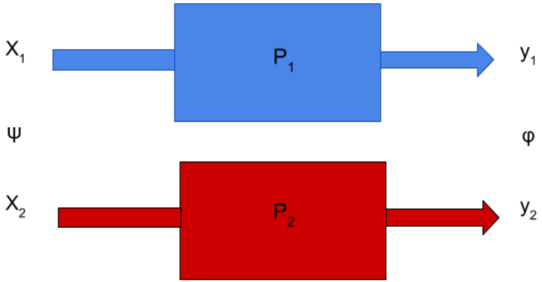


Program equivalence

Precondition: $x_1 = x_2$

Postcondition: $y_1 = y_2$

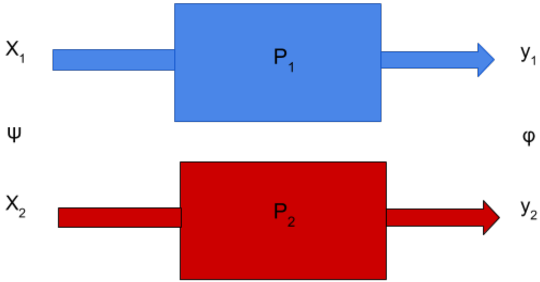
Relational properties



Co-safety

Precondition: $x_1 = x_2$
Postcondition: $y_1 \neq \mathrm{err} \Leftrightarrow y_2 \neq \mathrm{err}$

Relational properties

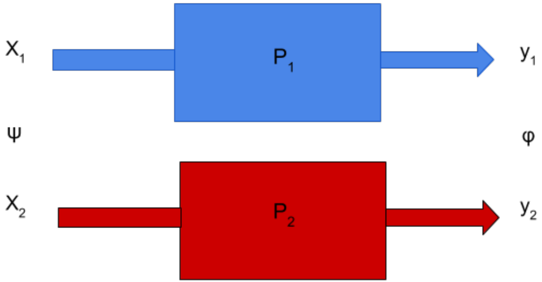


Lipschitz continuity

Precondition: inputs at distance $\leq d$

Postcondition: outputs at distance $\leq k \cdot d$

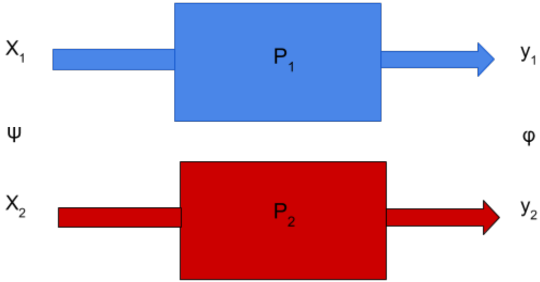
Relational properties



Truthfulness

Precondition: $x_1 = v_1 \wedge x_1' = x_2'$
Postcondition: $\mathrm{payoff}_2 \leq \mathrm{payoff}_1$

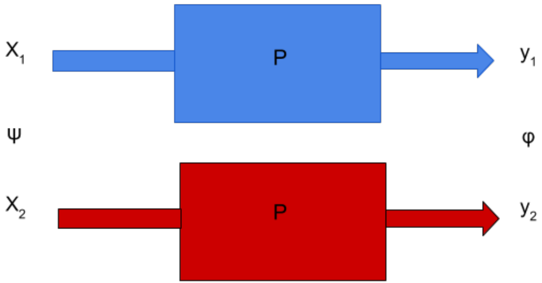
Relational properties



Algorithmic stability

Precondition: inputs adjacent
Postcondition: $|E(\ell_1) - E(\ell_2)| \leq \epsilon$

Relational properties

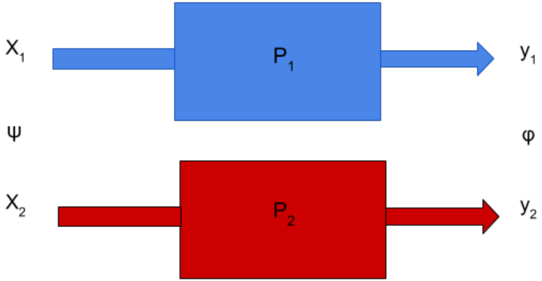


Relative cost

Precondition: $x_1 = x_2$

Postcondition: $\mathrm{cost}_1 - \mathrm{cost}_2 \leq n$

Relational properties



Uniformity

Postcondition: $\Pr[y_1 = a] = \Pr[y_2 = b]$

Verification by Relational Hoare Logic

$$\frac{\{\psi\}c_1 \sim c_2\{\Theta\} \quad \{\Theta\}c_1' \sim c_2'\{\phi\}}{\{\psi\}c_1; c_1' \sim c_2; c_2'\{\phi\}}$$

