



↑
スライドはここに置いてあります

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





第三日 帰着と完全問題



神託機械 (神託への質問機能があるチューリング機械)

「質問テープ」「回答テープ」があり 特別な状態「質問状態」がある

神託機械 M に神託として言語 $B \subseteq \Sigma^*$ を与えたもの M^B の計算では

質問テープに文字列 $y \in \Sigma^*$ が書かれてあるときに質問状態になると
瞬時に $y \in B$ の成否 (1 か 0) が回答テープに現れる

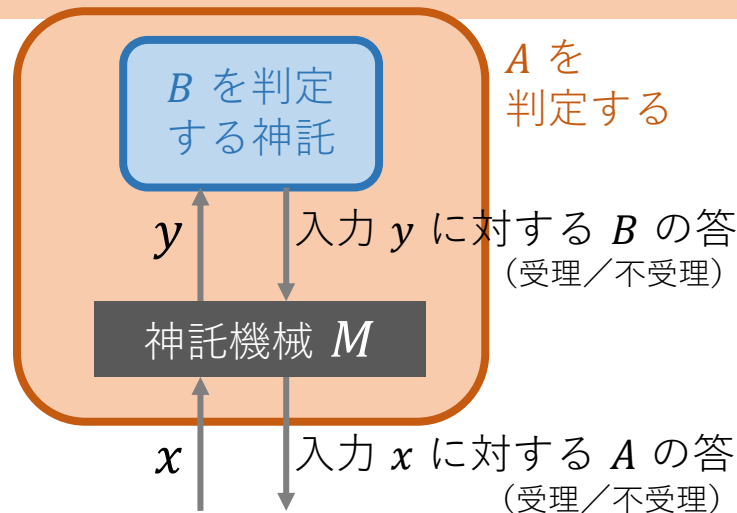
定義

言語 A が言語 B に**多項式時間帰着**するとは
多項式時間神託機械 M が存在して M^B が A を認識することをいう

$A \leq^P B$ あるいは $A \in \mathbf{P}^B$ と書く

「 B を使うと A が計算できる」

「 A の難しさは B 以下」





$A \leq^P B$ であるとする...

もし $B \in \mathbf{P}$ ならば $A \in \mathbf{P}$

もし $B \in \mathbf{PSPACE}$ ならば $A \in \mathbf{PSPACE}$

もし B が判定可能ならば A も判定可能

問題どうしの相対的な困難さの比較ができる

問題

入力

(M, x)

EVAL

答

M は x を受理するか

問題

入力

(R, w)

SR

答

$w \Rightarrow_R^* \varepsilon$ か

初日には $\text{EVAL} \leq^P \text{SR}$ を示すことで

EVAL の判定不能性から SR の判定不能性を導いた

(その時点では帰着が多項式時間どうかは気にしていなかったが)



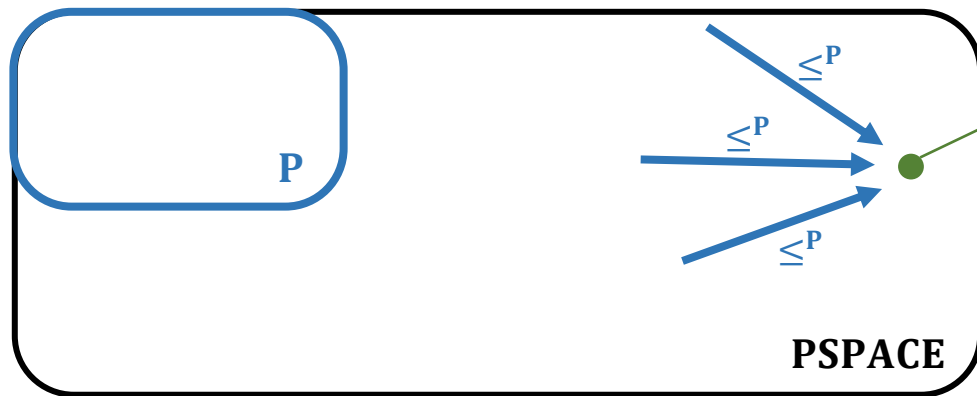
定義

言語 $B \in \mathbf{PSPACE}$ が **PSPACE 完全** であるとは
任意の言語 $A \in \mathbf{PSPACE}$ が B に
多項式時間帰着する ($A \leq^P B$) ことをいう

