サマースクール数理物理

計算量理論入門
―完全問題と「複雑さ」―

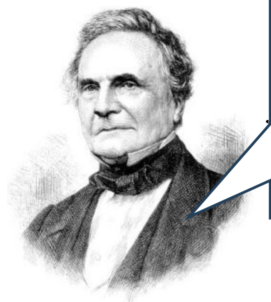
河村彰星

令和7年8月29日～31日

↑スライドはここに置いてあります

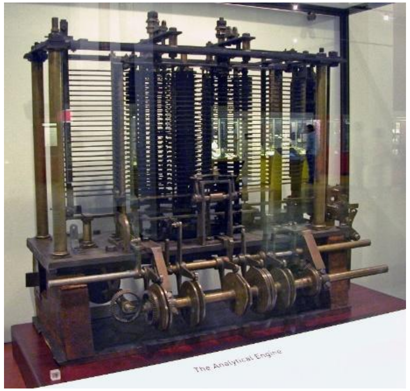
第二日
時間と空間の制限

計算量 (Computational Complexity)



解析機関は、実現すれば必ずや将来の科学に重要なものとなろう。この機械を用いて結果を得ようとするとき問われるのは、**いかなる手順で計算すれば最も短時間で結果に辿り着けるか**である。

— Charles Babbage (1864)



解析機関 (Analytical Engine)

判定可能な問題の間でも「どれほど効率よく判定可能か」による違いが重要

「解析機関」『ウィキペディア日本語版』https://ja.wikipedia.org/w/index.php?title=%E8%A7%A3%E6%9E%90%E6%A9%9F%E9%96%A2&oldid=82653674

計算量の考え方 (原則)　1950〜60年代

- チューリング機械での計算にかかる
  時間 (遷移の回数) や空間 (訪れる欄の数) を考える
  現実にかかる時間や空間をよく表している

- それが入力の長さに応じてどう変るか函数として表す
  「長さ n の入力なら必ず時間 (や空間) が $T(n)$ 以内で済む」
  ような函数 T が計算量の尺度 (最悪時評価)

- その函数の増大の速さに着目
  特に n の多項式以内か否かが重要

定義

機械 M が**多項式時間**であるとは　或る多項式 $p\colon \mathbf{N} \to \mathbf{N}$ が存在し
任意の長さ n の任意の入力に対する M の計算が
時間 $p(n)$ 以内に終る（遷移 $p(n)$ 回以下で停止する）ことをいう

言語 A を認識する多項式時間の機械 M が存在するとき
A は**多項式時間認識可能**である（$A \in \mathbf{P}$）という

n の多項式以内である例

n

$n\sqrt{n}$

$n^2 \log n$

$5n^3 + 4n$

n の多項式以内でない例

2^n

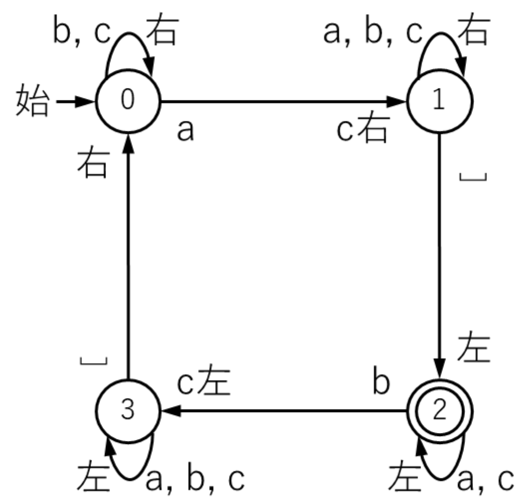
2^{2^n}

1.05^n

$n!$

$n^{\log n}$

例　昨日のこの機械は
時間 $(n+1)^2$ 以内に停止するので
MOREA $\in \mathbf{P}$



定義

機械 M が**多項式時間**であるとは　或る多項式 $p: \mathbf{N} \to \mathbf{N}$ が存在し
任意の長さ n の任意の入力に対する M の計算が
時間 $p(n)$ 以内に終る（遷移 $p(n)$ 回以下で停止する）ことをいう

言語 A を認識する多項式時間の機械 M が存在するとき
A は**多項式時間認識可能**である（$A \in \mathbf{P}$）という

「認識可能」の定義では $x \notin A$ のときは停止しないことを許していたのだから
それに合せて「多項式時間認識」の定義は　長さ n の入力 x について

- $x \in A$ のとき M は x を時間 $p(n)$ 以内に受理する
- $x \notin A$ のとき M は x を時間 $p(n)$ 以内に受理しない

とすべきでは？

そう定義しても
$\mathbf{P}$ の意味は変りません

受理せずに時間 $p(n)$ が経過したら
不受理と確定できるので

多項式時間では「認識可能」と「判定可能」の違いはない
(多項式時間認識可能な言語は補集合も多項式時間認識可能)

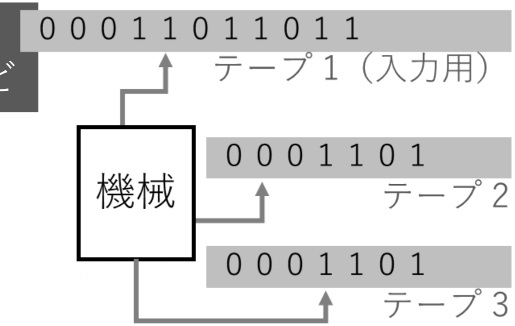
チャーチとチューリングの定立 (続)

「現実的な手間で計算できる」 ＝ (チューリング機械で)多項式時間認識可能

テープの形や読取り部の動き方など

チューリング機械の定義の細部は
計算法が多項式時間であるかないかに
影響を与えない

「$O(n^3)$ であるかないか」など細かい量には
それなりに影響がある



今後は一々チューリング機械を考えず　計算手順が解り易いように説明する

時間＝「基本的な操作の回数」と大雑把に考えてよい

- 四則演算やビットの操作など
- 機械語での命令数

なぜ多項式時間か否かが重要か

■ 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				
10^n	17分	3京年						
$n!$	3.6秒	800京年						

