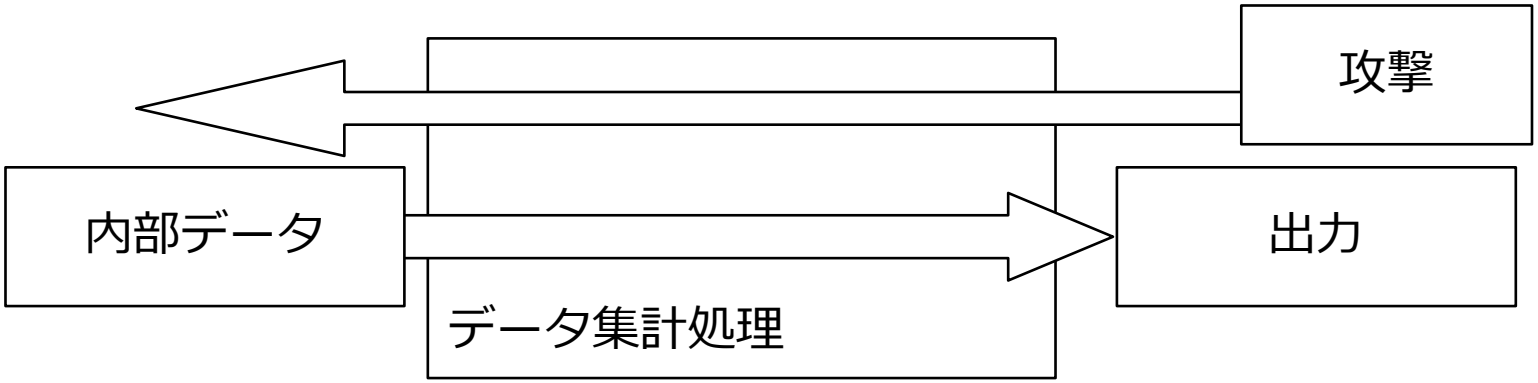
Isabelle/HOL を用いた
差分プライバシーの形式的検証

日本応用数理学会2024年度年会
2024年9月15日
佐藤哲也 (東京工業大学)

差分プライバシー

差分プライバシーの背景

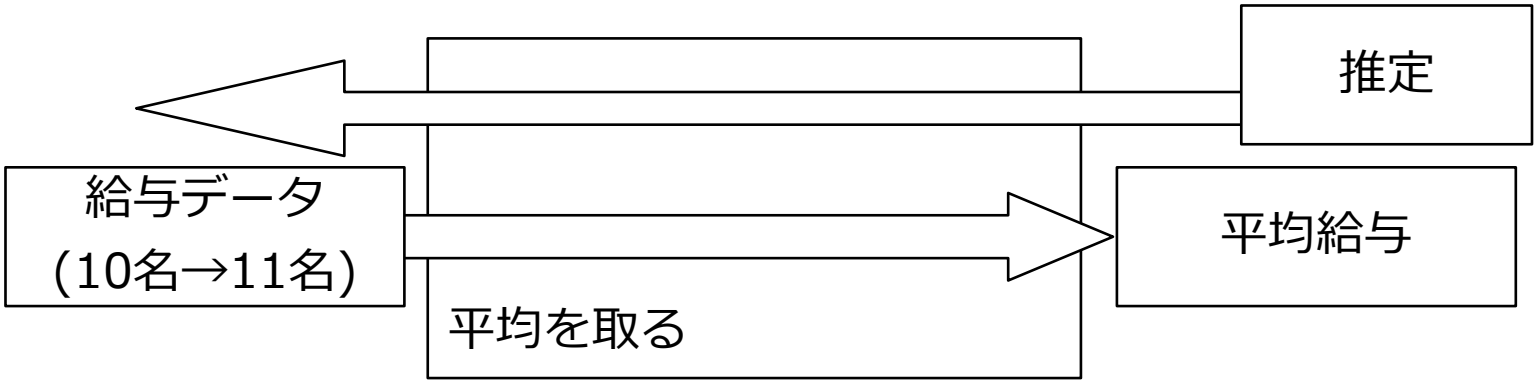
- 背景知識攻撃



  - 内部データ自体は保護されていたとしても、
  - 十分な背景知識があると、データベースの出力から内部データを(統計的に)逆算することが可能となる。

背景知識攻撃　極端な例

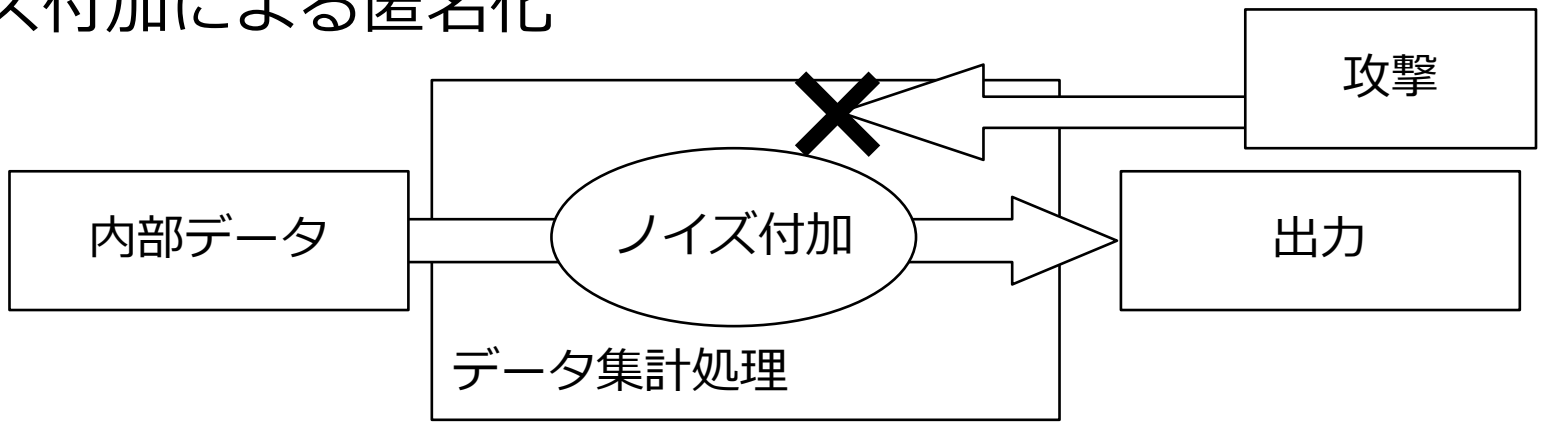
- 平均給与データベースへの攻撃



  - 10 名のとき、平均給与は 500万円だった
  - 一人加わり11名になると、平均給与は 600万円になった。
    - 11人目の人が「太郎さん」であることを知っているとき
    - 「太郎さん」の給与は1600万円であることがわかる。

ノイズ付加による匿名化

- ノイズ付加による匿名化

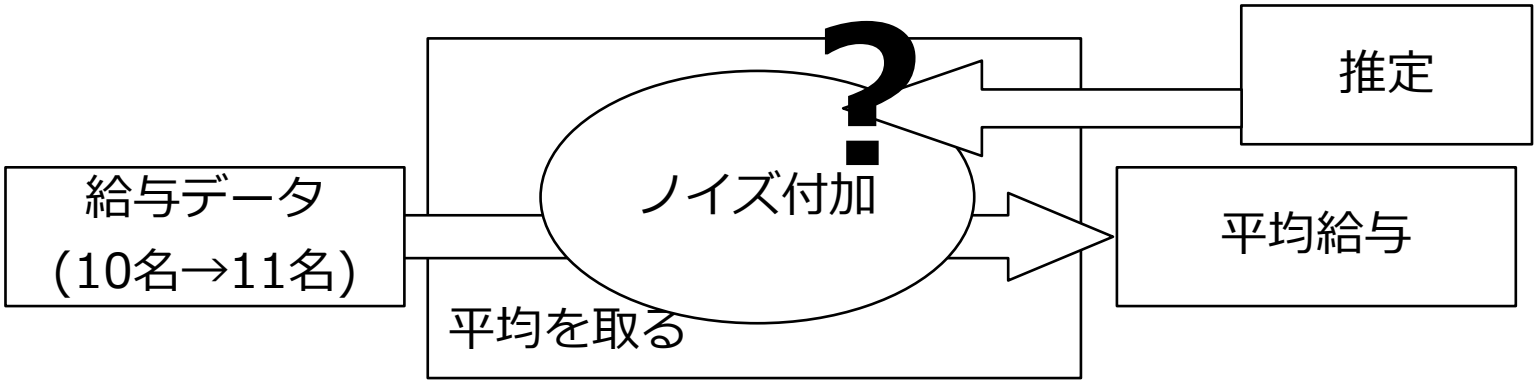


  - ノイズを付加し、内部データの推測を困難にする
  - 背景知識攻撃(任意の統計的攻撃)に対して頑健。

- 差分プライバシー(Differential Privacy, DP)は、
  このような匿名化における、プライバシーの基準である

背景知識攻撃　極端な例　の匿名化

- 平均給与データベースへの攻撃



  - 出力に(平均0の)ノイズが加算された状態で、
    10 名→平均500万円、11名→平均600万円、と出力
    - 攻撃者は11人目の人が「太郎さん」であることを知っている
    - 平均値については(ノイズによるが)おおむね正しく算出される。
    - が「太郎さん」個人の給与は1600万円かは非常に怪しい。

差分プライバシー

(Differential Privacy)

- 定義[Dwork+, TCC 2006]：
  - ランダム化されたメカニズム M: Datasets → Prob(Y) が(ε,δ)-差分プライバシー(DP)を満たすとは、
    - “隣接する”データセット D1～D2 について以下が成立：

$$\forall S \subseteq Y.$$
$$\Pr[M(D_1) \in S] \leq \exp(\varepsilon) \Pr[M(D_2) \in S] + \delta$$

    - 直観：確率δの場合を除き、確率比がεで押さえられる((ε,δ)=(0,0)のとき、確率分布は一致する)

“隣接する” データセット

- 差分プライバシーは、内部データが更新されたことを秘匿する。本来、内部データの構造に依存した概念である。

- 本講演では、以下の定式化を採用する。
  - データセットは自然数の配列とする $D \in \mathbb{N}^n$
    - 各成分 D[i] は i 番目の項目に対応するデータの個数。
  - データセットの”隣接関係”
    - １項目しか違いがないということの数学的表現。

$$D \sim D' \iff \|D - D'\|_1 \leq 1$$

(補)仮説検定的特徴づけ

[Kariouz+, ICML 2015]

- ランダム化されたメカニズム M: Datasets → Prob(Y) が(ε,δ)-差分プライバシー(DP)を満たすことは以下と同値：
  - 隣接する内部データ D1 ～ D2 について

$$\forall S \subseteq Y.\ (\Pr[M(D_1) \in S], \Pr[M(D_2) \notin S]) \in R(\varepsilon, \delta)$$

