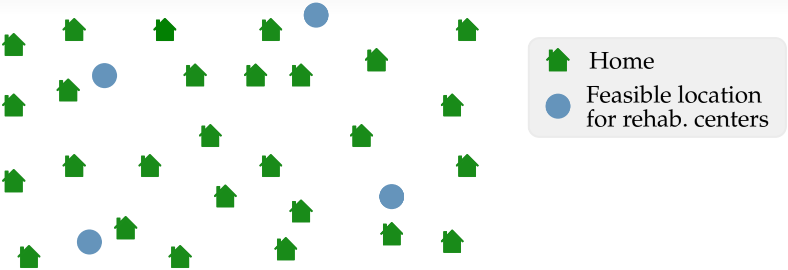
Approximate Relational Reasoning for Probabilistic Programs

PhD Candidate: Federico Olmedo
Supervisor: Gilles Barthe

IMDEA Software Institute

PhD Examination – Technical University of Madrid
January 9, 2014

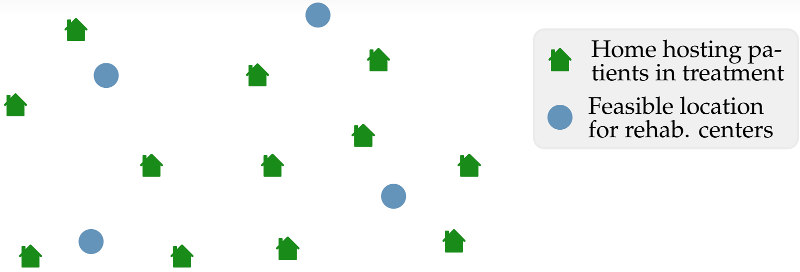
Selecting Locations for Rehabilitation Centers



Scenario: 2 new rehab. centers to be opened; 4 feasible locations.

Goal: select locations that minimize average patient commute time.

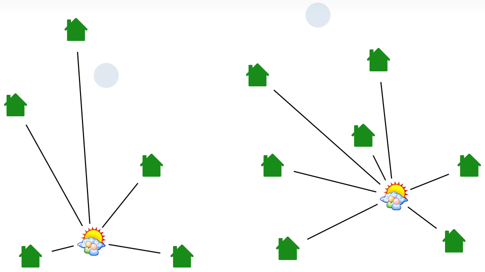
Selecting Locations for Rehabilitation Centers

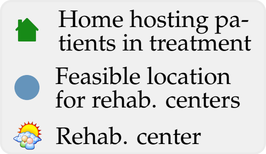


Scenario: 2 new rehab. centers to be opened; 4 feasible locations.

Goal: select locations that minimize average patient commute time.

Selecting Locations for Rehabilitation Centers

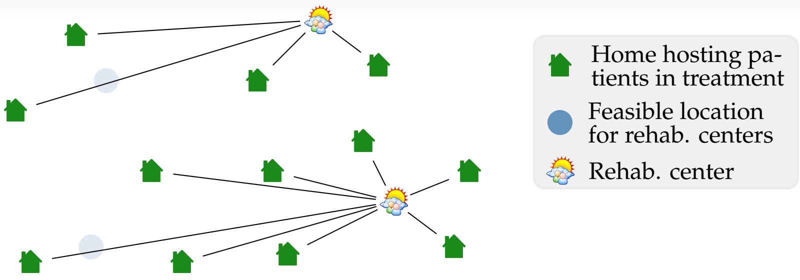




Scenario: 2 new rehab. centers to be opened; 4 feasible locations.

Goal: select locations that minimize average patient commute time.

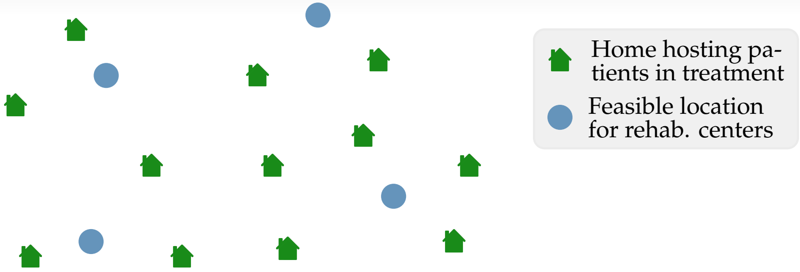
Selecting Locations for Rehabilitation Centers



Scenario: 2 new rehab. centers to be opened; 4 feasible locations.

Goal: select locations that minimize average patient commute time.

Selecting Locations for Rehabilitation Centers



Scenario: 2 new rehab. centers to be opened; 4 feasible locations.

Goal: select locations that minimize average patient commute time.

Optimum Solution Approach:

- Highest utility.
- Leakage of sensitive information.

The Privacy–Utility Conflict

privacy  utility

Differential Privacy (DP)
[Dwork+, ICALP '06]

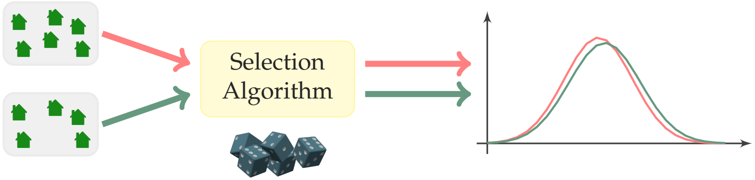
The Privacy–Utility Conflict

Differential Privacy (DP)