多項式時間

指数時間

1秒に100万回の処理が
できるとしたときにかかる時間

実際には　多項式時間の中での速さの差
(n^2 と n^3 の違いなど) も勿論重要なのですが
この講義では「多項式時間であるかないか」という
より大きな違いに着目します

なぜ多項式時間か否かが重要か

■ 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)

問題 PRIME 与えられた正整数 (十進法で n 桁) が素数か判定

それ未満の数 (10^n 個ほどある) で割り切れるかすべて調べれば判る

つまり　整数 x を文字列 0^x で表すという「非効率」な符号法で入力を受付ける素数判定問題なら自明に $\mathbf{P}$

それよりも劇的に速く　n の多項式時間で判定する方法が存在 [AKS05]

問題 HAMILTON 与えられたグラフ (頂点数 n) がハミルトン閉路をもつか判定

(全頂点を一度づつ通る辿り方)

頂点の並べ順 ($n!$ 通り) をすべて調べれば判る

n の多項式時間で判定する方法があるかどうか不明 (多分なさそう)

問題 SAT 与えられた命題論理式 (変数が n 個) が充足可能か判定

真理値割当 (2^n 通り) をすべて調べれば判る

n の多項式時間で判定する方法があるかどうか不明 (多分なさそう)

[AKS05] M. Agrawal, N. Kayal and N. Saxena. PRIMES is in P. *Annals of Mathematics*, 160: 781–793, 2004.

定義

機械 M が**多項式空間**であるとは　或る多項式 $p: \mathbf{N} \to \mathbf{N}$ が存在し
任意の長さ n の任意の入力に対する M の計算が　停止し
空間 $p(n)$ 以内である（テープ上で初めの欄の左右 $p(n)$ 個の範囲に留まる）ことをいう
言語 A を認識する多項式空間の機械 M が存在するとき
A は**多項式空間認識可能**である（$A \in \mathbf{PSPACE}$）という

定理

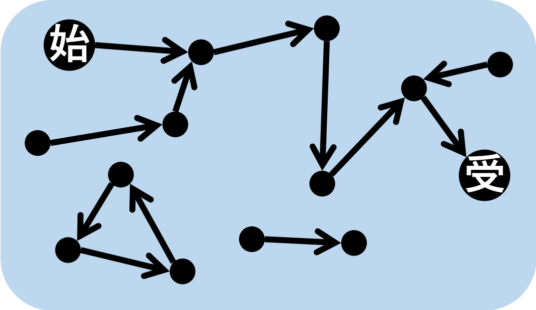
$\mathbf{P} \underset{①}{\subseteq} \mathbf{PSPACE} \underset{②}{\subseteq} \mathbf{EXP}$

多項式 q が存在して
入力長 n のとき $2^{q(n)}$ 時間以内

① 空間を 1 使う（新たな欄に移動する）には
時間 1 以上かかるので

② 多項式空間機械 M に対して多項式 q が存在し
長さ n の入力に対する M の状況として
あり得るものは $2^{q(n)}$ 個以内

したがって M は
「受理するときは時間 $\leq 2^{q(n)}$ で受理する」



状況間の遷移関係

問題 PRIME	入力	正の整数 X（を十進表示したもの）
	答	X は素数か

※ PRIME は本当は多項式時間認識可能であることが今では判っているが

問題 HAMILTON	入力	グラフ G
	答	G にハミルトン閉路はあるか

定理

PRIME, HAMILTON, SAT ∈ $\mathbf{PSPACE}$

n 桁の割り算ひとつひとつは容易（n の多項式時間）

入力

48581583386419 ÷ 2 = 24290791693209 あまり 1

n 桁

3 = 16193861128806 あまり 1

⋮

6306457 = 7703467 あまり 0

48581583386418

指数的に多くの手間がかかるが

以前の計算過程を覚えておく必要はない（同じ場所に上書きしてよい）

問題		
PRIME	入力	正の整数 X（を十進表示したもの）
	答	X は素数か

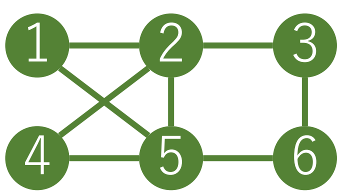
※ PRIME は本当は多項式時間認識可能であることが今では判っているが

問題		
HAMILTON	入力	グラフ G
	答	G にハミルトン閉路はあるか

定理

PRIME, HAMILTON, SAT $\in$ **PSPACE**

入力



n 頂点

辺がちゃんと繋がった経路になっているか確かめるのは容易

1 2 3 4 5 6　ダメ（3と4の間や6と1の間が繋がっていないので）

1 2 3 4 6 5　ダメ（3と4の間や4と6の間が繋がっていないので）

1 2 3 5 4 6

⋮

6 5 4 3 2 1

経路の候補 $n!$ 個

指数的に多くの手間がかかるが以前の計算過程を覚えておく必要はない（同じ場所に上書きしてよい）

しかし「指数個の可能性を調べる」と言っても
$\overline{\text{PRIME}}$ や HAMILTON や SAT は　調べた指数個の候補のうち
「適するものが存在するか」を問うという特殊な形をしている

定義

言語 A が **NP** に属するとは
或る多項式時間機械 M と多項式 $p\colon \mathbf{N} \to \mathbf{N}$ が存在し
任意の入力 x に対し
- もし $x \in A$ ならば　長さ $p(|x|)$ の文字列 r が存在して $M(x;r)$ は受理
- もし $x \notin A$ ならば　長さ $p(|x|)$ の任意の文字列 r で $M(x;r)$ は不受理

