脱乱択化の理論とメタ計算量

平原 秀一

情報学プリンシプル研究系・准教授

@組合せ最適化セミナー

２０２３年8月9日

概要

1. 脱乱択化の理論
   - 理論の概要
   - 疑似乱数生成器の概念
   - 疑似乱数生成器の構成

2. メタ計算量と平均時計算量について
   - メタ計算問題
   - 平均時計算量
   - 最悪時・平均時計算量の同値性
   - 暗号との関係

乱択アルゴリズム (Randomized algorithm)

例：モンテカルロ法、クイックソート、多項式同一性判定、…

- 乱択アルゴリズムは非常に有用である。しかし…

- どのようにして乱択アルゴリズムを実装すべきだろうか？
  - srand(time(NULL)); rand(); rand(); …　← 理論的な保証なし
  - 時間、マウスの動き、ノイズ、放射線などを使う
    ← デバイス依存＆一様分布とは限らない

- 計算量理論における二つのアプローチ
  1. Randomness extractor
  2. Derandomization （脱乱択化）

二つのアプローチ

1. Randomness extractor
   - ほぼ一様ランダムな乱数列を 「十分ランダムな」乱数列から抽出する。
     （「十分ランダムな」: 最小エントロピーが大きい）
   - 例：マウスの動きから一様ランダムな乱数列を抽出できる。

2. 脱乱択化（Derandomization）
   - 効率的な乱択アルゴリズムが用いる乱数の量を（理想的には）$O(\log n)$ビットまで減らす技術。
   - $O(\log n)$長の乱数列は、全探索により多項式時間で模倣できる。
   - 「**P = BPP予想**」の根拠となった理論。

P: Polynomial-time

- $f: \{0,1\}^* \to \{0,1\}$ ： 判定問題
- （決定性）多項式時間アルゴリズムAがfを解くとは、

全ての$n \in \mathbb{N}$、全ての入力$x \in \{0,1\}^n$について、

$$A(x) = f(x) \qquad x\text{: 入力}$$

が成立することをいう。

$\mathrm{P} :=$ {判定問題f | ある多項式時間アルゴリズムAがfを解く }

BPP: **B**ounded-error **P**robabilistic **P**olynomial-time

- $f: \{0,1\}^* \to \{0,1\}$ ： 判定問題
- **乱択**多項式時間アルゴリズムAがfを解くとは、

ある多項式pが存在し、全ての$n \in \mathbb{N}$、全ての入力$x \in \{0,1\}^n$について、

$$\Pr_{r \sim \{0,1\}^{p(n)}}[A(x;r) = f(x)] \geq \frac{3}{4}$$

x: 入力
r: 一様ランダムな文字列
（真の乱数列）

が成立することをいう。

BPP ≔ { 判定問題f | ある乱択多項式時間アルゴリズムAがfを解く }

Hardness versus Randomness framework

計算困難性　　乱択

[Yao'82], [Blum & Micali '84], [Nisan & Wigderson '94], …

- 困難性と乱雑性は両立しない。

Theorem [Impagliazzo & Wigderson 1997]

$$\mathrm{E} \not\subseteq \text{io-SIZE}\left(2^{0.01n}\right) \quad \Rightarrow \quad \mathrm{P} = \mathrm{BPP}.$$

計算困難性（回路下界）　　**脱乱択化**

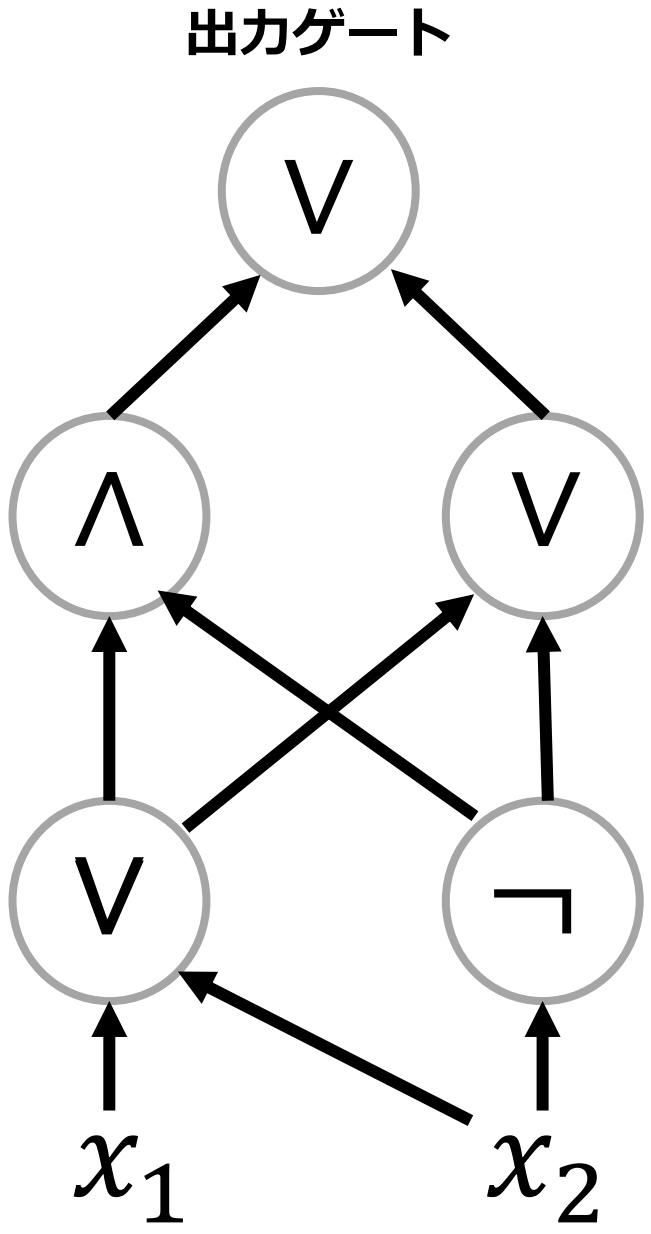
指数時間$2^{O(n)}$で計算可能だが、
$2^{0.01n}$サイズの論理回路では
計算不可能な判定問題が存在するならば

$$\mathrm{E} := \mathrm{DTIME}\left(2^{O(n)}\right).$$

乱択多項式時間アルゴリズムを
決定性多項式時間アルゴリズムに変換できる。

回路の定義

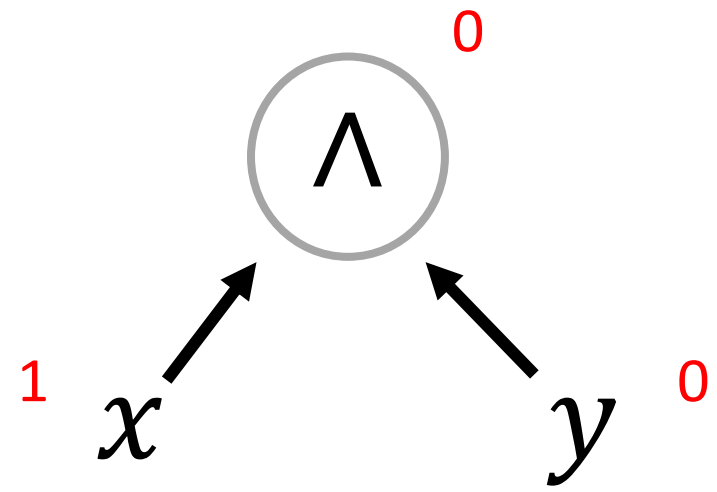
- 回路とは、ラベル付き有向非巡回グラフ（DAG; directed acyclic graph）であって、各頂点のラベルは$\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}$ゲートまたは入力ゲート$x_1, \dots, x_n$か定数$0, 1$のいずれかのラベル付けがされているもの。
  - 出次数0の頂点は**出力ゲート**と呼ばれる。
  - 入次数：$\mathrm{AND}, \mathrm{OR}$ゲートは$2$, NOTゲートは1, 入力ゲートと定数ゲートは0。
  - 回路サイズ $:=$ $(\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}$ゲートの個数$)$
- 最も基礎的かつ重要な計算モデルの一つ。
  - コンピュータを作るときにも用いられる。

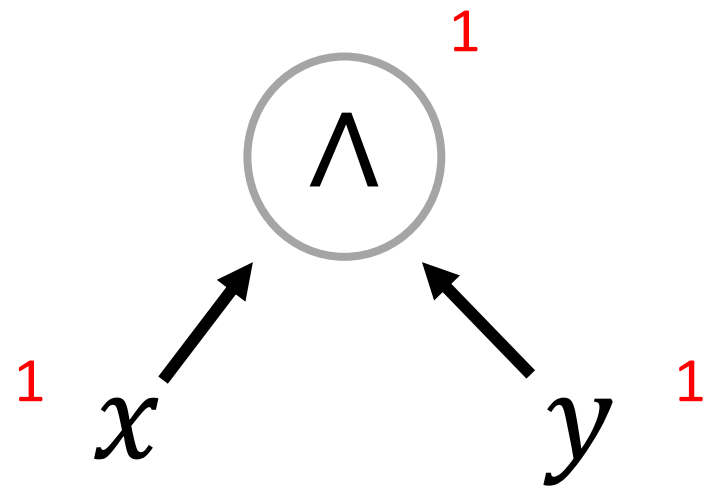


ANDゲート（論理積ゲート）

$$\wedge : \{0,1\}^2 \to \{0,1\}$$

$$x \wedge y = \begin{cases} 1, & x = y = 1 \\ 0, & \text{otherwise} \end{cases}$$

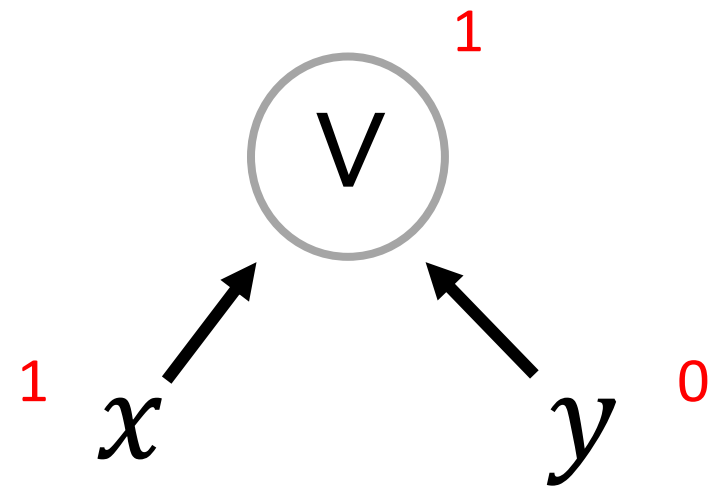


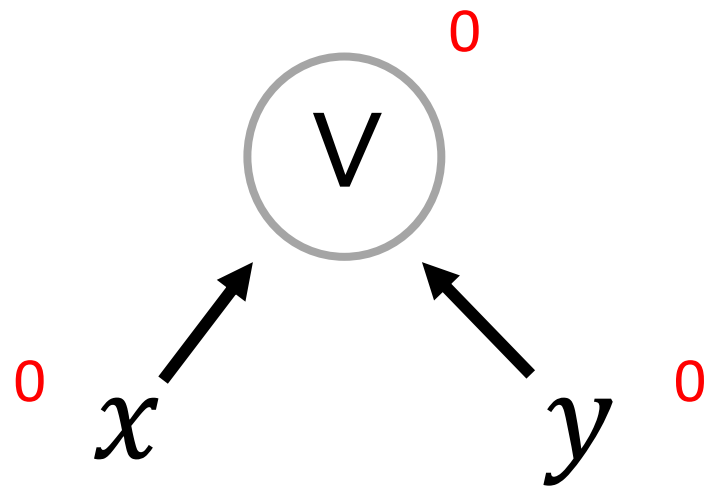


ORゲート（論理和ゲート）

$$\vee : \{0,1\}^2 \to \{0,1\}$$

$$x \vee y = \begin{cases} 1, & x = 1 \text{ or } y = 1 \\ 0, & \text{otherwise} \end{cases}$$