「 B は **PSPACE** な問題のうち最難」

「**PSPACE** = **P** でない限り B は **P** に属しない」

「**PSPACE** とは 複雑さが問題 B 以下であること」



PSPACE 完全な問題 B

そんな特別な問題が
存在するの？





問題 $\text{EVAL}_{\text{space}}$

入力 文字列の組 $(M, x, 0^s)$

答 機械 M は入力 x を空間 s 以下で受理するか

定理

$\text{EVAL}_{\text{space}}$ は **PSPACE** 完全

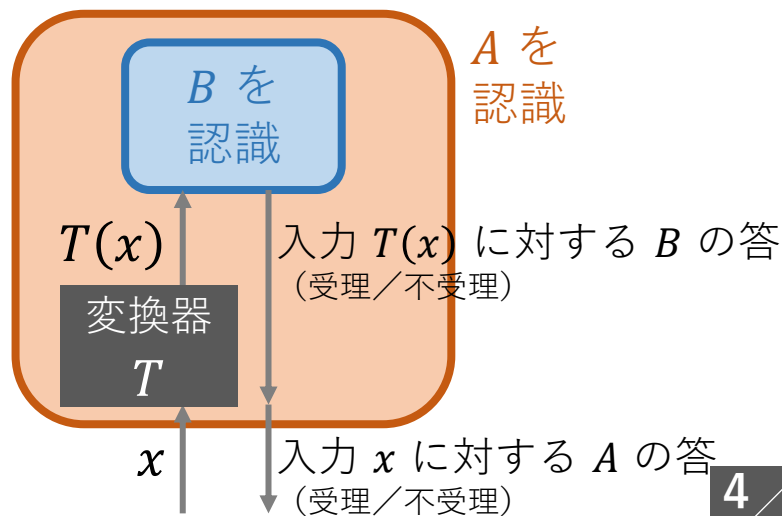
∴ 言語 $A \in \text{PSPACE}$ を考えると A を認識する機械 M と多項式 p が存在し 任意の入力 x に対して M は空間 $\leq p(|x|)$ したがって $x \in A \iff (M, x, 0^{p(|x|)}) \in \text{EVAL}_{\text{space}}$

この例のように 特殊な形の帰着で事足りることもあるので 記法を定めておく

定義

言語 A, B について $A \leq_m^P B$ とは 多項式時間機械 T が存在して 任意の入力 x に対する T の出力 $T(x)$ が $x \in A \iff T(x) \in B$ を満たすことをいう

停止したときの
テープ上の文字列





問題 $\text{EVAL}_{\text{space}}$

入力 文字列の組 $(M, x, 0^s)$

答 機械 M は入力 x を空間 s 以下で受理するか

定理

$\text{EVAL}_{\text{space}}$ は **PSPACE** 完全 (帰着 $\leq_m^{\mathbf{P}}$ の下でも)