明らかに $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE}$

定理

$\overline{\text{PRIME}}, \text{HAMILTON}, \text{SAT} \in \mathbf{NP}$

問題
QBF

与えられた量化命題論理式

$$Q_n X_n . Q_{n-1} X_{n-1} \cdots Q_1 X_1 . \varphi(X_1, \ldots, X_n)$$

の真偽を判定せよ

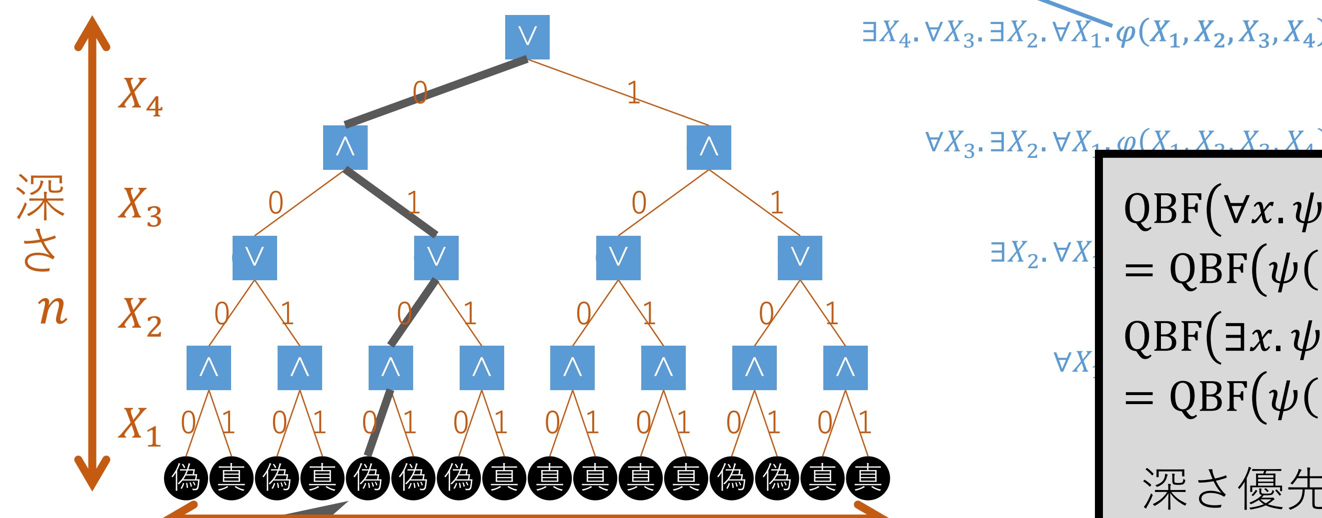
命題変数
真（1）か偽（0）の値をとる

量化子（∀ か ∃）

命題論理式
命題変数から論理記号
∧（かつ）
∨（または）
¬（否定）
を使って作られる式

「二人ゲームでどちらが必勝か判定する問題」

例 $\exists X_4 . \forall X_3 . \exists X_2 . \forall X_1 . \underline{(X_2 \vee \neg X_3) \wedge (X_1 \vee X_4)}$



$\varphi(0,0,1,0) =$ 偽
容易に計算できる

葉 2^n 枚

この例では $n = 4$

$\mathrm{QBF}\big(\forall x . \psi(x)\big)$
$= \mathrm{QBF}\big(\psi(0)\big) \wedge \mathrm{QBF}\big(\psi(1)\big)$
$\mathrm{QBF}\big(\exists x . \psi(x)\big)$
$= \mathrm{QBF}\big(\psi(0)\big) \vee \mathrm{QBF}\big(\psi(1)\big)$

深さ優先探索
➡ 多項式空間でできる

問題		
問題 SR	入力	書換え規則の集合 R と文字列 w
	答	R による書換えを次々と w に施して ε にできるか
問題 $\mathrm{SR}_{\geq}$	入力	書換え規則の集合 R と文字列 w 但し R の各規則 $u \to v$ において (u の長さ) $\geq$ (v の長さ)
	答	R による書換えを次々と w に施して ε にできるか
問題 $\mathrm{SR}_{>}$	入力	書換え規則の集合 R と文字列 w 但し R の各規則 $u \to v$ において (u の長さ) $>$ (v の長さ)
	答	R による書換えを次々と w に施して ε にできるか
問題 $\mathrm{SR}^{1}_{>}$	入力	書換え規則の集合 R と文字列 w 但し R の各規則 $u \to v$ において (u の長さ) $>$ (v の長さ)
	答	R による書換えを次々と w に施すと 可能な書換え方は毎回一通りしかなく　やがて ε に達する