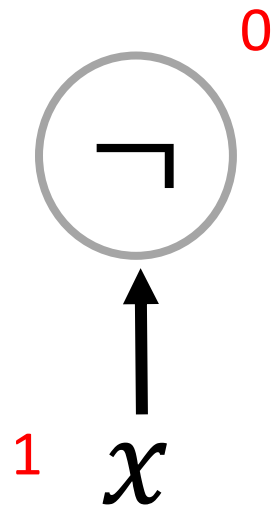


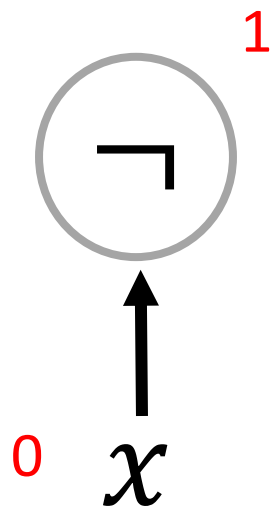


NOTゲート（論理否定ゲート）

$$\neg: \{0,1\}^1 \to \{0,1\}$$

$$\neg x = \begin{cases} 0, & x = 1 \\ 1, & \mathrm{x} = 0 \end{cases}$$



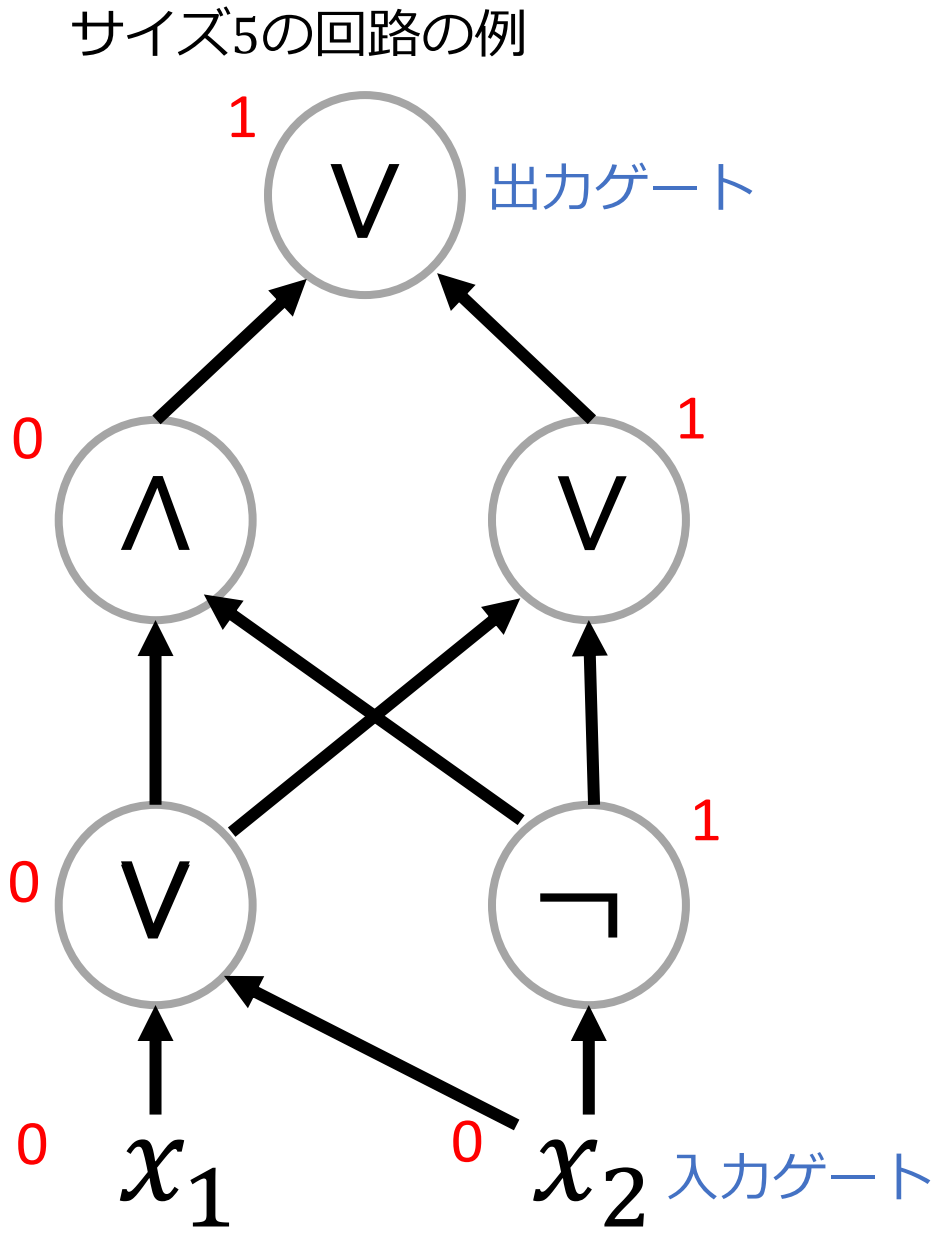


回路の例

- 以下の関数$f:\{0,1\}^2 \to \{0,1\}$を計算する。

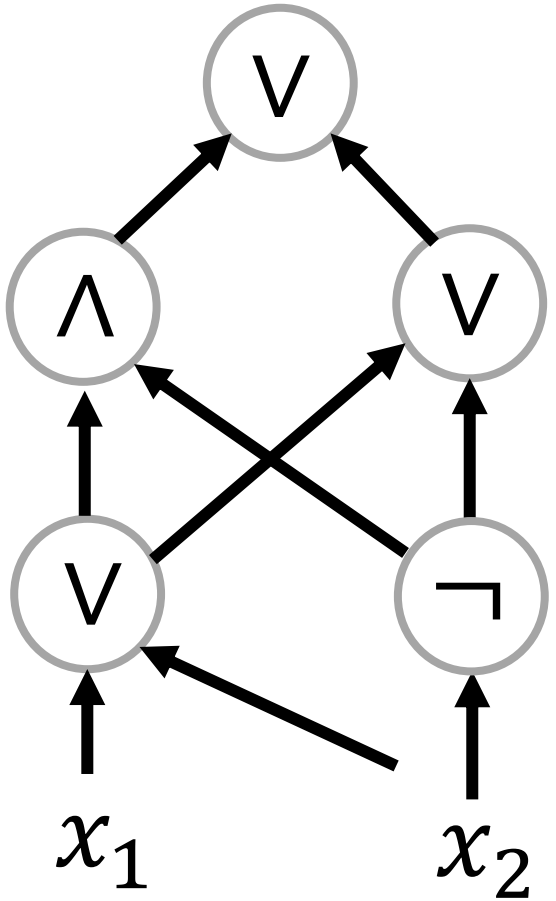
x_1	x_2	$f(x_1,x_2)$
0	0	1
0	1	
1	0	
1	1	

- 実は定数関数を計算する！
  - より小さい回路でも同じ関数が計算できる。



回路：非一様な計算モデル

- （論理）回路：チューリング機械よりも組合せ論的で扱いやすい
  - 回路は有限の関数 $f\colon\{0,1\}^n \to \{0,1\}$ を計算する。



- 判定問題$f\colon\{0,1\}^* \to \{0,1\}$を回路族$\{C_n\}_{n\in\mathbb{N}}$が計算する
  （定義）⇔ 各$n \in \mathbb{N}$について、C_nが$f_n \coloneqq f|_{\{0,1\}^n}$を計算する
  - 各nについて異なる回路を用いてよい（**非一様性**）

- 多項式時間チューリング機械は
  多項式サイズの回路で模倣できる。

$$\mathrm{P} \subseteq \mathrm{SIZE}\left(n^{O(1)}\right)$$

Hardness versus Randomness framework

計算困難性　　乱択

Theorem [Impagliazzo & Wigderson 1997]

$$\mathrm{E} \not\subseteq \text{io-SIZE}\left(2^{0.01n}\right) \quad \Rightarrow \quad \mathrm{P} = \mathrm{BPP}.$$

計算困難性（回路下界）　　**脱乱択化**

仮定： $\mathrm{E} = \mathrm{DTIME}\left(2^{O(n)}\right)$ を$2^{0.01n}$サイズの論理回路で計算できない。

結論： 乱択多項式時間アルゴリズムを脱乱できる。

不可能性 $\Rightarrow$ **可能性**

Hardness versus Randomness framework

計算困難性　　乱択

Theorem [Impagliazzo & Wigderson 1997]

$$\mathrm{E} \not\subseteq \text{io-SIZE}\left(2^{0.01n}\right) \quad \Rightarrow \quad \mathrm{P} = \mathrm{BPP}.$$

計算困難性（回路下界）　　**脱乱択化**

仮定： $\mathbf{E} = \mathbf{DTIME}\left(2^{O(n)}\right)$ を$2^{0.01n}$サイズの論理回路で計算できない。

結論： 効率的なアルゴリズムでは「識別できない」（に対して安全な）
「疑似乱数生成器」を計算できる。

不可能性 ⇒ 可能性

概要

1. 脱乱択化の理論
   - 理論の概要
   - 疑似乱数生成器の概念
   - 疑似乱数生成器の構成

2. メタ計算量と平均時計算量について
   - メタ計算問題
   - 平均時計算量
   - 最悪時・平均時計算量の同値性
   - 暗号との関係

rand()関数はよい疑似乱数列だろうか？

- 乱択アルゴリズム $A(x;r)$ を「実装」しよう。
  - どのように乱数列$r \sim \{0,1\}^*$を模倣すべきだろうか。

- rand()の実装例（線形合同法）

```
int rand () {
    rand_next = rand_next * 1103515245 + 12345;
    return rand_next & 0x7fffffff;
}
```

$$r_0 := \mathrm{srand}(\mathrm{time}(\mathrm{NULL}))$$
$$r_1 := \mathrm{rand}()$$
$$r_2 := \mathrm{rand}()$$
$$r_3 := \mathrm{rand}()$$
$$\ldots$$

rand()関数はよい疑似乱数列だろうか？

- 乱択アルゴリズム $A(x;r)$ を「実装」しよう。
  - どのように乱数列$r \sim \{0,1\}^*$を模倣すべきだろうか。

- rand()の実装例（線形合同法）

```
int rand () {
    rand_next = rand_next * 1103515245 + 12345;
    return rand_next & 0x7fffffff;
}
```

$$
\begin{aligned}
r_0 &:= \text{seed} \\
r_1 &:= (1103515245 \times r_0 + 12345) \bmod 2^{31} \\
r_2 &:= (1103515245 \times r_1 + 12345) \bmod 2^{31} \\
r_3 &:= (1103515245 \times r_2 + 12345) \bmod 2^{31} \\
&\dots
\end{aligned}
$$

$$
\begin{aligned}
r_1 &\in \{0,1\}^{31} \\
r_2 &\in \{0,1\}^{31} \\
r_3 &\in \{0,1\}^{31}
\end{aligned}
$$

$A(x;r)$をrand()関数で模倣する

- $G:\{0,1\}^s \to \{0,1\}^{31m}$を疑似乱数の種$z \in \{0,1\}^s$を受け取って、rand()関数によって生成される列を出力する関数と定める。

$$G(z) \coloneqq r_1 r_2 r_3 \ldots r_m$$

ここで、$r_{i+1} = (a r_i + c) \bmod 2^{31}, r_0 = z, a = 1103515245, c = 12345.$

- 以下の意味で$A(x;r)$を正しく模倣できるだろうか？

$$\forall x, \quad \Pr_{r \sim \{0,1\}^{31m}}[A(x;r) = f(x)] \approx \Pr_{z \sim \{0,1\}^{s}}[A(x;G(z)) = f(x)].$$

  - ここで、$f:\{0,1\}^* \to \{0,1\}$はAによって計算される判定問題とする。

$A(x;r)$をrand()関数で模倣する

- $G:\{0,1\}^s \to \{0,1\}^{31m}$を疑似乱数の種$z \in \{0,1\}^s$を受け取って、rand()関数によって生成される列を出力する関数と定める。

$$G(z) \coloneqq r_1 r_2 r_3 \ldots r_m$$

ここで、$r_{i+1} = (a r_i + c) \bmod 2^{31}, r_0 = z, a = 1103515245, c = 12345.$

- 以下の意味で$A(x;r)$を正しく模倣できるだろうか？

$$\forall x, \quad \Pr_{r \sim \{0,1\}^{31m}}[A(x;r) = 1] \approx \Pr_{z \sim \{0,1\}^{s}}[A(x;G(z)) = 1].$$

  - ここで、$f:\{0,1\}^* \to \{0,1\}$はAによって計算される判定問題とする。
  - 簡単のため、$f \equiv 1$とする。（実は回路については一般性を失わない。）

rand()はあるアルゴリズムAを**模倣できない**

- $G: \{0,1\}^s \to \{0,1\}^{31m}$

  $G(z) = r_1 r_2 r_3 \dots r_m$, where $r_{i+1} \coloneqq (a r_i + c) \bmod 2^{31}$, $r_0 \coloneqq z$.

- 次のアルゴリズム$A(;r)$を考える： $A(; r_1 r_2 r_3 \dots) \coloneqq \begin{cases} 1 & \text{if } r_2 = (a r_1 + c) \bmod 2^{31} \\ 0 & \text{otherwise} \end{cases}$

- すると：

$$\Pr_{r \sim \{0,1\}^{31m}}[A(;r) = 1] = \Pr_{r_1, r_2 \sim \{0,1\}^{31}}[r_2 = (a r_1 + c) \bmod 2^{31}] = 2^{-31} \approx 0.$$

$$\ll \Pr_{z \sim \{0,1\}^{s}}[A(;G(z)) = 1] = 1.$$

- $A(;-)$は疑似乱数列$G(z)$と真の乱数列rを**識別できる**。

統計的検定（Statistical Test）

- より一般に、$G:\{0,1\}^s \to \{0,1\}^m$を$s < m$を満たす関数とする。

  長さsの疑似乱数の種zを受け取って、
  「疑似乱数列」 $G(z)$を出力する関数だとみなす。

- 関数$T:\{0,1\}^m \to \{0,1\}$が$G(-)$を（一様分布から）ϵ-識別する とは、

$$\left| \Pr_{z\sim\{0,1\}^s}\left[T\left(G(z)\right)=1\right] - \Pr_{r\sim\{0,1\}^m}\left[T(r)=1\right] \right| \geq \epsilon$$

- Tは$G(-)$に対するϵ-統計的検定（または ϵ-distinguisher）とも呼ばれる。

- 通常は$\epsilon := 1/m$と定義して、単純にTが$G(-)$を識別するという。

疑似乱数生成器 (PRG; Pseudorandom Generator)

定義： 疑似乱数生成器

関数$G:\{0,1\}^s \to \{0,1\}^m$と統計的検定のクラス$\mathcal{C}$について、

$s < m$ かつ 任意の$T \in \mathcal{C}$がGをϵ-識別できない、つまり

$$\left| \Pr_{z\sim\{0,1\}^s}[T(G(z)) = 1] - \Pr_{r\sim\{0,1\}^m}[T(r) = 1] \right| < \epsilon$$

のとき、Gをクラス$\mathcal{C}$に対して**ϵ-安全な疑似乱数生成器**と呼ぶ。

- rand()関数は安全でない疑似乱数生成器の例。暗号には絶対に使ってはいけない！

➢ 乱択アルゴリズム$A(x;r)$を模倣するためには、
任意の入力$x \in \{0,1\}^*$について$A(x;-) \in \mathcal{C}$となる疑似乱数生成器Gを使えばよい。

$$\left| \Pr_{z\sim\{0,1\}^s}[A(x;G(z)) = 1] - \Pr_{r\sim\{0,1\}^m}[A(x;r) = 1] \right| < \epsilon.$$

パラメータの取り方について

- $G\colon\{0,1\}^s \to \{0,1\}^m$：クラス$\mathcal{C}$に対して安全な疑似乱数生成器
  - Gを使うことで、mビットの乱数列をsビットに減らせる。
  - 理想的には$s = O(\log m)$ビットに減らしたい。
- クラス$\mathcal{C}$:= {線形サイズの論理回路} = $\{C\colon m\text{-size circuits}\}$

$$\forall C \in \mathcal{C},\quad \left| \Pr_{z\sim\{0,1\}^s}\left[C\left(G(z)\right)=1\right] - \Pr_{r\sim\{0,1\}^m}\left[C(r)=1\right] \right| < \frac{1}{m}.$$