[Dwork+, ICALP ’06]

privacy  utility

1. Privacy definition

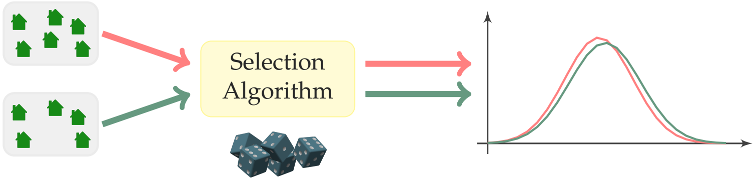


The Privacy–Utility Conflict

Differential Privacy (DP)
[Dwork+, ICALP '06]

privacy  **utility**

1. Privacy definition



2. Privacy realization
   - Basic mechanisms for numeric/discrete-valued computations.
   - Composition theorems.

Differentially Private Location Selection

[Gupta+, SODA'10]

function $\textsc{kMedian}(C, F_0)$
$i \leftarrow 0$;
while $i < T$ do
    $(x, y) \xleftarrow{\$}$ pick–swap$(F_i \times \overline{F_i})$;
    $F_{i+1} \leftarrow (F_i \setminus \{x\}) \cup \{y\}$;
    $i \leftarrow i + 1$
end;
$j \xleftarrow{\$}$ pick–solution$([1, \ldots, T], F)$;
return F_j

Verifying Differential Privacy

Dynamic verification:

- PINQ [McSherry '09]
- Airavat [Roy+ '10]

Static verification:

- *Fuzz* [Reed & Pierce '10] and *DFuzz* [Gaboardi+ '13]
- [Chaudhuri+ '11]

Limitations of theses techniques:

- Only programs that are combinations of basic mechanisms.
- Only standard differential privacy.
- Fixed set of domains and/or operations.

In this Dissertation

Our Goal

Verify differential privacy properties of probabilistic programs.

We want our technique to

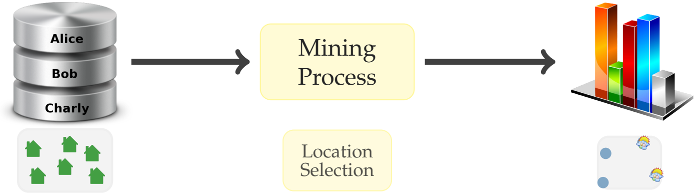
- Circumvent limitations of existing techniques.
- Provide strong evidence of correctness.
- Be extensible to reason about other quantitative properties of probabilistic programs.

Outline

Outline

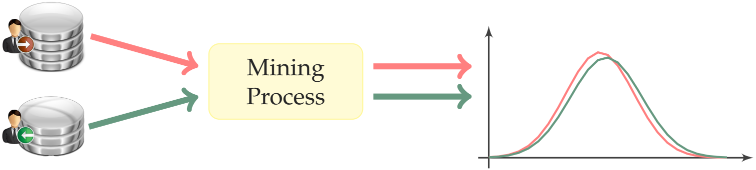
Differential Privacy – Definition

Differential Privacy – Definition

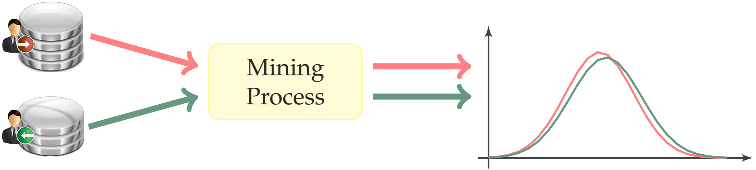


Differential Privacy – Definition

Differential Privacy – Definition



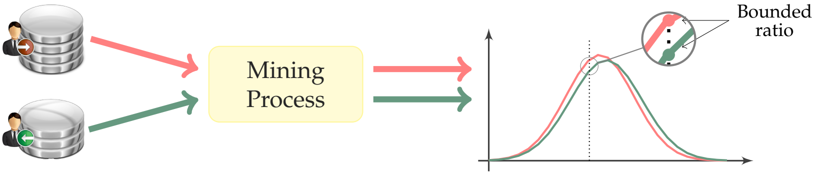
Differential Privacy – Definition



A randomized mechanism K is **ϵ-differentially private** iff for all databases d_1 and d_2, and all events A,

$$\Delta(d_1, d_2) \leq 1 \implies \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A]$$

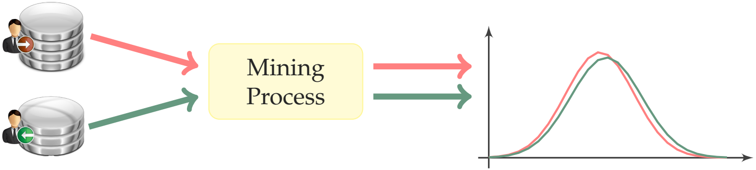
Differential Privacy – Definition



A randomized mechanism K is **ϵ-differentially private** iff for all databases d_1 and d_2, and all events A,

$$\Delta(d_1, d_2) \leq 1 \implies \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A]$$

Differential Privacy – Definition

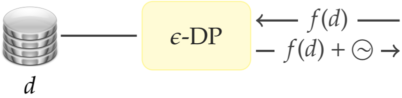


A randomized mechanism K is **(ϵ, δ)-differentially private** iff for all databases d_1 and d_2, and all events A,

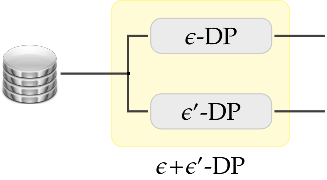
$$\Delta(d_1, d_2) \leq 1 \implies \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A] + \delta$$

Differential Privacy – Fundamentals

- Basic mechanism for numeric queries.



- Composition theorem.