困難そう（一般的な入力）

容易そう（特殊な入力）

問題 SR

入力	(R, w)
答	$w \Rightarrow_R^* \varepsilon$ か

定理（昨日）

SR ∈ **CE**（だが $\overline{\text{SR}} \notin$ **CE**）

問題 $\text{SR}_{\geq}$

SR を各規則が
(旧長さ) ≥ (新長さ)
の場合に制限したもの

?（次頁）

問題 $\text{SR}_{>}$

更に各規則が
(旧長さ) > (新長さ)
の場合に制限したもの

定理

$\text{SR}_{>} \in \mathbf{NP}$

書換えの列（出発文字列の長さ以内と
保証されている）の存在を問う問題だから

問題 $\text{SR}_{>}^{1}$

$\text{SR}_{>}$を更に
可能な書換え方が毎回一通り
な場合に制限したもの

定理

$\text{SR}_{>}^{1} \in \mathbf{P}$

書換えを実際に順次（機械の
テープ上で）行ってみればよい

定理

SR$_{\geq}$ ∈ **PSPACE**

※ $\Sigma = \{a, b\}$ として考える

問題 SR$_{\geq}$

入力	(R, w) 但し各規則が (旧長さ) ≥ (新長さ)
問	$w \Rightarrow_R^* \varepsilon$ か

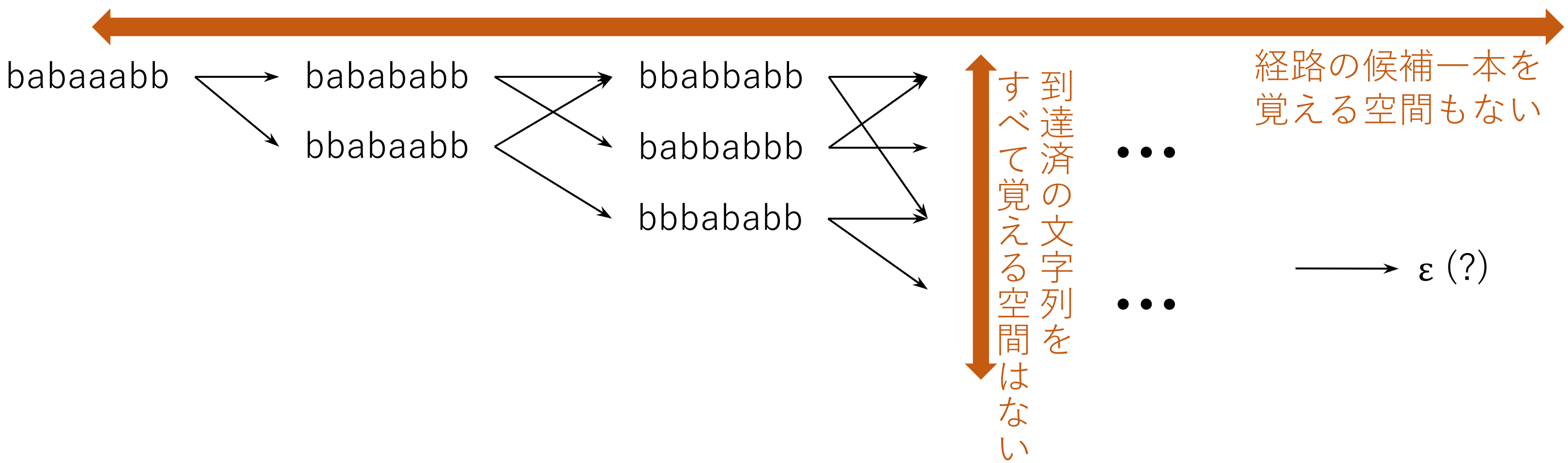
入力の長さが n である（w の長さ $< n$）とき

w から書換えにより生じ得る文字列も長さ $< n$ であり　その個数は $< 2^n$

したがって $w \Rightarrow_R^{\leq 2^n} \varepsilon$ かどうか調べればよい

書換え 2^n 回以内で w から ε が得られる という意味

しかし　単純に長さ $\leq 2^n$ の経路すべてを調べることはできない



定理

$\mathrm{SR}_{\geq} \in \mathbf{PSPACE}$

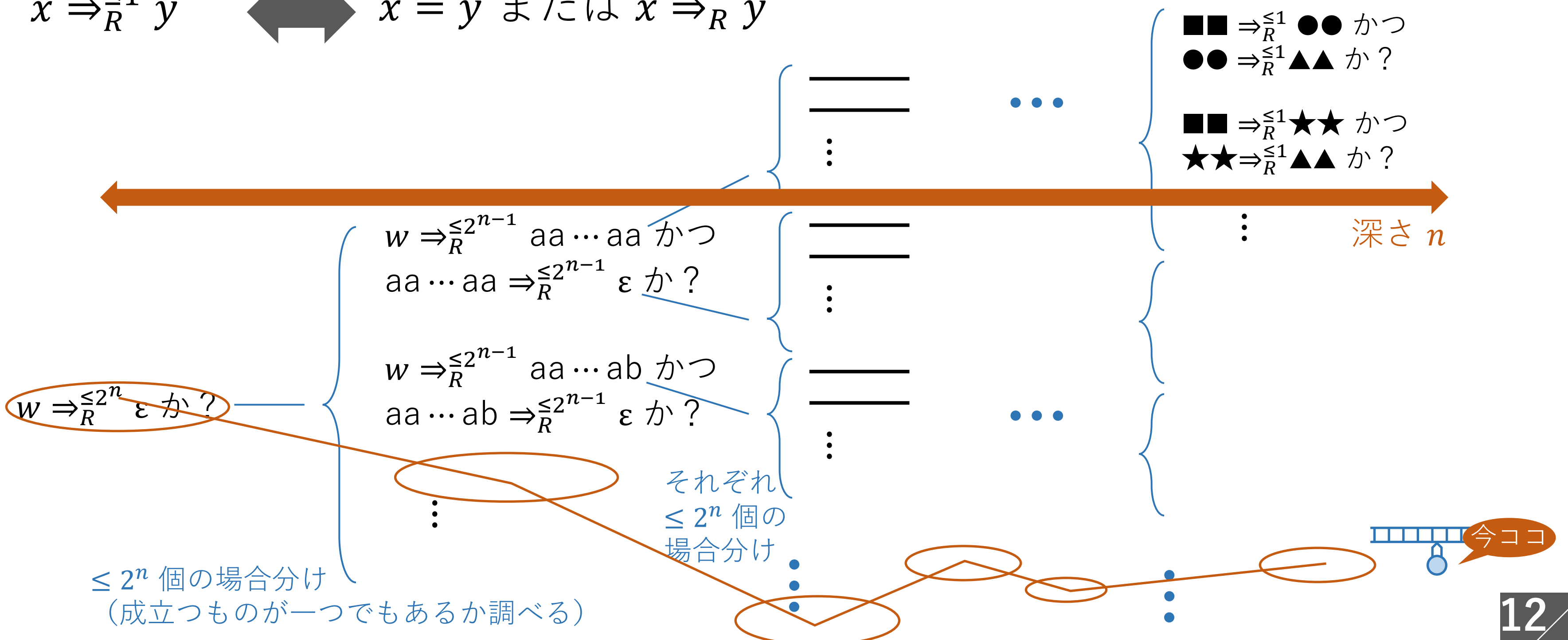
問題 $\mathrm{SR}_{\geq}$

入力	(R, w) 但し各規則が (旧長さ) $\geq$ (新長さ)
問	$w \Rightarrow_R^* \varepsilon$ か

