Isabelle/HOLによる 差分プライバシーの形式的検証 についての進捗報告

TPP 2023

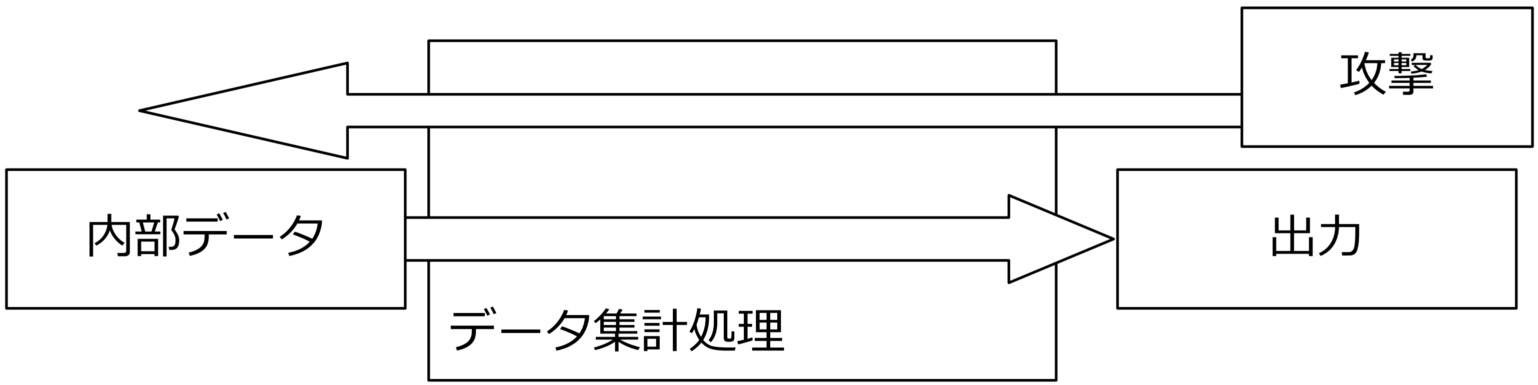
2023/10/31

佐藤哲也　(東京工業大学)

(special thanks: 勝股審也・南出靖彦)

差分プライバシーの背景

- 背景知識攻撃

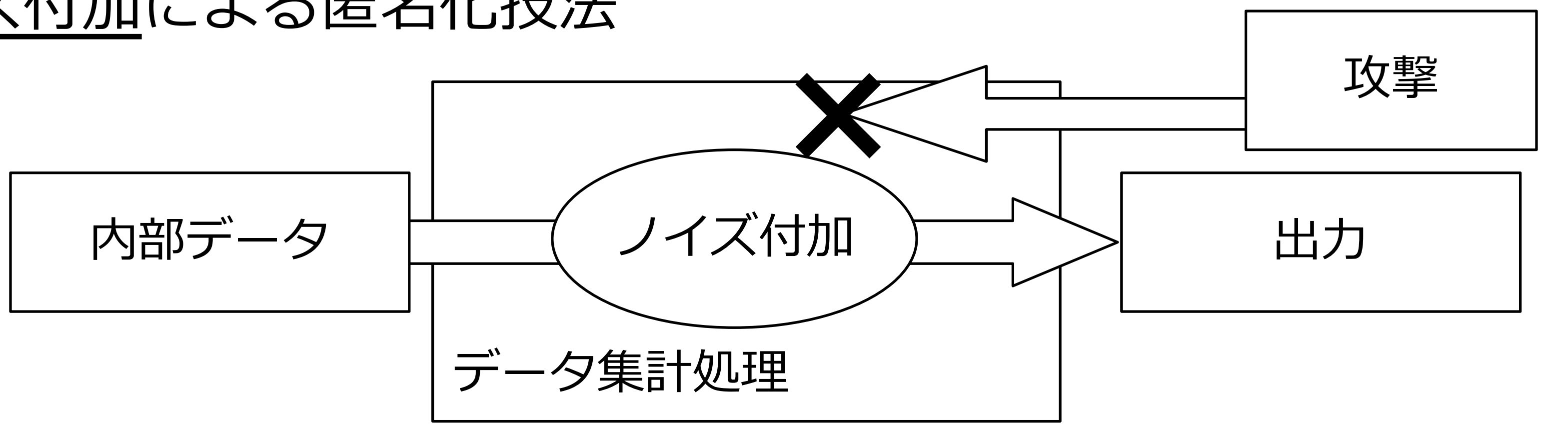


  - 内部データ自体は保護されていたとしても、

    十分な背景知識があると、データベースの出力を見て内部データを(統計的に)推測することが可能となる。

差分プライバシー

- ノイズ付加による匿名化技法



  - 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(もっと言えば任意の統計的攻撃)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

差分プライバシーの定義

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

      （隣接関係、データ更新 D1 → D2 1ステップの差分）

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

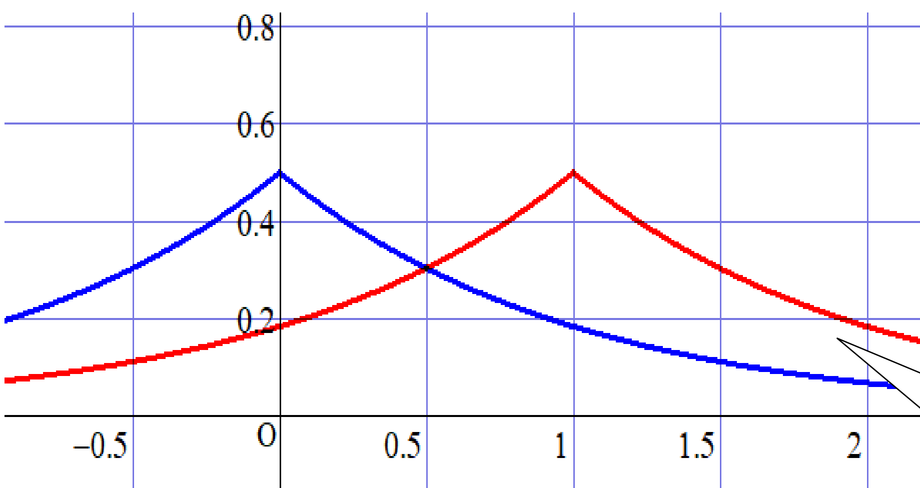
    - 直観：**確率 δ の場合を除き、確率比が ε で押さえられる**
      ((ε,δ)=(0,0) のとき、確率分布は一致する）