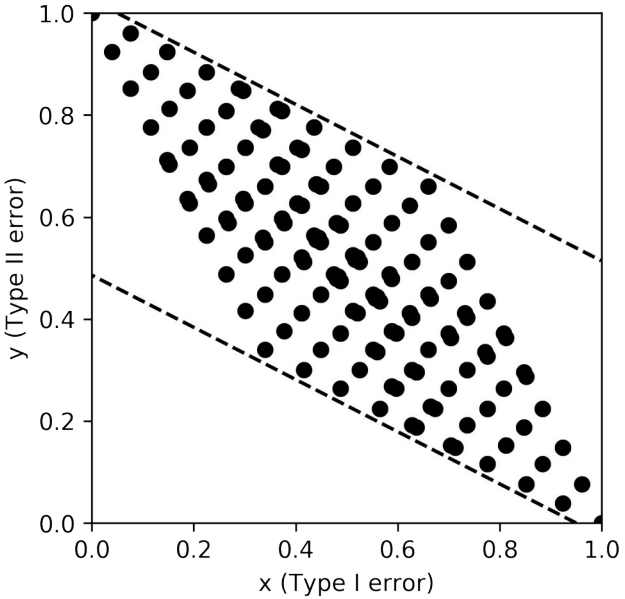
Rejection　Type I error　Type II error　privacy region

$$R(\varepsilon, \delta) = \{\ (s,t) \mid s + e^{\varepsilon} \cdot t \geq 1 - \delta, \quad t + e^{\varepsilon} \cdot s \geq 1 - \delta\ \}$$

Sは、内部データがD1かD2かのどちらかを決定する手法と等価。

仮説検定の棄却域を検定統計量で引き戻した逆像にあたる。

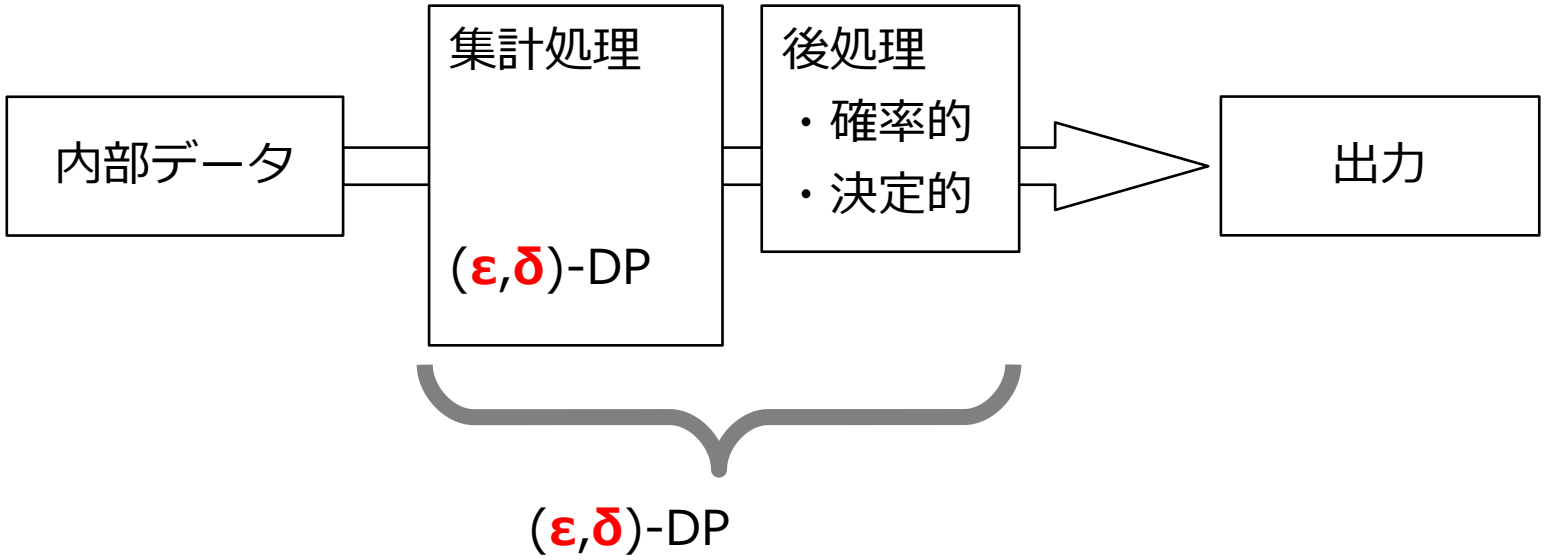
解釈：どんな検定手法で内部データを検定しても、誤りを一定以下にできない。



後処理に対する安定性

(Postprocessing)

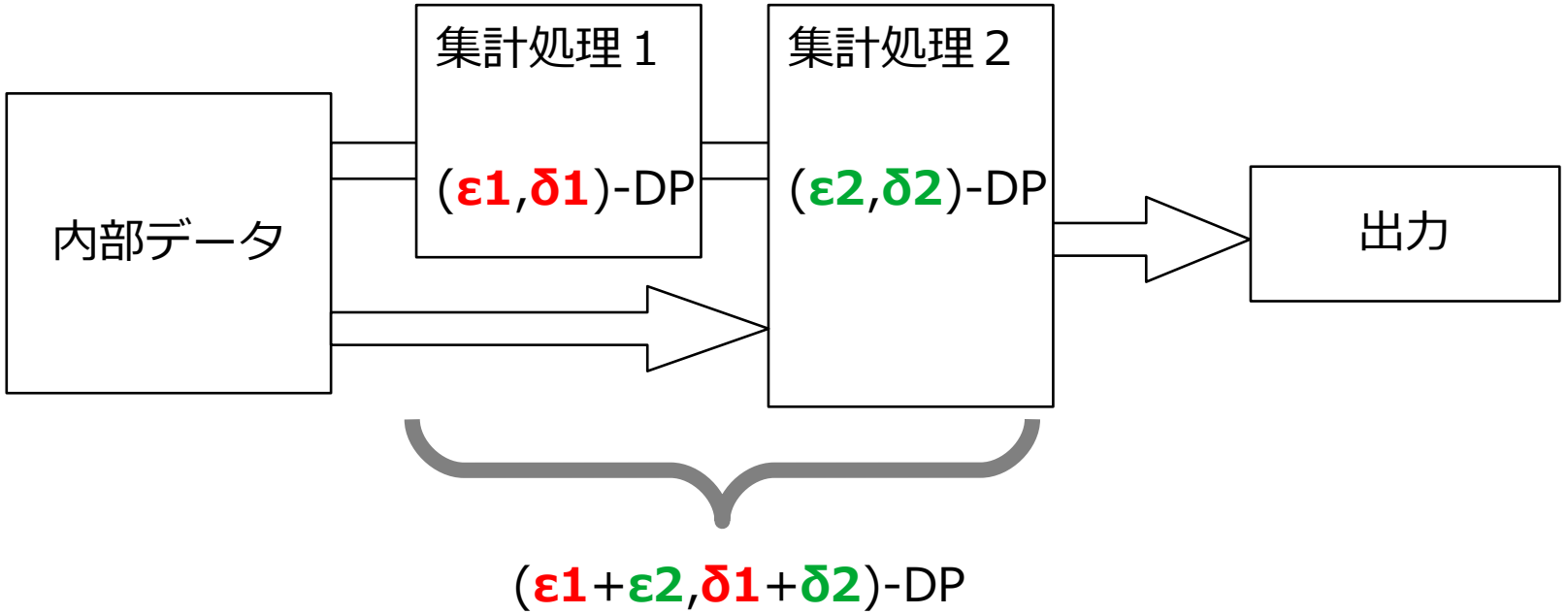
- 差分プライバシーの基本的性質に、
  後処理に対して安定的　というものがある。



差分プライバシーの合成性

(Composition theorem)

- 組み合わせたデータベースの差分プライバシーは、
  各ブロックごとに分割して評価できる。



※単純な和よりも厳密に評価する
Advanced Composition
という技法があるが割愛

ノイズ付加：Randomized Response

- ビットを一定確率(1/(exp(ε)+1))で反転。

$$\mathtt{RR}_\varepsilon : \{\top, \bot\} \to \mathrm{Prob}\{\top, \bot\}$$

$$\mathtt{RR}_\varepsilon(b) = \begin{cases} b & \text{with probability } \frac{e^\varepsilon}{e^\varepsilon+1} \\ \neg b & \text{with probability } \frac{1}{e^\varepsilon+1} \end{cases}$$

あきらかに(ε,0)-DPを満たす。

- 応用：QuickTypeの候補出力
  https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf
  → 収集したデータの各ビットにrandomized responseを適用。

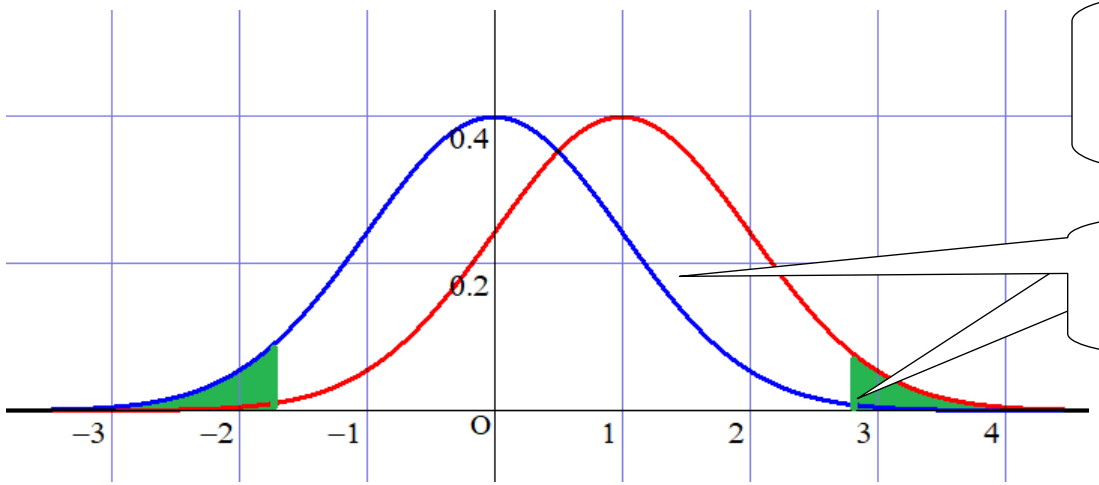
ノイズ付加：正規分布によるノイズ

- 平均0、分散 σ^2 の正規分布からなるノイズを加算

$\mathtt{Gauss}_\sigma : \mathbb{R} \to \mathrm{Prob}(\mathbb{R})$

$\mathtt{Gauss}_\sigma(x)$ は平均 x 分散 σ^2 の正規分布となる

  - 隣接関係 |x-y|≦1 に対して、(ε,δ)-DPが言える。
    (0<εに対して適当なδが決まる)