特に、（Aの計算時間 $\ll m$となるようにmを十分大きくとると）$A(x;-) \in \mathcal{C}$であるので、

$$\forall x,\quad \left| \Pr_{z\sim\{0,1\}^s}\left[A\left(x;G(z)\right)=1\right] - \Pr_{r\sim\{0,1\}^m}\left[A(x;r)=1\right] \right| < \frac{1}{m}.$$

∃ 疑似乱数生成器 $\Longrightarrow$ BPP = P (脱乱択化)

- ∃ 疑似乱数生成器 $G = \{G_m : \{0,1\}^{O(\log m)} \to \{0,1\}^m\}$ が $m^{O(1)}$時間で計算可能だとし、線形サイズの回路に対して安全だと仮定する。

- 判定問題$f \in$ BPPとfを解く乱択多項式時間アルゴリズム$A(x;r)$を取る。

$$\Pr_{r\sim\{0,1\}^m}[A(x;r) = f(x)] \geq \frac{3}{4}.$$

$m \coloneqq \mathrm{poly}(n)$ とする。

≈ （安全性より） ← $C_x(r) \coloneqq A(x;r) \oplus f(x) \oplus 1$という回路を考える。

$$\Pr_{z\sim\{0,1\}^{O(\log m)}}\left[A\left(x; G_m(z)\right) = f(x)\right] \geq \frac{3}{4} - 0.01 \geq \frac{2}{3}.$$

- 新しいアルゴリズム$A\left(x; G_m(z)\right)$は$O(\log m)$ビットの乱数列しか使わない！
  - 全ての乱数列を全探索することで、$2^{O(\log m)} = \mathrm{poly}(n)$ 時間で模倣できる。
  - 従って、$f \in$ P。

計算的識別不可能性 (computationally indistinguishable)

- より一般に、$\{0,1\}^*$上の二つの確率分布X, Yに対して、
  任意の小さい回路Cについて、
  $$\Pr_X[C(X) = 1] \approx \Pr_Y[C(Y) = 1]$$
  のとき、XとYは**計算的に識別不可能**という。このとき、
  $$X \approx_c Y$$
  と書く。（c: **c**omputationally）

- 疑似乱数生成器$G: \{0,1\}^s \to \{0,1\}^m$は
  $$G(\mathcal{U}_s) \approx_c \mathcal{U}_m$$
  を満たす。ただし、$\mathcal{U}_n$は$\{0,1\}^n$上の一様分布。

概要

1. 脱乱択化の理論
   - 理論の概要
   - 疑似乱数生成器の概念
   - 疑似乱数生成器の構成

2. メタ計算量と平均時計算量について
   - メタ計算問題
   - 平均時計算量
   - 最悪時・平均時計算量の同値性
   - 暗号との関係

PRGの構成のための３つの証明技法

1. 識別可能性 ⇔ 次のビットの予想可能性（Next-bit predictor）

   $G: \{0,1\}^n \to \{0,1\}^{n+1}$：1ビット延長できる

2. 混成分布に基づく議論（Hybrid argument）

   $G: \{0,1\}^{nk} \to \{0,1\}^{nk+k}$： kビット延長できる

3. 組合せデザイン（Combinatorial design）

   $G: \{0,1\}^{O(\log m)} \to \{0,1\}^m$：指数的に延長できる

もっとも単純な疑似乱数生成器の構成

- 非自明な疑似乱数生成器 $G\colon\{0,1\}^n \to \{0,1\}^{n+1}$ を構成する。

主張 [Yao'82]

$\mathrm{E} \not\subseteq \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n};\delta)$ならば、指数サイズの回路クラスに対して$\delta$-安全かつ指数時間で計算可能な疑似乱数生成器$G\colon\{0,1\}^n \to \{0,1\}^{n+1}$が存在する。

- $h \in \widetilde{\mathrm{SIZE}}(s(n);\delta)$: 関数$h\colon\{0,1\}^* \to \{0,1\}$は次を満たす。任意の$n$について、サイズ$s(n)$の回路$C$が存在して、$h_n := h|_{\{0,1\}^n}$を$\delta$-近似する、つまり

$$\Pr_{x\sim\{0,1\}^n}[C(x) = h_n(x)] \geq \frac{1}{2} + \delta\,.$$

- $h \in \mathrm{E} \setminus \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n};\delta)$なる関数$h\colon\{0,1\}^* \to \{0,1\}$をとる。

疑似乱数生成器の構成

構成:

$$G:\{0,1\}^n \to \{0,1\}^{n+1},$$
$$G(z) := \left(z, h_n(z)\right) \in \{0,1\}^{n+1}.$$

$h \in \mathrm{E}$: 回路で近似困難な関数

主張: Gに対するδ-統計的検定 $D:\{0,1\}^{n+1} \to \{0,1\}$ が存在するならば、
hが小さい回路で近似できる。 (D: 回路サイズ$2^{\epsilon n}$)

$\Longrightarrow h \notin \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n};\delta)$ に矛盾。

$\Longrightarrow G$は安全な疑似乱数生成器。

証明:

$$\left|\Pr_{z\in\{0,1\}^n}\left[D\left(G(z)\right)=1\right]-\Pr_{w\in\{0,1\}^{n+1}}\left[D(w)=1\right]\right| \geq \delta.$$

$\Downarrow$

$$\Pr_{z\in\{0,1\}^n}\left[D\left(G(z)\right)=1\right]-\Pr_{w\in\{0,1\}^{n+1}}\left[D(w)=1\right] \geq \delta$$

or

$$\Pr_{z\in\{0,1\}^n}\left[D\left(G(z)\right)=1\right]-\Pr_{w\in\{0,1\}^{n+1}}\left[D(w)=1\right] \leq -\delta.$$

$\Downarrow$ $D'(w) := \neg D(w)$と定義

$$\Pr_{z\in\{0,1\}^n}\left[D'\left(G(z)\right)=1\right]-\Pr_{w\in\{0,1\}^{n+1}}\left[D'(w)=1\right] \geq \delta.$$

一般性を失わず、左の場合のみを考えればよい。

識別可能 ⟹ 次のビットを予測可能

主張: Gに対するδ-統計的検定 $D:\{0,1\}^{n+1} \to \{0,1\}$ が存在するならば、
hを$2^{\epsilon n}$サイズの回路で近似できる。　（D: 回路サイズ$2^{\epsilon n}$）

$$\Pr_{z\in\{0,1\}^n}\left[D\left(z, h_n(z)\right) = 1\right] - \Pr_{w\in\{0,1\}^{n+1}}[D(w) = 1] \geq \delta.$$

Dは次の二つの分布を識別できる：

(1) $\left(z, h_n(z)\right)$　（ただし、$z \sim \{0,1\}^n$）

(2) (z, b)　（ただし、$z \sim \{0,1\}^n, b \sim \{0,1\}$）

[Yao'82]

⟹ Dを用いて "next-bit predictor" P^D を構成できる。

$G(z)$の最初のnビットが与えられたときに、次のビットを予測できる。

z　　$h_n(z)$

(Cf. rand()関数の次のビットを予測できる。)

識別可能 ⟹ 次のビットを予測可能

主張: Gに対するδ-統計的検定 $D:\{0,1\}^{n+1} \to \{0,1\}$ が存在するならば、
hを$2^{\epsilon n}$サイズの回路で近似できる。 （D: 回路サイズ$2^{\epsilon n}$）

$$\Pr_{z\in\{0,1\}^n}\left[D\left(z, h_n(z)\right) = 1\right] - \Pr_{\substack{z\in\{0,1\}^n \\ b\in\{0,1\}}}[D(z,b) = 1] \geq \delta.$$

“next-bit predictor” P^Dの構成方法：

$$P^D(z;b) = \begin{cases} b & \text{if } D(z,b) = 1 \\ b \oplus 1 & \text{otherwise} \end{cases}$$

（ここで、$b \sim \{0,1\}$）

アイディア: もし $D(z,b) = 1$ ならば $h_n(z) = b$ である確率が高いはず。

事実: $\Pr_{z,b}[P^D(z;b) = h_n(z)] \geq \frac{1}{2} + \delta$ ⟹ h_n が δ-近似できる。

事実の証明

$$P^D(z;b) = \begin{cases} b & \text{if } D(z,b)=1 \\ b \oplus 1 & \text{otherwise} \end{cases}$$

事実: $\Pr_z[P^D(z;b) = h_n(z)] \geq \frac{1}{2} + \delta$

仮定: $\Pr_{z\sim\{0,1\}^n}[D(z,h_n(z)) = 1] - \Pr_{\substack{z\sim\{0,1\}^n \\ b\sim\{0,1\}}}[D(z,b)=1] \geq \delta.$

$$\Pr_{\substack{z\sim\{0,1\}^n \\ b\sim\{0,1\}}}[D(z,b)=1] = \frac{1}{2}\Pr_z[D(z,h_n(z))=1] + \frac{1}{2}\Pr_z[D(z,\neg h_n(z))=1].$$

$(b = h_n(z)$ or $b = \neg h_n(z))$

$$\Rightarrow \quad \frac{1}{2}\Pr_{z\sim\{0,1\}^n}[D(z,h_n(z))=1] - \frac{1}{2}\Pr_z[D(z,\neg h_n(z))=1] \geq \delta.$$

$(b = h_n(z)$ or $b = \neg h_n(z))$

$$\Pr_{z,b}[P^D(z;b) = h_n(z)] = \frac{1}{2}\Pr[D(z,h_n(z))=1] + \frac{1}{2}\Pr[D(z,\neg h_n(z))=0]$$

$$= \frac{1}{2}\Pr[D(z,h_n(z))=1] + \frac{1}{2} - \frac{1}{2}\Pr[D(z,\neg h_n(z))=1] \geq \frac{1}{2} + \delta.$$

単純な疑似乱数生成器の構成のまとめ

主張 [Yao'82]

$\mathrm{E} \not\subseteq \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n}; \delta)$ならば、指数サイズの回路クラスに対して$\delta$-安全かつ指数時間で計算可能な疑似乱数生成器$G: \{0,1\}^n \to \{0,1\}^{n+1}$が存在する。

- 構成:
  - $h \in \mathrm{E} \setminus \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n}; \delta)$ となる計算困難な関数$h: \{0,1\}^* \to \{0,1\}$をとる。
  - $z \in \{0,1\}^n$ に対して $G(z) := \big(z, h(z)\big) \in \{0,1\}^{n+1}$ と定める。
- 証明のアイディア: 背理法

  D が $G(-)$に対する統計的検定 $\Longrightarrow$ P^D: "next-bit predictor"が構成できる。

PRGの構成のための３つの証明技法

1. 識別可能性 ⇔ 次のビットの予想可能性（Next-bit predictor）

   $G:\{0,1\}^n \to \{0,1\}^{n+1}$：1ビット延長できる

2. 混成分布に基づく議論（Hybrid argument）

   $G:\{0,1\}^{nk} \to \{0,1\}^{nk+k}$： kビット延長できる

3. 組合せデザイン（Combinatorial design）

   $G:\{0,1\}^{O(\log m)} \to \{0,1\}^m$：指数的に延長できる

kビットの拡張

- 計算困難な関数$h\colon \{0,1\}^n \to \{0,1\}$をとる。
- $G^h\colon \{0,1\}^n \to \{0,1\}^{n+1}$：1ビット拡張する疑似乱数生成器

$$G^h(z) \coloneqq \big(z, h(z)\big).$$

hの困難性$\Longrightarrow$ G^hの安全性

- これをkビットに拡張したい。

$$\mathrm{DP}_k^h\colon \{0,1\}^{nk} \to \{0,1\}^{nk+k}$$

$$\mathrm{DP}_k^h \coloneqq \left(G^h\right)^k$$

$$\mathrm{DP}_k^h(z_1, \dots, z_k) \coloneqq \big(z_1, \dots, z_k, h(z_1), \dots, h(z_k)\big)$$

kビットの拡張: k-wise Direct Product Genertor

主張

$\mathrm{E} \not\subseteq \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n}; \delta/k)$ ならば、ある疑似乱数生成器

$$\mathrm{DP}_k : \{0,1\}^{kn} \to \{0,1\}^{kn+k}$$

が存在して、$2^{\epsilon n}$サイズの回路に対してδ-安全で、$2^{O(n)}$時間で計算可能。

- 構成: k-wise **D**irect **P**roduct Generator （k直積生成器）
  - 関数$h: \{0,1\}^* \to \{0,1\}$ であって $h \in \mathrm{E} \setminus \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n}; \delta/k)$ なるものをとる。
  - $\mathrm{DP}_k^h : (\{0,1\}^n)^k \to \{0,1\}^{nk+k}$を次のように定める。

$$\mathrm{DP}_k^h(z_1, \dots, z_k) := \big(z_1, \dots, z_k, h(z_1), \dots, h(z_k)\big).$$

アイディア: Hybrid Argument

$$\mathrm{DP}_k^h : (\{0,1\}^n)^k \to \{0,1\}^{nk+k}$$
$$\mathrm{DP}_k^h(z_1, \dots, z_k) := \left(z_1, \dots, z_k, h(z_1), \dots, h(z_k)\right).$$

- $\mathrm{DP}_k^h(-)$に対する統計的テストDが存在したとする。すなわち：

$$\Pr_{z_1,\dots,z_k}\left[D\left(\mathrm{DP}_k^h(z_1, \dots, z_k)\right) = 1\right] - \Pr_{\substack{z_1,\dots,z_k \\ b_1,\dots,b_k}}[D(z_1, \dots, z_k, b_1, \dots, b_k) = 1] \geq \delta$$

- $\left(z_1, \dots, z_k, h(z_1), \dots, h(z_k)\right)$ と $(z_1, \dots, z_k, b_1, \dots, b_k)$ を直接比べるのは難しい。
- Hybrid argument ：二つの分布を補間する分布$H_0, H_1, \dots, H_k$を考える。

