ランダム性と計算量

令和5年11月24日

河村

http://www.kurims.kyoto-u.ac.jp/~kawamura/t/kyushu/

スライドURL→

概要

計算中にランダムな選択を行うことで高確率で効率的に正解を得る手法を乱択といい、多くのアルゴリズムで使われている。乱択が効率的に問題を解く上でどれほど本質的なのか（決定的に作った数列を乱数の代りに用いてはダメなのか）、という問は計算量理論の大きなテーマの一つであり、多項式時間計算など幾つかの重要な意味においては未解決である（BPP＝P予想）。この問がどのように定式化され、他の困難さ予想とどのように関わるのか解説する。

キーワード：計算量、多項式時間、乱択、脱乱択、擬乱数生成

スライドURL→

0.
準備
（問題・機械・計算量
特に多項式時間）

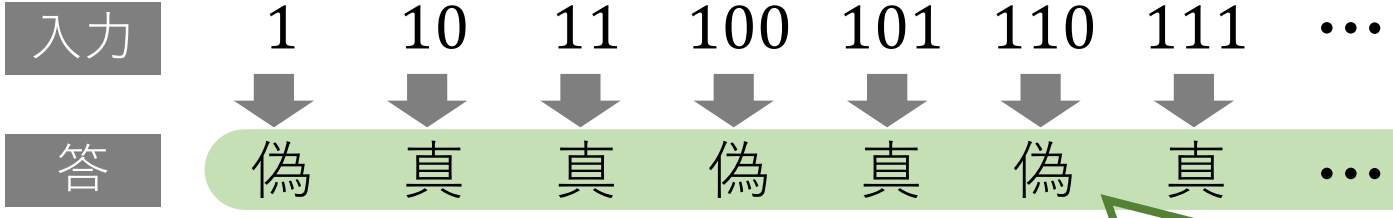
スライドURL→

問題とは
(problem)

各入力(input)に対し
真（1）か偽（0）かを定めたもの
（入力は $\{0,1\}$ 上の文字列で表される）

例えば

問題 与えられた正整数（二進法で書く）が素数か答えよ



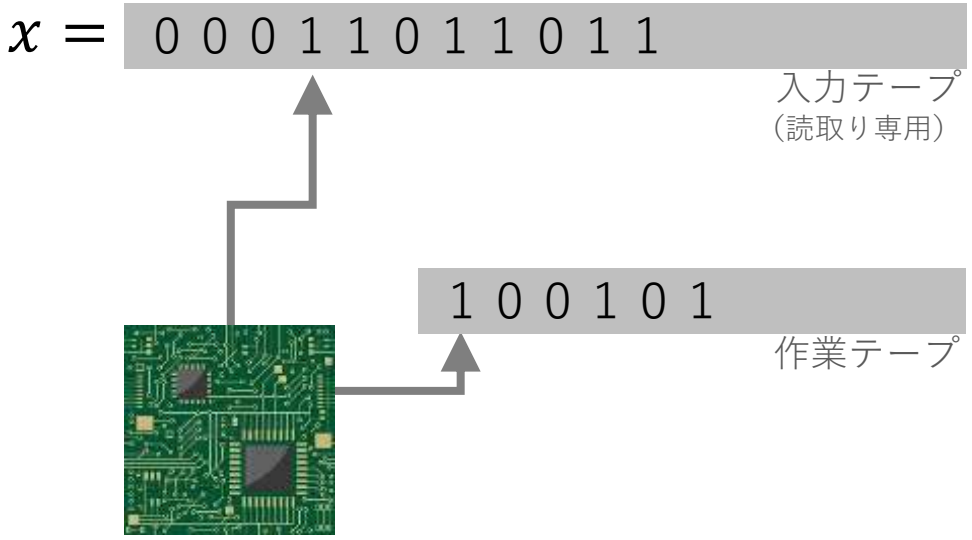
問題とは
コレ全体のこと！

個々の「110」などは「入力」と呼ぶ

このように答として 1 か 0 かを求める
問題を**判定問題**(decision problem)という（今日は主にこれだけを考える）
真となる入力全体の集合（**言語**(language)）と考えても同じ

スライドURL→

チューリング機械 (Turing machine)　（＝アルゴリズム）



機械は各時点で状態
（有限集合 Q の元）をもつ

現在の状態と読んでいる文字から
状態を変え　文字を書込み
各テープ上の注目点を一歩動かす

これを決める遷移規則 δ:

$$Q \times \{0,1\}^2 \to Q \times \{0,1\} \times \{\text{左}, \text{右}\}^2$$

計算結果

$M(x)$

受理（1）か
不受理（0）

(running time)
時間量が $p\colon \mathbf{N} \to \mathbf{N}$ とは　**任意の x について**
$p(|x|)$ 時間以内で停止すること
p を多項式とできるとき
M は**多項式時間**機械という
(polynomial time)