ラプラスメカニズム

- 平均0分散 2ε^2 のラプラス分布からなるノイズを加算
  これは(隣接関係を|x-y|≦1としたとき)　(ε,0)-DPを満たす。

$$\mathtt{Lap}_\varepsilon : \mathbb{R} \to \mathrm{Prob}(\mathbb{R})$$

$\mathtt{Lap}_\varepsilon(x)$ は平均 x 分散 $2\varepsilon^2$ のラプラス分布



$$\begin{aligned}
\Pr[\mathtt{Lap}_\varepsilon(x) = z] &= \frac{1}{2\varepsilon}\exp(-\frac{|x-z|}{\varepsilon}) \\
&\le \frac{1}{2\varepsilon}\exp(-\frac{|y-z|-|x-y|}{\varepsilon}) \\
&\le \exp(\varepsilon)\frac{1}{2\varepsilon}\exp(-\frac{|y-z|}{\varepsilon})]] \\
&= \exp(\varepsilon)\Pr[\mathtt{Lap}_\varepsilon(y) = z]
\end{aligned}$$

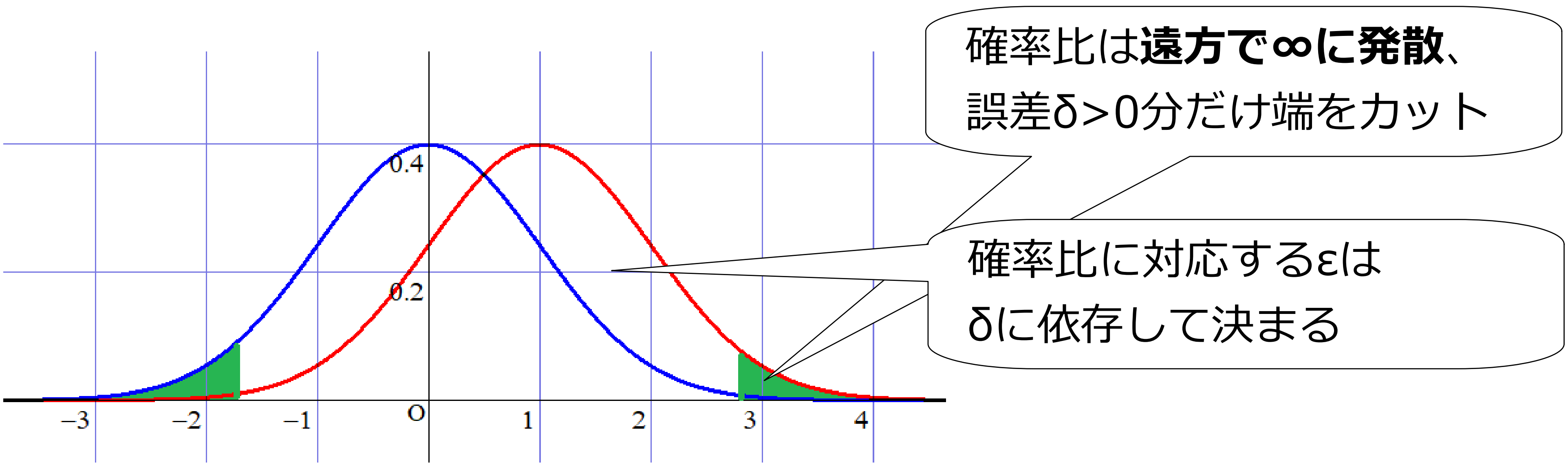
確率比(密度比)が一定以下

ガウシアンメカニズム

- 平均０の正規分布を加算。(隣接関係を|x-y|≦1としたとき)
  適当なパラメータにおける(ε,δ)-DPを満たす。

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

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



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: X → 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

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

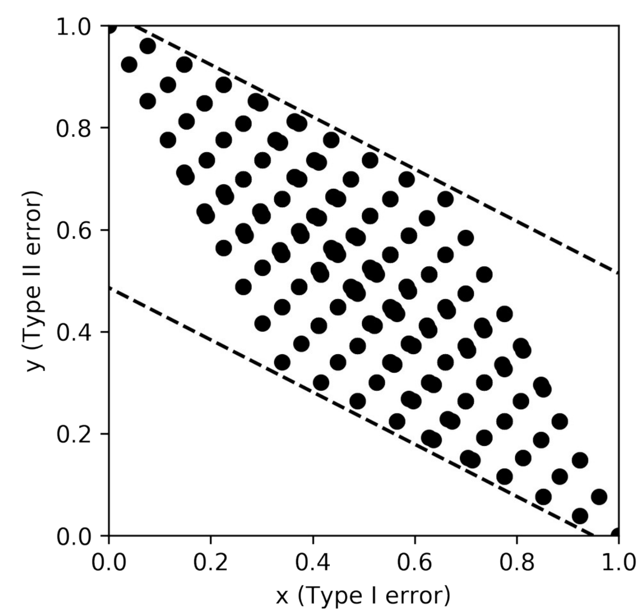
Rejection　Type I error　Type II error　privacy region