確率比は遠方で∞に発散、
誤差 0<δ だけ端をカット

残った中心部分の確率比e^ε

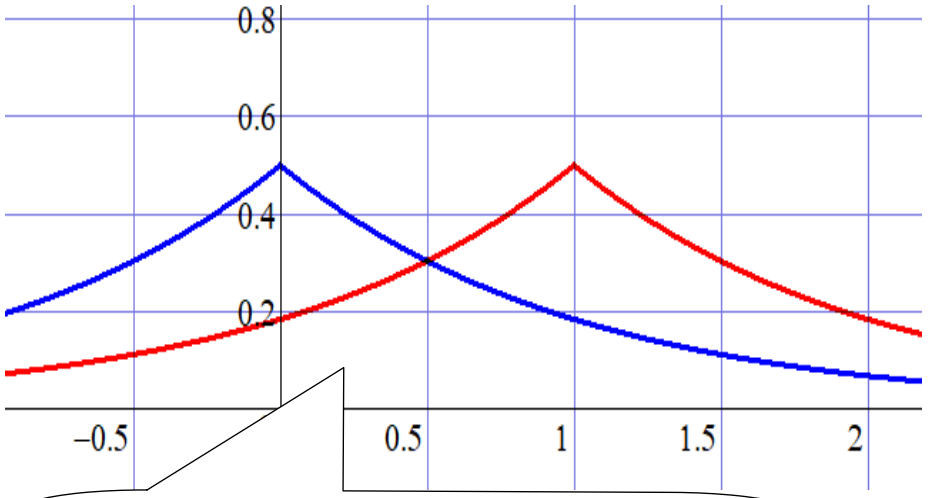
ノイズ付加：ラプラス分布によるノイズ

- 平均0、尺度 b のラプラス分布からなるノイズを加算

$\mathrm{Lap}_b : \mathbb{R} \to \mathrm{Prob}(\mathbb{R})$

$\mathrm{Lap}_b(x)$ は平均 x 尺度 b のラプラス分布となる

  - 隣接関係 |x-y|≦1 に対して、(1/b,0)-DPが得られる



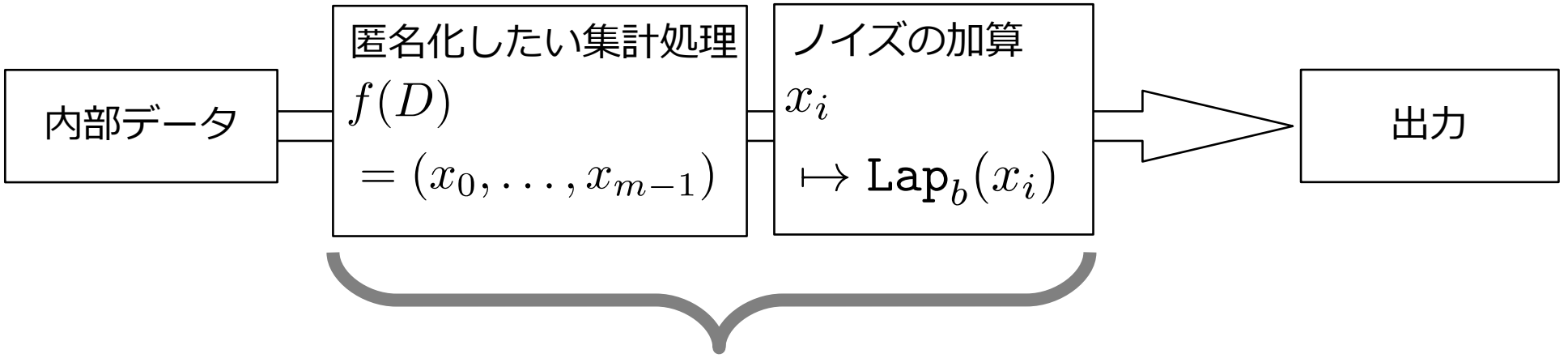
x=0, y=1　という２つの数値に
ラプラス分布によるノイズを加算した結果

$$\begin{aligned}\Pr[\mathrm{Lap}_b(x) = z] &= \frac{1}{2b}\exp(-\frac{|x-z|}{b}) \\ &\leq \frac{1}{2b}\exp(-\frac{|y-z|-|x-y|}{b}) \\ &\leq \exp(1/b)\frac{1}{2b}\exp(-\frac{|y-z|}{b})] \\ &= \exp(1/b)\Pr[\mathrm{Lap}_b(y) = z]\end{aligned}$$

確率比が一定で押さえられる

Laplace Mechanism

- 関数m次元の出力の各成分に、
  ラプラス分布によるノイズを独立に加算する。



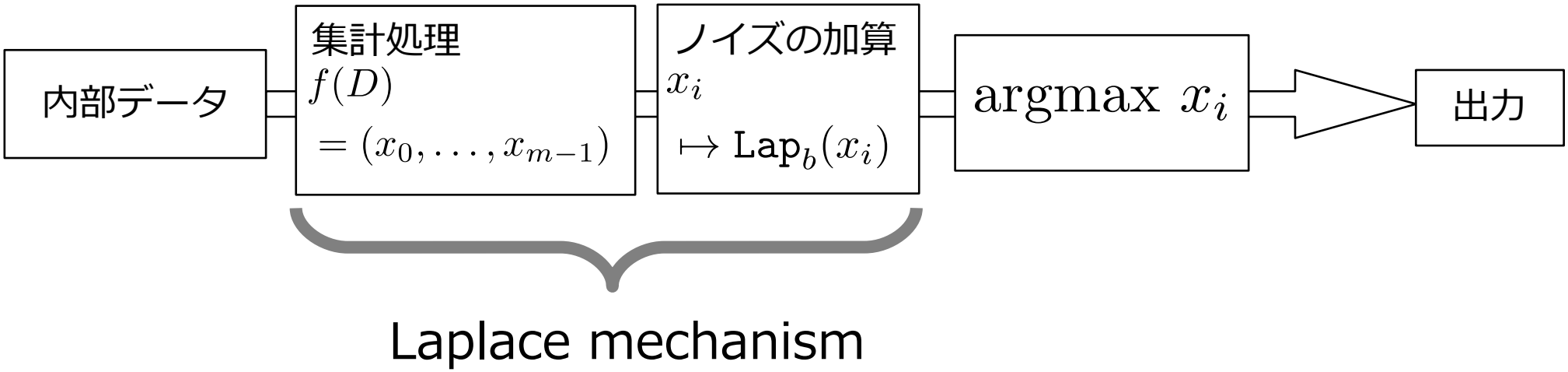
f の sensitivity

$$\Delta f = \sup\{\|f(D_1) - f(D_2)\|_1 \mid D_1 \sim D_2\}$$

に対して、(**Δf/b**,0)-DP

Report Noisy Max Mechanism

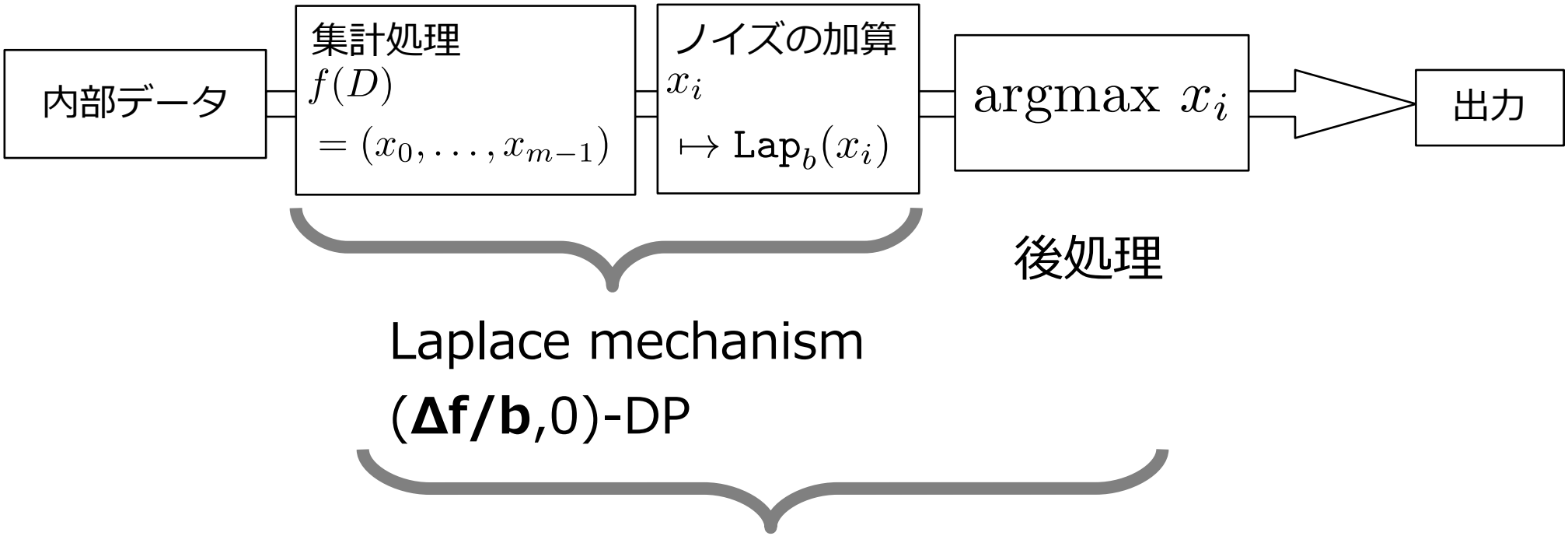
- Laplace Mechanism の結果の最大値の番号を返す。



  - 集計結果の最大値の番号を返す処理は例えば
    - 「来客集計において、最も混んでる時間帯」や
    - 「販売集計で、最も売れたメニュー」などを想定している。