スライドURL→

定義

言語 A が(class)級 **P** に属するとは
或る多項式時間機械 M が存在し
任意の入力 $x \in \{0,1\}^*$ に対し

$x \in A$ のとき $M(x) = 1$ (受理)

$x \notin A$ のとき $M(x) = 0$ (不受理)

(Church-Turing thesis)
拡張チャーチ・チューリングのテーゼ

P に属する ＝ まあ大体 「現実に解ける」

スライドURL→

何故「多項式以内か否か」が重要か

■入力が大きくなると手間が大違い

入力長 $n=$	10	30	50	100	1000	1万	100万	1億
$\sqrt{n}$	1秒以内	1秒以内	1秒以内	1秒以内	1秒以内	1秒以内	1秒以内	1秒以内
n	1秒以内	1秒以内	1秒以内	1秒以内	1秒以内	1秒以内	1秒	2分
n^2	1秒以内	1秒以内	1秒以内	1秒以内	1秒	2分	12日	3百年
n^3	1秒以内	1秒以内	1秒以内	1秒	17分	12日	3万年	3百億年
2^n	1秒以内	18分	36年	4京年				
$n!$	4秒	8百京年						

多項式時間

指数時間

1秒に1000000回の処理ができるとしたときにかかる時間

■指数個（以上）の組合せから何かを探す場面は多い（組合せ爆発）

例えば 問題 SAT 与えられた命題論理式（n 個の命題変数をもつ）が充足可能か判定

- すべての割当（2^n 個）を試すという愚直な方法は指数時間かかる
- n の多項式時間で解く方法は多分ない（**NP** 完全問題）

スライドURL→

1.
乱択算法と計算量
（**P** と **BPP**）

スライドURL→

乱択

乱数を利用した算法

例 与えられた（多変数の）整数係数多項式が零か判定

$$
\begin{aligned}
&(z-x)(x+y)(yy+zz)-xxyz\\
&+xy(z+x)(y-z)\\
&-(x-z)(xx+xy-yz)(x+y-z)\\
&+xyyz+(x+y)y(z-x+y)(x+y-z)\\
&-(x-z)(xy+xz+yz)(x+y-z)\\
&+(x-y)(x+z)(y+z)(y-z)+yyzz
\end{aligned}
$$