問題 $\text{EVAL}_{\text{ntime}}$

入力 文字列の組 $(M, x, 0^t)$

答 或る文字列 r が存在して
機械 M は入力 $x\#r$ を時間 t 以下で受理するか

定理

$\text{EVAL}_{\text{ntime}}$ は (帰着 $\leq_m^{\mathbf{P}}$ の下で) **NP** 完全

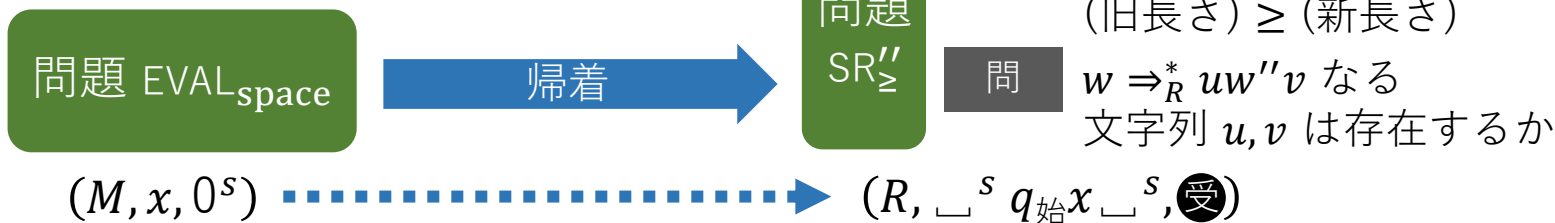
∴ 同様



定理

$SR_{\geq}$ は **PSPACE** 完全 (同じ証明により $SR_{\geq}^1$ も)

第一日の $EVAL \leq^P SR'' \leq^P SR$ と同様に
 $EVAL_{space} \leq^P SR''_{\geq} \leq^P SR_{\geq}$ を示す



R は M の各状態 $q \in Q$ と文字 $\sigma \in \Gamma$ について次の規則を加えて作る

- もし $\delta(q, \sigma) = (q', \sigma', \text{左})$ ならば 各 $\tau \in \Gamma$ に対し規則 $\tau q \sigma \rightarrow q' \tau \sigma'$
- もし $\delta(q, \sigma) = (q', \sigma', \text{右})$ ならば 規則 $q \sigma \rightarrow \sigma' q'$
- もし $\delta(q, \sigma) = \text{止}$ かつ $q \in Q_{\text{受理}}$ ならば 規則 $q \sigma \rightarrow \text{受}$

($EVAL \leq^P SR''$ のときに設けた「空白文字を端に付け足す規則」 $\triangleright \rightarrow \triangleright \sqcup$ および $\triangleleft \rightarrow \sqcup \triangleleft$ は設けない)

すると M は x を空間 $\leq s$ で受理 $\iff R$ の規則に従って $\sqcup^s q_{\text{始}} x \sqcup^s$ から 受 を含む文字列に到達できる が成立



定理

$SR_{\geq}$ は **NP** 完全

∴ 帰着 $EVAL_{ntime} \leq^P SR_{\geq}$ なら前頁と同様に考えて作れる



$(M, x, 0^t) \dashrightarrow (R \cup \{\square \rightarrow \sigma : \sigma \in \Sigma\}, q_{\text{始}}x\#\square^t, \text{受})$

長さを減らす規則だけにするには

もとの各 (長さを変えない) 規則 $a_1a_2 \cdots a_k \rightarrow b_1b_2 \cdots b_k$ の代りに
(長さを減らす) 規則 $a_1\$^{\tau_1}a_2\$^{\tau_2} \cdots \$^{\tau_{k-1}}a_k \rightarrow b_1\$^{\tau_1-1}b_2\$^{\tau_2} \cdots \$^{\tau_{k-1}}b_k$ を
設ける (各 $\tau_1, \tau_2, \dots, \tau_{k-1} \leq t$ について)

もとの初期文字列の各文字の間に $\t を挟んだものを初期文字列とする

□

定理

SAT や HAMILTON も **NP** 完全

∴ 頑張ると多項式時間帰着が作れる



問題
QBF

与えられた量化命題論理式

$$Q_n X_n \cdot Q_{n-1} X_{n-1} \dots Q_1 X_1 \cdot \varphi(X_1, \dots, X_n)$$

の真偽を判定せよ

量化子 (∀ か ∃)

命題変数

真 (1) か偽 (0) の値をとる

命題論理式

命題変数から論理記号

∧ (かつ)

∨ (または)