Report Noisy Max Mechanism

- 差分プライバシーはいくつになるか

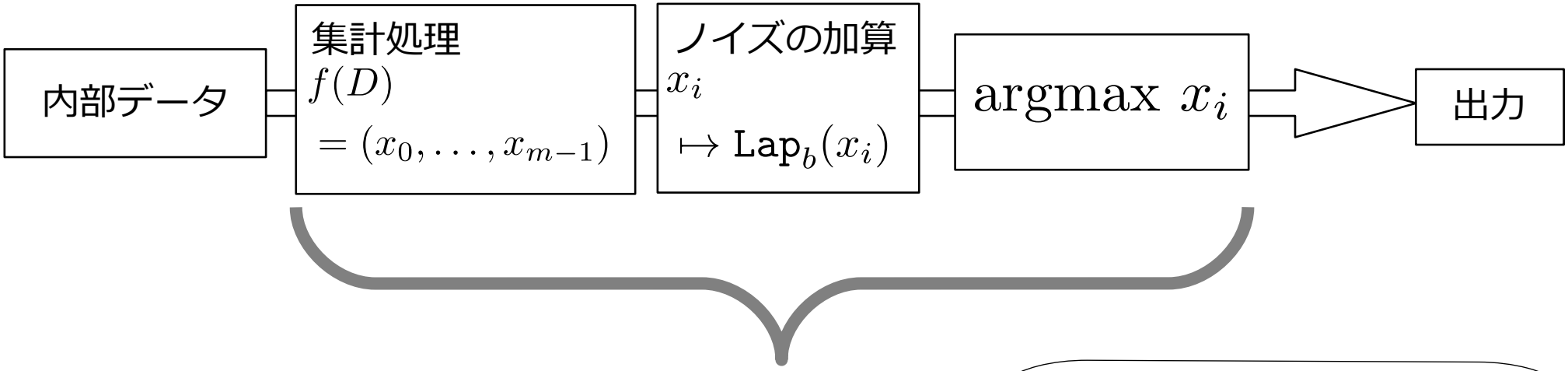


素朴に考えると、(**Δf/b**,0)-DP

b = 1/ε であれば (**Δf * ε**,0)-DP

Report Noisy Max Mechanism

- 差分プライバシーのより良い評価



f が数えあげクエリの時、

b = 1/ε について **(ε,0)**-DP が成立

データセット D において、ある属性を持ったデータがいくつあるかを数える処理

数え上げクエリと argmax に特有の性質を利用する。

差分プライバシーの
Isabelle/HOLによる形式的検証

形式的検証の必要性

- 差分プライバシーはプログラムの小さな変更で崩れ得る。

差分プライバシーを満たす
Above thresholdメカニズム

Algorithm 2 SVT in Dwork and Roth 2014 [8].

Input: $D, Q, \Delta, T, c.$
1: $\epsilon_1 = \epsilon/2, \quad \rho = \mathsf{Lap}\,(c\Delta/\epsilon_1)$
2: $\epsilon_2 = \epsilon - \epsilon_1, \quad \text{count} = 0$
3: **for** each query $q_i \in Q$ **do**
4: $\quad \nu_i = \mathsf{Lap}\,(2c\Delta/\epsilon_1)$
5: $\quad$ **if** $q_i(D) + \nu_i \geq T + \rho$ **then**
6: $\quad\quad$ Output $a_i = \top$, $\rho = \mathsf{Lap}\,(c\Delta/\epsilon_2)$
7: $\quad\quad$ count = count + 1, **Abort** if count $\geq c$.
8: $\quad$ **else**
9: $\quad\quad$ Output $a_i = \bot$

厳密で正確な検証を行う
「ツール」が必要

いかなる差分プライバシーも
満たさないことが後からわかった例

Algorithm 3 SVT in Roth's 2011 Lecture Notes [17].

Input: $D, Q, \Delta, T, c.$
1: $\epsilon_1 = \epsilon/2, \quad \rho = \mathsf{Lap}\,(\Delta/\epsilon_1),$
2: $\epsilon_2 = \epsilon - \epsilon_1, \quad \text{count} = 0$
3: **for** each query $q_i \in Q$ **do**
4: $\quad \nu_i = \mathsf{Lap}\,(c\Delta/\epsilon_2)$
5: $\quad$ **if** $q_i(D) + \nu_i \geq T + \rho$ **then**
6: $\quad\quad$ Output $a_i = q_i(D) + \nu_i$
7: $\quad\quad$ count = count + 1, **Abort** if count $\geq c$.
8: $\quad$ **else**
9: $\quad\quad$ Output $a_i = \bot$

Algorithm 5 SVT in Stoddard et al. 2014 [20].

Input: $D, Q, \Delta, T.$
1: $\epsilon_1 = \epsilon/2, \quad \rho = \mathsf{Lap}\,(\Delta/\epsilon_1)$
2: $\epsilon_2 = \epsilon - \epsilon_1$
3: **for** each query $q_i \in Q$ **do**
4: $\quad \nu_i = 0$
5: $\quad$ **if** $q_i(D) + \nu_i \geq T + \rho$ **then**
6: $\quad\quad$ Output $a_i = \top$
7:
8: $\quad$ **else**
9: $\quad\quad$ Output $a_i = \bot$

関係ホーア論理による検証

[Barthe+, POPL2012] (他多数)

- パラメータ付きの二項関係版ホーア論理

$$\{\Phi\}\ c_1 \sim^{(\varepsilon,\delta)} c_2\ \{\Psi\}$$

2つの確率的プログラムを考えている

差分プライバシーパラメータ

M上の**二項関係**
Φ…事前条件
Ψ…事後条件

$$\{\mathtt{Adj}\}c \sim^{\varepsilon,\delta} c\{x\langle 1\rangle = x\langle 2\rangle\}$$

という形の”Hoare Triple” が
差分プライバシーと同値

”Hoare Triple” の解釈がこのような巧妙な性質を持つための
意味論の構成法を研究してきた (relational lifting,coupling)

推論規則

- 次数付きホーア論理 [Gaboardi+,ESOP 2021]の一種で、ホーア論理に類似の（パラメータが付きの）推論規則を持つ

$$\{\Phi\}\texttt{skip} \sim^{(0,0)} \texttt{skip}\{\Phi\}$$

$$\frac{\{\Phi\}c_1 \sim^{(\varepsilon,\delta)} c_2\{\Phi'\} \qquad \{\Phi'\}d_1 \sim^{(\varepsilon',\delta')} d_2\{\Psi\}}{\{\Phi\}c_1; d_1 \sim^{(\varepsilon+\varepsilon',\delta+\delta')} c_2; d_2\{\Psi\}}$$

$$\frac{\Phi' \subseteq \Phi \quad \{\Phi\}c_1 \sim^{(\varepsilon,\delta)} c_2\{\Psi\} \quad \Psi \subseteq \Psi' \quad \varepsilon \leq \varepsilon' \quad \delta \leq \delta'}{\{\Phi'\}c_1 \sim^{(\varepsilon',\delta')} c_2\{\Psi'\}}$$

差分プライバシーを満たす勾配降下法

[Lee & Kifer, KDD 2018]

- 検証が難しい例(最終的にこれの形式的検証をやりたい)。

  - ノイズ持ち点 ρ を持っておく。
    - ρ がゼロになるまで以下を繰り返す：
      - ρから支払い、ノイズを付加した勾配ベクトルgtを取得
        - -gtが勾配を下る方向のときは
          -gtの方向に勾配降下。出力wをアップデートする。
        - そうでないときは、ρを払戻し、wをアップデートせず、
          若干大きなノイズを付加して方向ベクトルgtを取り直す。
    - ρ は最終的には必ずゼロになり、必要なノイズが付加される。
    - 持ち点 ρ のノイズを付加 ⇒ (ε,δ)-DPを保証。

- 付加されるノイズの量がこれまでの実行に依存する。
  ＋付加されるノイズと勾配に応じて条件がかわる。

着想

- 先ほどのような例の差分プライバシーを検証したいが、
  関係ホーア論理でやったような形式的検証はやりづらい。

- 差分プライバシーの研究では様々な解析的手法が使われているが、
  それらも含めて、厳密な検証ができるのではないだろうか。

- ラドン＝ニコディムの定理の形式化は定理証明支援系で実装済。
  - 差分プライバシーを一般的な形で(i.e. 連続的分布で)形式化

定理証明支援系上で、
差分プライバシーの直接的な
形式化(形式的定義・証明)を行う

Isabelle/HOLによる形式的検証

形式的定義と形式的証明を与える

- Isabelle/HOLで差分プライバシーの形式化を進める。
  - 測度論や確率論のライブラリが充実、自動証明機能

- 現在の進捗状況
  - (標準的な)差分プライバシーの形式化
    - 差分プライバシーの定義をIsabelle/HOLで記述
    - 合成性などの性質の形式的証明をあたえた
      - 差分プライバシーのための統計的ダイバージェンス
  - Laplace mechanismの形式化
    - ラプラス分布の実装(正規分布は既にあった)
    - 差分プライバシーの形式的証明をあたえる
  - Report noisy max mechanismの形式化
    - 差分プライバシーを示す(2000行程度)

Isabelle/HOL

- 公式サイト：https://isabelle.in.tum.de/
- 証明支援系(proof assistant)の一つ。
  - 定義や定理をなんらかのプログラミング言語で記述、誤りのない証明を支援。(他にも Coq、Lean など)

```
primrec rev :: "'a list ⇒ 'a list" where
"rev [] = []" |
"rev (x # xs) = rev xs @ [x]"
```

Isabelle/HOLでの
リストの反転と結合の定義

```
primrec append :: "'a list ⇒ 'a list ⇒ 'a list" (infixr "@" 65) where
append_Nil: "[] @ ys = ys" |
append_Cons: "(x#xs) @ ys = x # xs @ ys"
```

Isabelle/HOL上で証明された
リストの反転と結合の性質

```
lemma rev_append [simp]: "rev (xs @ ys) = rev ys @ rev xs"
by (induct xs) auto

lemma rev_rev_ident [simp]: "rev (rev xs) = xs"
by (induct xs) auto
```

Isabelle/HOLでの確率的プログラム

- 測度論ライブラリを使う
- 確率的プログラムを可測関数(Isabelle/HOLの項)として扱う。
  - 実行可能な確率的プログラムも取り扱い可能だが、触れない。