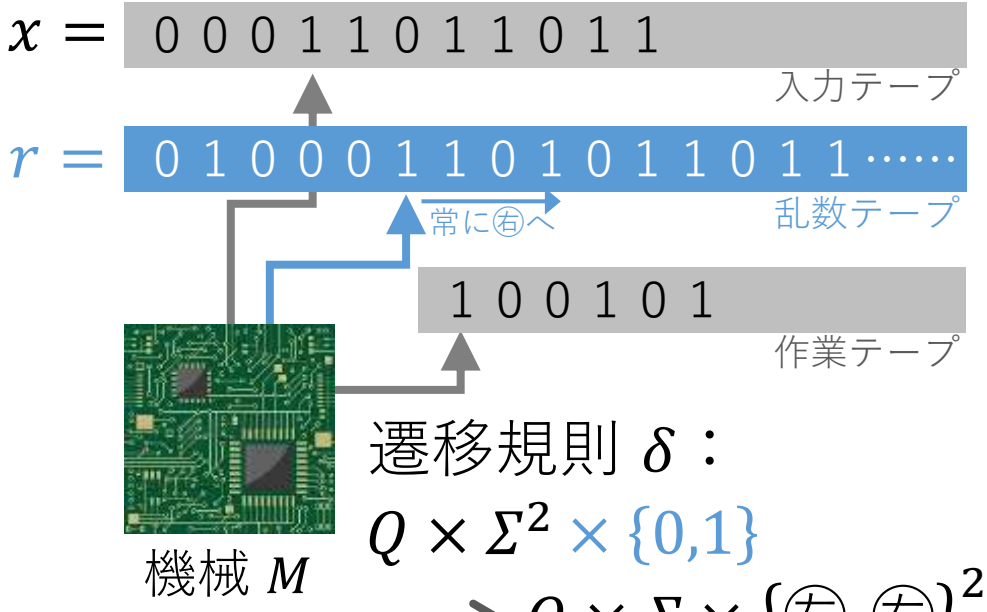
文字式のまま全部展開して計算 ➡ 大変（多項式時間でない）

適当に数値を代入して計算し
0 になるか調べる
$(x,y,z)=(1,2,3)$ とか

➡ ラク（高速・単純）
高確率で正解

スライドURL→

非決定性（＝乱択）機械



遷移規則 δ：

$Q \times \Sigma^2 \times \{0,1\}$

$\longrightarrow Q \times \Sigma \times \{㊧, ㊨\}^2$

次の遷移を二つの分岐から
非決定的に（等確率で）選ぶ

＝

初めに「乱数テープ」上に
乱数列が~~無限に~~供給される

$p(|x|)$ ビットで十分

計算結果

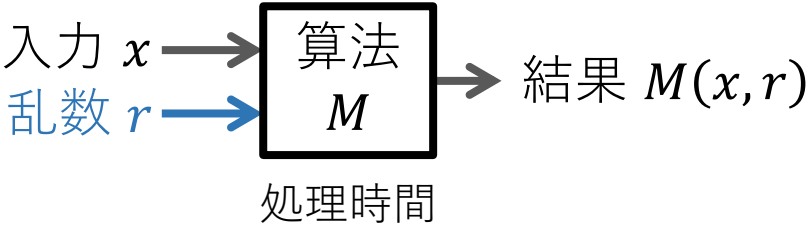
$M(x; r)$ 受理か不受理

入力　乱数　に依存

時間量が $p: \mathbf{N} \to \mathbf{N}$ とは
任意の x と**任意の r について**
$p(|x|)$ 時間以内で停止すること

スライドURL→

乱択算法では
計算結果が入力だけでなく
乱数にも依存するので……



乱数の選び方によって

- 出力が正しかったり正しくなかったりする
  しかし高い確率で正しいなら役に立つ

まずこちらを
考えます

- 計算時間が長かったり短かったりする
  しかし高い確率で短い
  （かかる時間の期待値が小さい）なら役に立つ

スライドURL→

RP

片側誤り
誤受理なし

定義

randomized の R

言語 A が級 **RP** に属するとは
或る多項式時間（乱択）機械 M が存在し
任意の入力 x に対し

$x \in A$ のとき $M(x;r)$ は**確率** $> \frac{1}{2}$ **で**受理

$x \notin A$ のとき $M(x;r)$ は**必ず**不受理

※ 非対称な定義であることに注意

明らかに
P ⊆ RP

$\frac{1}{2}$ の代りに $\frac{99}{100}$ にしたければ…
乱数を独立に 7 回取って r_1, …, r_7 とし
$M(x,r_1), \cdots, M(x,r_7)$ どれかが受理したら受理

スライドURL→

先程の例

問題 与えられた整数係数多項式 $p(X_1, \dots, X_m)$ が **非零** であるか判定せよ

この問題が **P** に属するかは未解決だが
次の算法により **RP** には属する（多項式**零**判定問題が **coRP** に属する）

d は p の全次数

算法 数 $r_1, \dots, r_m \in \{1, \dots, 2d\}$ を一様独立に乱択し
$p(r_1, \dots, r_m) \neq 0$ ならば受理

➡ p が零なら確実に不受理
非零なら次の補題により確率 $\geq \frac{1}{2}$ で受理

スライドURL→

補題

全次数 d の非零多項式 p について

$$\Pr[\, p(r_1, r_2, \dots, r_m) \neq 0 \,] \geq 1 - \frac{d}{B}$$

（但し Pr は $r_1, r_2, \dots, r_m \in \{1, \dots, B\}$ を一様独立に選ぶときの確率）

証明

m に関する帰納法　$m = 0$ のとき自明　$m > 0$ とする

X_1 の最高次数を i として　次のように書く

$$p(X_1, X_2, \dots, X_m) = X_1^i \cdot q(X_2, \dots, X_m) + \left(X_1 \text{の低次の項}\right)$$

帰納法の仮定より $\Pr[\, q(r_2, \dots, r_m) \neq 0 \,] \geq 1 - \frac{d-i}{B}$

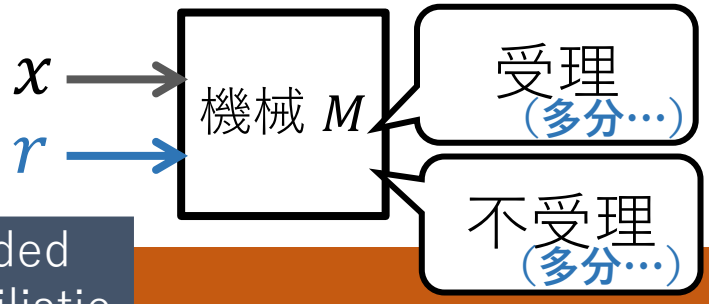
もし□ならば $p(X_1, r_2, \dots, r_m)$ は X_1 に関する i 次の非零な多項式で

その根は高々 i 個　故に

$$\Pr[\square] \geq \Pr[\square] \cdot \left(1 - \frac{i}{B}\right) \geq \left(1 - \frac{d-i}{B}\right)\left(1 - \frac{i}{B}\right) \geq 1 - \frac{d}{B}$$