Verifying Differential Privacy – Our Approach

Differential privacy is a **quantitative 2-safety property**:

$$\Delta(d_1, d_2) \leq 1 \implies \forall A \bullet \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A] + \delta$$

Verifying Differential Privacy – Our Approach

Differential privacy is a **quantitative 2-safety property**:

$$\Delta(d_1, d_2) \leq 1 \implies \forall A \bullet \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A] + \delta$$

relational
pre-condition

Verifying Differential Privacy – Our Approach

Differential privacy is a **quantitative 2-safety property**:

$$\Delta(d_1, d_2) \le 1 \implies \forall A \bullet \Pr[K(d_1) \in A] \le e^{\epsilon} \Pr[K(d_2) \in A] + \delta$$

relational pre-condition

quantitative relational post-condition

Verifying Differential Privacy – Our Approach

Differential privacy is a **quantitative 2-safety property**:

$$\Delta(d_1, d_2) \leq 1 \implies \forall A \cdot \Pr[K(d_1) \in A] \leq e^{\epsilon} \Pr[K(d_2) \in A] + \delta$$

relational pre-condition

quantitative relational post-condition

We propose a **quantitative probabilistic relational Hoare logic**

$$\{\Psi\}\, c_1 \sim_{\alpha,\delta} c_2\, \{\Phi\}$$

such that a program c is (ϵ, δ)-DP iff

$$\{\simeq\}\, c \sim_{e^{\epsilon},\delta} c\, \{\equiv\}$$

database adjacency

equality on observable output

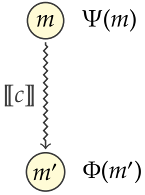
Standard Hoare Logic

$$\models \{\Psi\}\, c\, \{\Phi\}$$

$$\begin{tikzcd}
m \arrow[d, squiggly, "\llbracket c \rrbracket"'] & \Psi(m) \\
m' & \Phi(m')
\end{tikzcd}$$

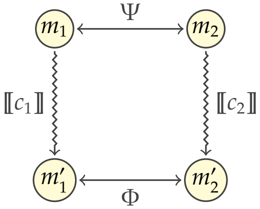
Standard Hoare Logic

$$\models \{\Psi\}\, c\, \{\Phi\}$$



Relational Hoare Logic

$$\models \{\Psi\}\, c_1 \sim c_2\, \{\Phi\}$$



Characterizing Differential Privacy

Our Goal

c is (ϵ, δ)-DP iff $\{\simeq\}\, c \sim_{e^\epsilon, \delta} c\, \{\equiv\}$

To achieve so we rely on a lifting operation and a distance measure.

Characterizing Differential Privacy

Our Goal

c is (ϵ, δ)-DP iff $\{\simeq\}\ c \sim_{e^\epsilon,\delta} c\ \{\equiv\}$

To achieve so we rely on a lifting operation and a distance measure.

$$\mathcal{L}_\alpha^\delta(\cdot) \qquad \Delta_\alpha(\cdot,\cdot) \qquad \alpha \geq 1, \delta \geq 0$$

$$\mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B) \qquad \mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0}$$

Characterizing Differential Privacy

Our Goal

c is (ϵ, δ)-DP iff $\{\simeq\}\, c \sim_{e^\epsilon,\delta} c\, \{\equiv\}$

To achieve so we rely on a lifting operation and a distance measure.

$\mathcal{L}_\alpha^\delta(\cdot)$	$\Delta_\alpha(\cdot,\cdot)$	$\alpha \geq 1, \delta \geq 0$
$\mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B)$	$\mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0}$	

1. Judgment $\{\Psi\}\, c_1 \sim_{\alpha,\delta} c_2\, \{\Phi\}$ is interpreted as

$$m_1\, \Psi\, m_2 \implies ([\![c_1]\!]\, m_1)\, \mathcal{L}_\alpha^\delta(\Phi)\, ([\![c_2]\!]\, m_2)$$

Characterizing Differential Privacy

Our Goal

c is (ϵ,δ)-DP iff $\{\simeq\}\, c \sim_{e^\epsilon,\delta} c\, \{\equiv\}$

To achieve so we rely on a lifting operation and a distance measure.

$$\mathcal{L}_\alpha^\delta(\cdot) \qquad \Delta_\alpha(\cdot,\cdot) \qquad \alpha \geq 1, \delta \geq 0$$

$$\mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B) \qquad \mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0}$$

1. Judgment $\{\Psi\}\, c_1 \sim_{\alpha,\delta} c_2\, \{\Phi\}$ is interpreted as

$$m_1\, \Psi\, m_2 \implies ([\![c_1]\!]\, m_1)\, \mathcal{L}_\alpha^\delta(\Phi)\, ([\![c_2]\!]\, m_2)$$

3. c is (ϵ,δ)-DP iff for all memories m_1 and m_2,

$$m_1 \simeq m_2 \implies \forall A \bullet\ \Pr[c(m_1) \in A] \leq e^\epsilon \Pr[c(m_2) \in A] + \delta$$

Characterizing Differential Privacy

Our Goal

c is (ϵ, δ)-DP iff $\{\simeq\}\, c \sim_{e^\epsilon,\delta} c\, \{\equiv\}$

To achieve so we rely on a lifting operation and a distance measure.

$$\mathcal{L}_\alpha^\delta(\cdot) \qquad \Delta_\alpha(\cdot,\cdot) \qquad \alpha \geq 1, \delta \geq 0$$

$$\mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B) \qquad \mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0}$$

1. Judgment $\{\Psi\}\, c_1 \sim_{\alpha,\delta} c_2\, \{\Phi\}$ is interpreted as

$$m_1\, \Psi\, m_2 \implies (\llbracket c_1 \rrbracket\, m_1)\, \mathcal{L}_\alpha^\delta(\Phi)\, (\llbracket c_2 \rrbracket\, m_2)$$

3. c is (ϵ, δ)-DP iff for all memories m_1 and m_2,

$$m_1 \simeq m_2 \implies \Delta_{e^\epsilon}(\llbracket c \rrbracket\, m_1, \llbracket c \rrbracket\, m_2) \leq \delta$$

Characterizing Differential Privacy

Our Goal

c is (ϵ,δ)-DP iff $\{\simeq\}\, c \sim_{e^\epsilon,\delta} c\, \{\equiv\}$

To achieve so we rely on a lifting operation and a distance measure.