- 例えば、

$$M : \mathbb{R} \to \mathrm{Prob}(\mathbb{R})$$

という可測関数は、

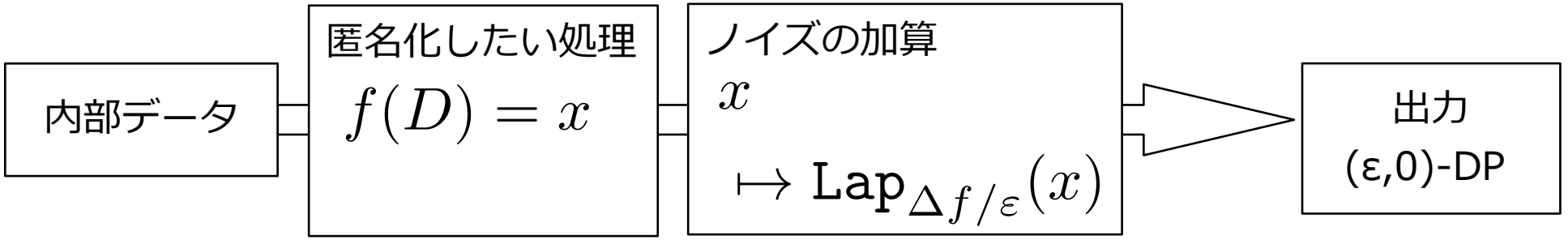
```
"M ∈ X →M prob algebra borel"
```

という条件を満たすものとして扱う

  - prob_algebra は確率分布全体の可測空間を構成(Giryモナド)
  - borel は実直線+ボレル代数

Isabelle/HOLでの確率的プログラム

- 1次元のラプラスメカニズムの場合



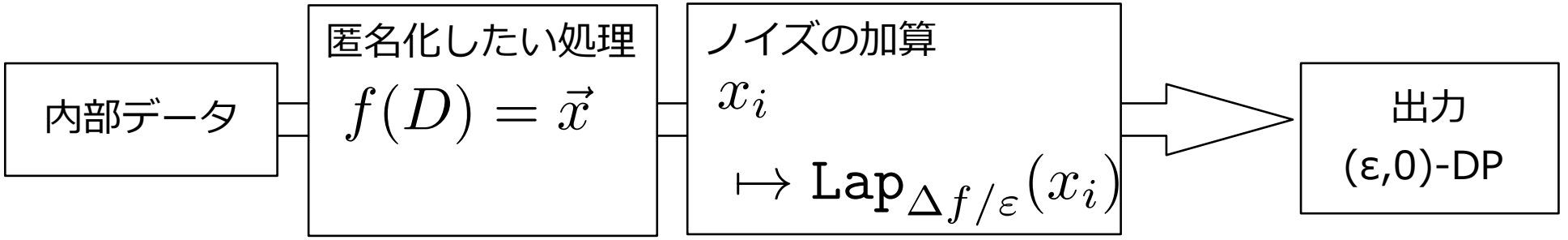

```
"LapMech_1dim ε x = Lap_dist ((real_of_ereal sensitivity) / ε) (f x)"

lemma Lap_dist_def2:
  shows "(Lap_dist b x) = do{r ← Lap_dist0 b; return borel (x + r)}"
```

平均 0 、尺度 b のラプラス分布

それを加算(return式；決定的)

Isabelle/HOLでの確率的プログラム

- m次元のラプラスメカニズムの場合




```
"LapMech_list ε x = Lap_dist_list ((real_of_ereal sensitivity) / ε) (f x)"
```

ラプラス分布を各成分に加算する処理(下記)

```
primrec Lap_dist_list :: "real list ⇒ (real list) measure" where
  "Lap_dist_list [] = return (listM borel) []"
  "Lap_dist_list (x # xs) =
    do{x1 ← (Lap_dist b x);
       x2 ← (Lap_dist_list xs);
       return (listM borel) (x1 # x2)}"
```

注意：Isabelle/HOLの測度・可測性

- Isabelle/HOLの測度と可測空間の両方が
  「測度空間」を表す型で書かれる。

三つ組$M = (|M|, \Sigma_M, \mu)$

```
typedef‹tag important› 'a measure =
  "{(Ω::'a set, A, μ). (∀a∈-A. μ a = 0) ∧ measure space Ω A μ }"
```

  - それぞれの要素を取り出す操作（ M :: "'a measure" ）：

```
"space M"
  :: "'a set"
```

```
"sets M"
  :: "'a set set"
```

```
"emeasure M"
  :: "'a set ⇒ ennreal"
```

- 関数の可測性は「可測関数全体の集合」を用いて書かれる。

```
M →M N =
{f ∈ space M → space N.
 ∀y∈sets N. f -` y ∩ space M ∈ sets M}
```

関数であって、可測集合
を引き戻すと可測になる

差分プライバシーの形式的定義

- データセットの型と隣接関係を抽象化した形式的定義を与える
  - 元の定義[Dwork+, TCC 2006]
    ランダム化されたメカニズム M: **Datasets** → Prob(Y) が
    (ε,δ)-差分プライバシー(DP)を満たすとは、
    隣接するデータセット **D1～D2** について以下が成立：

$$\Pr[M(D_1) \in S] \leq \exp(\varepsilon) \Pr[M(D_2) \in S] + \delta$$

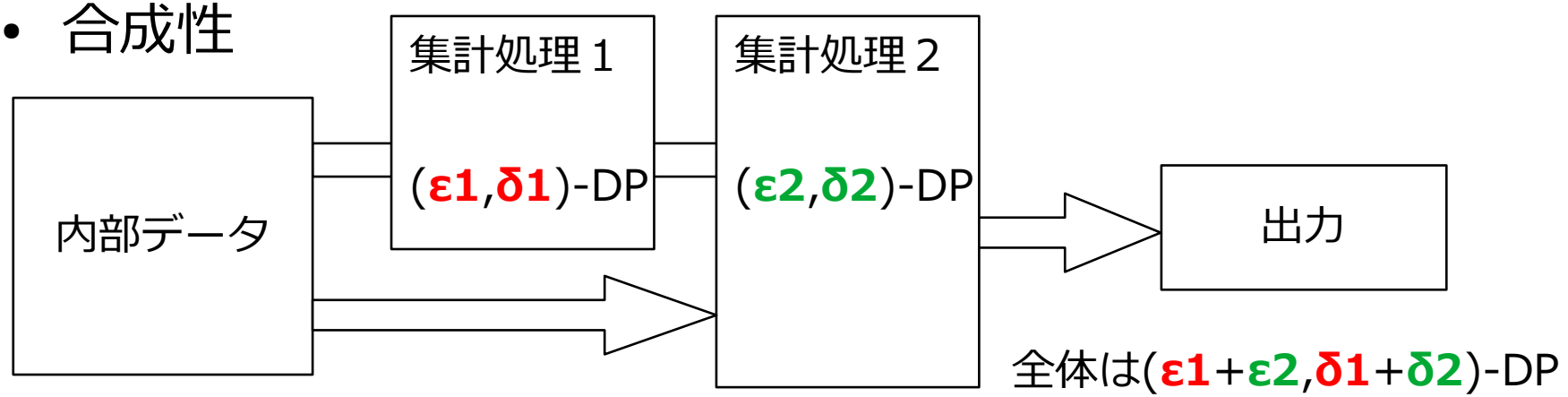
- Isabelle/HOLによる形式的定義

```
definition DP_inequality:: "'a measure ⇒ 'a measure ⇒ real ⇒ real ⇒ bool" where
  "DP_inequality M N ε δ ≡ (∀ A ∈ sets M. measure M A ≤ (exp ε) * measure N A + δ)"

definition differential_privacy :: "('a ⇒ 'b measure) ⇒ ('a rel) ⇒ real ⇒ real ⇒ bool "
  where
  "differential_privacy M adj ε δ ≡
  ∀(d1,d2)∈adj. DP_inequality (M d1) (M d2) ε δ ∧ DP_inequality (M d2) (M d1) ε δ"
```

隣接関係adjが対称的という条件も落としている

基本的性質の形式的表現

- 合成性



- Isabelle/HOLの命題として書き下す (Isabelle/HOL上で形式的に証明する)

```
proposition differential_privacy_composition_adaptive:
  assumes "ε ≥ 0"and "δ ≥ 0"
    and "ε' ≥ 0"and "δ' ≥ 0"
    and M: "M ∈ X →M (prob_algebra Y)"
    and DPM: "differential_privacy M adj ε δ"
    and N: "N ∈ (X ⊗M Y) →M (prob_algebra Z)"
    and DPN: "∀ y ∈ space Y. differential_privacy (λ x. N (x,y)) adj ε' δ'"
    and "adj ⊆ (space X) × (space X)"
  shows "differential_privacy (λx. do{y ← M x; N (x, y)}) adj (ε + ε') (δ + δ')"
```

基本的性質の形式的証明における工夫

- Isabelle/HOLで書き下した合成性

```
proposition differential_privacy_composition_adaptive:
  assumes "ε ≥ 0"and "δ ≥ 0"
    and "ε' ≥ 0"and "δ' ≥ 0"
    and M: "M ∈ X →M (prob_algebra Y)"
    and DPM: "differential_privacy M adj ε δ"
    and N: "N ∈ (X ⊗M Y) →M (prob_algebra Z)"
    and DPN: "∀ y ∈ space Y. differential_privacy (λ x. N (x,y)) adj ε' δ'"
    and "adj ⊆ (space X) × (space X)"
  shows "differential_privacy (λx. do{y ← M x; N (x, y)}) adj (ε + ε') (δ + δ')"
```

- 上記のような差分プライバシーの基本的性質を証明する工夫
  - 確率的プログラム(を表す確率分布モナド)の構造に
    もっと適合した数学的構造を考える

ダイバージェンスによる特徴づけ

[Barthe & Olmedo, ICALP 2013]

[Sato & Katsumata, MSCS 2023]

- 差分プライバシーの「ダイバージェンスによる定式化」
  - いくつかの定理の形式的証明が楽になる。
  - 実は、DPを検証する関係ホーア論理の本質的なデータ

- M: Datasets → Prob(Y) が(ε,δ)-DPを満たす

  ⇔ 隣接するデータD1～D2について

$$\forall S \subseteq Y.\quad \Pr[M(D_1) \in S] \leq \exp(\varepsilon)\Pr[M(D_2) \in S] + \delta$$

  ⇔ 隣接するデータD1～D2について

$$\underbrace{\sup_{S \subseteq Y}\left(\Pr[M(D_1) \in S] - \exp(\varepsilon)\Pr[M(D_2) \in S]\right)}_{\Delta^{\varepsilon}(M(D_1)||M(D_2))} \leq \delta$$

統計的ダイバージェンス(分布間の擬距離)

ダイバージェンス Δ^{ε} の基本的性質

- 反射性

$$\Delta_X^0(\mu, \mu) = 0$$

- 単調性

$$\varepsilon \geq \varepsilon_2 \implies \Delta_X^{\varepsilon}(\mu, \nu) \leq \Delta_X^{\varepsilon_2}(\mu, \nu)$$

- 合成性

$$\Delta_X^{\varepsilon}(\mu, \nu) \leq \delta \text{ and } \forall x \in X. \Delta_Y^{\varepsilon_2}(f(x), g(x)) \leq \delta_2$$
$$\implies \Delta_X^{\varepsilon+\varepsilon_2}(\mu \gg\!= f, \nu \gg\!= g) \leq \delta + \delta_2$$