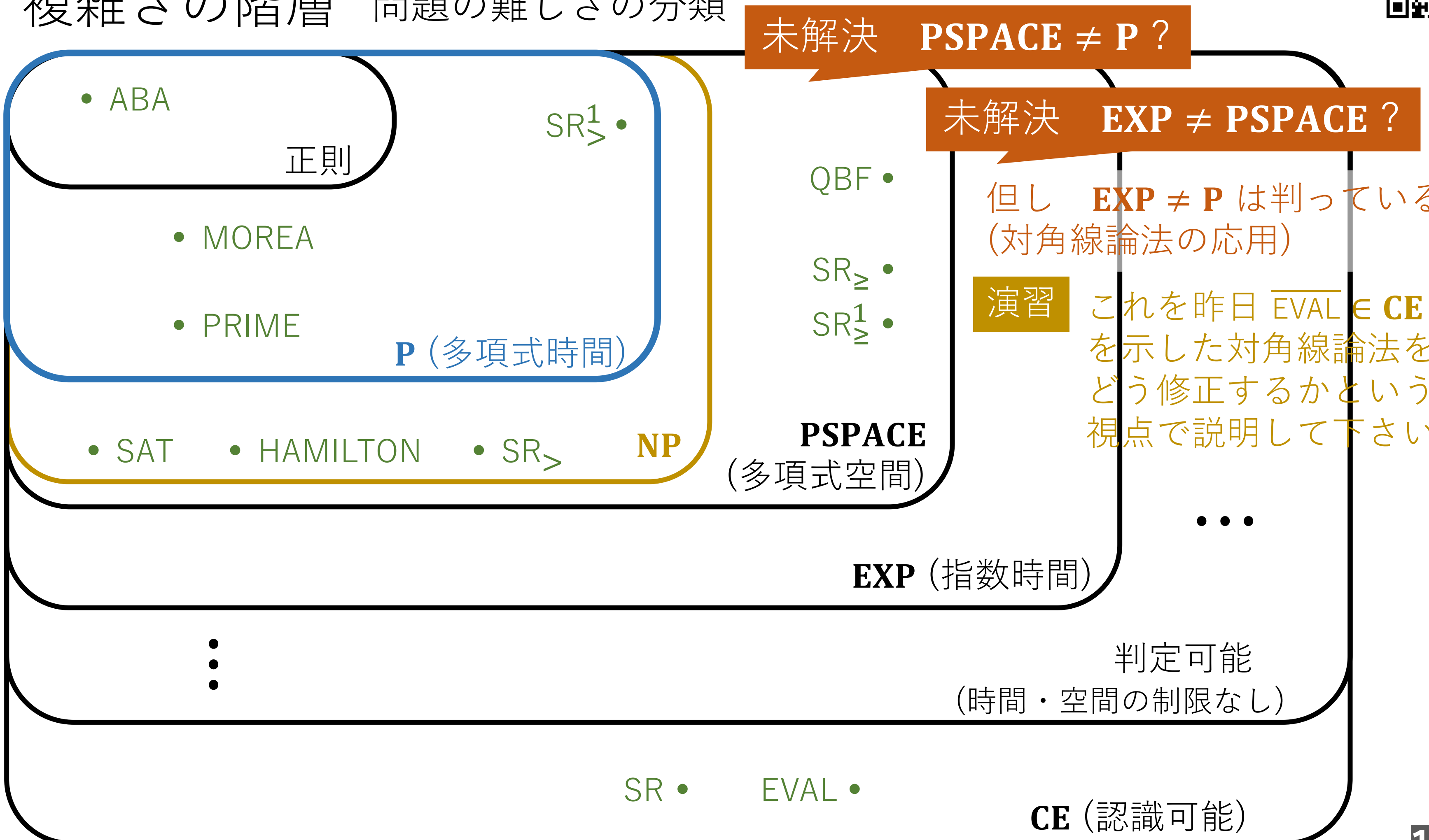
$w \Rightarrow_R^{\leq 2^n} \varepsilon$ かどうか　次の関係を再帰的に用いて調べればよい

$x \Rightarrow_R^{\leq 2^{i+1}} y \iff$ 或る $z \in \Sigma^{<n}$ が存在して $x \Rightarrow_R^{\leq 2^i} z$ かつ $z \Rightarrow_R^{\leq 2^i} y$

$x \Rightarrow_R^{\leq 1} y \iff x = y$ または $x \Rightarrow_R y$



複雑さの階層　問題の難しさの分類



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

非決定性（＝乱択）機械

$x =$ 0 0 0 1 1 0 1 1 0 1 1

入力テープ

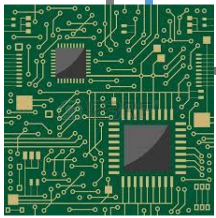
$r =$ 0 1 0 0 0 1 1 0 1 0 1 1 0 1 1 ……

常に㊨へ

乱数テープ

1 0 0 1 0 1

作業テープ



機械 M

遷移規則 δ：

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

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

計算結果

$M(\underline{x}; \underline{r})$ 受理か不受理

入力　乱数　に依存

次の遷移を二つの分岐から非決定的に（等確率で）選ぶ

＝

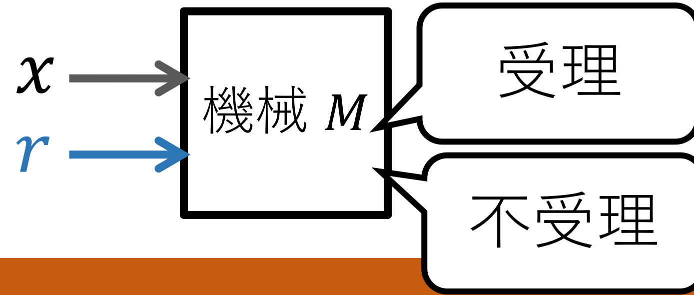
初めに「乱数テープ」上に乱数列が~~無限に~~供給される

$p(|x|)$ ビットで十分

時間量が $p\colon \mathbf{N} \to \mathbf{N}$ 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

nondeterministic の N

NP の定義の言い換え



定義

言語 A が級 **NP** に属するとは
或る多項式時間 (非決定性) 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**或る r で**受理