$\mathcal{L}^{\delta}_{\alpha}(\cdot)$ $\qquad$ $\Delta_{\alpha}(\cdot,\cdot)$ $\qquad$ $\alpha \geq 1, \delta \geq 0$

$\mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B)$ $\qquad$ $\mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0}$

1. Judgment $\{\simeq\}\, c \sim_{e^\epsilon,\delta} c\, \{\equiv\}$ is interpreted as

$$m_1 \simeq m_2 \implies (\llbracket c \rrbracket\, m_1)\ \mathcal{L}^{\delta}_{e^\epsilon}(\equiv)\ (\llbracket c \rrbracket\, m_2)$$

3. c is (ϵ,δ)-DP iff for all memories m_1 and m_2,

$$m_1 \simeq m_2 \implies \Delta_{e^\epsilon}(\llbracket c \rrbracket\, m_1, \llbracket c \rrbracket\, m_2) \leq \delta$$

Characterizing Differential Privacy

Our Goal

c is (ϵ, δ)-DP iff $\{\simeq\}\, c \sim_{e^\epsilon,\delta} c\, \{\equiv\}$

To achieve so we rely on a lifting operation and a distance measure.

$\mathcal{L}_\alpha^\delta(\cdot)$ $\quad \Delta_\alpha(\cdot,\cdot)$ $\quad \alpha \geq 1, \delta \geq 0$

$\mathcal{P}(A \times B) \to \mathcal{P}(\mathcal{D}_A \times \mathcal{D}_B)$ $\quad \mathcal{D}_A \times \mathcal{D}_A \to \mathbb{R}^{\geq 0}$

1. Judgment $\{\simeq\}\, c \sim_{e^\epsilon,\delta} c\, \{\equiv\}$ is interpreted as
$$m_1 \simeq m_2 \implies (\llbracket c \rrbracket\, m_1)\, \mathcal{L}_{e^\epsilon}^\delta(\equiv)\, (\llbracket c \rrbracket\, m_2)$$
2. The lifting $\mathcal{L}_\alpha^\delta(\equiv)$ of equality is characterized as
$$\mu_1\, \mathcal{L}_\alpha^\delta(\equiv)\, \mu_2 \iff \Delta_\alpha(\mu_1, \mu_2) \leq \delta$$
3. c is (ϵ, δ)-DP iff for all memories m_1 and m_2,
$$m_1 \simeq m_2 \implies \Delta_{e^\epsilon}(\llbracket c \rrbracket\, m_1, \llbracket c \rrbracket\, m_2) \leq \delta$$

Characterizing Differential Privacy – Cont'd

- Definition of the α-distance is straightforward.

$$\Delta_\alpha(\mu_1, \mu_2) \triangleq \max_A \Pr[\mu_1 \in A] - \alpha \Pr[\mu_2 \in A]$$

- Definition of the (α, δ)-lifting is somewhat intricate (in the general case),

  ... but simpler characterization for equiv. relations.

The Programming Language

$\mathcal{C}$	::=	skip	nop
	$\mid$	$\mathcal{C};\ \mathcal{C}$	sequence
	$\mid$	$\mathcal{V} \leftarrow \mathcal{E}$	assignment
	$\mid$	$\mathcal{V} \xleftarrow{\$} \mathcal{D}$	random sampling
	$\mid$	if $\mathcal{E}$ then $\mathcal{C}$ else $\mathcal{C}$	conditional
	$\mid$	while $\mathcal{E}$ do $\mathcal{C}$	while loop
	$\mid$	$\mathcal{V} \leftarrow \mathcal{P}(\mathcal{E}, \ldots, \mathcal{E})$	procedure call

Weakening

$$\frac{\models \{\Psi'\}\, c_1 \sim_{\alpha',\delta'} c_2\, \{\Phi'\} \qquad \Psi \Rightarrow \Psi' \quad \Phi' \Rightarrow \Phi \quad \alpha' \leq \alpha \quad \delta' \leq \delta}{\models \{\Psi\}\, c_1 \sim_{\alpha,\delta} c_2\, \{\Phi\}}$$

Weakening

$$\frac{\models \{\Psi'\}\, c_1 \sim_{\alpha',\delta'} c_2 \,\{\Phi'\} \qquad \Psi \Rightarrow \Psi' \quad \Phi' \Rightarrow \Phi \quad \alpha' \leq \alpha \quad \delta' \leq \delta}{\models \{\Psi\}\, c_1 \sim_{\alpha,\delta} c_2 \,\{\Phi\}}$$

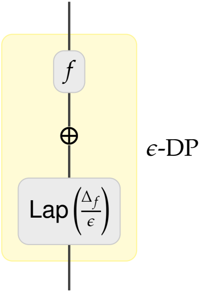
Sequential composition

$$\frac{\models \{\Psi\}\, c_1 \sim_{\alpha_1,\delta_1} c_2 \,\{\Phi'\} \qquad \models \{\Phi'\}\, c_1' \sim_{\alpha_2,\delta_2} c_2' \,\{\Phi\}}{\models \{\Psi\}\, c_1; c_1' \sim_{\alpha_1\alpha_2,\delta_1+\delta_2} c_2; c_2' \,\{\Phi\}}$$

The Proof System – Cont'd

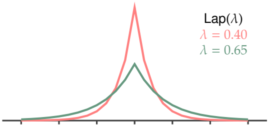
Laplacian mechanism

Output perturbation makes numerical queries ϵ-DP



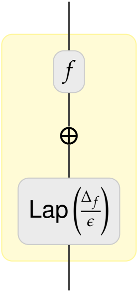
The **sensitivity** of a numerical query $f : \mathcal{D} \to \mathbb{R}$ is defined as:

$$\Delta_f \triangleq \max_{\substack{d_1, d_2 \\ d_1 \simeq d_2}} |f(d_1) - f(d_2)|$$