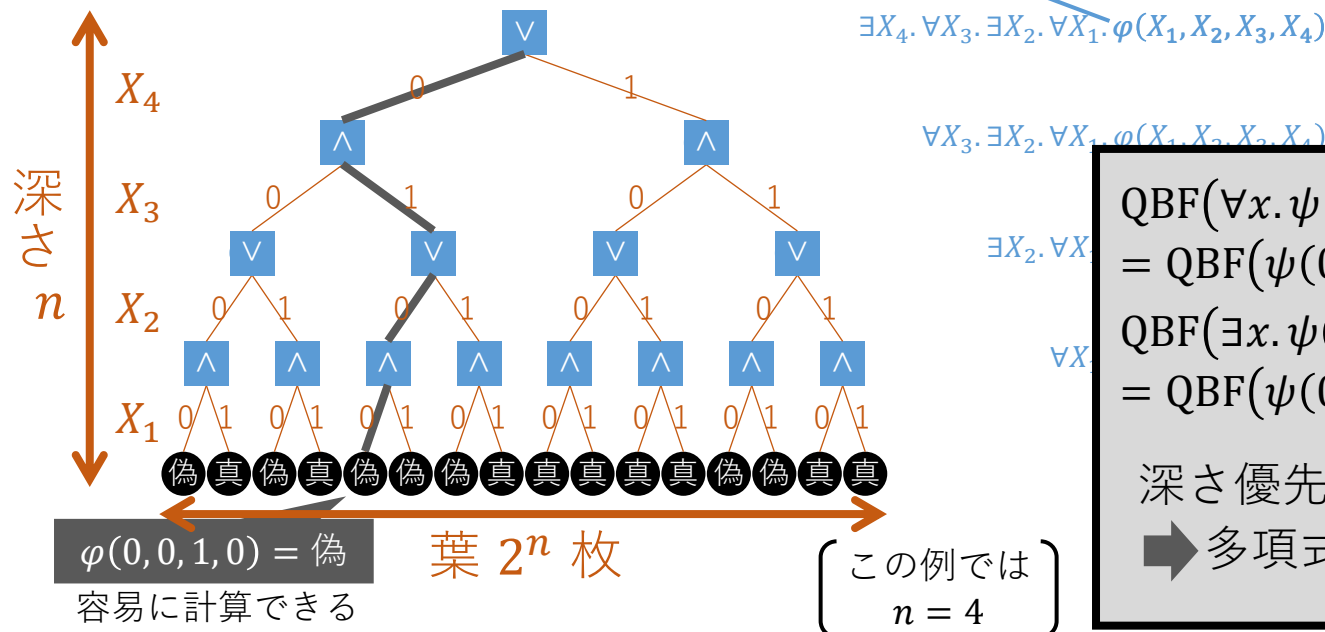
¬ (否定)

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

$$\exists X_4. \forall X_3. \exists X_2. \forall X_1. (X_2 \vee \neg X_3) \wedge (X_1 \vee X_4)$$



$$\begin{aligned} \text{QBF}(\forall x. \psi(x)) &= \text{QBF}(\psi(0)) \wedge \text{QBF}(\psi(1)) \\ \text{QBF}(\exists x. \psi(x)) &= \text{QBF}(\psi(0)) \vee \text{QBF}(\psi(1)) \end{aligned}$$

深さ優先探索

➡ 多項式空間でできる



定理

QBF は **PSPACE** 完全

問題
 $SR_{\geq}$

入力

(R, w) 但し各規則が
(旧長さ) $\geq$ (新長さ)

問

$w \Rightarrow_R^* \varepsilon$ か

$w \Rightarrow_R^{\leq 2^n} \varepsilon$ かどうかを
調べれば良い (昨日)

$SR_{\geq} \leq^P$ QBF を示す

昨日 $SR_{\geq}$ の多項式空間認識可能性を示したときと同じ次の関係を用いる

$$\begin{aligned}
 x \Rightarrow_R^{\leq 2^{i+1}} y &\iff \exists z (x \Rightarrow_R^{\leq 2^i} z \text{ かつ } z \Rightarrow_R^{\leq 2^i} y) \\
 &\iff \exists z \forall u, v \\
 &\quad (\text{もし } (u, v) = (x, z) \text{ または } = (z, y) \text{ ならば } u \Rightarrow_R^{\leq 2^i} v)
 \end{aligned}$$

「 $\exists w$ 」は「或る $w \in \Sigma^{<n}$ が存在して」
「 $\forall w$ 」は「任意の $w \in \Sigma^{<n}$ に対し」の意