混合分布に基づく議論（Hybrid Argument）

$$\Pr_{z_1,\dots,z_k}\left[D\left(z_1,\dots,z_k,h(z_1),\dots,h(z_k)\right)=1\right]-\Pr_{\substack{z_1,\dots,z_k\\ b_1,\dots,b_k}}\left[D(z_1,\dots,z_k,b_1,\dots,b_k)=1\right]\geq\delta$$

$\equiv H_k$ $\equiv H_0$

- i番目の混合分布 $H_i \equiv (z_1,\dots,z_k,h(z_1),\dots,h(z_i),b_{i+1}\dots,b_k)$
  （ここで、$i\in\{0,\dots,k\}, z_1,\dots,z_k\sim\{0,1\}^n, b_1,\dots,b_k\sim\{0,1\}$）

$$\delta\leq\Pr[D(H_k)=1]-\Pr[D(H_0)=1]=\sum_{i=1}^{k}(\Pr[D(H_i)=1]-\Pr[D(H_{i-1})=1])$$

$$\Rightarrow\quad\frac{\delta}{k}\leq\Pr[D(H_i)=1]-\Pr[D(H_{i-1})=1]$$ （ある$i\in\{1,\dots,k\}$について）

$$=\Pr[D(z_1,\dots,z_k,h(z_1),\dots,h(z_i),b_{i+1},\dots,b_k)=1]-\Pr[D(z_1,\dots,z_k,h(z_1),\dots,b_i,b_{i+1}\dots,b_k)=1]$$

$$=\mathbb{E}[D(z_1,\dots,z_k,h(z_1),\dots,h(z_i),b_{i+1},\dots,b_k)-D(z_1,\dots,z_k,h(z_1),\dots,b_i,b_{i+1}\dots,b_k)].$$

$\Rightarrow$ 期待値を最大化するようなz_j,b_jを全ての$j\in\{1,\dots,k\}\setminus\{i\}$について固定し、
$D'(z_i,b_i):=D(z_1,\dots,z_k,h(z_1),\dots,b_k)$ と定めると、

Yao's next-bit predictor

$$\Pr_{z_i}\left[D'\left(z_i,h(z_i)\right)=1\right]-\Pr_{z_i,b_i}\left[D'(z_i,b_i)=1\right]\geq\delta/k\quad\Rightarrow\quad h\in\widetilde{\mathrm{SIZE}}(2^{\epsilon n};\delta/k).$$

PRGの構成のための３つの証明技法

1. 識別可能性 ⇔ 次のビットの予想可能性（Next-bit predictor）

   $G: \{0,1\}^n \to \{0,1\}^{n+1}$：1ビット延長できる

2. 混成分布に基づく議論（Hybrid argument）

   $G: \{0,1\}^{nk} \to \{0,1\}^{nk+k}$： kビット延長できる

3. 組合せデザイン（Combinatorial design）

   $G: \{0,1\}^{O(\log m)} \to \{0,1\}^m$：指数的に延長できる

指数的に乱数の種を延長する

- k直積生成器では、独立な入力で計算困難関数hを計算する。

$$\mathrm{DP}_k^h\colon\ (z_1,\dots,z_k)\mapsto\left(z_1,\dots,z_k,h(z_1),\dots,h(z_k)\right)$$

h は計算困難 $\Longrightarrow$ DP_k^h は安全。

- hを相関のある指数個の入力で計算する。

$$\mathrm{NW}^h\colon\ z\ \mapsto\ \left(z_{S_1},\dots,z_{S_m}\right)\mapsto\left(h\left(z_{S_1}\right),\dots,h\left(z_{S_m}\right)\right)$$

指数的に乱数の種を伸ばす

定理 [Nisan & Wigderson '94]

ある定数$\epsilon > 0$について$\mathrm{E} \not\subseteq \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n}; 2^{-\epsilon n})$ならば、ある疑似乱数生成器 $\mathrm{NW} = \left\{\mathrm{NW}_N : \{0,1\}^{O(\log N)} \to \{0,1\}^N\right\}_{N \in \mathbb{N}}$ であって、

- サイズN回路に対して安全で、かつ
- $N^{O(1)}$時間で計算可能なもの

が存在する。特に、$\mathrm{P} = \mathrm{BPP}$。

- $h : \{0,1\}^n \to \{0,1\}$: 計算困難な関数として、

$$\mathrm{NW}_N : \{0,1\}^{O(\log N)} \to \{0,1\}^N$$

$$\mathrm{NW}^h(z) := \left(h\left(z_{S_1}\right), \ldots, h\left(z_{S_N}\right)\right)$$

詳細は演習問題

ただし、$n = O(\log N)$。

Eに対する最悪時・平均時計算量の同値性

[Nisan-Wigderson '94] $\mathrm{E} \not\subseteq \bigcap_{\epsilon>0} \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n}; 2^{-\epsilon n}) \implies \mathrm{P} = \mathrm{BPP}$

⇕ 局所リスト復号可能誤り訂正符号

[Impagliazzo-Wigderson '97] $\mathrm{E} \not\subseteq \bigcap_{\epsilon>0} \text{io-}\mathrm{SIZE}(2^{\epsilon n}) \implies \mathrm{P} = \mathrm{BPP}$

局所リスト復号可能誤り訂正符号 $\mathrm{Enc}: f \mapsto \mathrm{Enc}(f)$ の性質

[Sudan-Trevisan-Vadhan '01]

1. $f \in \mathrm{E} \implies \mathrm{Enc}(f) \in \mathrm{E}.$
2. $\mathrm{Enc}(f) \in \text{io-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n}; 2^{-\epsilon n}) \implies f \in \text{io-}\mathrm{SIZE}\left(2^{\epsilon' n}\right).$

Hardness versus Randomness Trade-off

$$\mathrm{EXP} \not\subseteq \mathrm{ioSIZE}\left(n^{O(1)}\right) \implies \mathrm{BPP} \subseteq \mathrm{SUBEXP} := \bigcap_{\epsilon>0} \mathrm{DTIME}\left(2^{n^{\epsilon}}\right).$$

(∃ PRG $G: \{0,1\}^{m^{\epsilon}} \to \{0,1\}^{m}$, computable in time $2^{m^{\epsilon}}$)

$$\mathrm{EXP} \not\subseteq \bigcap_{\epsilon>0} \mathrm{ioSIZE}\left(2^{n^{\epsilon}}\right) \implies \mathrm{BPP} \subseteq \mathrm{QuasiP} := \mathrm{DTIME}\left(2^{(\log n)^{O(1)}}\right).$$

(∃ PRG $G: \{0,1\}^{(\log m)^{O(1)}} \to \{0,1\}^{m}$, computable in time $2^{(\log m)^{O(1)}}$)

$$\mathrm{E} \not\subseteq \bigcap_{\epsilon>0} \mathrm{ioSIZE}\left(2^{\epsilon n}\right) \implies \mathrm{BPP} \subseteq \mathrm{P}.$$

(∃ PRG $G: \{0,1\}^{O(\log m)} \to \{0,1\}^{m}$, computable in time $m^{O(1)}$)

疑似乱数生成器の理論の応用

- Randomness extractor の構成 [Trevisan ’01]
- $\mathrm{AC}^0[\oplus]$（PARITY付き定数段回路 ）を準指数時間で学習。
  [Carmosino, Impagliazzo, Kabanets, Kolokolova (CCC’16)]
- 非ブラックボックスな最悪時・平均時帰着 [H. FOCS’18]

これらの結果は、疑似乱数生成器の理論を**メタ的**にみることによって得られる。

脱乱択化の理論のまとめ

- どのように乱択アルゴリズム$A(x;r)$を模倣すべきか
  - $A(x;r)$に対して安全な疑似乱数生成器を使うべき。

    rand()関数で振る舞いが変わりそうなときは使ってはダメ
  - 回路下界（計算困難性） $\Longrightarrow$ 安全な疑似乱数生成器が構成可能。

参考文献：Salil Vadhan, “Pseudorandomness”, 2012

より最近の発展については：Lijie Chen, Roei Tell ”New ways of studying the BPP = P conjecture”, 2023

概要

1. 脱乱択化の理論
   - 理論の概要
   - 疑似乱数生成器の概念
   - 疑似乱数生成器の構成

2. メタ計算量と平均時計算量について
   - メタ計算問題
   - 平均時計算量
   - 最悪時・平均時計算量の同値性
   - 暗号との関係

メタ計算量と平均時計算量

- メタ計算量：「計算量を問う問題の計算量」
  - メタ計算問題の代表例：MCSP（回路最小化問題）、MINKT
  - これらのメタ計算問題に対して最悪時・平均時帰着が存在 [H. FOCS'18]。
- 平均時計算量
  - 中心的未解決問題：NPの最悪時・平均時計算量は同値か？

    Cf. EXPやPSPACEの最悪時・平均時計算量は同値（∵ 局所リスト復号可能誤り訂正符号）。
  - メタ計算問題がNP完全であれば、肯定的に解決できる！

メタ計算量（Meta-complexity）

- **メタ計算量**：**計算量**を問う計算問題の**計算量**のこと。
  - 近年の理論計算機科学におけるホットトピックのひとつ

- **計算量**（complexity, 複雑さ）
  - 計算に必要な資源のこと
    例: 計算時間、メモリ、回路サイズ、記述量

- **P ≠ NP**予想の同値な言いかえ：
  - 三彩色問題を解くための計算時間は超多項式的か？
  - 特に、三彩色問題 $\notin \mathrm{SIZE}(n^{O(1)})$ を示すことができれば、$\mathrm{P} \neq \mathrm{NP}$。

回路計算量の歴史

- 関数fを計算する最小の回路サイズ（回路計算量; Circuit Complexity）を$\mathrm{CC}(f)$と書く。
- 一様ランダムな関数$f:\{0,1\}^n \to \{0,1\}$について、$\mathrm{CC}(f) \approx 2^n/n$ [Shannon 1949]
  - $\mathrm{CC}(\text{三彩色問題}) \geq n^{\omega(1)} \implies \mathrm{P} \neq \mathrm{NP}$

（ある関数$f \in \mathrm{P}$について、$B_2 = \{g:\{0,1\}^2 \to \{0,1\}\}$基底に対して）

$\mathrm{CC}(f) \geq 2n - O(1)$ [Kloss & Malyshev 1965]

$\mathrm{CC}(f) \geq 2.5n - O(1)$ [Stockmeyer 1977]

$\mathrm{CC}(f) \geq 3n - o(n)$ [Blum 1984]

$\mathrm{CC}_{U_2}(f) \geq 5n - o(n)$ [Iwama & Morizumi 2002]

… 約３０年の時が流れる

$\mathrm{CC}(f) \geq 3.011n - o(n)$ [Find, Golovnev, Hirsch, and Kulikov (FOCS 2016)]

$\mathrm{CC}(f) \geq 3.1n - o(n)$ [Li & Yang (STOC 2022)]

回路計算量をメタ的にみる

- ここまでは具体的な関数$f:\{0,1\}^n \to \{0,1\}$の回路計算量を考えた。
  (explicit; $f \in \mathrm{P}$ や $f \in \mathrm{NP}$ などを意味する)
- しかし、回路計算量の計算は（人間には）難しい！
  - アルゴリズムにとっても難しいはず。
- **回路計算量**を計算するための**計算量**は？ ⟹ 回路最小化問題
- 回路計算量の計算が多項式時間アルゴリズムに対しても
  難しいことを示せれば、$\mathrm{P} \neq \mathrm{NP}$。
  - Andrey Kolmogorov: 同様のアプローチをコルモゴロフ記述量に基づいて提唱 （1960年代）

回路計算量を**計算する**問題

MCSP (Minimum Circuit Size Problem; 回路最小化問題)

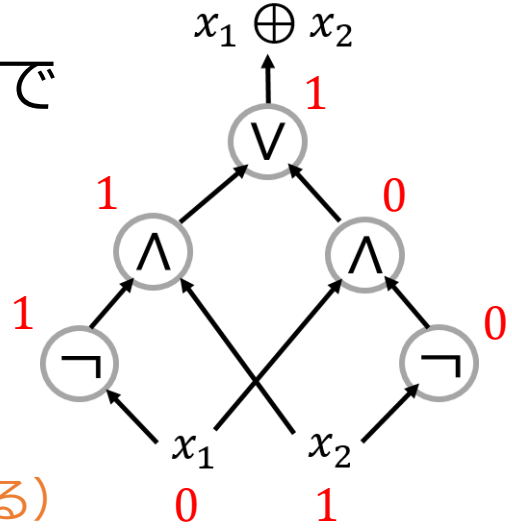
入力

- ブール値関数 $f:\{0,1\}^n \to \{0,1\}$ の真理値表　($f \in \{0,1\}^{2^n}$ と同一視)
- 閾値 $s \in \mathbb{N}$

出力

$\mathrm{CC}(f) \leq s$?