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

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。

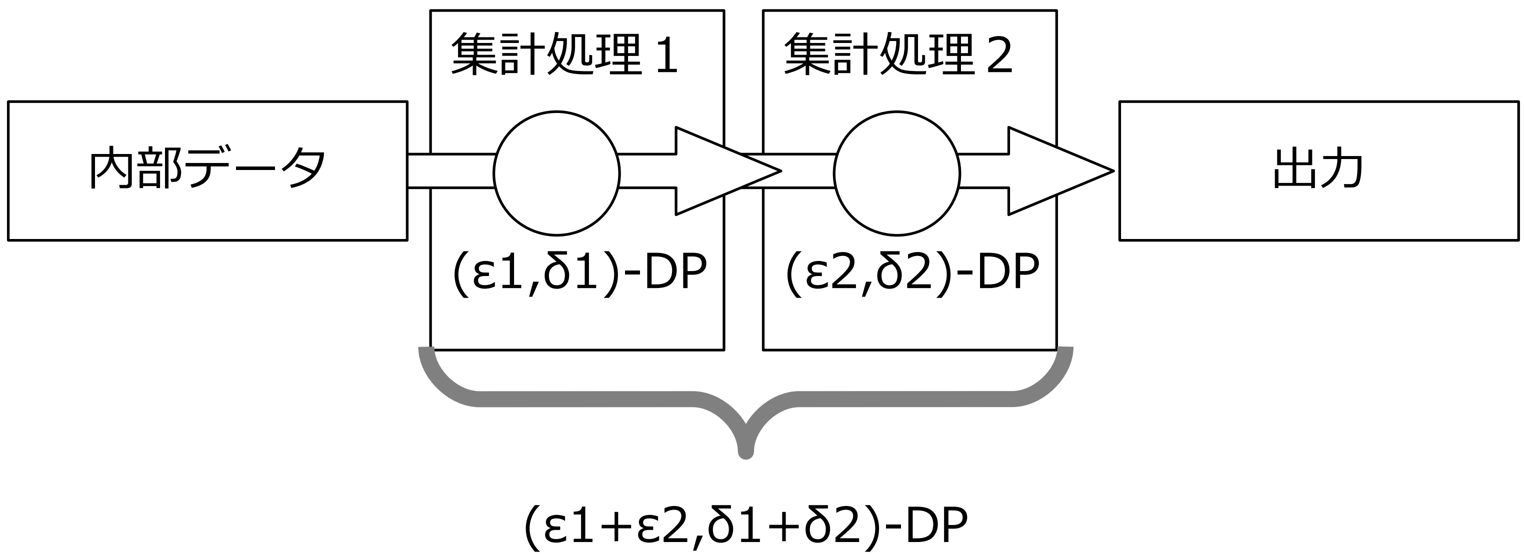
仮説検定の棄却域を検定統計量で引き戻した逆像。

どんな検定手法で内部データの弁別を試みても、誤りを一定以下にできない。



差分プライバシーの合成性

- データベースの差分プライバシーは、
  処理ブロックごとに分割して評価することができる。



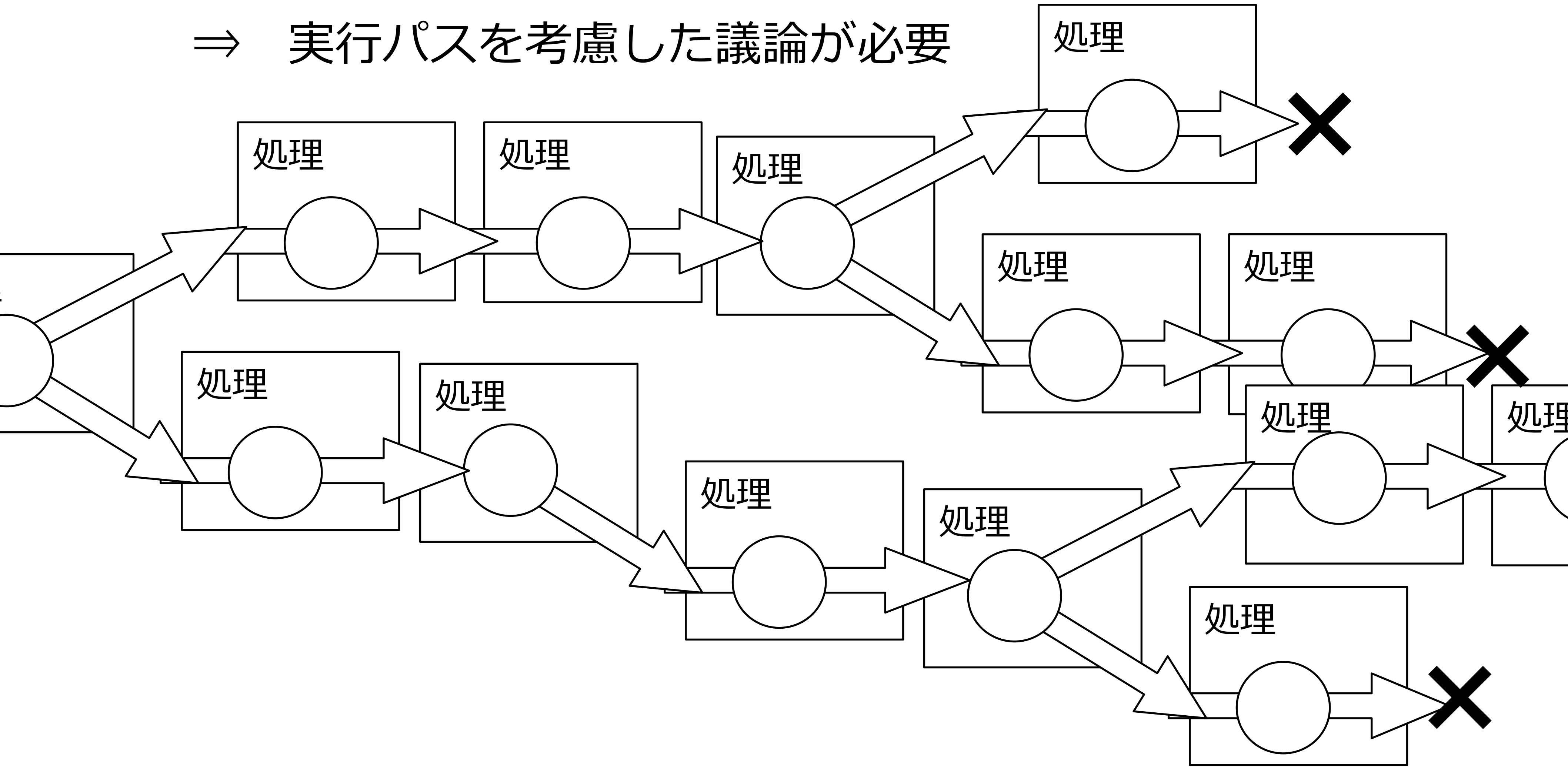
  - 特に、決まった回数の繰り返しの差分プライバシーは、
    繰り返し回数とループ本体の差分プライバシーで求まる。