$x \notin A$ のとき $M(x; r)$ は**すべての r で**不受理

注意

NP (や次頁の RP) の定義について

理くつ上はこの動詞の方を
(「∃ 認識する」とかに？) 変えるべきだが
「**NP** は非決定性多項式時間で解ける問題」
のように言ってしまうことも多い

定義
機械 M が**多項式時間**であるとは…

定義
機械 M が言語 A を**認識**するとは…

→ **P**
(多項式時間限定な機械によって認識される言語の全体)

定義
非決定性機械 M が**多項式時間**であるとは…

定義
機械 M が言語 A を●●するとは…

→ **NP**
(多項式時間限定な機械によって●●される言語の全体)

定義
非決定性機械 M が**多項式時間**であるとは…

定義
機械 M が言語 A を▲▲するとは…

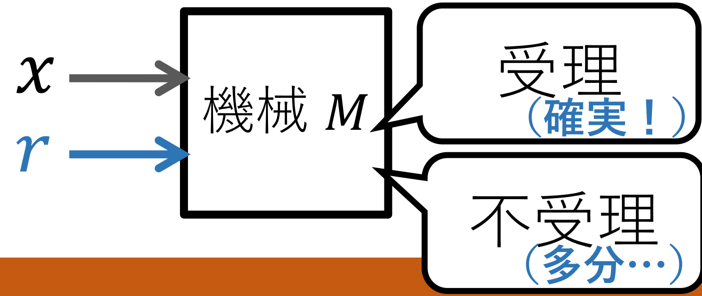
→ **RP**
(多項式時間限定な機械によって▲▲される言語の全体)

↑ **こっちは共通**
(**PSPACE** や **EXP** の定義ではこれを変える)

↑ **ココを変えた！**

randomized の R

RP



片側誤り
誤受理なし

定義

言語 A が級 **RP** に属するとは
或る多項式時間 (非決定性) 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** $> \boxed{\frac{1}{2}}$ **で**受理

$x \notin A$ のとき $M(x;r)$ は**必ず**不受理

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

明らかに
$$\mathbf{P} \subseteq \mathbf{RP} \subseteq \mathbf{NP}$$

先程の例

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

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

d は p の全次数

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

補題

全次数 d の非零多項式 p について

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

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

証明

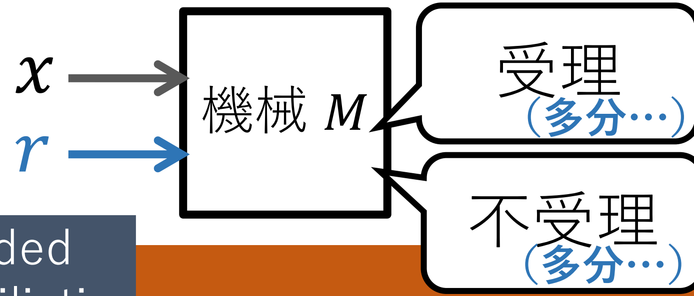
m に関する帰納法　$m = 0$ のとき自明　$m > 0$ とする
X_1 の最高次数を i として　次のように書く

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

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

もし$\boxed{}$ならば $p(X_1, r_2, \ldots, r_m)$ 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