ゲート数s以下の論理回路で
関数fを計算するものが
存在するかどうか。



例
- $f = \oplus_2$ (2入力XOR)
- $s = 5$

YES

(∵ 右図のような論理回路が存在する)

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

簡単な事実: $\mathrm{MCSP} \in \mathrm{NP} = \mathrm{NTIME}\left(N^{O(1)}\right)$

$\mathrm{MCSP} \in \mathrm{DTIME}\left(2^{O(N)}\right)$　($N := 2^n$, 入力長)

重要な未解決問題: MCSPはNP完全か？

NP完全性の概念がCookとLevinによって導入された1970年代初頭から未解決。

回路最小化問題は「圧縮」の問題

- 回路最小化問題は、
  「入力文字列$f \in \{0,1\}^{2^n}$を小さい回路Cでfを圧縮表現できるか」
  という問題だとみなせる。
  - サイズsの回路は$O(s\log(s+n)) \ll 2^n$ビットで表現できるから。
- コルモゴロフ記述量：最適な圧縮の方法
  - より効率的に圧縮できるので、メタ計算量の研究ではより扱いやすい。
  - $\mathrm{K}(x) =$ (xを出力する最小のプログラムの長さ)

コルモゴロフ記述量

解凍機の例：unzip、 Pythonのインタプリタ

(decompressor)

- 解凍機Mに関する文字列$x \in \{0,1\}^*$のコルモゴロフ記述量

$$\mathrm{K}_M(x) = \min\{\, |d| : M(d) = x \,\}.$$

- 普遍チューリング機械Uをとると、コルモゴロフ記述量はほぼ最小になる。
  - 任意のチューリング機械Mに対して、ある$d_M \in \{0,1\}^*$が存在して、
$$U(d_M, x) = M(x).$$
  - 従って、任意の機械Mについて、ある定数c_Mが存在して、
$$\mathrm{K}_U(x) \leq \mathrm{K}_M(x) + c_M.$$

- 以降、Uを一つ固定して、$\mathrm{K}(x) := \mathrm{K}_U(x)$と略記する。

時間制限付きコルモゴロフ記述量

- 文字列xのt時間制限付きコルモゴロフ記述量とは

$$\mathrm{K}^t(x) \coloneqq \min\{|d| : \text{入力}d\text{で}U\text{が}x\text{を}t\text{時間以内で出力する}\}.$$

例

- $\mathrm{K}^t(\underbrace{00\ldots0}_{n\text{回繰り返し（}0^n\text{と書く）}}) = \log n + O(1)$　（ただし、$t \gg n$.）　← print “0” × n
- $\mathrm{K}^t(x) \leq n + O(1)$　（ただし、$t \gg n$、$x \in \{0,1\}^n$）　← print “x”
- $\Pr_{x\sim\{0,1\}^n}[\mathrm{K}(x) \geq n-2] \geq \frac{3}{4}$.　← 数え上げの議論（演習問題）

コルモゴロフ記述量を**計算する**問題

MINKT （時間制限付きコルモゴロフ記述量問題）

入力

- 文字列$x \in \{0,1\}^*$
- サイズパラメタ$s \in \{0,1\}^*$
- 時間パラメタ1^t
  （一進法で書く）

出力　　$\mathrm{K}^t(x) \leq s$?

文字列xを時間制限t以内で出力するsビット以下のプログラムが存在するか。

➢ 言語として定義すると、$\mathrm{MINKT} := \{(x, 1^t, 1^s) \mid \mathrm{K}^t(x) \leq s\}$

簡単な事実: $\mathrm{MINKT} \in \mathrm{NP}$

$\mathrm{MINKT} \in \mathrm{DTIME}\left(2^{O(N)}\right)$　($N = \Theta(|x| + t)$: 入力長)

重要な未解決問題: MINKTはNP完全か？

概要

1. 脱乱択化の理論
   - 理論の概要
   - 疑似乱数生成器の概念
   - 疑似乱数生成器の構成

2. メタ計算量と平均時計算量について
   - メタ計算問題
   - 平均時計算量
   - 最悪時・平均時計算量の同値性
   - 暗号との関係

平均時計算量と最悪時計算量

悲観的すぎる
ことが多い

- 二つの計算量の概念
  - 最悪時計算量：**計算時間が最も大きくなる入力**での計算時間　　P　　$\mathrm{SIZE}(s)$
  - 平均時計算量：**入力をランダムに**とったときの平均計算時間　　AvgP　　$\widetilde{\mathrm{SIZE}}(s)$

より現実に
即している

- 例：NP完全問題
  - 実応用上は、NP完全問題であってもSATソルバーで高速に解けることが多い。
  - 理論的にも、いくつかのNP完全問題は自然な入力分布上で平均時多項式時間で解ける。
    - ✓ ハミルトンパス問題はErdős–Rényiランダムグラフ上で期待線形時間で解ける。
      [Gurevich & Shelah (1987)]
- 脱乱択化や暗号理論においても平均時計算量が重要となる。

DistNP: 平均時計算量版のNP

- $(L, \mathcal{D})$: 分布問題 (distributional problem)

  判定問題Lと入力分布$\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$の組のこと。

- どのような入力分布を考えるべきか？

  → 実用上現れる入力は**効率的にサンプル可能な分布**から生成されているはず

  $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$が**多項式時間サンプル可能である**とは、

  ある乱択多項式時間アルゴリズムAが存在して、

  $$\Pr_A[A(1^n) = x] = \Pr_{X\sim\mathcal{D}_n}[X = x]$$

  となることを言う。

- $\mathrm{DistNP} := \{(L, \mathcal{D}) \mid L \in \mathrm{NP},\ \mathcal{D}$: 多項式時間サンプル可能$\}$

  （NPの分布問題版）

効率的サンプル可能な分布だけを考える！

AvgP: 平均時多項式時間計算可能な分布問題

$\mathrm{AvgP} := \{(L,\mathcal{D}) \mid$ 平均時多項式時間計算可能な分布問題$(L,\mathcal{D})\}$ [Levin'86]

Aが$(L,\mathcal{D})$に対する平均時多項式時間アルゴリズムであるとは、
任意の$n \in \mathbb{N}$について、以下の二つの条件を満たす。

1. 任意の$x \in \mathrm{supp}(\mathcal{D}_n)$について、 $A(x,1^n) = L(x)$.
2. ある定数$\epsilon > 0$に対して、$\mathbb{E}_{x\sim\mathcal{D}_n}[t_A(x,1^n)^\epsilon] \leq n^{O(1)}$.
   ただし、$t_A(x,1^n)$はAの入力$(x,1^n)$における計算時間。

定数ϵ:
定義をロバストにするために必要。

誤りなしヒューリスティクスを用いて、同値な定義が与えられる。（演習問題）

※ 本講演の全ての分布では、$x \in \mathrm{supp}(\mathcal{D}_n)$から$1^n$が計算可能なので、以降では$1^n$は省略する。

例：$\mathcal{U} = \{\mathcal{U}_n\}_{n\in\mathbb{N}}$：$\{0,1\}^n$上の一様分布。

NPの最悪時計算量 と 平均時計算量

中心的未解決問題 Heuristicaを除外できるか

$$\mathrm{P} \neq \mathrm{NP} \quad \overset{?}{\Longrightarrow} \quad \mathrm{DistNP} \not\subseteq \mathrm{AvgP}$$

（NPが最悪時計算量の意味で計算困難であれば、
ある多項式時間サンプル可能な分布上において
NPが平均時計算量の意味で計算困難か？）

本質的には一様分布だけを考えても一般性を失わない
[Impagliazzo-Levin'90]

- より安全な暗号を構築するための重要な一歩
- なぜ解決するのが難しいのか？ ← 理論的障壁

✗「ブラックボックス帰着の限界」　✗「困難性増幅の不可能性」　✗「相対化のバリア」

概要

1. 脱乱択化の理論
   - 理論の概要
   - 疑似乱数生成器の概念
   - 疑似乱数生成器の構成

2. メタ計算量と平均時計算量について
   - メタ計算問題
   - 平均時計算量
   - 最悪時・平均時計算量の同値性
   - 暗号との関係

MCSPの最悪時・平均時計算量の同値性

- ブラックボックス帰着の限界 [Bogdanov & Trevisan '06]
  - 問題LがDistNPに（非適応的）乱択多項式時間で帰着できるならば、$L \in \mathrm{coNP/poly}$。
  - 帰着を構成するような標準的な手法では、NP完全問題をDistNPに帰着できない。
    （$\mathrm{NP} \subseteq \mathrm{coNP/poly}$でない限り）

定理 [H. (FOCS 2018)]

$\mathcal{U}$: 一様分布として、以下は同値。

1. ある定数$\epsilon > 0$について、$(\mathrm{MCSP}[2^{\epsilon n}], \mathcal{U}) \in \mathrm{AvgBPP}$。
   $\mathrm{MCSP}[2^{\epsilon n}]$: サイズパラメタが$2^{\epsilon n}$に固定された回路最小化問題
2. ある定数$\epsilon > 0$について、$\mathrm{Gap}_{\epsilon}\mathrm{MCSP} \in \mathrm{BPP}$。
   $\mathrm{Gap}_{\epsilon}\mathrm{MCSP}$: 回路計算量の$2^{(1-\epsilon)n}$倍近似を計算する問題

- ブラックボックス帰着の限界の限界を突破する初めての結果。
  - [Rudich'97]の予想：$\mathrm{Gap}_{\epsilon}\mathrm{MCSP} \notin \mathrm{coNP/poly}$

約束付き問題$\mathrm{Gap}_\epsilon\mathrm{MCSP}$の定義

- 近似問題は約束付き問題$\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$によって定式化される。
- アルゴリズムAが$\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$を**解く**とは、全ての$x$で以下が成立すること。

$$(1)\ x \in \Pi_{\mathrm{YES}} \implies A(x) = 1,$$

$$(2)\ x \in \Pi_{\mathrm{NO}} \implies A(x) = 0.$$

- $\mathrm{Gap}_\epsilon\mathrm{MCSP} \coloneqq (\mathrm{Gap}_\epsilon\mathrm{MCSP}_{\mathrm{YES}}, \mathrm{Gap}_\epsilon\mathrm{MCSP}_{\mathrm{NO}})$

$$\mathrm{Gap}_\epsilon\mathrm{MCSP}_{\mathrm{YES}} = \{f \mid \mathrm{CC}(f) \le 2^{\epsilon n}\}$$

$$\mathrm{Gap}_\epsilon\mathrm{MCSP}_{\mathrm{NO}} = \left\{f \;\middle|\; \mathrm{CC}(f) \ge 2^{(1-\epsilon)n}\right\}$$

- 事実：ある定数$\epsilon > 0$について、$\mathrm{Gap}_\epsilon\mathrm{MCSP} \in \mathrm{P}$ $\Leftrightarrow$ ある定数$\epsilon > 0$と多項式時間アルゴリズムAが存在し、$\mathrm{CC}(f) \le A(f) \le \mathrm{CC}(f) \cdot 2^{(1-\epsilon)n}$ を満たす。

証明のアイディア

- “Hardness versus Randomness”の枠組みを**メタ的**に適用する。
- ほとんど全ての疑似乱数生成器の構成は**「ブラックボックス」**である。

[Impagliazzo-Wigderson’97]

任意の計算困難な関数 $f \notin \mathrm{SIZE}\left(2^{0.01n}\right)$ $\mapsto$ 疑似乱数生成器 $\mathrm{IW}^f : \{0,1\}^{O(\log N)} \to \{0,1\}^N$

  - このような写像$\mathrm{IW}^{(\cdot)}$を**疑似乱数生成器の構成**と呼ぶ。

- 実は、疑似乱数生成器の構成が平均時・最悪時帰着になっている。

“Hardness versus Randomness” をメタ的に適用

- 伝統的な “Hardness versus Randomness” の枠組みでは

仮定：**固定された** 計算困難関数 $f \in \mathbf{E} \setminus \text{io-SIZE}(2^{0.01n})$ $\mapsto$ 疑似乱数生成器 $G^f : \{0,1\}^{O(\log N)} \to \{0,1\}^N$

- この枠組みを**メタ計算量的**に解釈する。
  - fを**MCSP**の**入力**だとする。
  - $f \mapsto G^f(z)$ をMCSPに対する乱択帰着だとみなす。

$G^f(z) \in \{0,1\}^N$ を関数$\{0,1\}^{\log N} \to \{0,1\}$の真理値表と同一視する。

NOインスタンス $f \notin \text{SIZE}(2^{0.9n})$ $\mapsto$ $G^f(z) \approx_c \mathcal{U} \notin \text{SIZE}(N^{0.99})$ (w.h.p)

最悪の入力　　**平均的な**（疑似的に一様な）入力

YESインスタンス $f \in \text{SIZE}(2^{0.1n})$ $\mapsto$ $G^f(z) \in \text{SIZE}(N^{0.99})$

平均時多項式時間アルゴリズムをAとして、最悪時多項式時間アルゴリズムBをつくる。

主張：$\left(\mathrm{MCSP}[2^{n/2}], \mathcal{U}\right) \in \mathrm{AvgP} \Longrightarrow \mathrm{Gap}_{\gamma}\mathrm{MCSP} \in \mathrm{BPP}$

補題 [Yao'82, Nisan-Wigderson'94, Carmosino-Impagliazzo-Kabanets-Kolokolova'16]

任意の定数$\gamma > 0$と$m \in \mathbb{N}$に対し、以下のような疑似乱数生成器の構成$G^{(\cdot)}$が存在する。

1. $f:\{0,1\}^n \to \{0,1\}$ を $G^f:\{0,1\}^d \to \{0,1\}^{2^m}$ に写す（ある$d \in \mathbb{N}$に対して）。
2. $\mathrm{CC}(f) \geq 2^{O(m+\gamma n)} + 2^{(1-\gamma)n-1}$ならば、$G^f$は$2^{O(m)}$サイズ回路に対して安全な疑似乱数生成器。
3. 任意のzに対して、$\mathrm{CC}\left(G^f(z)\right) \leq \mathrm{CC}(f) \cdot 2^{\gamma n}$。　（$G^f(z):\{0,1\}^m \to \{0,1\}$と同一視する。）

構成：$G^f(z) \coloneqq \mathrm{NW}^{f^{\oplus k}}(z)$　ただし、$f^{\oplus k}(x) = f(x_1) \oplus \cdots \oplus f(x_k), k = 2^{\Theta(\gamma n)}$.