差分プライバシーの
(いくつかある)合成性や
後処理に対する安定性は
この３つから導かれる

確率分布モナドのbind：

$$\{x \leftarrow \mu; f(x)\}$$

μからサンプリングした x をfに適用、
一連の結果の確率分布を取る

合成性の証明（スケッチ）

$$\Pr[\mu \gg\!= f \in S] - \exp(\varepsilon_1 + \varepsilon_2) \Pr[\nu \gg\!= g \in S]$$

$$= \int f(x)(S) d\mu - \exp(\varepsilon) \int g(x)(S) d\nu(x)$$

$$= \int f(x)(S) \cdot \frac{d\mu}{d\pi}(x)\ d\pi - \exp(\varepsilon_1 + \varepsilon_2) \int g(x)(S) \cdot \frac{d\nu}{d\pi}(x)\ d\pi(x)$$

$$= \int f(x)(S) \cdot \frac{d\mu}{d\pi}(x) - \exp(\varepsilon_1 + \varepsilon_2) g(x)(S) \cdot \frac{d\nu}{d\pi}(x)\ d\pi(x)$$

$$\leq \int (\max(0, f(x)(S) - \delta_2) + \delta_2) \cdot \frac{d\mu}{d\pi}(x)$$

$$- \exp(\varepsilon_1) \min(1, \exp(\varepsilon_2) g(x)(S)) \cdot \frac{d\nu}{d\pi}(x)\ d\pi(x)$$

$$= \int \max(0, f(x)(S) - \delta_2) \cdot \frac{d\mu}{d\pi}(x)$$

$$- \exp(\varepsilon_1) \min(1, \exp(\varepsilon_2) g(x)(S)) \cdot \frac{d\nu}{d\pi}(x)\ d\pi(x) + \int \delta_2 \frac{d\mu}{d\pi}(x)\ d\pi$$

$$\leq \int_B \left(\frac{d\mu}{d\pi}(x) - \exp(\varepsilon_1) \frac{d\nu}{d\pi}(x) \right) \cdot \min(1, \exp(\varepsilon_2) \cdot g(x)(S))\ d\pi + \int \delta_2 \frac{d\mu}{d\pi}(x)\ d\pi(x)$$

$$\leq \delta_1 + \delta_2$$

Giryモナドのbindを積分に展開

共通の測度 π を考え
密度関数を取る変形
(Radon-Nikodym)

同じ議論を、Isabelle/HOL
上で形式的に行った結果、
約180行かかった。

移項や不等式変形が
たくさんある。

ダイバージェンス Δ^{ε} の形式化

- 定義　(※確率測度であることはまだ言ってない)

```
definition DP_divergence :: "'a measure ⇒ 'a measure ⇒ real ⇒ ereal " where
  "DP_divergence M N ε
    = Sup {ereal(measure M A - (exp ε) * measure N A) | A::'a set. A ∈ (sets M)}"
```

- 反射性・単調性・合成性（※定理側で確率測度であることを言う）

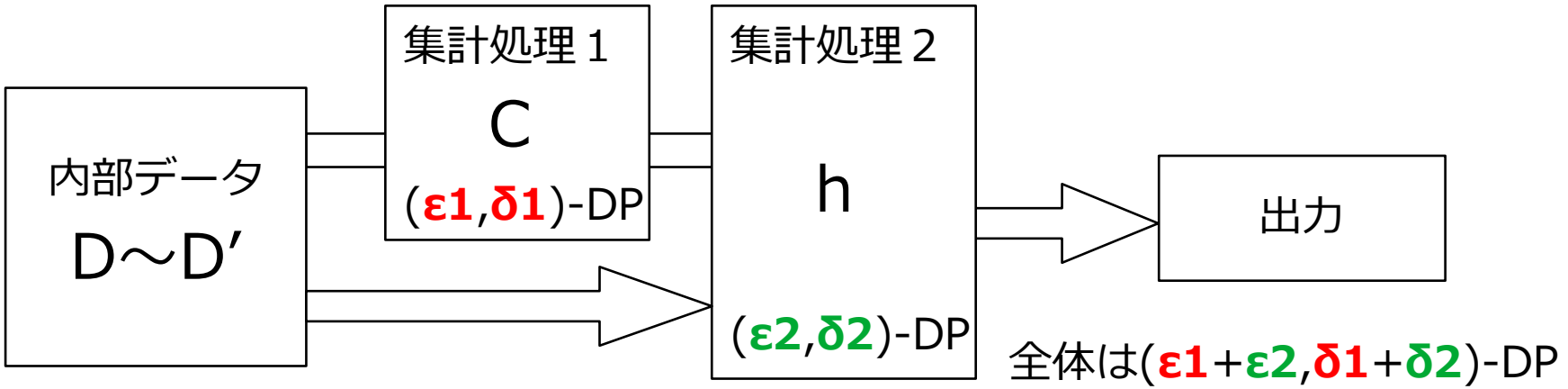
```
lemma DP_divergence_monotonicity:
  assumes M: "M ∈ space (prob_algebra L)"
    and N: "N ∈ space (prob_algebra L)"
    and "ε1 ≤ ε2"
  shows "DP_divergence M N ε2 < DP_divergence M N ε1"
```

```
lemma DP_divergence_reflexivity:
  shows "DP_divergence M M 0 = 0"
```

```
proposition DP_divergence_composability:
  assumes M: "M ∈ space (prob_algebra L)"
    and N: "N ∈ space (prob_algebra L)"
    and f: "f ∈ L →M prob_algebra K"
    and g: "g ∈ L →M prob_algebra K"
    and div1: "DP_divergence M N ε1 ≤ (δ1 :: real)"
    and div2: "∀x ∈ (space L). DP_divergence (f x) (g x) ε2 ≤ (δ2 :: real)"
    and "0 ≤ ε1" and "0 ≤ ε2"
  shows "DP_divergence (M ≫= f) (N ≫= g) (ε1 + ε2) ≤ δ1 + δ2"
```

DPの合成性とダイバージェンスの合成性

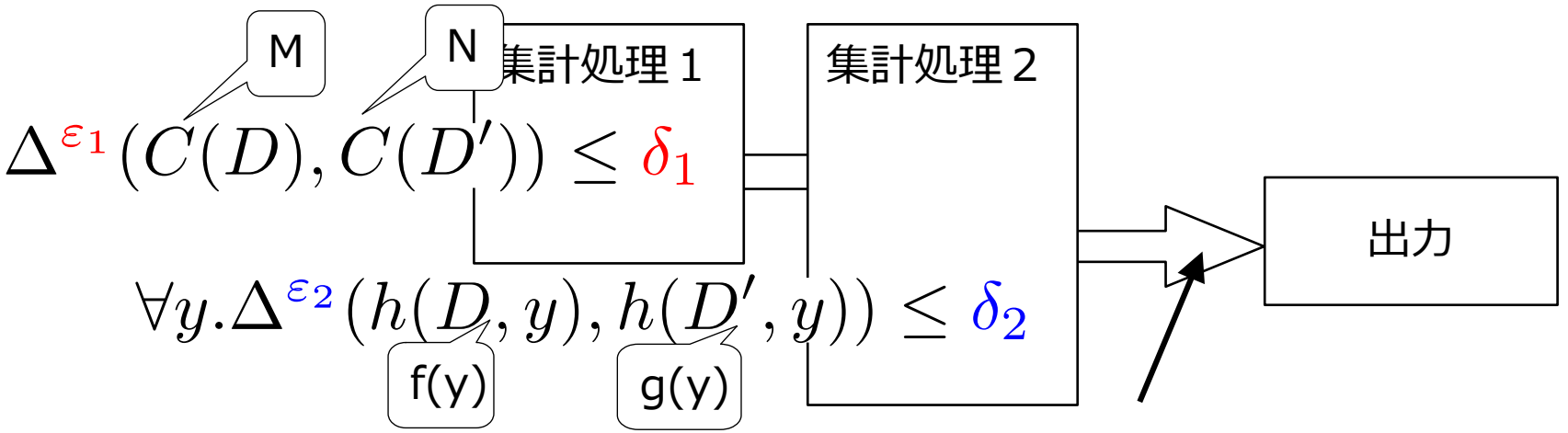
- DPの合成性




```
proposition differential_privacy_composition_adaptive:
  assumes "ε ≥ 0"and "δ ≥ 0"
    and "ε' ≥ 0"and "δ' ≥ 0"
    and M: "M ∈ X →M (prob_algebra Y)"
    and DPM: "differential_privacy M adj ε δ"
    and N: "N ∈ (X ⊗M Y) →M (prob_algebra Z)"
    and DPN: "∀ y ∈ space Y. differential_privacy (λ x. N (x,y)) adj ε' δ'"
    and "adj ⊆ (space X) × (space X)"
  shows "differential_privacy (λx. do{y ← M x; N (x, y)}) adj (ε + ε') (δ + δ')"
```

DPの合成性とダイバージェンスの合成性

- 隣接するD～D′ を固定



$$\Delta^{\varepsilon_1}(C(D), C(D')) \leq \delta_1$$

$$\forall y. \Delta^{\varepsilon_2}(h(D, y), h(D', y)) \leq \delta_2$$

$$\Delta_X^{\varepsilon_1+\varepsilon_2}(C(D) \gg\!= (\lambda y. h(D, y)), C(D') \gg\!= (\lambda y. h(D', y)))$$

$$\leq \delta_1 + \delta_2$$

ダイバージェンスの合成性で
DPの合成性を導ける

```
proposition DP_divergence_composability:
  assumes M: "M ∈ space (prob_algebra L)"
    and N: "N ∈ space (prob_algebra L)"
    and f: "f ∈ L →M prob_algebra K"
    and g: "g ∈ L →M prob_algebra K"
    and div1: "DP_divergence M N ε1 ≤ (δ1 :: real)"
    and div2: "∀x ∈ (space L). DP_divergence (f x) (g x) ε2 ≤ (δ2 :: real)"
    and "0 ≤ ε1" and "0 ≤ ε2"
  shows "DP_divergence (M ≫= f) (N ≫= g) (ε1 + ε2) ≤ δ1 + δ2"
```

Laplace Mechanismの形式化

- ラプラス分布の密度関数・累積分布関数を実装

```
definition laplace_density :: "real ⇒ real ⇒ real ⇒ real" where
  "laplace_density l m x = (if l > 0 then (exp(-¦ x - m¦ / l) / (2* l)) else 0)"

definition laplace_CDF :: "real ⇒ real ⇒ real ⇒ real" where
  "laplace_CDF l m x = (if l > 0
      then (if x < m then (exp((x - m) / l) / 2) else (1 - exp(-(x - m) / l) / 2)) else 0)"
```

・・・（様々な性質の形式的証明を与える）・・・

- ラプラス分布によるノイズ付加を実装

$\mathrm{Lap}_b : \mathbb{R} \to \mathrm{Prob}(\mathbb{R})$

$\mathrm{Lap}_b(x)$ は平均 x 尺度 b のラプラス分布

```
definition Lap_dist :: "real ⇒ real ⇒ real measure" where
  "Lap_dist b μ =
    (if b ≤ 0 then return borel μ
      else density lborel (laplace_density b μ))"

lemma Lap_dist_def2:
  shows "(Lap_dist b x) = do{r ← Lap_dist0 b; return borel (x + r)}"
```

Laplace Mechanismの形式化

- m次元のノイズ付加の実装　(リストとして形式化)