スライドURL→

BPP



両側誤り

定義

bounded probabilistic

言語 A が級 **BPP** に属するとは
或る多項式時間（乱択）機械 M が存在し
任意の入力 x に対し

$\frac{1}{2}$ と $\frac{1}{2}$ ではダメ

$x \in A$ のとき $M(x;r)$ は**確率** $> \frac{2}{3}$ **で**受理

$x \notin A$ のとき $M(x;r)$ は**確率** $< \frac{1}{3}$ **で**受理

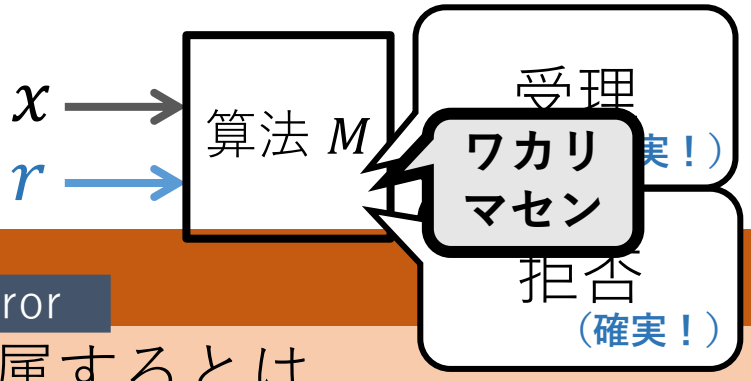
$\frac{2}{3}$ の代りに $\frac{99}{100}$ や $1-\frac{1}{2^{|x|}}$ などにしたければ…

RP ⊆ BPP

乱数を独立に十分な回数取って $r_1, \ldots, r_k$ とし（k は $|x|$ の多項式以内）
$M(x,r_1), \cdots, M(x,r_k)$ の多数決で受理・不受理を決める

スライドURL→

ZPP



無誤り
誤受理なし
誤拒否なし

定義

zero-error

言語 A が級 **ZPP** に属するとは
或る多項式時間（乱択）機械 M が存在し
任意の入力 x に対し

$x \in A$ のとき　$M(x;r)$ は**必ず**受理または「？」

$x \notin A$ のとき　$M(x;r)$ は**必ず**不受理または「？」

「？」の確率は常に $< \frac{1}{2}$

繰返せば $< \frac{1}{100}$ にもできる

決着がつくまで繰返す　→　時間の期待値 $\mathrm{poly}(|x|)$

スライドURL→

定理

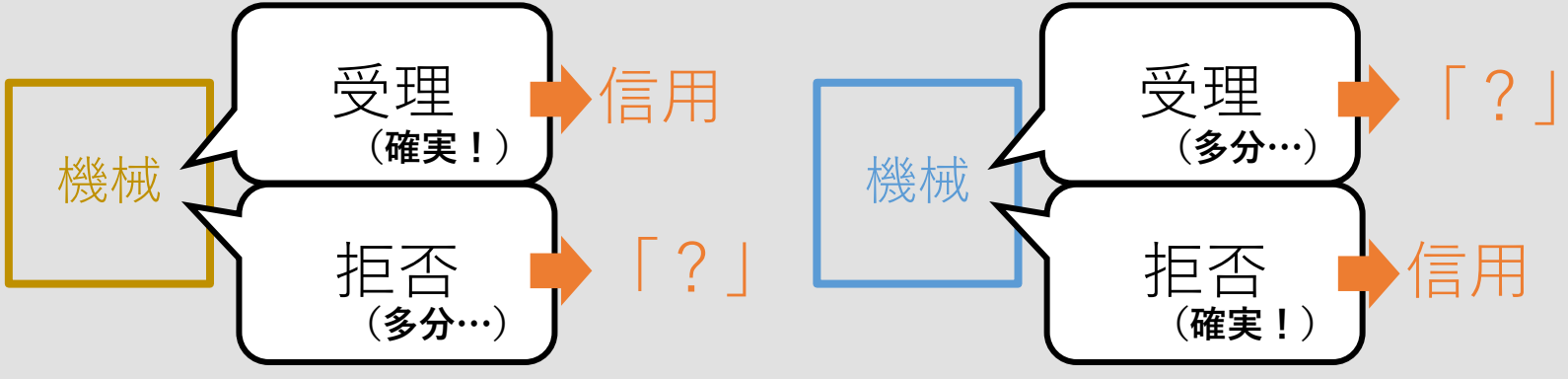
$$\mathbf{ZPP} = \mathbf{RP} \cap \mathbf{coRP}$$