Laplacian mechanism

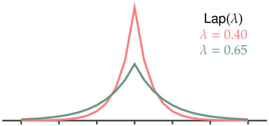
Output perturbation makes numerical queries ϵ-DP



ϵ-DP

The **sensitivity** of a numerical query $f : \mathcal{D} \to \mathbb{R}$ is defined as:

$$\Delta_f \triangleq \max_{\substack{d_1, d_2 \\ d_1 \simeq d_2}} |f(d_1) - f(d_2)|$$



$$\frac{m_1 \, \Psi \, m_2 \implies |[\![r]\!] \, m_1 - [\![r]\!] \, m_2| \leq k}{\models \{\Psi\} \, x \xleftarrow{\$} \mathcal{L}(r, {}^{k}/_{\epsilon}) \sim_{e^{\epsilon}, 0} x \xleftarrow{\$} \mathcal{L}(r, {}^{k}/_{\epsilon}) \, \{x\langle 1 \rangle = x\langle 2 \rangle\}}$$

Machine-Checked Proofs of Differential Privacy

CertiPriv: framework proving interactive support for the logic built on top of the Coq proof assistant.

- Delivers machine-checked proofs of differential privacy.
- Built as an extension of CertiCrypt.
  - α-distance + (α, δ)-lifting + logic soundness (+6.500 lines of Coq proof-script)
- Several case studies:
  - Laplacian, Exponential and Gaussian basic mechanisms.
  - k-Median, Minimum Vertex Cover, streaming algorithm.

Case Study: k-Median Problem

```
function kMedian(C, F_0)
  i ← 0;
  while i < T do
    (x, y) ←$ pick−swap(F_i × F̄_i);
    F_{i+1} ← (F_i \ {x}) ∪ {y};
    i ← i + 1
  end;
  j ←$ pick−solution([1, ..., T], F);
  return F_j
```

Differential privacy is captured by judgment

$$\{\Psi\}\ \textsc{kMedian} \sim_{\alpha,0} \textsc{kMedian}\ \{\Phi\}$$

Ψ: $C\langle 1\rangle \simeq C\langle 2\rangle \wedge F_0\langle 1\rangle = F_0\langle 2\rangle$

α: $e^{2\epsilon\Delta(T+1)}$

Φ: $F_j\langle 1\rangle = F_j\langle 2\rangle$

Judgment derivation + Verification of side conditions $\approx$ 450 lines proof-script

Verifying Differential Privacy – Summary

- Program logic for reasoning about DP.
- Framework for building machined-checked proofs of DP

With G. Barthe, B. Köpf and S. Zanella Béguelin
[POPL ’12] [TOPLAS ’13]

Our Goal

Verify differential privacy properties of probabilistic programs.

We want our technique to

- Provides strong evidence of correctness. ✓
- Circumvent limitations of existing techniques. ✓
- Be extensible to reason about other quantitative properties of probabilistic programs.

Outline

Scope of our Approach

Differential privacy is a **quantitative relational property** of probabilistic programs:

$$m_1 \Psi m_2 \implies \Delta\left(\llbracket c_1 \rrbracket\, m_1, \llbracket c_2 \rrbracket\, m_2\right) \leq \delta$$

But it is not the only one!

- Indifferentiability
- Zero Knowledge
- Pseudo-randomness
- ...

Can we use our logic as it is to reason about these properties as well?

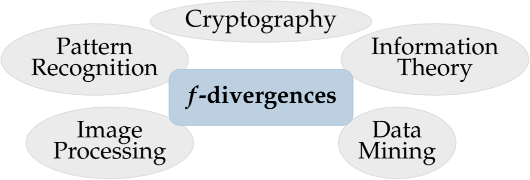
NO. They use distance measures different from the α-distance.

Extending our Logic

Our logic is extensible to the class of f-divergences.

The class of f-divergences comprises **well-know examples** of distance measures and finds **applications in multiple areas**:

- Statistical distance
- Hellinger distance
- Relative entropy
- α-distance
- χ^2-distance



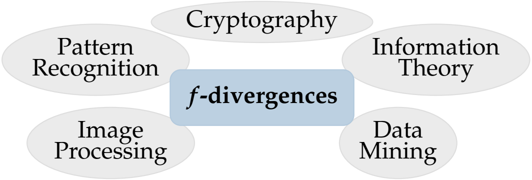
Extending our Logic

Our logic is extensible to the class of f-divergences.

With G. Barthe [ICALP '13]

The class of f-divergences comprises **well-know examples** of distance measures and finds **applications in multiple areas**:

- Statistical distance
- Hellinger distance
- Relative entropy
- α-distance
- χ^2-distance



Our Goal

Verify differential privacy properties of probabilistic programs.

We want our technique to

- Provides strong evidence of correctness. ✓
- Circumvent limitations of existing technique. ✓
- Be extensible to reason about other quantitative properties of probabilistic programs. ✓

What else is in the dissertation?

Crypto Case Study: Secure Hash Functions into Elliptic Curves [Brier+ ’10]

Security is captured by formula

$$\forall \mathcal{D} \cdot \Delta_{\mathsf{SD}}\left(\mathcal{D}^{H,h}, \mathcal{D}^{\mathcal{RO},S}\right) \leq \epsilon$$

Our machine-checked proof

- Approximate observational equivalence (specialization of our Hoare logic) + adversary rule.
- Requires heavy algebraic reasoning (elliptic curves and group theory).
- 10.000+ lines of Coq proof-script.

With G. Barthe, B. Grégoire, S. Heraud, S. Zanella [POST ’12, JCS ’14]

Conclusions

Summary of contributions

- Quantitative relational Hoare logic for approximate reasoning about probabilistic programs.
- Framework for building machined-checked proofs of differential privacy (and other quantitative properties).
- Verification of several constructions from the recent literature.

