Relational Verification of Probabilistic Programs

Gilles Barthe
IMDEA Software Institute, Madrid, Spain

July 27, 2018
Relational properties

- programs $P_1$ and $P_2$
- $\psi$-related inputs yield $\phi$-related outputs
Relational properties

Non-interference
Precondition: $x_1 \equiv_L x_2$
Postcondition: $y_1 \equiv_L y_2$
Relational properties

Side-channel leakage

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

Cryptographic proofs

Postcondition:
\[ \Pr[A \text{ breaks scheme}] \leq \lambda \Pr[S \text{ solves hard problem}] + \epsilon \]
Relational properties

$\epsilon$-differential privacy

Precondition: $x_1$ and $x_2$ are adjacent (at distance $\leq 1$)
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: $y_1 \neq \text{err} \iff y_2 \neq \text{err}$
Relational properties

Lipschitz continuity
Precondition: inputs at distance $\leq d$
Postcondition: outputs at distance $\leq k \cdot d$
Relational properties

Truthfulness
Precondition: \( x_1 = v_1 \land x'_1 = x'_2 \)
Postcondition: \( \text{payoff}_2 \leq \text{payoff}_1 \)
Relational properties

Algorithmic stability
Precondition: inputs adjacent
Postcondition: $|E(l_1) - E(l_2)| \leq \epsilon$
Relational properties

Relative cost
Precondition: $x_1 = x_2$
Postcondition: $\text{cost}_1 - \text{cost}_2 \leq n$
Relational properties

Uniformity

Postcondition: $\Pr[y_1 = a] = \Pr[y_2 = b]$
Verification by Relational Hoare Logic

\[
\begin{align*}
\{\psi\} c_1 & \sim c_2 \{\Theta\} & \{\Theta\} c'_1 & \sim c'_2 \{\phi\} \\
\{\psi\} c_1; c'_1 & \sim c_2; c'_2 \{\phi\}
\end{align*}
\]

\[
\begin{align*}
\{\psi \land b_1\} c_1 & \sim c_2 \{\phi\} & \{\psi \land \neg b_1\} c'_1 & \sim c'_2 \{\phi\} & \psi \implies b_1 = b_2 \\
\{\psi\} \text{if } b_1 \text{ then } c_1 \text{ then } c'_1 & \sim \text{if } b_2 \text{ then } c_2 \text{ then } c'_2 \{\phi\}
\end{align*}
\]

\[
\begin{align*}
\{\psi \land b_1\} c_1 & \sim c_2 \{\psi\} & \psi \implies b_1 = b_2 \\
\{\psi\} \text{while } b_1 \text{ do } c_1 & \sim \text{while } b_2 \text{ do } c_2 \{\psi \land \neg b_1\}
\end{align*}
\]

\[
\begin{align*}
\{\psi \land b_1\} c_1 & \sim c_2 \{\phi\} & \{\psi \land \neg b_1\} c'_1 & \sim c'_2 \{\phi\} \\
\{\psi\} \text{if } b_1 \text{ then } c_1 \text{ then } c'_1 & \sim c_2 \{\phi\}
\end{align*}
\]

\[
\begin{align*}
\{\psi \land b_1\} c_1 & \sim \text{skip}\{\psi\} \\
\{\psi\} \text{while } b_1 \text{ do } c_1 & \sim \text{skip}\{\psi \land \neg b_1\}
\end{align*}
\]
Verification by product constructions

\[
\begin{align*}
&c_1 \times c_2 \rightarrow c_1; c_2 \\
&c_1 \times c_2 \rightarrow c \quad c'_1 \times c'_2 \rightarrow c' \\
&\quad \rightarrow c_1; c'_1 \times c_2; c'_2 \rightarrow c; c'
\end{align*}
\]

while \( b_1 \) do \( c_1 \) \( \times \) while \( b_2 \) do \( c_2 \) \( \rightarrow \) assert\((b_1 \iff b_2)\); while \( b_1 \) do \((c; \text{assert}(b_1 \iff b_2))\)

\[
\begin{align*}
&c_1 \times c_2 \rightarrow c \\
&c_1 \times c_2 \rightarrow c \quad c'_1 \times c'_2 \rightarrow c' \\
&\quad \rightarrow \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 \rightarrow \text{assert}(b_1 = b_2)\); \text{if } b_1 \text{ then } c \text{ then } c'
\end{align*}
\]

\[
\begin{align*}
&c_1 \times c_2 \rightarrow c \\
&c_1 \times c_2 \rightarrow c \quad c'_1 \times c'_2 \rightarrow c' \\
&\quad \rightarrow \text{if } b_1 \text{ then } c_1 \text{ then } c'_1 \times c_2 \rightarrow \text{if } b_1 \text{ then } c \text{ then } c'
\end{align*}
\]

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

$\mu \in D(C_1 \times C_2)$ is a $\Psi$-coupling for $(\mu_1, \mu_2) \in D(C_1) \times D(C_2)$ iff:
- $\pi_1(\mu) = \mu_1$ and $\pi_2(\mu) = \mu_2$ (coupling)
- $\text{supp}(\mu) \subseteq \Psi$ (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