合成性に基づく形式的検証が厄介な例

- 差分プライバシーを満たす勾配降下法
  [Lee & Kifer, KDD 2018]
  - 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。

なぜ厄介なのか

- 分岐結果で付加するノイズ量やステップ数が<u>動的に変化</u>
  ＋付加するノイズ量が多いとストッパーが働く
  ⇒　実行パスを考慮した議論が必要



着想

- 先ほどのような例の差分プライバシーを検証しようとすると、合成性を通した(保守的な)形式的検証はやりづらい。
  - 実行パスを考慮した議論が上手く扱えるかがネック。

- 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、それらについても厳密な検証もできるのではないか。

- ラドン＝ニコディムの定理の形式化が様々な定理証明支援系で実装済。
  - <u>過去の差分プライバシーの研究で、いかにこの定理に頼ってきたか。</u>
  - <u>連続的な確率分布に対応した形で差分プライバシーを形式化できそう。</u>

差分プライバシーを直接形式化しよう！

Isabelle/HOLによるDPの形式化

- **Isabelle/HOL**で差分プライバシーの形式化を進める。
  - 測度論ライブラリがそろっている。
  - 自動証明ツールsledgehammerが便利。

- 現在の進捗：差分プライバシーの議論のための部品の形式化を進行中：
  - <u>差分プライバシーに対応する統計的ダイバージェンス</u>
    - 定義・反射性・合成性など
  - <u>ラプラスメカニズム</u>
    - ラプラス分布 (正規分布は標準ライブラリにある)
  - <u>有限リストの可測空間(データセットの空間)</u>
    - 位相的圏構造、直和可測空間、リスト操作の可測性

DPの統計的ダイバージェンスによる定式化

[Barthe & Olmedo, ICALP 2013]

- M: X → Prob(Y) が(ε,δ)-DPを満たす

⇔ 隣接する内部データD1～D2について

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

⇔隣接する内部データD1～D2について

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

パラメータεのついたGiryモナド上のダイバージェンス
として解釈することができる(合成性・反射性)。

合成性の証明スケッチ

$$
\begin{aligned}
&\Pr[\mu \gg\!= f \in S] - \exp(\varepsilon_1 + \varepsilon_2)\Pr[\nu \gg\!= g \in S] \\
&= \int f(-)(S) d\mu - \exp(\varepsilon) \int g(-)(S) d\nu \\
&= \int f(-)(S) \cdot \frac{d\mu}{d\pi}\ d\pi - \exp(\varepsilon_1 + \varepsilon_2) \int g(-)(S) \cdot \frac{d\nu}{d\pi}\ d\pi \\
&= \int f(-)(S) \cdot \frac{d\mu}{d\pi} - \exp(\varepsilon_1 + \varepsilon_2) g(-)(S) \cdot \frac{d\nu}{d\pi}\ d\pi \\
&\le \int (\max(0, f(-)(S) - \delta_2) + \delta_2) \cdot \frac{d\mu}{d\pi} - \exp(\varepsilon_1)\min(1, \exp(\varepsilon_2) g(-)(S)) \cdot \frac{d\nu}{d\pi}\ d\pi \\
&= \int \max(0, f(-)(S) - \delta_2) \cdot \frac{d\mu}{d\pi} - \exp(\varepsilon_1)\min(1, \exp(\varepsilon_2) g(-)(S)) \cdot \frac{d\nu}{d\pi}\ d\pi + \int \delta_2 \frac{d\mu}{d\pi}\ d\pi \\
&\le \int_B \left(\frac{d\mu}{d\pi} - \exp(\varepsilon_1) \frac{d\nu}{d\pi} \right) \cdot \min(1, \exp(\varepsilon_2) \cdot g(-)(S))\ d\pi + \int \delta_2 \frac{d\mu}{d\pi}\ d\pi \\
&\le \delta_1 + \delta_2 \\
&= \Delta^{\varepsilon_1}(\mu || \nu) + \sup_x \Delta^{\varepsilon_2}(f(x) || g(x))
\end{aligned}
$$

合成性の証明スケッチ

$$
\begin{aligned}
&\Pr[\mu \gg= f \in S] - \exp(\varepsilon_1 + \varepsilon_2)\Pr[\nu \gg= g \in S] \\
&= \int f(-)(S) d\mu - \exp(\varepsilon) \int g(-)(S) d\nu \\
&= \int f(-)(S) \cdot \frac{d\mu}{d\pi}\ d\pi - \exp(\varepsilon_1 + \varepsilon_2) \int g(-)(S) \cdot \frac{d\nu}{d\pi}\ d\pi \\
&= \int f(-)(S) \cdot \frac{d\mu}{d\pi} - \exp(\varepsilon_1 + \varepsilon_2) g(-)(S) \cdot \frac{d\nu}{d\pi}\ d\pi \\
&\le \int (\max(0, f(-)(S) - \delta_2) + \delta_2) \cdot \frac{d\mu}{d\pi} - \exp(\varepsilon_1)\min(1, \exp(\varepsilon_2) g(-)(S)) \cdot \frac{d\nu}{d\pi}\ d\pi \\
&= \int \max(0, f(-)(S) - \delta_2) \cdot \frac{d\mu}{d\pi} - \exp(\varepsilon_1)\min(1, \exp(\varepsilon_2) g(-)(S)) \cdot \frac{d\nu}{d\pi}\ d\pi + \int \delta_2 \frac{d\mu}{d\pi}\ d\pi \\
&\le \int_B \left(\frac{d\mu}{d\pi} - \exp(\varepsilon_1) \frac{d\nu}{d\pi} \right) \cdot \min(1, \exp(\varepsilon_2) \cdot g(-)(S))\ d\pi + \int \delta_2 \frac{d\mu}{d\pi}\ d\pi \\
&\le \delta_1 + \delta_2 \\
&= \Delta^{\varepsilon_1}(\mu || \nu) + \sup_x \Delta^{\varepsilon_2}(f(x) || g(x))
\end{aligned}
$$