Future work

- Improve automation (e.g. inference of loop invariants).
- Lipschitz continuity of probabilistic programs.
- Combination of different techniques.

The (α, δ)-lifting

Witness distributions in $\mathcal{D}_{A\times B}$

$$\mu_1\ \mathcal{L}^{\delta}_{\alpha}(R)\ \mu_2 \triangleq \exists\, \mu_L, \mu_R \bullet \begin{cases} \Delta_\alpha(\mu_L, \mu_R) \leq \delta \\ \pi_1(\mu_L) = \mu_1 \ \wedge\ \pi_2(\mu_R) = \mu_2 \\ supp(\mu_L) \subseteq R \ \wedge\ supp(\mu_R) \subseteq R \end{cases}$$

$\subseteq (A \times B)$

- Admits an inductive characterization.
- For equivalence relations, it can be characterized as a closeness condition.

$$\mu_1\ \mathcal{L}^{\delta}_{\alpha}(R)\ \mu_2 \iff \Delta_\alpha(\mu_1/R, \mu_2/R) \leq \delta$$

- For finite relations, it can be modeled as network-flow problem.

Generalized Data Processing Theorem

For any distribution transformer $h : \mathcal{D}_A \rightarrow \mathcal{D}_B$

$$\Delta_f\left(h(\mu_1), h(\mu_2)\right) \leq \Delta_f\left(\mu_1, \mu_2\right)$$

$$\frac{\forall m_1, m_2 \bullet\ m_1\ \Psi\ m_2 \implies (m_1\{[\![e_1]\!]\,m_1/x_1\})\ \Phi\ (m_2\{[\![e_2]\!]\,m_2/x_2\})}{\vdash \{\Psi\}\, x_1 \leftarrow e_1 \sim_{f,0} x_2 \leftarrow e_2\, \{\Phi\}}\,[\text{assn}]$$

$$\frac{\forall m_1, m_2 \bullet\ m_1\ \Psi\ m_2 \implies \Delta_f\,([\![\mu_1]\!]\,m_1, [\![\mu_2]\!]\,m_2) \leq \delta}{\vdash \{\Psi\}\, x_1 \stackrel{\$}{\leftarrow} \mu_1 \sim_{f,\delta} x_2 \stackrel{\$}{\leftarrow} \mu_2\, \{x_1\langle 1\rangle = x_2\langle 2\rangle\}}\,[\text{rand}]$$

$$\frac{\begin{array}{c}\Psi \implies b\langle 1\rangle \equiv b'\langle 2\rangle \\ \vdash \{\Psi \wedge b\langle 1\rangle\}\, c_1 \sim_{f,\delta} c'_1\, \{\Phi\} \qquad \vdash \{\Psi \wedge \neg b\langle 1\rangle\}\, c_2 \sim_{f,\delta} c'_2\, \{\Phi\}\end{array}}{\vdash \{\Psi\}\, \text{if } b \text{ then } c_1 \text{ else } c_2 \sim_{f,\delta} \text{if } b' \text{ then } c'_1 \text{ else } c'_2\, \{\Phi\}}\,[\text{cond}]$$

$$\frac{\begin{array}{c}(f_1, \ldots, f_n) \text{ composable and monotonic} \\ \Theta \triangleq b\langle 1\rangle \equiv b'\langle 2\rangle \qquad \Psi \wedge e\langle 1\rangle \leq 0 \Longrightarrow \neg b\langle 1\rangle \\ \vdash \{\Psi \wedge b\langle 1\rangle \wedge b'\langle 2\rangle \wedge e\langle 1\rangle = k\}\, c \sim_{f_1,\delta} c'\, \{\Psi \wedge \Theta \wedge e\langle 1\rangle < k\}\end{array}}{\vdash \{\Psi \wedge \Theta \wedge e\langle 1\rangle \leq n\}\, \text{while } b \text{ do } c \sim_{f_n, n\delta} \text{while } b' \text{ do } c'\, \{\Psi \wedge \neg b\langle 1\rangle \wedge \neg b'\langle 2\rangle\}}\,[\text{while}]$$

$$\frac{}{\vdash \{\Psi\}\, \text{skip} \sim_{f,0} \text{skip}\, \{\Psi\}}\,[\text{skip}]$$

$$\frac{\begin{array}{c}(f_1, f_2) \text{ is } f_3\text{-composable} \\ \vdash \{\Psi\}\, c_1 \sim_{f_1,\delta_1} c_2\, \{\Phi'\} \quad \vdash \{\Phi'\}\, c'_1 \sim_{f_2,\delta_2} c'_2\, \{\Phi\}\end{array}}{\vdash \{\Psi\}\, c_1; c'_1 \sim_{f_3,\delta_1+\delta_2} c_2; c'_2\, \{\Phi\}}\,[\text{seq}]$$

$$\frac{\begin{array}{c}\vdash \{\Psi \wedge \Theta\}\, c_1 \sim_{f,\delta} c_2\, \{\Phi\} \\ \vdash \{\Psi \wedge \neg\Theta\}\, c_1 \sim_{f,\delta} c_2\, \{\Phi\}\end{array}}{\vdash \{\Psi\}\, c_1 \sim_{f,\delta} c_2\, \{\Phi\}}\,[\text{case}]$$