$\mathrm{Lap}_b^m : \mathbb{R}^m \to \mathrm{Prob}(\mathbb{R}^m)$

$\mathrm{Lap}_b^m(\vec{x})$ は各成分が平均 x[i] 尺度 b のラプラス分布

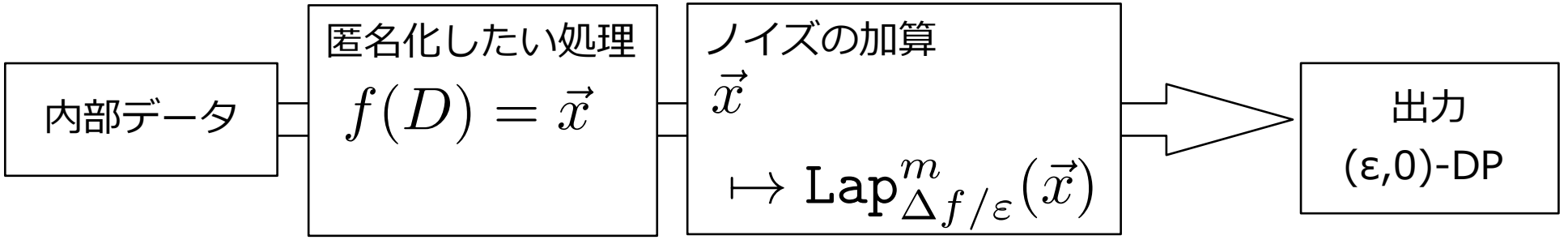
```
primrec Lap_dist_list :: "real list ⇒ (real list) measure" where
  "Lap_dist_list [] = return (listM borel) []"
  "Lap_dist_list (x # xs) = do{x1 ← (Lap_dist b x);
     x2 ← (Lap_dist_list xs); return (listM borel) (x1 # x2)}"
```

  - 所望のノイズ付加であること

```
lemma Lap_dist_list_def2:
  shows "Lap_dist_list xs =
    do{ys ← (Lap_dist0_list (length xs));
      return (listM borel) (map2 (+) xs ys)}"
```

２つのリストの各成分を足したリストを作る

Laplace Mechanismの形式化



- f の感度を定義する

```
definition sensitivity:: ereal where
  "sensitivity = Sup{ ereal ( ∑ i∈{1..m}. ¦ nth (f x) (i-1) - nth (f y) (i-1) ¦)
                      | x y::'a. x ∈ space X ∧ y ∈ space X ∧ (x,y) ∈ adj}"
```

- Laplace Mechanismを定義

```
"LapMech_list ε x = Lap_dist_list ((real_of_ereal sensitivity) / ε) (f x)"
```

- 差分プライバシーの形式的証明 ((ε,0)-DPであることを示す)

```
proposition differential_privacy_LapMech_list:
  assumes pose: "ε > 0"
    and "sensitivity > 0"
    and "sensitivity < ∞"
  shows "differential_privacy (LapMech_list ε) adj ε 0"
```

具体的なデータセットの隣接関係の形式化

- データセットは自然数列、隣接関係は以下のように定義された。

$$D \in \mathbb{N}^n \qquad D \sim D' \iff \|D - D'\|_1 \leq 1$$

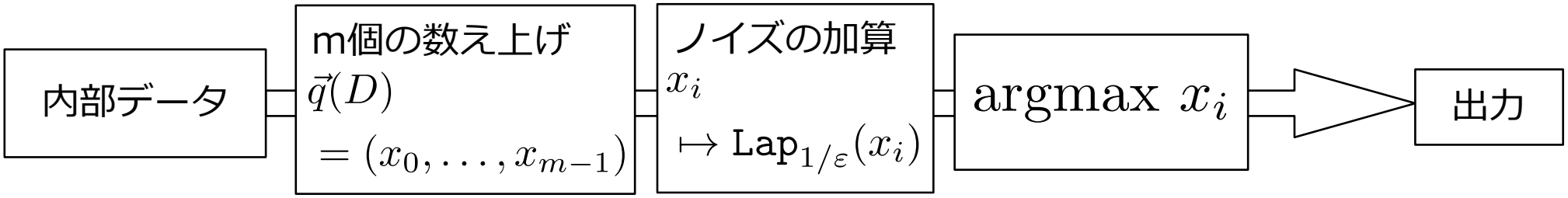
  - locale というIsabelle/HOLの機能を使ってこれを実装する。

```
interpretation L1_norm_list "(UNIV::nat set)"
  "(λ x y. ¦int x - int y¦)" n
```

    - L1_norm_list という自作のlocaleは、距離空間と n から長さ n のリストの距離空間(L1-norm風の)を構成する。
      - locale の中身は(メタな)定義・補題の集まり。
    - 自然数上の普通の距離と n を与えてL1_norm_list を実体化。
      - N^n と隣接性の形式的表現が得られる。

Report Noisy Max Mechanismの形式化

- 数え上げクエリ＋ラプラス分布によるノイズ加算 + argmax

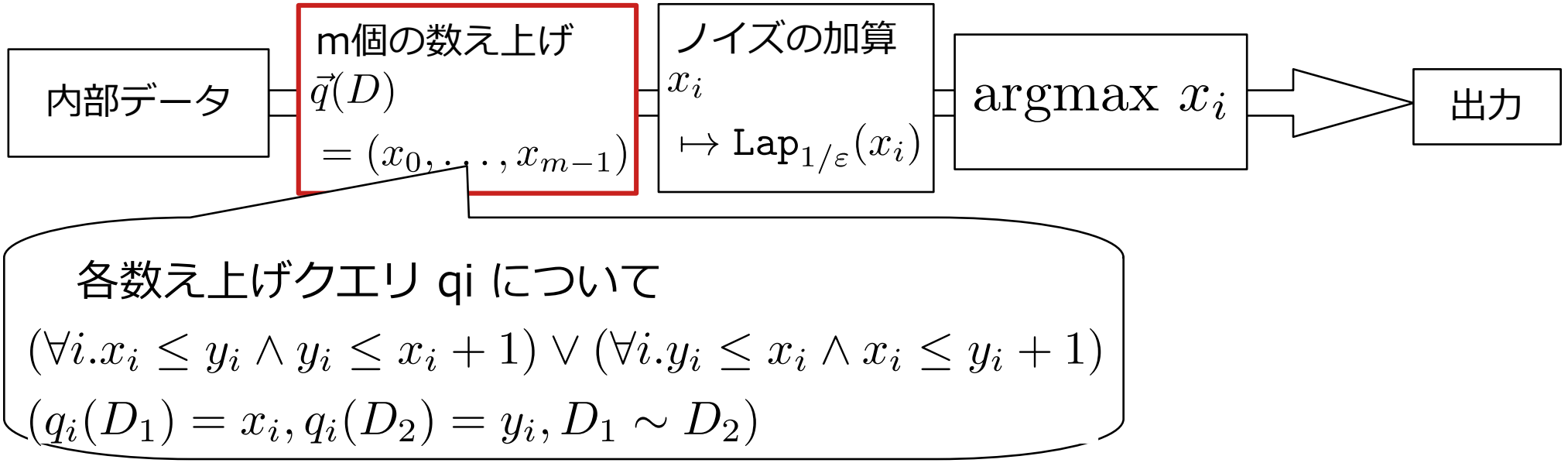



```
definition RNM_counting :: "real ⇒ nat list ⇒ nat measure" where
  "RNM_counting ε x = do {
 y ← Lap_dist_list (1 / ε) (counting_query x);
 return (count_space UNIV) (argmax_list y)
 }"

theorem differential_privacy_LapMech_RNM_AFDP:
  assumes pose: "(ε::real) > 0"
  shows " differential_privacy (RNM_counting ε) adj_L1_norm ε 0"
```

数え上げクエリの形式化

(形式的定義は省く)

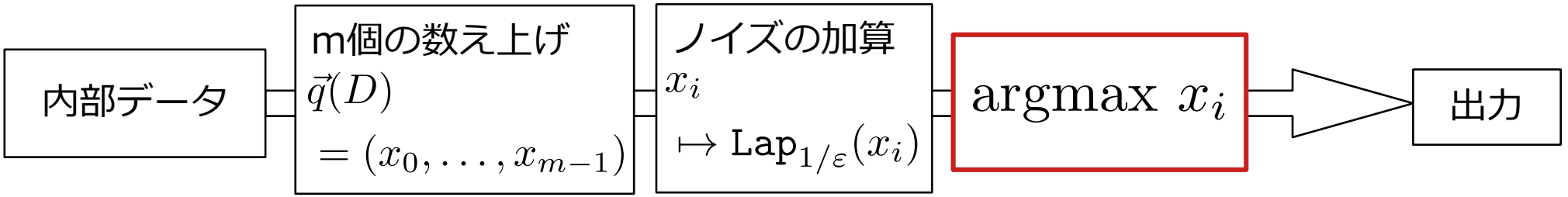
- **数え上げクエリ**＋ラプラス分布によるノイズ加算 + argmax




```
lemma finer_sensitivity_counting_query:
  assumes "(xs,ys) ∈ adj_L1_norm"
  shows "list_all2 (λ x y. x ≥ y ∧ x ≤ y + 1)
           (counting_query xs) (counting_query ys)
         ∨ list_all2 (λ x y. x ≥ y ∧ x ≤ y + 1)
           (counting query ys) (counting query xs)"
```

argmaxの形式化

(形式的定義は省く)



$$\operatorname*{argmax}_j x_j = i \iff \max_{j \neq i} x_j \leq x_i \wedge \max_{j<i} < x_i$$

- この命題を形式化するために、argmax+リストの挿入を定義

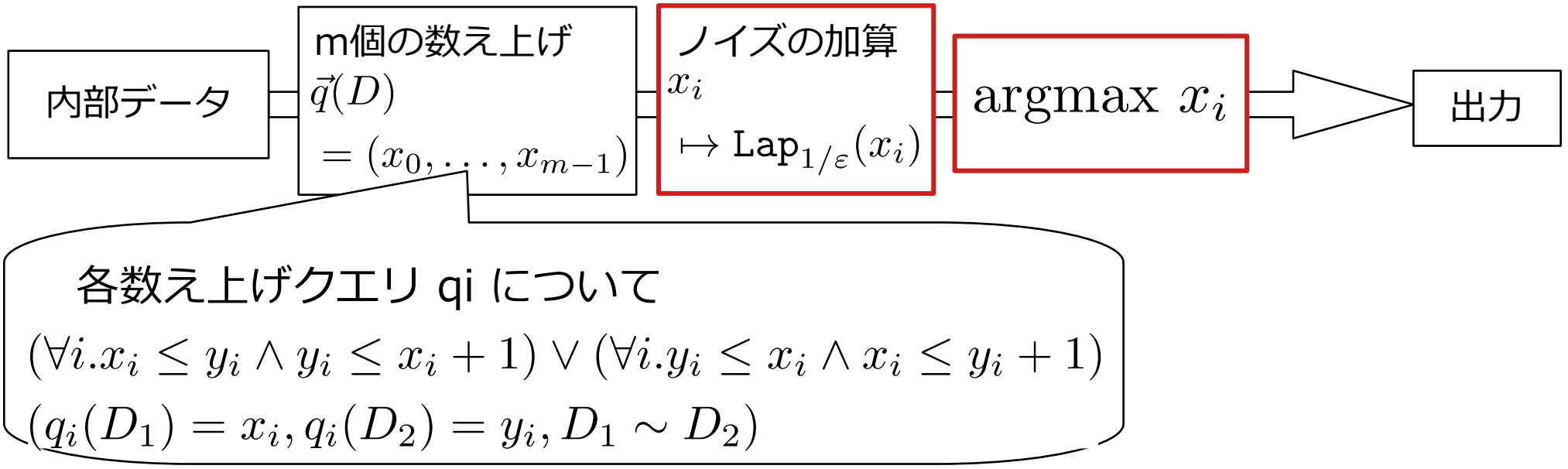
```
definition argmax_insert :: "real ⇒ real list ⇒ nat ⇒ nat" where
  "argmax_insert k ks i = argmax_list (list_insert k ks i)"
```