- 乱択アルゴリズム$B(f)$：$z \sim \{0,1\}^m$ を一様ランダムにとる。$s \leq 2^{\gamma n}$としてよい。

  $B(f) \coloneqq 1 \Leftrightarrow A\left(G^f(z)\right)$が多項式時間で停止しない、または1と出力する。

- $\mathrm{CC}(f) \leq 2^{\gamma n}$のとき、$\mathrm{CC}\left(G^f(z)\right) \leq 2^{\gamma n} \cdot 2^{\gamma n} \leq 2^{m/2}$。ただし、$m \coloneqq 4\gamma n$。特に、$\Pr_B[B(f) = 1] = 1$。
- $\mathrm{CC}(f) \geq 2^{(1-\gamma)n}$のとき、$\Pr_B[B(f) = 0] \approx 1$。（次ページ）

主張：$\mathrm{CC}(f) \geq 2^{(1-\gamma)n}$のとき、$\Pr_B[B(f)=0] \approx 1$。

$\because 2^{O(\gamma n)} + 2^{(1-\gamma)n-1} \leq 2^{(1-\gamma)n}$ （十分小さい$\gamma > 0$をとる）

$$\mathrm{CC}(f) \geq 2^{(1-\gamma)n} \implies \mathrm{CC}(f) \geq 2^{O(\gamma n)} + 2^{(1-\gamma)n-1} \implies G^f(\mathcal{U}_d) \approx_c \mathcal{U}_{2^m}$$

$\because$ 性質2

$\because$ “$\approx_c$”より

$$\Pr_B[B(f)=1] = \Pr_z\left[A'\left(G^f(z)\right)=1\right] \leq \Pr_{\mathcal{U}}[A'(\mathcal{U})=1] + o(1)$$

$A'(x) \coloneqq 1 \overset{\text{定義}}{\Leftrightarrow}$ $A(x)=1$ または
$A(x)$が多項式時間pで停止しない。

$$\Pr_{\mathcal{U}}[A'(\mathcal{U})=1] \leq \Pr[t_A(\mathcal{U}) \geq p] + \Pr[A(\mathcal{U})=1].$$

$\because A$は平均時多項式時間

- $\Pr[t_A(\mathcal{U}) \geq p] \leq \Pr[t_A(\mathcal{U})^\epsilon \geq p^\epsilon] \leq \dfrac{\mathbb{E}[t_A(\mathcal{U})^\epsilon]}{p^\epsilon} \leq \dfrac{2^{O(m)}}{p^\epsilon} \leq o(1).$

  $\because$ Markovの不等式　　$\because$ 十分大きい$p = 2^{O(m)}$をとる。

- $\Pr[A(\mathcal{U})=1] \leq \Pr_{f \sim \{0,1\}^{2^m}}\left[\mathrm{CC}(f) \leq 2^{m/2}\right] \leq o(1).$

  $\because A$は誤りなし　　$\because$ 数え上げの議論（演習問題）

MINKTの最悪時・平均時帰着

- MINKTの場合には、近似精度が大幅に改善できる。

定理 [H. (FOCS 2018, CCC 2020)]

DistNP ⊆ AvgPのもとで、GapMINKT ∈ P。ただし、

$$\mathrm{GapMINKT}_{\mathrm{YES}} = \{(x, 1^t, 1^s) \mid \mathrm{K}^t(x) \leq s\}.$$

$$\mathrm{GapMINKT}_{\mathrm{NO}} = \left\{(x, 1^t, 1^s) \;\middle|\; \mathrm{K}^{\mathrm{poly}(t+|x|)}(x) > s + O(\log(t + |x|))\right\}.$$

- ここでは$O(\log n)$倍近似問題をDistNPに帰着できることを示す。

$$\mathrm{GapMINKT}'_{\mathrm{NO}} = \left\{(x, 1^t, 1^s) \;\middle|\; \mathrm{K}^{\mathrm{poly}(t+|x|)}(x) > s \cdot O(\log |x|)\right\}.$$

- 主張：$\mathrm{DistNP} \subseteq \mathrm{AvgP} \Longrightarrow \mathrm{GapMINKT}' \in \mathrm{BPP}$.

疑似乱数生成器の構成：k直積生成器

補題 (リスト復号 [Sudan'97])

$\mathrm{poly}\left(\frac{n}{\epsilon}\right)$時間計算可能リスト復号可能誤り訂正符号$\mathrm{Enc}\colon \{0,1\}^n \to \{0,1\}^{2^\ell}$と
復号アルゴリズム$\mathrm{Dec}\colon \{0,1\}^{2^\ell} \to (\{0,1\}^n)^{\mathrm{poly}\left(\frac{n}{\epsilon}\right)}$が存在して、以下を満たす。
$\Pr_{i\sim[2^\ell]}[\mathrm{Enc}(x)_i = y_i] \geq \frac{1}{2} + \epsilon$なる$y \in \{0,1\}^{2^\ell}$が与えられたとき、$\mathrm{Dec}(y)$は
リスト$L \subseteq \{0,1\}^n$であって$x \in L$なるものを出力できる。ただし、y_iはyのi番目のビット。

補題 (k直積生成器 [H. STOC'20])

$\mathrm{DP}_k\colon \{0,1\}^n \times \left(\{0,1\}^\ell\right)^k \to \{0,1\}^{\ell k+k}$を、
$\mathrm{DP}_k(x; z_1, \dots, z_k) = \left(z_1, \dots, z_k, \mathrm{Enc}(x)_{z_1}, \dots, \mathrm{Enc}(x)_{z_k}\right)$ と定める。（特に、$\ell = O(\log n)$）
$\mathrm{K}^{\mathrm{poly}(n,t)}(x) \geq k \cdot O(\log n)$ ならば、任意のt時間アルゴリズムTに対して、

$$\Pr_z\left[T\left(\mathrm{DP}_k(x; z)\right) = 1\right] \approx \Pr_{\mathcal{U}}[T(\mathcal{U}) = 1].$$

主張：DistNP ⊆ AvgP ⟹ GapMINKT′ ∈ BPP

- $L \coloneqq \{(x, 1^t) \mid \mathrm{K}^t(x) \leq |x| - 2\}$ とおく。（MINKTの閾値を$s \coloneqq n - 2$に固定した問題）
- 分布$\mathcal{D} = \{\mathcal{D}_{\langle n,t\rangle}\}_{n,t\in\mathbb{N}}$を、$\mathcal{D}_{\langle n,t\rangle} \equiv (\mathcal{U}_n, 1^t)$として定める。　$\langle\cdot,\cdot\rangle: \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, 効率的全単射
- $(L, \mathcal{D}) \in \mathrm{DistNP} \subseteq \mathrm{AvgP}$より、平均時多項式時間アルゴリズム$A$を得る。
- 乱択多項式時間アルゴリズムBを次のように定める。

$$B(x, 1^t, 1^s; z) \coloneqq 1 \iff A\left(\mathrm{DP}_k(x; z), 1^{t'}\right) = 1 \text{ または 多項式時間}p\text{で停止しない。}$$

ただし、$k = s + \Theta(\log n), t' = t + \mathrm{poly}(n)$。

- BがGapMINKT′を解くことを示す。
- $(x, 1^t, 1^s) \in \mathrm{GapMINKT}_{\mathrm{YES}}$のとき、つまり$\mathrm{K}^t(x) \leq s$のとき、　十分大きい$k = s + O(\log n)$をとる。
  - $\mathrm{K}^{t'}\left(\mathrm{DP}_k(x; z)\right) \leq \mathrm{K}^t(x) + |z| + O(\log n) \leq s + k\ell + O(\log n) \leq k\ell + k - 2.$

$$\implies \left(\mathrm{DP}_k(x; z), 1^{t'}\right) \in L.$$ 従って、Bは受理する。

- $(x, 1^t, 1^s) \in \mathrm{GapMINKT}'_{\mathrm{NO}}$のとき、つまり$\mathrm{K}^{\mathrm{poly}(n,t)}(x) > s \cdot O(\log n) = k \cdot O(\log n)$のとき、

（一般性を失わず$s \geq O(\log n)$としてよい。）

$$\Pr_z\left[A\left(\mathrm{DP}_k(x; z), 1^{t'}\right) = 1 \text{ or } t_A\left(\mathrm{DP_k(x; z)}, 1^{t'}\right) \geq p\right]$$

$(\because \mathrm{DP}_k(x; z) \approx_c \mathcal{U})$

$$\lesssim \Pr\left[A\left(\mathcal{U}, 1^{t'}\right) = 1 \text{ or } t_A\left(\mathcal{U}, 1^{t'}\right) \geq p\right] \leq \Pr\left[\left(\mathcal{U}, 1^{t'}\right) \in L\right] + \Pr\left[t_A\left(\mathcal{U}, 1^{t'}\right) \geq p\right] \leq \frac{1}{4} + o(1).$$

概要

1. 脱乱択化の理論
   - 理論の概要
   - 疑似乱数生成器の概念
   - 疑似乱数生成器の構成

2. メタ計算量と平均時計算量について
   - メタ計算問題
   - 平均時計算量
   - 最悪時・平均時計算量の同値性
   - 暗号との関係

P ≠ NP予想と暗号の関係

P = NP or **P ≠ NP**

☺ 任意のNPの問題が効率的に解ける

☺ 数学の証明を自動化できる

☹ どのような公開鍵暗号でも効率的に解読可能

☹ ビットコインが無価値になる

「フェルマーの最終定理の証明のうち、
n文字以内で記述できる証明を一つ見つけよ」
という問題は（適切な定式化の下で）NPに属する。

存在する

（例：素因数分解）

☺ 安全な暗号が存在する（？）

公開鍵から秘密鍵を計算できる。
（一方向性関数が存在しないので、
鍵生成の手順を逆方向に計算できる。）

公開鍵暗号方式を使うことにより、
持ち主だけがコインを使えるようにしている

Impagliazzoの５つの可能世界

Russell Impagliazzo

我々の知識と一貫性の
ある世界を5つに分類

Cryptomania

Minicrypt

Pessiland

Heuristica

$$P \neq NP$$

Algorithmica

$$P = NP$$

画像出展：https://www.simonsfoundation.org/people/russell-impagliazzo/

Impagliazzoの５つの可能世界

Russell Impagliazzo

我々の知識と一貫性の
ある世界を5つに分類

Cryptomania

Minicrypt

Pessiland

Heuristica

$P \neq NP$

☺任意のNPの最適化問題が
高速に解ける。
証明が自動化できる。
☹安全な暗号は作れない。

Algorithmica

$P = NP$

画像出展：https://www.simonsfoundation.org/people/russell-impagliazzo/

Impagliazzoの５つの可能世界

Russell Impagliazzo

我々の知識と一貫性のある世界を5つに分類

Cryptomania

$\exists$ 公開鍵暗号

Minicrypt

$\exists$ 秘密鍵暗号 & $\nexists$ 公開鍵暗号

Pessiland

$\mathrm{DistNP} \not\subseteq \mathrm{AvgP}$
（平均時計算量の意味で$\mathrm{P} \neq \mathrm{NP}$）
& $\nexists$ 秘密鍵暗号

Heuristica

$\mathrm{P} \neq \mathrm{NP}$ & $\mathrm{DistNP} \subseteq \mathrm{AvgP}$

Algorithmica

$\mathrm{P} = \mathrm{NP}$

画像出展：https://www.simonsfoundation.org/people/russell-impagliazzo/

Impagliazzoの５つの可能世界

Russell Impagliazzo

Cryptomania

☹難しい問題が存在するが、
☺安全な公開鍵暗号系が
存在する。

∃ 公開鍵暗号

Minicrypt

∃ 秘密鍵暗号 & ∄ 公開鍵暗号

Pessiland

$\mathrm{DistNP} \not\subseteq \mathrm{AvgP}$
（平均時計算量の意味で$\mathrm{P} \neq \mathrm{NP}$）
& ∄ 秘密鍵暗号

Heuristica

$\mathrm{P} \neq \mathrm{NP}$ & $\mathrm{DistNP} \subseteq \mathrm{AvgP}$

Algorithmica

$\mathrm{P} = \mathrm{NP}$

画像出展：https://www.simonsfoundation.org/people/russell-impagliazzo/

Impagliazzoの５つの可能世界

Russell Impagliazzo

我々の知識と一貫性のある世界を5つに分類

Cryptomania

$\exists$ 公開鍵暗号

Minicrypt

$\exists$ 秘密鍵暗号

考えられうる「最悪の世界」

☹暗号が作れない

☹NPも簡単に解けない

（Pessimistic = 悲観的）

Pessiland

$\mathrm{DistNP} \not\subseteq \mathrm{AvgP}$

（平均時計算量の意味で$\mathrm{P} \neq \mathrm{NP}$）

&

$\nexists$ 秘密鍵暗号

Heuristica

$\mathrm{P} \neq \mathrm{NP}$ & $\mathrm{DistNP} \subseteq \mathrm{AvgP}$

Algorithmica

$\mathrm{P} = \mathrm{NP}$

画像出展：https://www.simonsfoundation.org/people/russell-impagliazzo/

Impagliazzoの５つの可能世界

Russell Impagliazzo

Cryptomania

$\exists$ 公開鍵暗号

最小限の暗号が作れる世界
☺安全な秘密暗号方式が存在
☹公開鍵暗号方式は安全でない

Minicrypt

$\exists$ 秘密鍵暗号 & $\nexists$ 公開鍵暗号

Pessiland

$\mathrm{DistNP} \not\subseteq \mathrm{AvgP}$
（平均時計算量の意味で$\mathrm{P} \neq \mathrm{NP}$）
& $\nexists$ 秘密鍵暗号