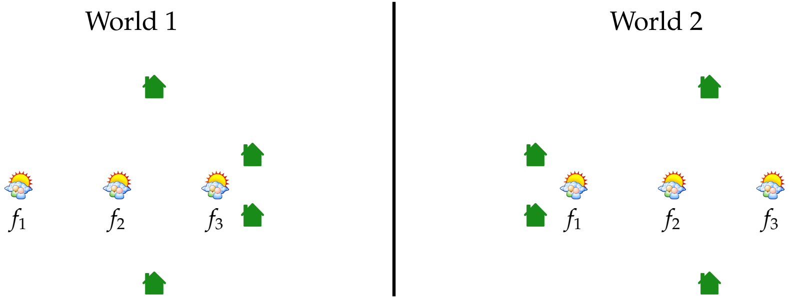
$$\frac{\begin{array}{c}\vdash \{\Psi'\}\, c_1 \sim_{f',\delta'} c_2\, \{\Phi'\} \\ \Psi \Rightarrow \Psi' \quad \Phi' \Rightarrow \Phi \quad f \leq f' \quad \delta' \leq \delta\end{array}}{\vdash \{\Psi\}\, c_1 \sim_{f,\delta} c_2\, \{\Phi\}}\,[\text{weak}]$$

Trusted Code Base

- You need to:
  - trust the type checker of Coq;
  - trust the language semantics;
  - make sure the security statement (a few lines in Coq) is as expected.
- You don't need to
  - understand or even read the proof;
  - trust program logics,

Case Study: k-Median Problem

Problem's solution may leak the presence/absence of clients



Assume $k = 2$

$$\text{Solution} = \{f_2, f_3\} \implies \text{World 1}$$
$$\text{Solution} = \{f_1, f_2\} \implies \text{World 2}$$

Case Study: k-Median Problem

```
function kMedian(C, F_0)
1  i ← 0;
2  while i < T do
3      (x, y) ←$ pick−swap(F_i × F̄_i);
4      F_{i+1} ← (F_i \ {x}) ∪ {y};
5      i ← i + 1
6  end;
7  j ←$ pick−solution([1, ..., T], F);
8  return F_j
```

$\Pr(x, y) \propto e^{-\epsilon\, c(F_i - x + y)}$

$\Pr(j) \propto e^{-\epsilon\, c(F_j)}$

Each iteration of the loop (3-5) $\rightsquigarrow$ $2\epsilon\Delta$-DP

Selection of the solution (7) $\rightsquigarrow$ $2\epsilon\Delta$-DP

$2\epsilon\Delta(T+1)$-DP

In our formalism,

$$\{\Psi\}\ \text{kMedian} \sim_{\alpha,0} \text{kMedian}\ \{\Phi\}$$

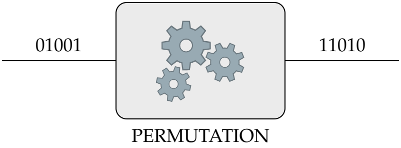
$C\langle 1\rangle \simeq C\langle 2\rangle \wedge F_0\langle 1\rangle = F_0\langle 2\rangle$

$e^{2\epsilon\Delta(T+1)}$

$F_j\langle 1\rangle = F_j\langle 2\rangle$

Improving security bounds for ***Key-Alternating Cipher*** *via Hellinger Distance.*

$$E_P(k, \cdot) : \{0,1\}^n \to \{0,1\}^n$$



f-divergences

The f*-divergence* between two distributions μ_1 and μ_2 over a set A is defined as

$$\Delta_f(\mu_1, \mu_2) \triangleq \sum_{a \in A} \mu_2(a)\, f\left(\frac{\mu_1(a)}{\mu_2(a)}\right)$$

where $f : \mathbb{R}^{\geq 0} \to \mathbb{R}$ is a continuous convex function s.t. $f(1) = 0$.

Some examples

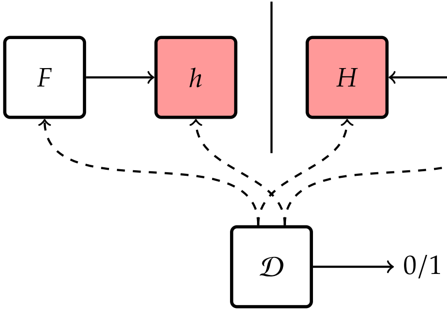
- Statistical distance (Δ_{SD}) $\quad f(t) = \frac{1}{2}|t-1|$
- Kullback-Leibler (Δ_{KL}) $\quad f(t) = t\ln(t)$
- Hellinger distance (Δ_{HD}) $\quad f(t) = \frac{1}{2}(\sqrt{t}-1)^2$
- α-distance (Δ_α) $\quad f(t) = \max\{t-\alpha, 0\}$

Indifferentiability

F with access to a RO h is $(t_{\mathcal{S}}, q, \epsilon)$-indifferentiable from a RO H if

$\exists \mathcal{S}$ that runs in time $t_{\mathcal{S}}$, $\forall \mathcal{D}$ that makes at most q queries,

$$\left| \Pr\left[b \leftarrow \mathcal{D}^{F,h} : b = 1 \right] - \Pr\left[b \leftarrow \mathcal{D}^{H,\mathcal{S}} : b = 1 \right] \right| \leq \epsilon$$

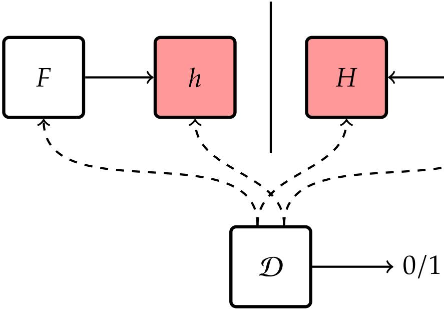


Indifferentiability

F with access to a RO h is $(t_{\mathcal{S}}, q, \epsilon)$-indifferentiable from a RO H if

$\exists\mathcal{S}$ that runs in time $t_{\mathcal{S}}$, $\forall\mathcal{D}$ that makes at most q queries,

$$\left|\Pr\left[b \leftarrow \mathcal{D}^{F,h} : b = 1\right] - \Pr\left[b \leftarrow \mathcal{D}^{H,\mathcal{S}} : b = 1\right]\right| \leq \epsilon$$