  - k を i 番目の要素と考えると、以下のように形式化される

```
lemma argmax_insert_i_i:
  assumes "m ≤ n"
    and "length xs = n"
  shows "(argmax_insert k xs m = m) ⟷
    (ereal k > (fst (max_argmax (drop m xs))) ∧ (ereal k ≥ (fst (max_argmax (take m xs)))))"
```

本体の差分プライバシーの形式化

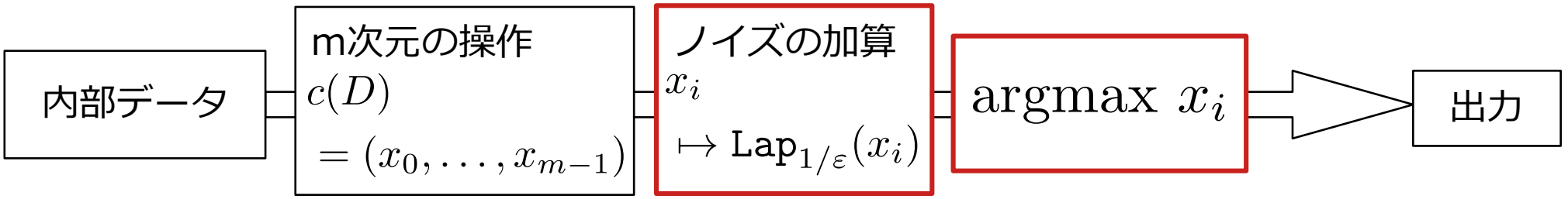
- ノイズ加算 + argmax を評価



  - Laplace mehcanism によく似ているが前提条件が異なる。
    (sensitivityよりもシビアな条件をcounting queriesが満たす)
  - 最後が argmax であることが本質的に効いてくる。

本体の差分プライバシーの形式化

- 数え上げクエリをいったん抽象化し、本体だけを見る。

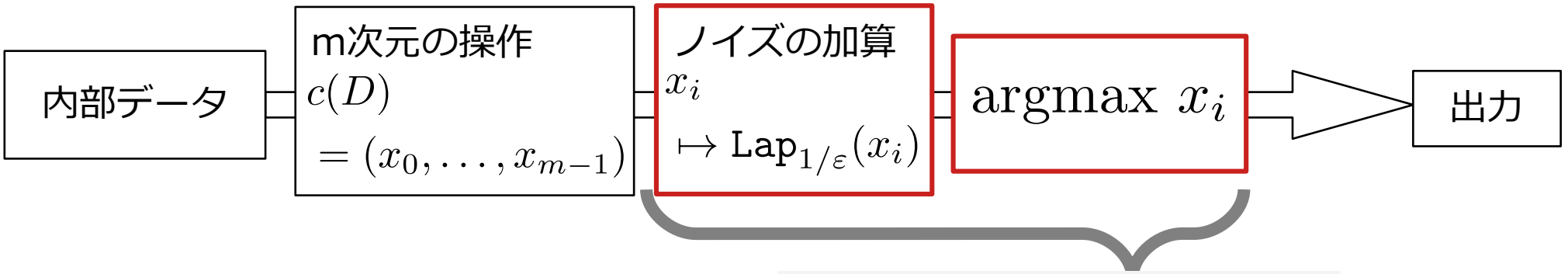


  - 抽象化した c を扱うために証明用のlocaleを自作。

```
locale Lap_Mechanism_RNM_mainpart =
  fixes M::"'a measure"
    and adj::"'a rel"
    and c::"'a ⇒ real list"
  assumes c: "c ∈ M →M (listM borel)"
    and cond: "∀ (x,y) ∈ adj.
          list_all2 (λx y. y ≤ x ∧ x ≤ y + 1) (c x) (c y)
        ∨ list_all2 (λx y. y ≤ x ∧ x ≤ y + 1) (c y) (c x)"
    and adj: "adj ⊆ (space M) × (space M)"
begin
```

数え上げクエリが満たす条件

Report Noisy Max Mechanism
証明の要点(１)



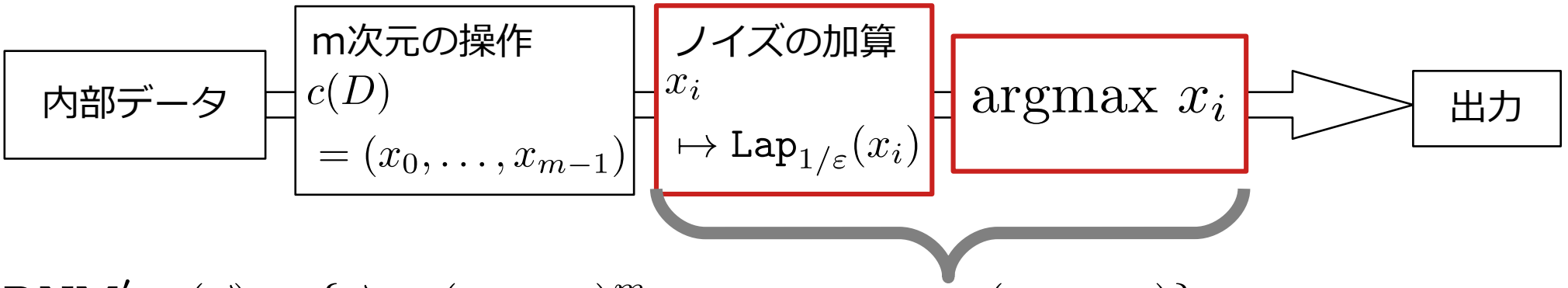

```
definition RNM' :: "real list ⇒ nat measure" where
  "RNM' zs = do {y ← Lap_dist_list (1 / ε) (zs);
    return (count_space UNIV) (argmax list y)}"
```

各 i が出る確率について不等式を証明

```
lemma DP_RNM'_M_i:
  fixes xs ys :: "real list" and i n :: nat
  assumes lxs: "length xs = n"
    and lys: "length ys = n"
    and adj: "list_all2 (λ x y. x ≥ y ∧ x ≤ y + 1) xs ys"
  shows "𝒫(j in (RNM' xs). j = i) ≤ (exp ε) * 𝒫(j in (RNM' ys). j = i)
    ∧ 𝒫(j in (RNM' ys). j = i) ≤ (exp ε) * 𝒫(j in (RNM' xs). j = i)"
```

Report Noisy Max Mechanism
証明の要点(２)

- 計算のためにプログラムを変形する



$$\mathrm{RNM}'_{m,\varepsilon}(\vec{x}) = \{\vec{r} \leftarrow (\mathtt{Lap}_{1/\varepsilon})^m; \mathrm{return}\ \ \underset{j}{\mathrm{argmax}}(x_j + r_j)\}$$

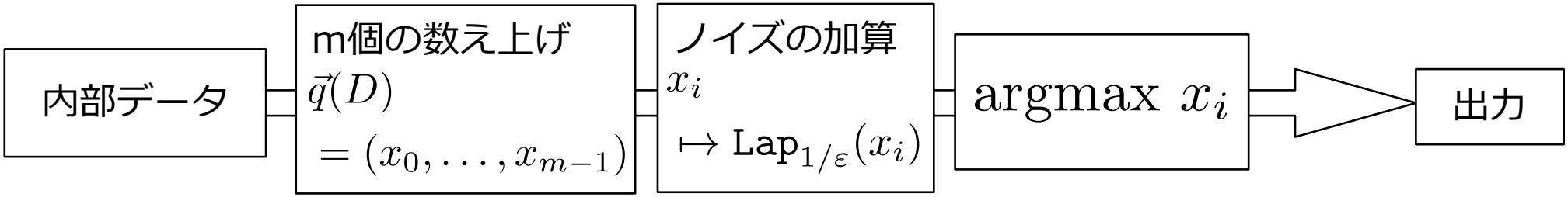
$$= \{\vec{r}_{\neq i} \leftarrow (\mathtt{Lap}_{1/\varepsilon})^{m-1}; r_i \leftarrow \mathtt{Lap}_{1/\varepsilon}; \mathrm{return}\ \ \underset{j}{\mathrm{argmax}}(x_j + r_j)\}$$

この後は、
ri 以外を固定して不等式を評価
→ ri 以外の分布で両辺を積分

```
lemma RNM'_expand:
  fixes n :: nat
  assumes "length xs = n" and "i ≤ n"
  shows "(RNM' (list_insert x xs i))
    = do{rs ← (Lap_dist0_list (1 / ε) (length xs)); (RNM_M xs rs x i)}"
```

最終的に得られるもの

- 数え上げクエリで抽象化した c を具体化し、
  Report noisy max の差分プライバシーの形式化を得る




```
definition RNM_counting :: "real ⇒ nat list ⇒ nat measure" where
  "RNM_counting ε x = do {
 y ← Lap_dist_list (1 / ε) (counting_query x);
 return (count_space UNIV) (argmax_list y)
 }"

theorem differential_privacy_LapMech_RNM_AFDP:
  assumes pose: "(ε::real) > 0"
  shows " differential_privacy (RNM_counting ε) adj_L1_norm ε 0"
```

隣接関係
$\|D - D'\|_1 \leq 1$

現在の形式化の状況

- Isabelle/HOLで差分プライバシーの形式化を行った
  - 連続的分布を扱うことが可能。

- 現在の進捗状況
  - (標準的な)差分プライバシーの形式化
    - 差分プライバシーの定義をIsabelle/HOLで記述
    - 合成性などの性質の形式的証明をあたえた
      - 差分プライバシーのための統計的ダイバージェンス
  - Laplace mechanismの形式化
    - ラプラス分布の実装(正規分布は既にあった)
    - 差分プライバシーの形式的証明をあたえる
  - Report noisy max mechanismの形式化
    - 差分プライバシーを示す(2000行程度)

- 最新の研究の形式化まではまだ遠く、やることが多い。

ありがとうございました！