証明

$\mathbf{ZPP} \subseteq \mathbf{RP}$　「？」の代りに拒否

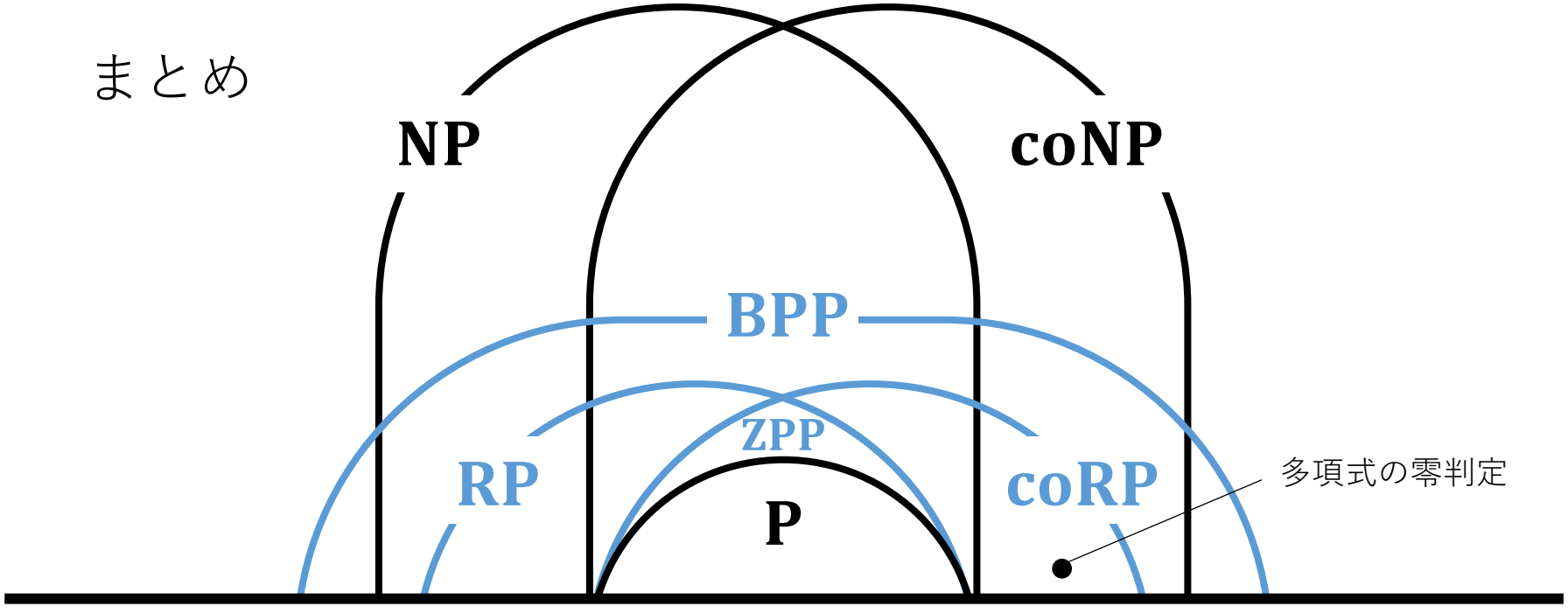
$\mathbf{ZPP} \subseteq \mathbf{coRP}$　「？」の代りに受理