Giryモナドのbindを展開

共通の測度 π に関する
Radon-Nikodym 微分

積分の移項や変形を
たくさん行う。

Bは前半の(...)が正になる
領域

２つの確率測度を比較する補題群(966行)

2つの確率測度をその和で微分したRadon-Nikodym微分の諸性質をまとめたlocaleを作る。

```
locale comparable_probability_measures =
  fixes L M N :: "'a measure"
  assumes M: "M ∈ space (prob_algebra L)" and N: "N ∈ space (prob_algebra L)"
begin
```

$\mu_M, \mu_N \in \mathrm{Prob}(L, \Sigma_L)$

(中略、細かい補題たち。「MとNがL上の測度」とか)

```
definition "dM = real_RN_deriv (sum_measure M N) M"

definition "dN = real_RN_deriv (sum_measure M N) N"
```

$$dM = \frac{d\mu_M}{d(\mu_M + \mu_N)}$$

$$dN = \frac{d\mu_N}{d(\mu_M + \mu_N)}$$

有限測度の、実数値のRadon-Nikodym微分をとる操作。
sigma_finite_measure.real_RN_deriv をもとにSOMEを使って構成。

(中略、Radon-Nikodym微分 dM, dNの非負性、可積分性、bindによる変形)

```
lemma dM_dN_partition_1_AE:
  shows "AE x in (sum_measure M N). (dM x + dN x) = 1"
proof- [26 lines]
qed

end (* end of locale *)
```

１の分割をなす。

DP(のダイバージェンス)の形式化(597行)

計算上の都合で、ダイバージェンスの値の型を
ennreal=[0,∞]ではなくereal = [-∞,∞]でとる。

(定義)

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

(非負性)

```
lemma DP_divergence_nonnegativity:
  assumes M: "M ∈ space (prob_algebra L)" and N: "N ∈ space (prob_algebra L)"
  shows "0 ≤ DP_divergence M N ε "
```

(Giryモナド上のダイバージェンス[Sato&Katsumata,2023]の公理：単調性・反射性・合成性)

```
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_reflexivity:
  shows " DP_divergence M M 0 = 0"
```

先ほどのlocaleを使う

```
theorem (in comparable_probability_measures) DP_composability:
  assumes f: "f ∈ measurable L (prob_algebra K)"
    and g: "g ∈ measurable L (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" "0≤ε2"
  shows "DP_divergence (bind M f) (bind N g) (ε1 + ε2) ≤ δ1 + δ2"
```

合成性の形式的証明(主要部分外観)

- ほぼスケッチ通りの内容。

```
have "(measure (M ≫= f) A) - exp (ε1 + ε2) * (measure (N ≫= g) A) [3 lines]
also have "... = (∫ x. (dM x) * (measure (f x) A) ∂(sum_measure M N)) - (∫ x. (exp (ε1 + ε2)) * (dN x) * (measure (g x) A) ∂(sum_measure M N)) " [1 lines]
also have "... = (∫ x. (dM x) * (measure (f x) A) - (exp (ε1 + ε2)) * (dN x) * (measure (g x) A) ∂(sum_measure M N)) " [1 lines]
also have "... = (∫ x. (dM x) * (measure (f x) A) - (exp ε1) * (exp ε2) * (dN x) * (measure (g x) A) ∂(sum_measure M N)) " [1 lines]
also have "... ≤ (∫ x. (dM x) * (max 0 (measure (f x) A - δ2) + δ2) - (exp ε1) * (dN x) * min 1 ((exp ε2) * (measure (g x) A)) ∂(sum_measure M N)) "  [15 lines]
also have "... = (∫ x. (dM x) * (max 0 (measure (f x) A - δ2)) - (exp ε1) * (dN x) * min 1 ((exp ε2)* (measure (g x) A)) + (dM x) *δ2 ∂(sum_measure M N)) " [1 lines]
also have "... ≤ (∫ x. (dM x) * (min 1 ((exp ε2)* (measure (g x) A))) - (exp ε1) * (dN x) * min 1 ((exp ε2)* (measure (g x) A)) + dM x *δ2 ∂(sum_measure M
N)) "  [12 lines]
also have "... =(∫ x. ((dM x) - (exp ε1) * (dN x)) * min 1 ((exp ε2) * (measure (g x) A)) + dM x *δ2 ∂(sum_measure M N)) " [1 lines]
also have "... =(∫ x. ((dM x) - (exp ε1) * (dN x)) * min 1 ((exp ε2) * (measure (g x) A))∂(sum_measure M N)) +(∫ x. dM x *δ2 ∂(sum_measure M N)) " [1 lines]
finally have *: "(measure (M ≫= f) A) - exp (ε1 + ε2) * (measure (N ≫= g) A)≤(∫ x. ((dM x) - (exp ε1) * (dN x)) * min 1 ((exp ε2)* (measure (g x) A))∂
(sum_measure M N)) +(∫ x. dM x *δ2 ∂(sum_measure M N))".

have "(∫ x. dM x *δ2 ∂(sum_measure M N))=(∫ x. δ2 ∂(density (sum_measure M N) dM))" [1 lines]
also have "... =(∫ x. δ2 ∂(density (sum_measure M N)(ennreal o dM)))" [1 lines]
also have "... =(∫ x. δ2 ∂M)" [1 lines]
also have " ... = δ2 * measure M (space M)" [1 lines]
also have " ... ≤ δ2" [1 lines]
finally have **: "(∫ x. dM x *δ2 ∂(sum_measure M N)) ≤ δ2".

let ?B = "{x∈ space (sum_measure M N). 0 ≤ ((dM x) - (exp ε1) * (dN x)) }"

have mble10: " ?B ∈ sets (sum_measure M N)" [1 lines]

have "(∫ x. ((dM x) - (exp ε1) * (dN x)) * min 1 ((exp ε2)* (measure (g x) A))∂(sum_measure M N)) ≤ (∫ x∈ ?B. ( ((dM x) - (exp ε1) * (dN x)) * min 1 ((exp
ε2)* (measure (g x) A)) )∂(sum_measure M N))"
proof(rule integral_drop_negative_part2) [17 lines]
qed
also have "... ≤ (∫ x∈ ?B. ((dM x) - (exp ε1) * (dN x)) ∂(sum_measure M N))" [11 lines]
also have "... = (∫ x∈ ?B. (dM x) ∂(sum_measure M N)) - (∫ x∈ ?B. ((exp ε1) * (dN x)) ∂(sum_measure M N))" [8 lines]
also have "... = (∫ x∈ ?B. (dM x) ∂(sum_measure M N)) - (exp ε1) * (∫ x∈ ?B. (dN x) ∂(sum_measure M N))" [1 lines]
also have "... = measure M ?B - (exp ε1) * (measure N ?B)" [42 lines]
also have "... ≤ δ1" [1 lines]
finally have ***:"(∫ x. ((dM x) - (exp ε1) * (dN x)) * min 1 ((exp ε2)* (measure (g x) A))∂(sum_measure M N)) ≤ δ1".

show "measure (M ≫= f) A - exp (ε1 + ε2) * measure (N ≫= g) A ≤ δ1 + δ2"
  using * ** *** by auto
qed
```