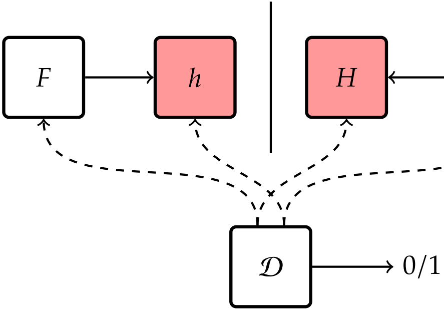


In **any** secure cryptosystem, a random oracle H
can be replaced with the construction F, which uses a random oracle h

Indifferentiability

F with access to a RO h is $(t_{\mathcal{S}}, q, \epsilon)$-indifferentiable from a RO H if

$$\exists \mathcal{S} \text{ that runs in time } t_{\mathcal{S}},\ \forall \mathcal{D} \text{ that makes at most } q \text{ queries,}$$
$$\left| \Pr\left[b \leftarrow \mathcal{D}^{F,h} : b = 1 \right] - \Pr\left[b \leftarrow \mathcal{D}^{H,\mathcal{S}} : b = 1 \right] \right| \leq \epsilon$$

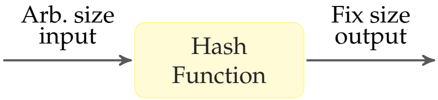


In **any** secure cryptosystem, a random oracle H into $EC(\mathbb{F}_p)$ can be replaced with the construction F, which uses a random oracle h into $\mathbb{F}_p \times \mathbb{Z}_N$

Constructing Secure Hash Functions into Elliptic Curves (EC)

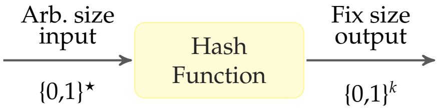
A Crypto Case Study

Constructing Secure Hash Functions into Elliptic Curves (EC)



- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.

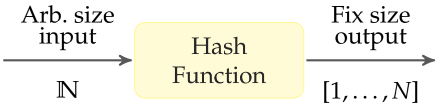
Constructing Secure Hash Functions into Elliptic Curves (EC)



- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.

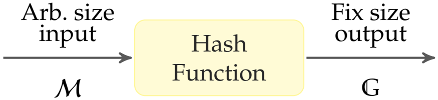
A Crypto Case Study

Constructing Secure Hash Functions into Elliptic Curves (EC)



- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.

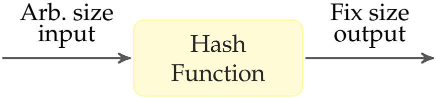
Constructing Secure Hash Functions into Elliptic Curves (EC)



- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.

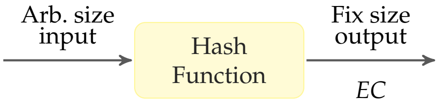
A Crypto Case Study

Constructing Secure Hash Functions into Elliptic Curves (EC)



- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.
- Their output should "look like" uniformly distributed.

Constructing Secure Hash Functions into Elliptic Curves (EC)



- Building blocks of numerous cryptosystems: encryption schemes, signature schemes, etc.
- Their output should "look like" uniformly distributed.
- Hash functions into elliptic curve allow an efficient implementation of some functionalities.

What is an elliptic curve?

Given a finite field $\mathbb{F}$ and two scalars $a, b \in \mathbb{F}$,

$$EC(\mathbb{F}) \triangleq \{(X, Y) \in \mathbb{F} \times \mathbb{F} \mid Y^2 = X^3 + aX + b\}$$

What is an elliptic curve?

Given a finite field $\mathbb{F}$ and two scalars $a, b \in \mathbb{F}$,

$$EC(\mathbb{F}) \triangleq \{(X, Y) \in \mathbb{F} \times \mathbb{F} \mid Y^2 = X^3 + aX + b\} \cup \{\mathcal{O}\}$$

Theorem: the points in $EC(\mathbb{F})$ have a group structure.

What is an elliptic curve?

Given a finite field $\mathbb{F}$ and two scalars $a, b \in \mathbb{F}$,

$$EC(\mathbb{F}) \triangleq \{(X, Y) \in \mathbb{F} \times \mathbb{F} \mid Y^2 = X^3 + aX + b\} \cup \{\mathcal{O}\}$$

Theorem: the points in $EC(\mathbb{F})$ have a group structure.

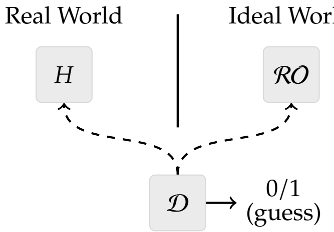
How to securely hash into an elliptic curve $EC(\mathbb{F})$?

[Brier+ '10]

$$H(m) = \underbrace{f}_{:\mathbb{F}\to EC(\mathbb{F})}(\underbrace{h_1}_{:\mathcal{M}\to\mathbb{F}}(m)) \otimes g^{\overbrace{h_2}^{:\mathcal{M}\to[1,\ldots,N]}(m)}$$

A Crypto Case Study – Cont'd II

Indifferentiability from a Random Oracle



H is called **ϵ-indifferentiable** from a random oracle iff

$$\forall \mathcal{D} \cdot \; \Delta_{\mathsf{SD}}\left(\mathcal{D}^{H}, \mathcal{D}^{\mathcal{RO}}\right) \leq \epsilon$$

Machine-checked version of Brier et al's proof

- Equational theory for approximate observational equivalence (specialization of our Hoare logic) + adversary rule.
- Requires heavy algebraic reasoning (elliptic curves and group theory).
- 10.000+ lines of Coq proof-script.

With G. Barthe, B. Grégoire, S. Heraud, S. Zanella [POST '12, JCS '14]