$\mathbf{RP} \cap \mathbf{coRP} \subseteq \mathbf{ZPP}$　両方の機械を実行してみて



スライドURL→

まとめ



「含まれる」の関係を図示したが
等しいか否かは証明されていない
（図中の集合がすべて等しいかもしれない）

未解決 $\mathbf{BPP} \stackrel{?}{=} \mathbf{P}$

（等しいと予想する人が多い）

スライドURL→

2.
乱択は本当に必要なのか
（**BPP＝P** 予想）

スライドURL→

まあ
大体

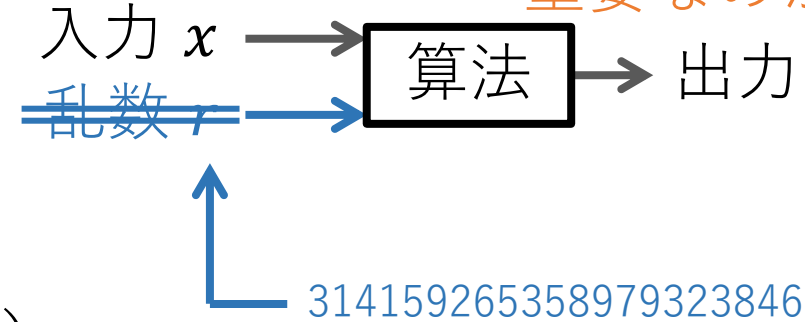
P ＝ 「現実に解ける」

BPP ＝ 「現実に解ける」
（真の乱数があれば）

本当に
異なるのか？

どれほど
重要なのか？

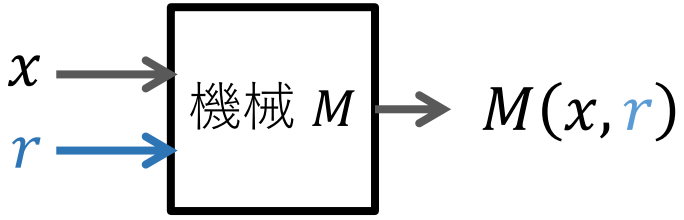
そもそも
乱数って作れるの？



→よくわからない
（計算量理論の未解決問題）

決定的に作られた「擬似乱数」では真の乱数を代用できないのか？

スライドURL→



定義「助言（advice）つき **P**」

言語 A が級 **P/poly** に属するとは
或る多項式時間（乱択）機械 M と
文字列 $r_0, r_1, r_2, \ldots$ が存在し
任意の入力 x に対し