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



両側誤り

BPP

bounded probabilistic

定義

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

$\geq \frac{1}{2}$ と $< \frac{1}{2}$ とかではダメ

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

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？

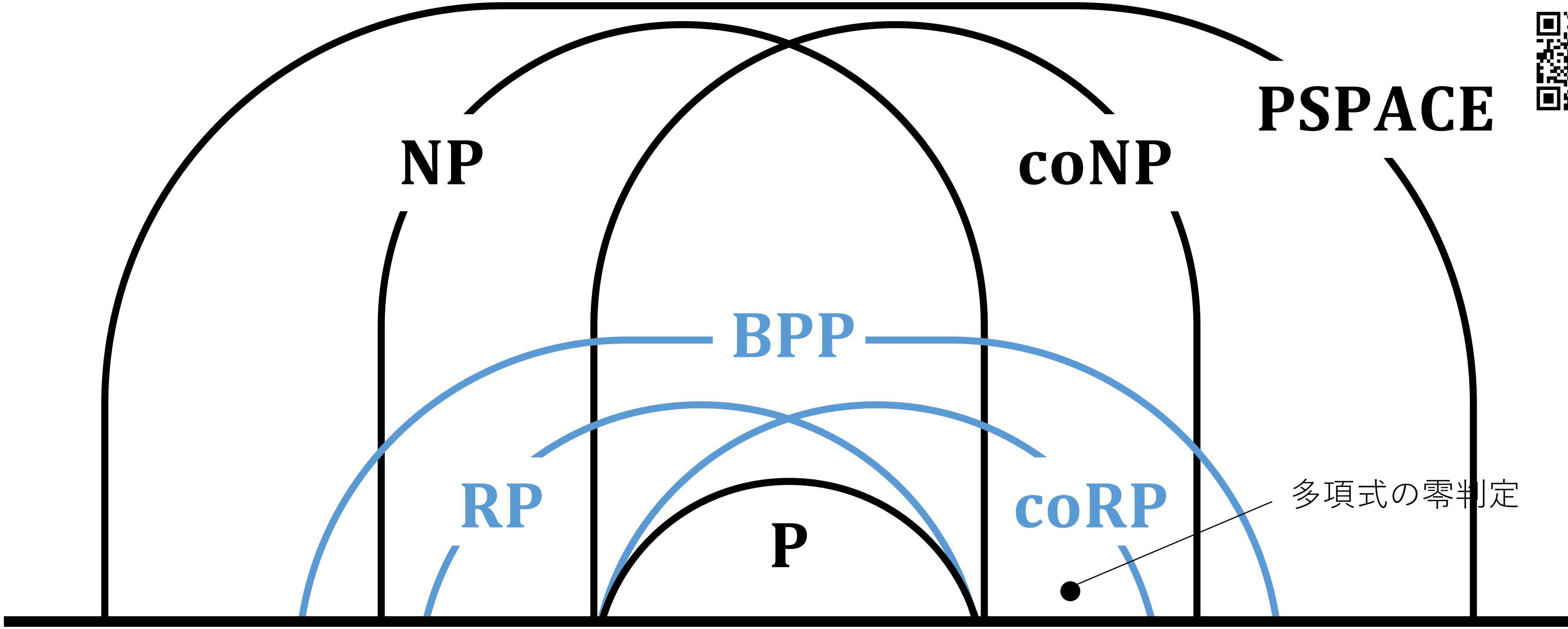
入力 x → 算法 → 出力

~~乱数 r~~ →

3141592653589793238462643383279
擬似乱数

→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？



「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
（図中の級がすべて等しいかもしれない）

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

（等しいと予想する人が多い）

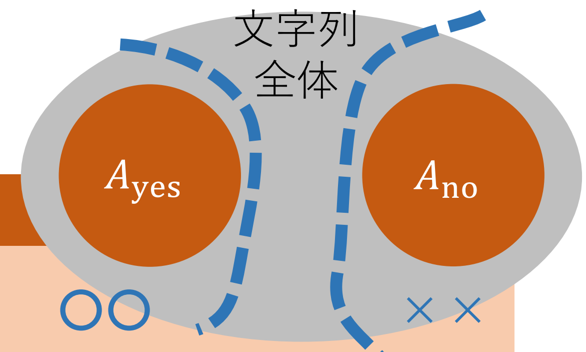
BPP (や RP) の定義の微妙にキモい点について

定義（再掲）

言語 A が **BPP** に属するとは
多項式時間の乱択機械 M が存在し　任意の入力 x に対し
- $x \in A$ ならば $M(x;\cdot)$ は確率 $> 2/3$ で受理
- $x \notin A$ ならば $M(x;\cdot)$ は確率 $< 1/3$ で受理

機械のうち「どの x に対しても受理確率が $1/3$ と $2/3$ の間にはならない」ようなもののみが意味ある計算と見なされる

本当は「約束つき問題」を考えた方がよい



定義

約束つき判定問題 $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ とは
互いに交わらない集合 $A_{\mathrm{yes}}, A_{\mathrm{no}} \subseteq \Sigma^*$ の組