これを再帰的に用いて

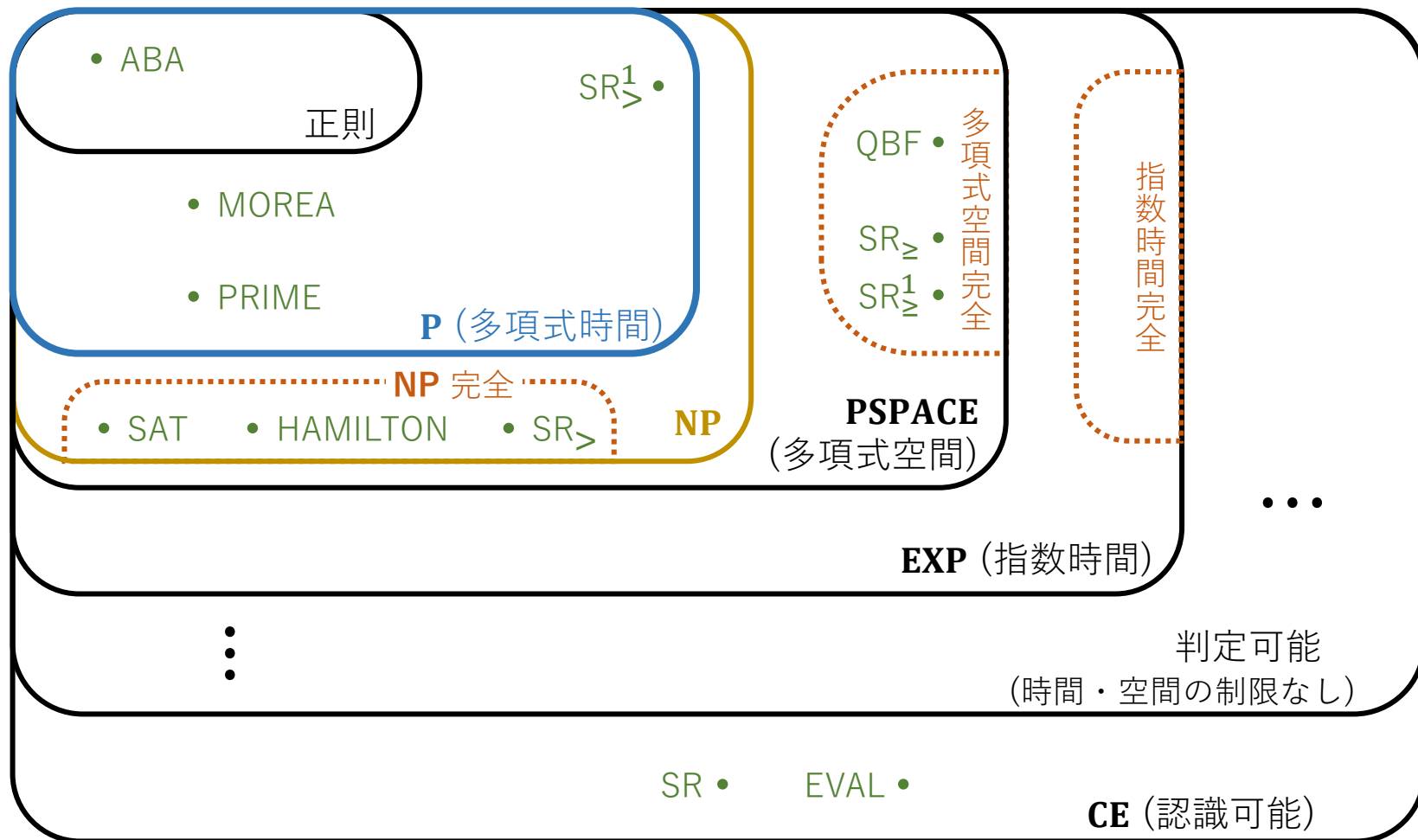
$$\underline{w \Rightarrow_R^{\leq 2^n} \varepsilon} \iff \exists z_n \forall u_n, v_n \quad \exists z_{n-1} \forall u_{n-1}, v_{n-1} \quad \cdots \exists z_1 \forall u_1, v_1$$