  - 積分の移項が多いので非負積分でなく実関数の積分を採用。
    (可積分性の証明の手間より、移項が楽になるほうが大きい)

ラプラス分布・ラプラスメカニズムの形式化

- 標準ライブラリ(HOL/Probability/Distributions)に正規分布の形式化があった(指数分布もあった、ラプラス分布はなかった)のでそれに倣った。
  - 密度関数
  - 累積分布関数
    - 同時に形式化する。密度関数と累積分布関数を用意して、微分積分学の基本定理(と広義積分)を適用する。
それから、合計が１になることを証明する。
  - ｎ次モーメント
    - 一般形を先に与え、帰納法で証明する。
n+1のケースを部分積分法を使ってnのケースに帰着。
- ラプラスメカニズムの構成と差分プライバシーの形式的証明は比較的安直にやれた。

ラプラス分布の形式化(864行)

(密度関数と累積分布関数)

```
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)"
```

(中略、可測性とか非負性とか)

中心より右側の累積分布
(左側も同様)

```
lemma nn_integral_laplace_density_pos:
  assumes pos[arith]: "0 < l"
    and 1: "a ≥ m"
  shows "(∫+ x ∈ {a..}. ennreal (laplace_density l m x)  ∂lborel) = 1 - laplace_CDF l m a"
proof-
  from 1 have "(∫+ x ∈ {a..}. ennreal (laplace_density l m x)  ∂lborel) = (∫+ x ∈ {a..}. (exp ((m - x) / l)/  (2 * l))  ∂lborel)"
  also have "... = 0 - (- exp ((m - a) / l) / 2)"
  proof(rule nn_integral_FTC_atLeast) [49 lines]
  qed(auto)
  also have " ... = ennreal (exp ((m - a) / l) / 2) "  [1 lines]
  also have " ... = ennreal (1 - laplace_CDF l m a) " [1 lines]
  finally show Q: " ∫+x∈{a..}. ennreal (laplace_density l m x)∂lborel = ennreal (1 - laplace_CDF l m a)".
qed
```

[a,∞)型の広義積分の計算

```
lemma nn_integral_laplace_density_neg:
  assumes pos[arith]: "0 < l"
    and 1: "a ≤ m"
  shows "(∫+ x ∈ {..a}. ennreal (laplace_density l m x)  ∂lborel) = laplace_CDF l m a"
proof- [54 lines]
qed
```

- 密度関数と累積分布関数に入っているif式を払うために、中心より右側と左側に分割して証明した。

ラプラス分布の形式化

(n次のモーメント)

n次モーメントの中心より右側部分の計算

```
lemma laplace_moment_0:
  fixes k::nat
  assumes pos[arith]: "0 < l"
  shows "has_bochner_integral lborel (λ x. (indicator {0..} x *R ((laplace_density l 0 x) * x^k )))(fact k  * l^k/2)"
    and "(λa. LBINT x. indicator {0..a} x *R ((laplace_density l 0 x) * x^k))
    ⟶ (LBINT y. (indicator {0..} y *R ((laplace_density l 0 y) * y^k)))"
proof(induction k) [366 lines]
qed
```

主に $\displaystyle \int_0^\infty x^{k+1} e^{-\frac{x}{l}}\ dx = \underbrace{[x^{k+1} \cdot (-l) e^{-\frac{x}{l}}]_0^\infty}_{=0} + l(k+1) \int_0^\infty x^k e^{-\frac{x}{l}}\ dx$

```
lemma laplace_moment_even:
  fixes k::nat
  assumes pos[arith]: "0 < l"
  shows "has_bochner_integral lborel (λ x. ((laplace_density l m x) * (x - m)^(2 * k) ))(fact (2 * k)  * l^(2 * k))"
proof- [11 lines]
qed

lemma laplace_moment_odd:
  fixes k::nat
  assumes pos[arith]: "0 < l"
  shows "has_bochner_integral lborel (λ x. ((laplace_density l m x) * (x - m)^(2 * k + 1) ))( 0 )"
proof- [11 lines]
qed

lemma laplace_moment_abs_odd:
  fixes k::nat
  assumes pos[arith]: "0 < l"
  shows "has_bochner_integral lborel (λ x. ((laplace_density l m x) * ¦x - m¦^(2 * k + 1) ))( fact (2 * k + 1) * l ^ (2 * k + 1) )"
proof- [13 lines]
qed
```

対称性からn次モーメントの計算をつくる。

ラプラスメカニズムの形式化(196行)

(定義・基本的構造・可測性)

```
definition Lap_mechanism ::"real ⇒ real ⇒ real measure"
  where "Lap_mechanism ε x = (density lborel (laplace_density (1/ε) x))"

lemma
  shows prob_space_Lap_mechanism: "ε > 0 ⟹ prob_space (Lap_mechanism ε x)"
    and sets_Lap_mechamism: "sets (Lap_mechanism ε x) = sets lborel"
    and space_Lap_mechamism: "space (Lap_mechanism ε x) = UNIV"
proof- [6 lines]
qed

lemma meausrable_Lap_mechanism[measurable]:
  assumes "ε > 0"
  shows "Lap_mechanism ε ∈ measurable lborel (prob_algebra lborel)"
proof(rule measurable_prob_algebraI) [31 lines]
qed
```