入力長 n に応じた
（多項式長の）助言 r_n を聞けば
すべての $x \in \{0,1\}^n$ で
正解する

$x \in A$ のとき $M(x; r_{|x|})$ は受理

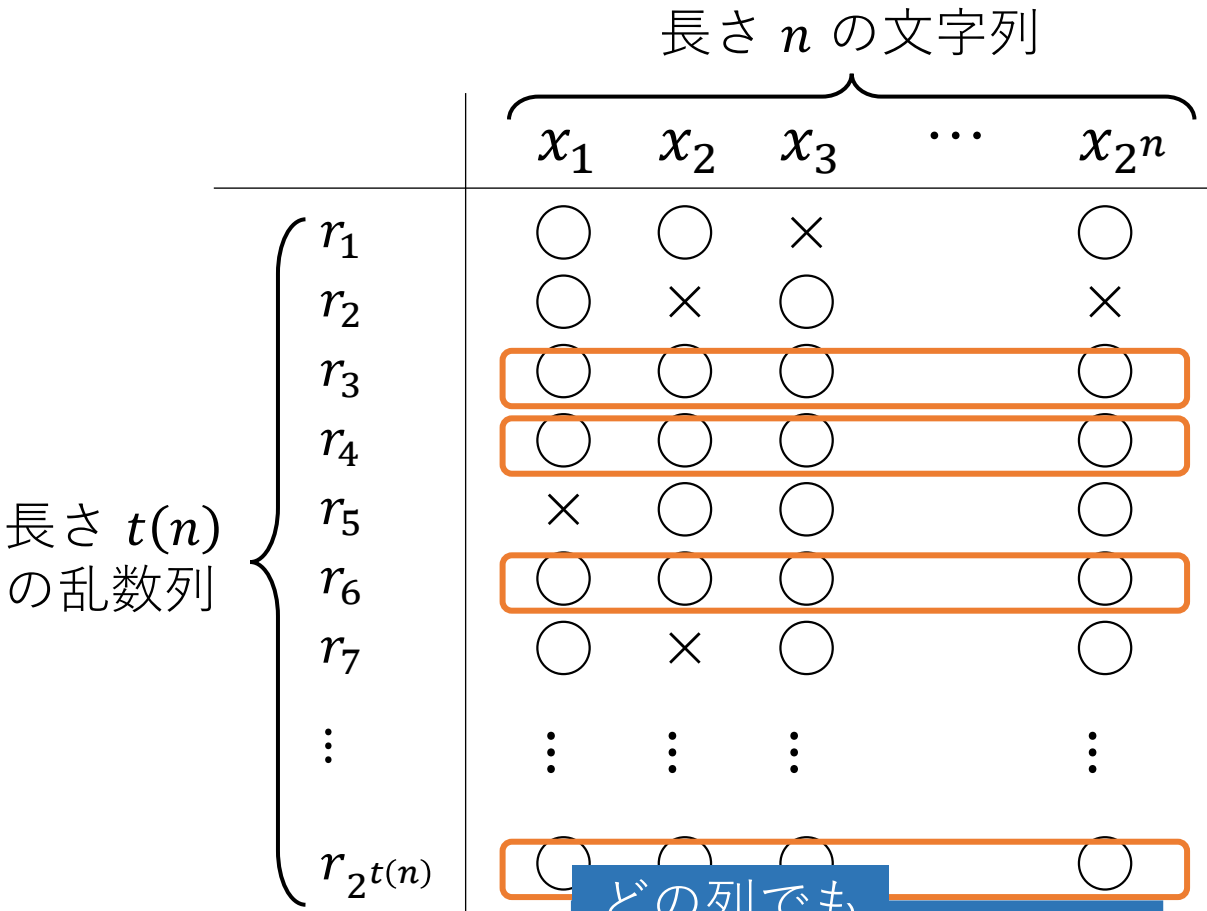
$x \notin A$ のとき $M(x; r_{|x|})$ は不受理

定理［Adleman 1978］（次頁に証明概略）

$$\mathbf{BPP} \subseteq \mathbf{P/poly}$$

スライドURL→

$A \in \mathbf{BPP}$ とすると　多項式時間算法 M が存在して



全部○の行
（長さ n の入力すべてで
M を正解させる乱数）
が存在！

➡ $A \in \mathbf{P/poly}$

BPP は **P** に
かなり近い？
実は等しいかも…？

どの列でも
誤り率 $<\frac{1}{2^{n+1}}$
（$M(x,r) \neq A(x)$ なる r の割合）

「ランダム性は（時間や空間と並んで）計算のための資源」

考察　（乱数の何が重要なのか）

0 と 1 が確率ちょうど $\frac{1}{2}$ で出る乱数であること

どうでもよい。

例えば「確率 $\frac{1}{3}$ で 1 が出る乱数」があれば代用できる。

乱数の各ビットが独立であること

それなりに重要。

前のビットから完全に決ってしまうようでは役立たず。

乱数が十分たくさんあること

それなりに重要。もし $O(\log n)$ ビットしか使わないなら
簡単に「脱乱択」（derandomize）できる
（代りに乱数を全部試しても $2^{O(\log n)}$ =多項式時間なので）。

スライドURL→

定義

G は　長さ k の乱数から
長さ $k+1$ の擬乱数を作る

多項式時間函数 G（出力長は常に 入力長 $+1$ であるとする）が
擬乱生成器（pseudorandom generator）であるとは
任意の多項式 p と多項式時間乱択機械 D に対し
有限個を除くすべての $k \in \mathbf{N}$ と　すべての x について

D は「弁別器」
（distinguisher）

$$\left| \Pr_{s\in\{0,1\}^k}[D(G(s);x)=1] - \Pr_{r\in\{0,1\}^{k+1}}[D(r;x)=1] \right| < \frac{1}{p(k)}$$

D は　入力されたのが
真の乱数 r なのか
擬乱数 $G(s)$ なのか
を判断しようとする

しかし　殆ど区別できない
（助言 x を与えられてさえも）

※　文献によっては弁別器の能力や弁別の定義などに若干の差異あり

擬乱生成器は存在するのか？
➡判っていないが　多分存在するだろうと考えられている

スライドURL→

定義

$l(k) > k$：伸長度

多項式時間函数 G（入力長が k のとき出力長は $l(k)$ とする）が
擬乱生成器（pseudorandom generator）であるとは
任意の多項式 p と多項式時間乱択機械 D に対し
有限個を除くすべての $k \in \mathbf{N}$ と　すべての x について

$$\left| \Pr_{s \in \{0,1\}^k}[D(G(s); x) = 1] - \Pr_{r \in \{0,1\}^{l(k)}}[D(r; x) = 1] \right| < \frac{1}{p(k)}$$

定理

伸長度 $l(k) = k + 1$ の擬乱生成器が存在　$\Longrightarrow$
任意の多項式 $l: \mathbf{N} \to \mathbf{N}$ に対し伸長度 $l(k)$ の擬乱生成器が存在

証明

伸長度 $k + 1$ の生成器を繰返し使って擬乱数を伸ばす（詳細略）

スライドURL→

定義

多項式時間函数 f（入力長＝出力長で全単射とする）が
一方向函数（one-way function）であるとは
任意の多項式 p と多項式時間乱択機械 M に対し
有限個を除くすべての $k \in \mathbf{N}$ と　すべての x について

$$\Pr_{s \in \{0,1\}^k}[\, M(f(s); x) = s \,] < \frac{1}{p(k)}$$

逆像を見つけようとしても

殆ど当らない

一方向函数の存在は $\mathbf{P} \neq \mathbf{NP}$ より強い仮定だが
存在すると多くの人は思っている（暗号の基礎）

定理

一方向函数が存在　$\Longrightarrow$　擬乱生成器が存在

スライドURL→

一方向函数が存在　⟹　擬乱生成器が存在

一方向函数 f から次で定まる G が擬乱生成器であることを示す

$G(s,u) = f(s)\, u\, \langle s,u \rangle$　（$s,u \in \{0,1\}^k$　$\langle s,u \rangle$ は $s_i = u_i = 1$ なる i の個数 mod 2）

$2k$ ビット　　$2k+1$ ビット

証明概略（1/2）

G が弁別器 D によって見破られるとする
すなわち多項式 p が存在し　無限個の k について或る x が存在して

$$\Pr_{s,u\in\{0,1\}^k}[D(G(s,u);x)=1] - \Pr_{r\in\{0,1\}^{2k+1}}[D(r;x)=1] \geq \frac{1}{p(k)}$$