もし $(u_n, v_n) = (w, z_n)$ または $= (z_n, \varepsilon)$ ならば
もし $(u_{n-1}, v_{n-1}) = (u_n, z_{n-1})$ または $= (z_{n-1}, v_n)$ ならば
.....
もし $(u_1, v_1) = (u_2, z_1)$ または $= (z_1, v_2)$ ならば $u_1 \Rightarrow_R^{\leq 1} v_1$

これを
量化命題論理式として
書くことができる



複雑さの階層と完全問題





計算可能解析 (Computable Analysis)

整数など ➡ 文字列で表現 (〇型)

実数など ➡ 文字列の関数で表現 (一型)

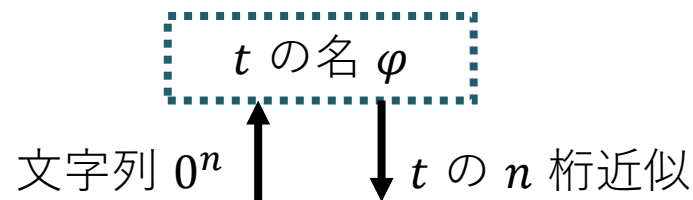
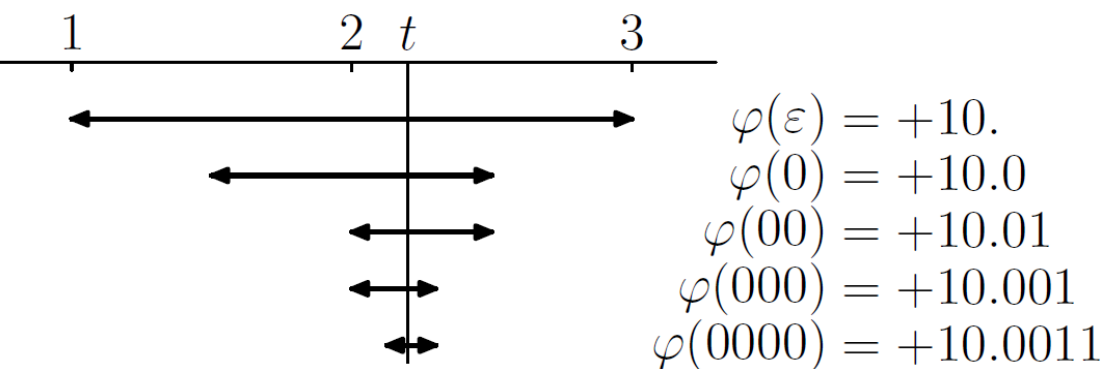
実数を入力とする関数など ➡ **二型の計算**

以下で述べる計算可能性・計算量の枠組は
Type-2 Theory of Effectivity
(二型の計算理論) と呼ばれる



定義

函数 $\varphi: \Sigma^* \rightarrow \Sigma^*$ が実数 $t \in \mathbf{R}$ の**名**であるとは 各 $n \in \mathbf{N}$ に対し $\varphi(0^n)$ が t を二進表示して小数点以下 n 桁で切上げ又は切捨てたものであること



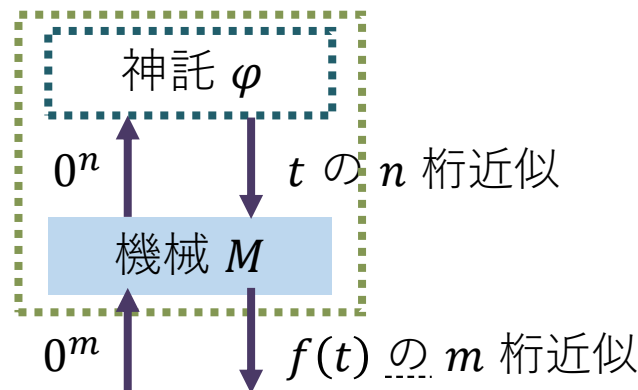
コーシー列 (収束の速さが保証されている) による表現

「数学に普通に出てくる実数」はだいたい多項式時間計算可能



定義

機械 M が $f: [0, 1] \rightarrow \mathbf{R}$ を**計算**するとは
 任意の $t \in [0, 1]$ の任意の名 φ について
 M^φ が $f(t)$ の或る名を計算することをいう