(差分プライバシー)

```
proposition DP_Lap_mechanism:
  fixes x y ε ::real
  assumes "ε > 0"
    and "¦ x - y ¦ ≤ 1"
  shows "DP_divergence (Lap_mechanism ε x) (Lap_mechanism ε y)  ε ≤ (0::real)"
proof(subst DP_divergence_forall[THEN sym],unfold Lap_mechanism_def,safe) [59 lines]
qed
```

$$
\begin{aligned}
&|x-y| \leq 1 \\
&\quad\Longrightarrow \forall z \in \mathbb{R}.\, \Pr[\mathsf{Lap}_\varepsilon(x) = z] \leq \exp(\varepsilon)\Pr[\mathsf{Lap}_\varepsilon(y) = z] \\
&\quad\Longrightarrow \forall S \in \Sigma_{\mathbb{R}}.\, \Pr[\mathsf{Lap}_\varepsilon(x) \in S] \leq \exp(\varepsilon)\Pr[\mathsf{Lap}_\varepsilon(y) \in S] \\
&\quad\Longrightarrow \Delta^\varepsilon(\mathsf{Lap}_\varepsilon(x) || \mathsf{Lap}_\varepsilon(y)) = 0
\end{aligned}
$$

リスト可測空間の形式化(506+717+961行)

- 構成はリスト準ボレル空間[Hirata+, ITP2023] を参考にした：

  a)可算直積空間 $\prod_{i \in I} A_i$ の形式化(標準ライブラリにある)

  b)可算直和空間 $\coprod_{i \in I} A_i$ の形式化(新しく作る)

  c) 直積と可算直和の分配律 $A \times \coprod_{n=0}^{\infty} B_n \cong \coprod_{n=0}^{\infty} A \times B_n$ (新しく作る)

  d)同型写像 $\varphi\colon \mathrm{lists}(A) \cong \coprod_{n=0}^{\infty} \prod_{k=0}^{n} A$

  で引き戻して $\mathrm{lists}(A)$ のσ代数を構成する(同型射にもなる)。

- そのうえでリスト処理の可測性を形式的に証明する(進行中)：
  - 形式化済み：Cons, append, fold, foldr, foldl, rev, concat

ラムダ抽象や関数適用が一般に可測とは限らない。rec_listの可測性はうまく使えない。uncurryが随所に必要でかなりの手間が生じる。

「source」の形式化と直積

- 可測空間 (X_i, Σ_{X_i}) と関数 $f_i\colon A \to X_i$ の族を考える $(i \in I)$ 。
  すべての $f_i\colon A \to X_i$ を可測ならしめる**最もσ粗い代数**

$$\Sigma_A = \sigma(\{f_i^{-1}(D) \mid D \in \Sigma_{X_i}, i \in I\})$$

  を備えた可測空間 (A, Σ_A) は、**source**と呼ばれる。
- 直積空間は射影 $\pi_i\colon \prod_{i\in I} X_i \to X_i$ が可測になるよう
  引き戻したsourceに他ならない。
  これに限らず、Set上の任意の極限を可測空間化できる。

$f_i\colon A \to X_i$の型は
制限されている。
（依存型がないため）

- Isablleにおける、sourceの定義と直積の定義

```
definition⌂‹tag important› source_algebra :: "'a set ⇒ (('a ⇒ 'b) × 'b measure) set ⇒ 'a measure" where
  "source_algebra X F = sigma X {f -` A ∩ X | A f M. (f,M) ∈ F ∧ A ∈ sets M }"

definition prod_source_algebra⌂‹tag important›::"'i set ⇒ ('i ⇒ 'a measure) ⇒ ('i ⇒ 'a) measure"
  where "prod_source_algebra I M = source_algebra (ΠE i∈I. space (M i)) ({((λ f. f i),M i) |i. i ∈ I}) "

proposition⌂‹tag important› PiM_is_source:
  shows "space (PiM I M) = space (prod_source_algebra I M)"
    and "sets (PiM I M) = sets (prod_source_algebra I M)"
```

標準ライブラリの直積
空間と可測空間は一致。

「sink」の形式化

- 可測空間 (X_i, Σ_{X_i}) と関数 $g_i\colon X_i \to A$ の族を考える $(i \in I)$ 。
  すべての $g_i\colon X_i \to A$ を可測ならしめる**最も細かいσ代数**
  $$\Sigma_A = \{D \subseteq A \mid g_i^{-1}(D) \in \Sigma_{X_i}, i \in I\}$$
  を備えた可測空間 (A, Σ_A) は**sink**と呼ばれる。
- 形式化

$g_i\colon X_i \to A$ の型は制限されている。
（依存型がないため）

```
definition‹tag important› sink_algebra :: "'b set ⇒ (('a ⇒ 'b) × 'a measure) set ⇒ 'b measure" where
  "sink_algebra Y G = sigma Y { A | A. A ∈ Pow Y ∧ (∀ (g,N) ∈ G. (g -` A ∩ (space N)  ∈ sets N)) }"

lemma measurable_sink_algebra1:
  "∀(g,N) ∈ G . g ∈ space N → Y ⟹ (g,N) ∈ G ⟹  g ∈ measurable N (sink_algebra Y G)"
  unfolding measurable_def by (auto intro: in_sink_algebra)

lemma measurable_sink_algebra2:
  assumes "f ∈ Y → space M"
    and "∀(g,N) ∈ G . g ∈ space N → Y"
    and "∀(g,N) ∈ G . (λx. f (g x)) ∈ measurable N M "
  shows "f ∈ measurable (sink_algebra Y G) M"
  unfolding sink_algebra_def
proof (rule measurableI) [28 lines]
qed
```

圏論的なsinkの定義を満足する：
(1) $g_i\colon X_i \to A$ は可測
(2) $f \circ g_i\colon X_i \to Y$ が可測なら、
$f\colon A \to Y$ は可測
※「最も細かい」ことも導ける。