各級の定義では
- $x \in A$ ならば○○
- $x \notin A$ ならば× ×

の代りに
- $x \in A_{\mathrm{yes}}$ ならば○○
- $x \in A_{\mathrm{no}}$ ならば× ×

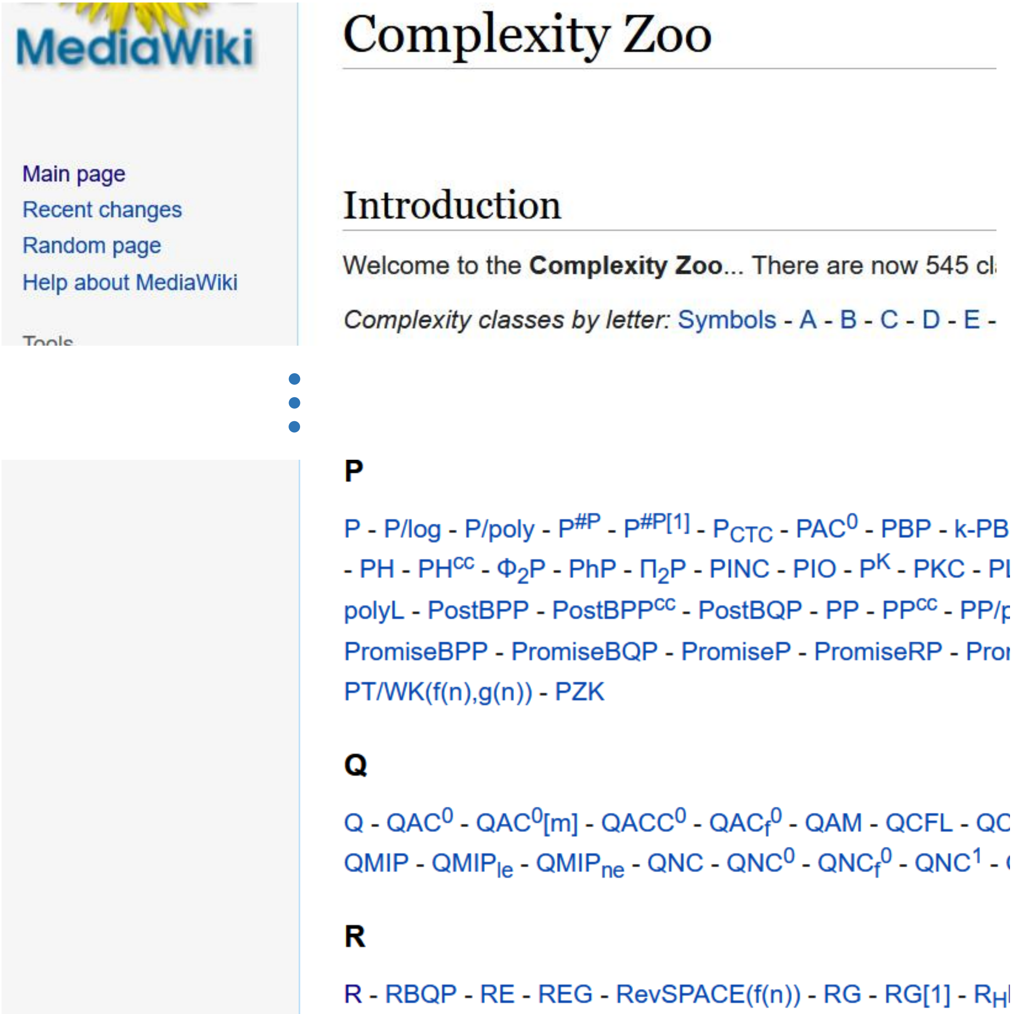
を要請

様々な計算量級（クラス）

ウェブサイト「Complexity Zoo」 500以上の計算量級を紹介
https://complexityzoo.net/Complexity_Zoo

本講義で扱った
- 正則（REG）
- 多項式時間認識可能（P）
- 多項式空間認識可能（PSPACE）
- 判定可能（R または Decidable）
- 認識可能（RE または CE）

以外にも問題の複雑さの度合を色々な側面から捉えた基準が数多く定義され研究されている



様々な計算量級（クラス）

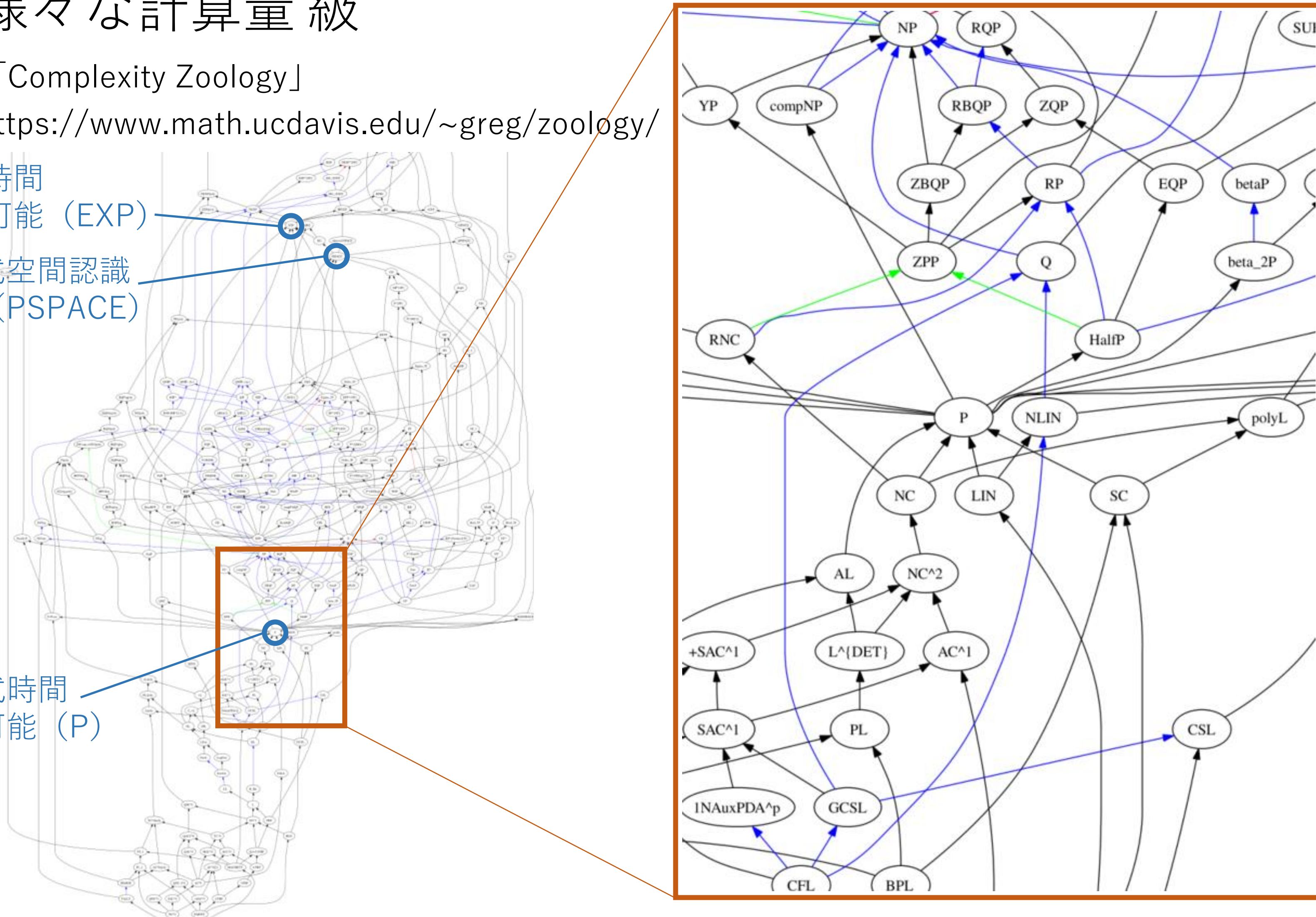
「Complexity Zoology」

https://www.math.ucdavis.edu/~greg/zoology/

指数時間
認識可能（EXP）

多項式空間認識
可能（PSPACE）

多項式時間
認識可能（P）



まとめ　第二日　時間と空間の制限

- 時間計算量は「入力長 n のとき時間 $T(n)$ 以内で計算できる」という形で測る (空間計算量も同様)
- 多項式時間 (**P** や **BPP**)　≒　現実的な手間での計算
- 多項式空間 (**PSPACE**)　指数個の調べ尽しができる
- **NP**　指数個の候補中の「存在」判定型の問題
- しかし本当に **PSPACE** $\neq$ **P** かは未解決 (不可能性の証明は難しい)