$$\frac{\{\psi \wedge b_1\}c_1 \sim c_2\{\phi\} \quad \{\psi \wedge \neg b_1\}c_1' \sim c_2'\{\phi\} \quad \psi \implies b_1 = b_2}{\{\psi\}\text{if } b_1 \text{ then } c_1 \text{ then } \ c_1' \sim \text{if } b_2 \text{ then } c_2 \text{ then } \ c_2'\{\phi\}}$$

$$\frac{\{\psi \wedge b_1\}c_1 \sim c_2\{\psi\} \quad \psi \implies b_1 = b_2}{\{\psi\}\text{while } b_1 \text{ do } c_1 \sim \text{while } b_2 \text{ do } c_2\{\psi \wedge \neg b_1\}}$$

$$\frac{\{\psi \wedge b_1\}c_1 \sim c_2\{\phi\} \quad \{\psi \wedge \neg b_1\}c_1' \sim c_2\{\phi\}}{\{\psi\}\text{if } b_1 \text{ then } c_1 \text{ then } \ c_1' \sim c_2\{\phi\}}$$

$$\frac{\{\psi \wedge b_1\}c_1 \sim \text{skip}\{\psi\}}{\{\psi\}\text{while } b_1 \text{ do } c_1 \sim \text{skip}\{\psi \wedge \neg b_1\}}$$

Verification by product constructions

$$\frac{}{c_1 \times c_2 \to c_1 ; c_2}$$

$$\frac{c_1 \times c_2 \to c \quad c_1' \times c_2' \to c'}{c_1 ; c_1' \times c_2 ; c_2' \to c ; c'}$$

$$\frac{c_1 \times c_2 \to c}{\text{while } b_1 \text{ do } c_1 \times \text{while } b_2 \text{ do } c_2 \to \text{assert}(b_1 \Leftrightarrow b_2); \text{while } b_1 \text{ do } (c; \text{assert}(b_1 \Leftrightarrow b_2))}$$

$$\frac{c_1 \times c_2 \to c \quad c_1' \times c_2' \to c'}{\text{if } b_1 \text{ then } c_1 \text{ then } c_1' \times \text{if } b_2 \text{ then } c_2 \text{ then } c_2' \to \text{assert}(b_1 = b_2); \text{if } b_1 \text{ then } c \text{ then } c'}$$

$$\frac{c_1 \times c_2 \to c \quad c_1' \times c_2' \to c'}{\text{if } b_1 \text{ then } c_1 \text{ then } c_1' \times c_2 \to \text{if } b_1 \text{ then } c \text{ then } c'}$$

For deterministic languages

Product programs and relational Hoare logic are equivalent

Probabilistic programs

- Sample from continuous distributions
- Condition wrt boolean-valued or real-valued function

Verification via probabilistic couplings

$\mu \in \mathsf{D}(C_1 \times C_2)$ is a Ψ-coupling for $(\mu_1, \mu_2) \in \mathsf{D}(C_1) \times \mathsf{D}(C_2)$ iff:

- $\pi_1(\mu) = \mu_1$ and $\pi_2(\mu) = \mu_2$ (coupling)
- $\mathsf{supp}(\mu) \subseteq \Psi$ (satisfaction)

probabilistic Relational Hoare Logic

$$\vdash \{\Psi\} c_1 \sim c_2 \{\Phi\}$$

- Validity states existence of coupling
- Probabilities are kept under the hood

Applications

- cryptographic proofs
- side-channel analysis
- differential privacy
- machine learning

The Jasmin project

- High-assurance and high-speed crypto libraries
- Assembly-level guarantees
  - cryptographic strength
  - side-channel resistance
  - functional correctness

  all using relational verification (and compiler correctness)
- Faster than record breaking (unverified) code

Conclusion

- Many properties of interest are relational
- Lots of research opportunities:
  - New properties
  - Generalizations
  - New paradigms
  - Tools
  - Theory of couplings