Heuristica

$\mathrm{P} \neq \mathrm{NP}$ & $\mathrm{DistNP} \subseteq \mathrm{AvgP}$

Algorithmica

$\mathrm{P} = \mathrm{NP}$

画像出展：https://www.simonsfoundation.org/people/russell-impagliazzo/

Impagliazzoの５つの可能世界

Russell Impagliazzo

我々の知識と一貫性のある世界を5つに分類

Cryptomania

∃ 公開鍵暗号

Minicrypt

∃ 秘密鍵暗号 & ∄ 公開鍵暗号

Pessiland

$\mathrm{DistNP} \not\subseteq \mathrm{AvgP}$

（平均時計算量の意味で$\mathrm{P} \neq \mathrm{NP}$）

ヒューリスティクスが上手くいく世界

☺多くの入力でNPが高速に解ける

☹安全な暗号は作れない

Heuristica

$\mathrm{P} \neq \mathrm{NP}$ & $\mathrm{DistNP} \subseteq \mathrm{AvgP}$

（平均時計算量の意味で$\mathrm{P} = \mathrm{NP}$）

Algorithmica

$\mathrm{P} = \mathrm{NP}$

画像出展：https://www.simonsfoundation.org/people/russell-impagliazzo/

Impagliazzoの５つの可能世界

Russell Impagliazzo

Cryptomania

ヨ公開鍵暗号

我々の知識と一貫性の
ある世界を5つに分類

Minicrypt

計算量理論の究極的な使命

我々の世界がどの世界であるかを決定すること！

（特に、Cryptomaniaであるという予想を解決し、
絶対的に安全な暗号を確立すること。）

Heuristica

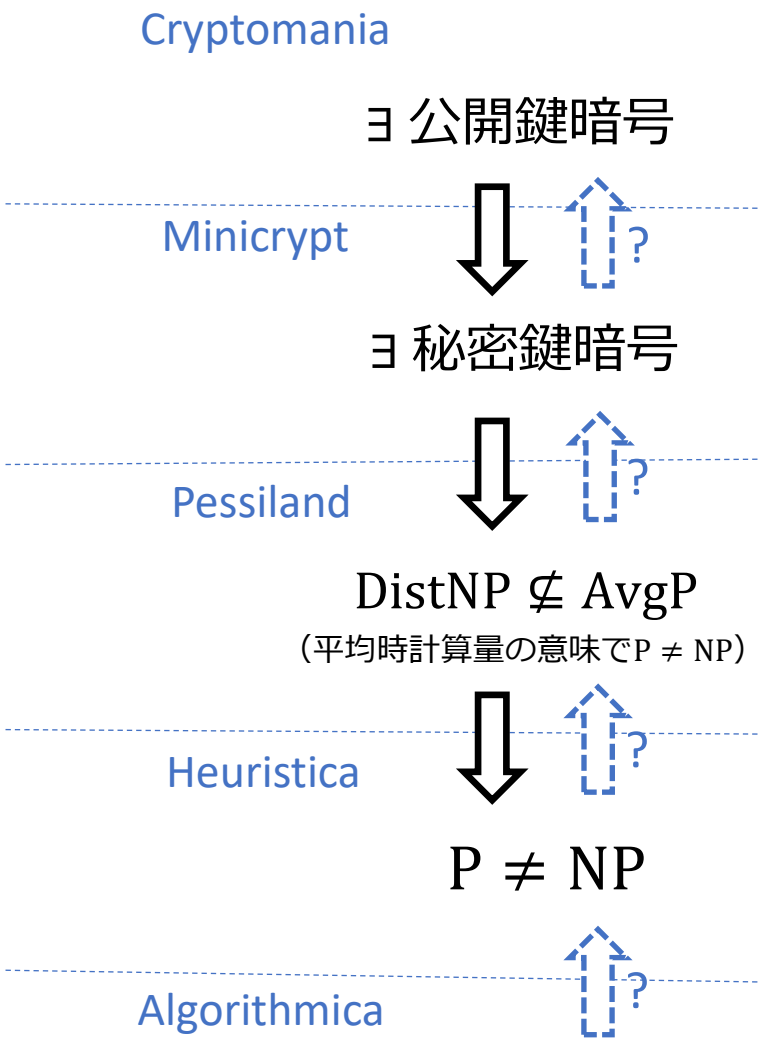
$P \neq NP$ & $DistNP \subseteq AvgP$

Algorithmica

$P = NP$

画像出展：https://www.simonsfoundation.org/people/russell-impagliazzo/

既知の事実と未解決問題



⇨：既知の事実

?⇢：重要な未解決問題

公開鍵暗号構築への道

⇨：既知の事実

?⇢：重要な未解決問題

Cryptomania

ヨ 公開鍵暗号

Minicrypt ⇩ ?⇧

重要な未解決問題

Minicryptを除外できるか？

ヨ 秘密鍵暗号

Pessiland ⇩ ?⇧

重要な未解決問題

Pessilandを除外できるか？

$\mathrm{DistNP} \not\subseteq \mathrm{AvgP}$

（平均時計算量の意味で$\mathrm{P} \neq \mathrm{NP}$）

Heuristica ⇩ ?⇧

重要な未解決問題

Heuristicaを除外できるか？

$\mathrm{P} \neq \mathrm{NP}$

Algorithmica ?⇧

重要な未解決問題

$\mathrm{P} \neq \mathrm{NP}$予想（Algorithmicaを除外できるか？）

全ての矢印を証明

⇔

我々の世界はCryptomania！

１つの矢印を証明

⇔

１つの可能世界を除外

現在の証明手法の限界

⇨：既知の事実

?⇢：重要な未解決問題

Cryptomania

ヨ 公開鍵暗号

Minicrypt

ヨ 秘密鍵暗号

Pessiland

DistNP ⊄ AvgP

（平均時計算量の意味でP ≠ NP）

Heuristica

P ≠ NP

Algorithmica

✕：理論的な障壁

ある種の証明手法では
その未解決問題を解決できない

相対化のバリア [Baker-Gill-Solovay'75]

代数化のバリア [Aaronson-Wigderson'09]

自然な証明のバリア [Razborov-Rudich'97]

局所性のバリア [Chen-H.-Oliveira-Pich-Rajgopal-Santhanam (ITCS'20)]

全ての矢印を証明
⇔
我々の世界はCryptomania！

１つの矢印を証明
⇔
１つの可能世界を除外

現在の証明手法の限界

⇨：既知の事実

?⇢：重要な未解決問題

Cryptomania

∋ 公開鍵暗号

Minicrypt

∋ 秘密鍵暗号

Pessiland

$\mathrm{DistNP} \not\subseteq \mathrm{AvgP}$

（平均時計算量の意味で$\mathrm{P} \neq \mathrm{NP}$）

Heuristica

$\mathrm{P} \neq \mathrm{NP}$

Algorithmica

✕：理論的な障壁

ある種の証明手法では
その未解決問題を解決できない

相対化のバリア

[Impagliazzo (2011)]

ブラックボックス帰着の限界

[Feigenbaum & Fortnow (1993)]
[Bogdanov & Trevisan (2006)]

困難性増幅の不可能性

[Viola (2005)]

全ての矢印を証明
⇔
我々の世界はCryptomania！

１つの矢印を証明
⇔
１つの可能世界を除外

障壁を突破する

Theorem [H. (FOCS 2018)]
GapMCSPの**最悪時**・**平均時**計算量は同値。

- それぞれの障壁は乗り越えられる。

$(\mathrm{MCSP}, \mathcal{U}) \notin \mathrm{AvgBPP} \iff \mathrm{GapMCSP} \notin \mathrm{BPP}$ [H. (FOCS 2018)]

ブラックボックス帰着の限界

[Bogdanov & Trevisan (2006)]

$\mathrm{UP} \not\subseteq \mathrm{DTIME}(2^{O(n/\log n)}) \Rightarrow \mathrm{DistNP} \not\subseteq \mathrm{AvgP}$ [H. (STOC 2021)]

困難性増幅の「不可能性」

相対化のバリア

$\mathrm{NP} \not\subseteq \mathrm{BPP} \Rightarrow \mathrm{GapMCSP}^* \notin \mathrm{BPP}$ [H. (FOCS 2022)]

- 上手く組み合わせることができれば、原理的にはHeurisitcaが除外できる。

MCSP^* (Partial MCSP; 部分関数版回路最小化問題)

入力

- 部分関数 $f:\{0,1\}^n \to \{0,1,*\}$ の真理値表

（$f(x) =*$ なる入力xは定義されていないとみなす。）

- 閾値$s \in \mathbb{N}$

出力

ゲート数s以下の論理回路で
入力$x \in f^{-1}(\{0,1\})$において
関数$f(x)$を出力するものが
存在するかどうか。

YES

（$\because C(x_1, x_2) = x_1$なる回路はゲート数0で$f$を計算できる。）

例

- $s = 0$
- $f =$

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	*
1	0	1
1	1	*

定理 [H. (FOCS 2022)]

（乱択多項式時間帰着の下で）MCSP^*はNP完全。

- 招待講演：Oxford & Warwick (8月), Columbia (9月), UT Austin (11月), Oxford (11月) Simons Institute (1月)
- Complexity result of the year 2022 受賞

MINKT* （部分関数版MINKT）

入力

- 部分文字列$x \in \{0,1,*\}^n$
- 閾値$s \in \mathbb{N}$
- 時間パラメタ$t \in \mathbb{N}$
  （一進法で記述する）

出力

ある文字列 $y \in \{0,1\}^n$ であって、
xと一貫性をもち、

$$\mathrm{K}^t(y) \leq s$$

となるものが存在するか？

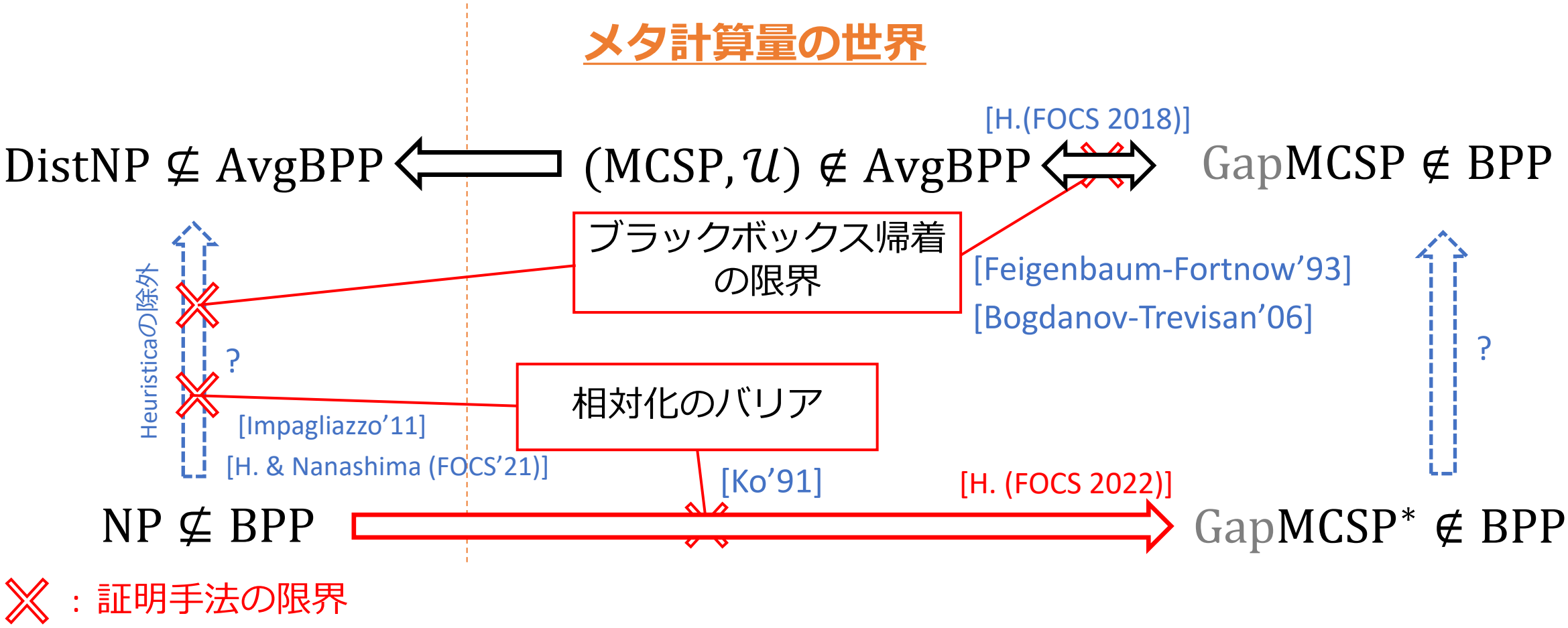
Theorem [H. FOCS 2022]

GapMINKT*は乱択多項式時間帰着の下でNP完全。

$(\log n)^{\Omega(1)}$倍近似問題

➢ これをGapMINKTに拡張することができれば、Heuristicaを除外できる。

次のステップ：相対化のバリアの先へ



後半の参考文献

- 平均時計算量のサーベイ

Andrej Bogdanov, Luca Trevisan, "Average-Case Complexity", Foundations and Trends in Theoretical Computer Science, 2006

- メタ計算量のサーベイ

Shuichi Hirahara, "Meta-Computational Average-Case Complexity: A New Paradigm Toward Excluding Heuristica", Bull. EATCS, 2022

演習問題リスト

1. 回路の上界
2. 回路とコルモゴロフ記述量
3. 直積生成器の安全性
4. 組合せデザインの存在
5. 貪欲法に基づく組合せデザインの構成
6. 多項式に基づく組合せデザインの構成
7. Nisan-Wigderson疑似乱数生成器
8. AveMCSPに対する最悪時・平均時帰着

   --- 以下はおまけ ---
9. アダマール符号の復号
10. 誤りなしヒューリスティクスと平均時多項式時間アルゴリズム
11. 平均時多項式時間のロバスト性
12. GapMCSPの特徴付け

前書き：確率集中不等式（Chernoff bound）

以下のよく知られた確率集中不等式および非復元抽出への拡張を使ってよい。

数列$\Omega = (\Omega_1, \cdots, \Omega_N) \in \{0,1\}^N$から復元抽出で$n$個選んだものを$X_1, \ldots, X_n$とする。
（すなわち、各$X_1, \ldots, X_n \in \{0,1\}$はベルヌーイ独立同一分布に従う。）

$\mu \coloneqq \mathbb{E}[\sum_{i=1}^{n} X_i]$ とすると、任意の$t > 0$に対して、以下が成立する。

$$\Pr\left[\sum_{i=1}^{n} X_i \geq t\right] \leq \exp\left(-t\left(\ln\left(\frac{t}{\mu}\right) + \frac{\mu}{t} - 1\right)\right).$$

同様に、Ωから非復元抽出でn個選んだものを$Y_1, \ldots, Y_n$とすると、

$$\Pr\left[\sum_{i=1}^{n} Y_i \geq t\right] \leq \exp\left(-t\left(\ln\left(\frac{t}{\mu}\right) + \frac{\mu}{t} - 1\right)\right).$$ [Hoeffding 1963]

演習問題１：回路の上界

任意のブール値関数$f:\{0,1\}^n \to \{0,1\}$がサイズ$O(2^n \cdot n)$の回路で
計算できることを証明せよ。

※ サイズ$O\left(\frac{2^n}{n}\right)$の回路で計算できることが知られている。[Lupanov 1958]

演習問題２：回路とコルモゴロフ記述量

以下の事実を証明せよ。

1. あるチューリング機械Mが存在して、任意の$f \in \{0,1\}^{2^n}$について、
$$\mathrm{K}_M(f) = \Theta(\mathrm{CC}(f)\log(\mathrm{CC}(f)+n)).$$

2. 任意の機械Mについて、$\Pr_{x\sim\{0,1\}^N}[\mathrm{K}_M(x) \geq n-2] \geq \frac{3}{4}$.