直和可測空間の構成(245行)

- 直和集合の余射影(埋め込み、coprojection) $\iota_i\colon X_i \to \coprod_{i\in I} X_i$ で各(X_i,Σ_{X_i})を押し出したsinkは、直和空間に他ならない。

  これに限らず、Set上の任意の余極限を可測空間化できる。

- 形式化(直和<u>集合</u>の構成は割愛)

```
definition‹tag important› coprod_sink_algebra‹tag important›::"'i set ⇒ ('i ⇒ 'a measure) ⇒ ('i × 'a) measure"
  where "coprod_sink_algebra I M = sink_algebra (∐ i∈I. space (M i)) ({(coProj i,M i) |i. i ∈ I}) "

syntax
  "_coprod_sink_algebra" :: "pttrn ⇒'i set ⇒ ('i ⇒ 'a measure) ⇒ ('i × 'a) measure"  ("(3∐M _∈_./ _)"  10)
translations
  "∐M i∈I. M" ⇌ "CONST coprod_sink_algebra I (λi. M)"

lemma coProj_measurable[measurable]:
  assumes "i ∈ I"
  shows "coProj i  ∈ (M i)→M (∐M i∈I. M i)"
  unfolding coprod_sink_algebra_def
proof(rule measurable_sink_algebra1) [4 lines]
qed

lemma coPair_measurable[measurable]:
  assumes "∀ i ∈ I. F i ∈ measurable (M i) N"
  shows "coPair I (λ i. space (M i)) F  ∈ (∐M i∈I. M i
  unfolding coprod_sink_algebra_def
proof(rule measurable_sink_algebra2) [22 lines]
qed
```

$\iota_i\colon X_i \to \coprod_{i\in I} X_i$ の可測性

$[f_i]_{i\in I}\colon \coprod_{i\in I} X_i \to Y$ の可測性
$(f_i\colon X_i \to Y)$

（二項）直積と可算直和の分配律(266行)

- 互いに可逆な可測関数を構成する(定義は省く)。

```
proposition dist_law_A_measurable:
  shows "dist_law_A I N M ∈  (∐M i∈I. (M ⨂M N i)) →M ( M ⨂M (∐M i∈I. N i))"
  unfolding dist_law_A_def space_pair_measure[THEN sym]
proof(intro coPair_measurable, subst Ball_def,intro allI impI) [8 lines]
qed

lemma sets_generator_product_of_M_and_coprodNi: [58 lines]

lemma sets_generator_coproduct_of_prod_M_Ni: [90 lines]

proposition dist_law_B_measurable:
  assumes I: "countable I"
  shows "dist_law_B I N M ∈ ( M ⨂M (∐M i∈I. N i)) →M (∐M i∈I. (M ⨂M N i))"
proof(rule measurableI) [43 lines]
qed

lemma dist_laws_mutually_inverse:
  shows "⋀x. (dist_law_A I N M o dist_law_B I N M) x = x"
    and  "⋀y. (dist_law_B I N M o dist_law_A I N M) y = y"
  by(auto simp add:dist_law_A_def2  dist_law_B_def)
```

$A \times \coprod_{n=0}^{\infty} B_n$ と $\coprod_{n=0}^{\infty} A \times B_n$
のσ代数の生成元を突き合わせる。

有限リストの可測空間の形式化(961行)

- 同型写像 $\varphi\colon \mathrm{lists}(A) \cong \coprod_{n=0}^{\infty} \prod_{k=0}^{n} A$から作る(source/vimage)

```
fun pair_to_list ::"(nat × (nat ⇒ 'a)) ⇒ 'a list" where
  Zero: " pair_to_list (0,_) = [] "
| Suc:" pair_to_list (Suc n,f) = (f 0) # pair_to_list (n, λ n. f (Suc n)) "

fun list_to_pair ::" 'a list ⇒ (nat × (nat ⇒ 'a)) " where
  Nil:" list_to_pair [] = (0,λ_.undefined) "
| Cons:" list_to_pair (x#xs) = (Suc (fst(list_to_pair xs)), λn. if n = 0 then x else (snd(list_to_pair
xs))(n - 1))"

definition‹tag important› listM::"'a measure ⇒ 'a list measure"
  where "listM M = source_algebra (lists (space M)) {( list_to_pair, ∐M n∈UNIV. ∏S i∈{..<n}. M )}"
```

  - 上記の2つのfunが実際に同型写像であることを示し、
  - sourceであることからそれらの可測性を証明する。

- Cons(のuncurrying)が可測写像であること(補題含め合計150行ほど)

```
lemma
  measurable_Cons[measurable]: " (λ (x,xs). x # xs) ∈ M ⨂M (listM M) →M (listM M)"
proof- [25 lines]
qed
```

  - 可測性形式化済み：map, append, fold, foldr, foldl, rev, concat,

Future Work

- まずは簡単な差分プライバシーの検証例を構成する

```
primrec LapMech:: "real ⇒ (real list) ⇒ real list measure" where
"LapMech ε [] = (return (listM lborel) [])"
| "LapMech ε (x#xs) = ( (Lap_mechanism ε x) ⨂M  (LapMech ε xs)
   ⤜ (λp. return (listM lborel) (Cons (fst p) (snd p))))"

lemma
  assumes "(ε::real) > 0"
  shows  "list_mechanism.DP lborel (listM lborel)  (LapMech ε) ε 0"
  unfolding list_mechanism.DP_def
proof(intro conjI ballI impI)
  sorry

  proof (state)
  goal (2 subgoals):
   1. LapMech ε ∈ listM lborel →M prob_algebra (listM lborel)
   2. ⋀(x::real list) y::real list.
         x ∈ space (listM lborel) ⟹
         y ∈ space (listM lborel) ⟹
         Hamming_distance x y ≤ (1::real) ∧ length x = length y ⟹
         DP_divergence (LapMech ε x) (LapMech ε y) ε ≤ (0::ereal)
```

実数のリストの全要素に
ラプラスメカニズムを適用

入力リストの隣接関係を定義し、
差分プライバシーを示す。