D を用いて次を満す多項式時間機械 D' を作れる

$$\Pr_{s,u\in\{0,1\}^k}[\, D'(f(s)\,u;x) = \langle s,u \rangle\,] \geq \frac{1}{2} + \frac{1}{p(k)}$$

D は入力が $f(s)\,u\,\langle s,u \rangle$ という形である可能性が高いことを（当てずっぽうよりは良く）検出できる

D' は入力 $f(s)\,u$ から $\langle s,u \rangle$ を（当てずっぽうよりは良く）言い当てる

D' から次のような多項式時間機械 M（と多項式 q）を作れる（次頁）

$$\Pr_{s\in\{0,1\}^k}[\, M(f(s);x) = s\,] \geq \frac{1}{q(k)}$$

スライドURL→

D' は $\Pr_{s,u\in\{0,1\}^k}[\, D'(f(s),u;x)=\langle s,u\rangle \,] \geq \frac{1}{2}+\frac{1}{p(k)}$ を満す

それを用いて次のような M を作る　$\Pr_{s\in\{0,1\}^k}[\, M(f(s);x)=s \,] \geq \frac{1}{q(k)}$

証明概略（2/2）

M は入力 $y\in\{0,1\}^k$ を受取り
$l=\mathrm{O}(\log_2 k)$ 個の文字列 $u^{(0)},\dots,u^{(l-1)}\in\{0,1\}^k$ と l 個のビット $\sigma^{(0)},\dots,\sigma^{(l-1)}$ を乱
空でない各 $J\subseteq\{0,\dots,l-1\}$ に対し　$u^{(J)}:=\bigoplus_{j\in J}u^{(j)}$ とし
$\sigma^{(J)}:=\bigoplus_{j\in J}\sigma^{(j)}$ と $D'\left(y,u^{(J)}\oplus \mathrm{e}^{(i)};x\right)$ が一致しているか調べる
一致する方が多ければ $s_i=0$　一致しない方が多ければ $s_i=1$ と答える

すると多項式 q で表される確率 $1/q(k)$ で　各 $\sigma^{(J)}$ は $\langle s,u^{(J)}\rangle$ を言い当てており
このときどの i についても s_i は正しく言い当てられている（計算略）

定理［Haastad, Impagliazzo, Levin, Luby］

一方向函数が存在　⇒　擬乱生成器が存在

※「長さを保つ全単射」という条件は外せる
　また実は逆も言える

スライドURL→

（「擬乱生成器が存在すれば $\mathbf{BPP} = \mathbf{P}$」と言いたい所だが　そこまでは言えないので）

定理

擬乱生成器が存在すれば $\mathbf{BPP} \subseteq \mathbf{SUBEXP} := \bigcap_{\varepsilon>0} \mathbf{Time}\left(2^{n^{\varepsilon}}\right)$

証明概略

$A \in \mathbf{BPP}$（多項式時間乱択機械 M により）とし　$A \in \mathbf{Time}\left(2^{n^{\varepsilon}}\right)$ を示そう

$k: \mathbf{N} \to \mathbf{N}$ と多項式 $l: \mathbf{N} \to \mathbf{N}$ を「$k(n) < \mathrm{o}(n^{\varepsilon})$」かつ
「M が入力長 n に対して使う乱数長は $l\left(k(n)\right)$ 以内」になるように取る
伸長度 l の擬乱生成器 G を仮定より用意する

M と G から次の機械 N を作る

　N は入力 $x \in \{0,1\}^n$ を受取ると
　全 $s \in \{0,1\}^{k(n)}$ について $M\left(x; G(s)\right)$ を求め　その多数決で出力する

すると N は計算時間 $\mathrm{O}\left(2^{n^{\varepsilon}}\right)$ であり　有限個を除くすべての x について

$$N(x) = 1 \Leftrightarrow \Pr_{s\in\{0,1\}^{k(n)}}\left[M\left(x, G(s)\right) = 1\right] > \frac{1}{2} \Leftrightarrow \Pr_{r\in\{0,1\}^{l(k(n))}}[M(x,r)] > \frac{1}{2} \Leftrightarrow x \in A$$

擬乱生成器の定義より

スライドURL→

本講義ではやらないが
より強い意味での擬乱生成器の存在を仮定すると **BPP = P** も言える
これを用いて

定理［Impagliazzo, Wigderson］［Nisan, Wigderson 1988］

級 **E** に属する（＝指数時間計算可能）言語であって
指数的よりも小さい回路で計算できないものが
存在すれば **BPP = P**

「計算の困難さ」を「ランダム性」に転換
（hardness-randomness implication）

スライドURL→