3. 一様ランダムなブール値関数$f:\{0,1\}^n \to \{0,1\}$を取ると、
高い確率で$\mathrm{CC}(f) \geq \Omega\left(\frac{2^n}{n}\right)$となる。

演習問題３：直積生成器の安全性

$\left(\frac{1}{2}-\epsilon\right)$の誤りを訂正できるリスト復号可能誤り訂正符号$\mathrm{Enc}\colon\{0,1\}^n \to \{0,1\}^{2^\ell}$であって、
$\ell = O\left(\log\left(\frac{n}{\epsilon}\right)\right)$となるようなものをとる。

$k \le n$に対して、$\mathrm{DP}_k\colon\{0,1\}^n \times \left(\{0,1\}^\ell\right)^k \to \{0,1\}^{\ell k+k}$を、
$\mathrm{DP}_k(x; z_1, \dots, z_k) = \left(z_1, \dots, z_k, \mathrm{Enc}(x)_{z_1}, \dots, \mathrm{Enc}(x)_{z_k}\right)$ と定める。

文字列xのDオラクル時間制限付きコルモゴロフ記述量とは、
最小のDオラクル付きプログラムの長さをいう。正確には、

$$\mathrm{K}^{t,D}(x) \coloneqq \min\{\, |d| : U^D(d)\text{が}t\text{時間以内で}x\text{を出力} \,\}$$

と定義される。

仮に関数$D\colon\{0,1\}^{\ell k+k} \to \{0,1\}$が$\mathrm{DP}_k(x;\cdot)$を$\epsilon$-識別するとき、十分大きい$t = \left(\frac{n}{\epsilon}\right)^{O(1)}$について、

$$\mathrm{K}^{t,D}(x) \le k\ell + k + O\left(\log\left(\frac{n}{\epsilon}\right)\right)$$

であることを証明せよ。

演習問題３の補遺

特に、対偶をとると、
$\mathrm{K}^{t,D}(x) > k\ell + k + O(\log(n/\epsilon))$ ならば $\left|\Pr_{z}\left[D\left(DP_k(x;z)\right)=1\right] - \Pr_{\mathcal{U}}[D(\mathcal{U})=1]\right| \leq \epsilon$
となる。

実は$\mathrm{E} \not\subseteq \bigcap_{\alpha>0} \mathrm{io\text{-}SIZE}(2^{\alpha n})$の下で、$k\ell + k$ を kまで改善できる。具体的には：

任意の$\mathrm{poly}\left(\frac{n}{\epsilon}\right)$サイズ回路$D\colon \{0,1\}^{k\ell+k} \to \{0,1\}$に対して、
もしDが$\mathrm{DP}_k(x;\cdot)$をϵ-識別するのであれば、

$$\mathrm{K}^{t,D}(x) \leq k + O(\log(n/\epsilon)).$$

さらに、$\mathrm{DistNP} \subseteq \mathrm{AvgP} \implies \mathrm{E} \not\subseteq \bigcap_{\alpha>0} \mathrm{io\text{-}SIZE}(2^{\alpha n})$ である。[Buhrman-Fortnow-Pavan’05]
これを用いて、$\mathrm{DistNP} \subseteq \mathrm{AvgP}$ ならば $\mathrm{GapMINKT} \in \mathrm{P}$を示せる。[H. CCC’20]

演習問題４：組合せデザインの存在

(d, ℓ, ρ)-組合せデザイン（combinatorial design）とは、
$[d] = \{1, \ldots, d\}$上の集合族$\{S_1, \ldots, S_m\}$であって、
以下の条件を満たすものをいう。

- 任意の$i \in [m]$について、$|S_i| = \ell$
- 相異なる$i, j \in [m]$について、$\left|S_i \cap S_j\right| \leq \rho$

$d = \frac{\ell^2}{\rho} \cdot \exp\left(1 + \frac{2 \ln m}{\rho}\right)$のとき、
(d, ℓ, ρ)-組合せデザインが存在することを証明せよ。

演習問題５：貪欲法に基づく組合せデザインの構成

次のアルゴリズムによって(d, ℓ, ρ)-組合せデザイン$S_1, \dots, S_m$を構築する。

For $i = 1, \dots, m$:
    For $S_i \in \binom{[d]}{\ell}$:
        If $|S_i \cap S_j| \leq \rho$ for every $j \in \{1, \dots, i-1\}$:
            break　　# 次のiへ
Output $(S_1, \dots, S_m)$

任意の定数$\gamma > 0$に対し、ある定数$c \in \mathbb{N}$が存在し、

$$d = c\ell, \rho = \gamma\ell, m = 2^{\ell}$$

としたとき、上述のアルゴリズムは$2^{O(\ell)}$時間で
(d, ℓ, ρ)-組合せデザインを正しく計算することを示せ。

演習問題６：多項式に基づく組合せデザインの構成

ℓを素数として、体$\mathbb{Z}_\ell$上のρ次1変数多項式$g \in \mathbb{Z}_\ell[X]$を考えると、
根は高々ρ個である。この事実を用いて、(d, ℓ, ρ)-組合せデザインを構成しよう。

$g_1, \ldots, g_{2^{\rho+1}} \in \mathbb{Z}_\ell[X]$ を係数が0か1のρ次多項式を辞書順に並べたものとする。
各多項式g_iを用いて集合$S_i \subseteq [d]$を上手く定義することにより、
$(d = O(\ell^2), \ell, \rho)$-組合せデザイン$S_1, \ldots, S_{2^{\rho+1}}$を構成せよ。

演習問題７：Nisan-Wigderson疑似乱数生成器

(d, n, ρ)-組合せデザイン$S_1, \dots, S_m$を固定する。
$h: \{0,1\}^n \to \{0,1\}$に対し、$\mathrm{NW}^h(z)$は次のように定義される。

$$\begin{array}{ccc} \mathrm{NW}^h: \{0,1\}^d & \to & \{0,1\}^m \\ \cup & & \\ z & \mapsto & h(z_{S_1}) \cdots h(z_{S_m}) \end{array}$$

ここで、大きさnの集合$S = \{i_1 < \cdots < i_n\} \subseteq [d]$に対し、$z_S \coloneqq z_{i_1} \cdots z_{i_n}$と定める。

以下を証明せよ。

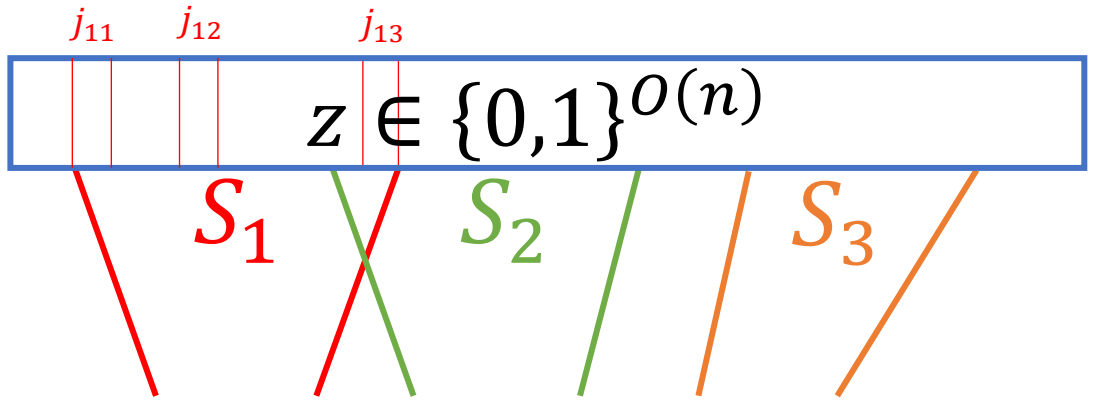
1. サイズmの回路Dが$\mathrm{NW}^h(\cdot)$をδ-識別するならば、
$$h \in \widetilde{\mathrm{SIZE}}\left(O(m \cdot 2^\rho \cdot \rho); \frac{\delta}{m}\right).$$
2. $\mathrm{E} \not\subseteq \bigcap_{\epsilon>0} \mathrm{io}\text{-}\widetilde{\mathrm{SIZE}}(2^{\epsilon n}; 2^{-\epsilon n}) \implies \mathrm{P} = \mathrm{BPP}.$

Nisan-Wigderson疑似乱数生成器 NW^h

- 計算困難関数$h: \{0,1\}^n \to \{0,1\}$
- 疑似乱数生成器$\mathrm{NW}^h: \{0,1\}^{O(n)} \to \{0,1\}^{2^{\epsilon n}}$

相異なる$i \neq j \in \{1, \dots, m\}$に対して
$S_i \subseteq \{1, \dots, O(n)\}, |S_i \cap S_j| \leq \epsilon n.$

乱数の種 z



$S_1 = \{j_{11} < j_{12} < j_{13}\}$

z_{S_1} z_{S_2} z_{S_3} z_{S_4} …

$\in \{0,1\}^n$

$z_{S_1} = z_{j_{11}} z_{j_{12}} z_{j_{13}}$

$(n = 3)$

↧ ↧ ↧

出力 $h(z_{S_1})$ $h(z_{S_2})$ $h(z_{S_3})$ … $=: \mathrm{NW}^h(z) \in \{0,1\}^{2^{\epsilon n}}$

$\in \{0,1\}$

演習問題８：AveMCSPの最悪時・平均時帰着

多項式pに対して、約束付き問題$\mathrm{Ave}_p\mathrm{MCSP}$を次のように定める。

$$\mathrm{Ave}_p\mathrm{MCSP}_{\mathrm{YES}} \coloneqq \{(f,s) \mid f \in \mathrm{SIZE}(s)\} = \{(f,s) \mid \mathrm{CC}(f) \leq s\},$$

$$\mathrm{Ave}_p\mathrm{MCSP}_{\mathrm{NO}} \coloneqq \left\{(f,s) \;\middle|\; f \notin \widetilde{\mathrm{SIZE}}\left(p(s,n); \frac{1}{p(s,n)}\right)\right\}.$$ （ただし、$f \in \{0,1\}^{2^n}$）

$\mathrm{DistNP} \subseteq \mathrm{AvgP}$ ならば、ある多項式pが存在して、
$\mathrm{Ave}_p\mathrm{MCSP} \in \mathrm{P}$であることを証明せよ。

演習問題９：アダマール符号の局所復号

アダマール符号$\mathrm{Had}: \{0,1\}^n \to \{0,1\}^{2^n}$は、任意の$r \in \{0,1\}^n$について

$$\mathrm{Had}(x)_r := \left(\sum_{i=1}^{n} x_i r_i\right) \bmod 2$$

として定義される。

問：オラクル付き乱択多項式時間アルゴリズムDecであって、
任意の$f \in \{0,1\}^{2^n}$であって

$$\Pr_{r \sim \{0,1\}^n}[f(x)_r = \mathrm{Had}(x)_r] > 0.76$$

なるものに対し、

$$\Pr_{\mathrm{Dec}}\left[\mathrm{Dec}^f(1^n) = x\right] \geq 0.999$$

であるDecを構成せよ。

※ アダマール符号は$\left(\frac{1}{2} - \epsilon\right)$の誤りから局所リスト復号可能であることが知られている。[Goldreich-Levin'89]

演習問題１０：誤りなしヒューリスティクス

- 平均時多項式時間AvgPは**誤りなしヒューリスティクス**を用いて特徴づけることができる。 (errorless heuristics)

事実 次は同値。

1. $(L, \mathcal{D}) \in \mathrm{AvgP}$
2. あるアルゴリズムAであって、以下の条件を満たすものが存在する。
   （A: 誤りなしヒューリスティクス）
   - (a) $A(x;\delta,n) \in \{L(x), \text{“失敗”}\}$（アルゴリズムは正しい答えを返す or 失敗）
   - (b) $\Pr_{x\sim\mathcal{D}_n}[A(x;\delta,n) = \text{"失敗"}] \leq \delta$ （アルゴリズムの失敗確率はδ以下）
   - (c) $t_A(x;\delta,n) \leq \mathrm{poly}(n/\delta)$（アルゴリズムの実行時間は多項式時間）

アルゴリズムの失敗
⇔
計算時間が超多項式時間
（計算を打ち切る）

演習問題１１：平均時多項式時間のロバスト性

ある関数$t: \{0,1\}^n \to \mathbb{N}$であって、

1. $\mathbb{E}_{x\sim\{0,1\}^n}[t(x)] \leq n$ であるが、
2. $\mathbb{E}_{x\sim\{0,1\}^n}[t(x)^2] \geq 2^n$ となるもの

が存在することを証明せよ。

※ 従って、期待多項式時間を$\mathbb{E}_{x\sim\{0,1\}^n}[t(x)] \leq n^{O(1)}$として定義すると、
計算モデル（チューリング機械やRAM）に依存した定義になってしまう。

演習問題１２：GapMCSPの特徴付け

以下の条件の同値性を示せ。

1. ある定数$\epsilon > 0$について、$\mathrm{Gap}_{\epsilon}\mathrm{MCSP} \in \mathrm{P}$。

2. ある定数$\epsilon > 0$と多項式時間アルゴリズムAが存在し、
$$\mathrm{CC}(f) \leq A(f) \leq \mathrm{CC}(f) \cdot 2^{(1-\epsilon)n}$$
を満たす。

あとがき：暗号的疑似乱数生成器との関係

- $\approx_c$の記号は非標準的な使い方であることを注意しておく。 $X \approx_c Y$は暗号の文脈でよく使われる。

- 任意の多項式サイズ回路についてXとYが識別不可能なとき、$X \approx_c Y$と書く。
  （線形サイズ回路に対する識別不可能性ついては、普通この記号を用いない。）

- 暗号学的疑似乱数生成器Gは、
  $$(\text{統計的テストの計算時間}) \gg (G\text{の計算時間})$$
  であるようなもの。一方向性関数の存在と同値 [Hastad-Impagliazzo-Levin-Luby'99]。

- 本講演で扱ったものは計算量理論的疑似乱数生成器Gで、
  $$(\text{統計的テストの計算時間}) \ll (G\text{の計算時間})。$$