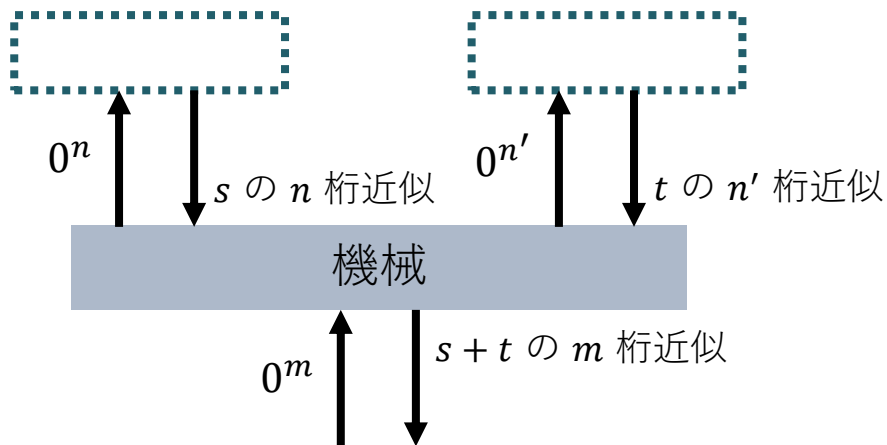
「 t が任意の精度でわかるならば $f(t)$ を任意の精度で求められる」

M が多項式時間というのは (通常の神託機械での定義の通り) m の多項式をさす
 これにより実関数の多項式時間計算可能性を定義する

- 例 $f: \mathbf{R} \rightarrow \mathbf{R}$ を計算するには：
1. 入力 0^m を受け取る
 2. $n = m + 3$ として質問 0^n を発する
 3. その答として神託から 有理数 r を受取る (この r は $|r - t| < 2^{-n}$ を満す)
 4. 出力として $3 \cdot r$ を最も近い 2^{-m} の倍数に丸めたものを書く



例 加算 $+: [0, 1]^2 \rightarrow \mathbf{R}$ は
多項式時間計算可能



$\therefore n = n' = m + 2$ とすればよい

例 $\sin: [0, 1] \rightarrow \mathbf{R}$ は
多項式時間計算可能

$$\sin t = \frac{t}{1!} - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \dots$$

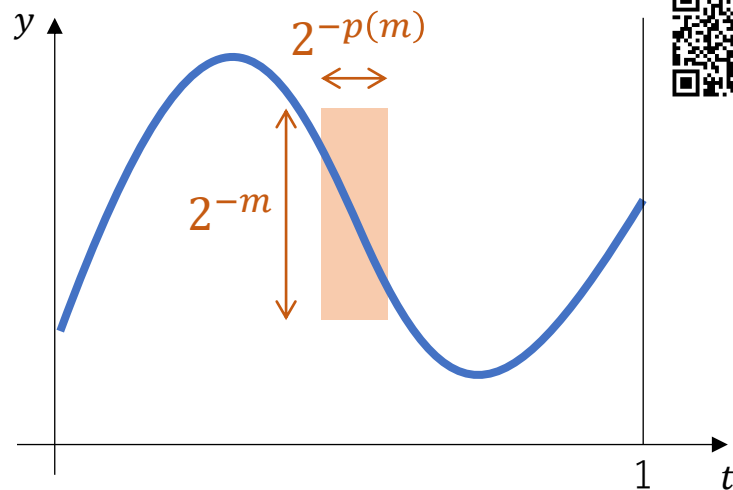
$\therefore \sin$ は傾き ≤ 1 なので
 $\sin t$ を近似的に求めるには
近くの点 t' で $\sin t'$ を求めればよい
上記の級数は収束が十分早いので
和を精度 m ビット求めるには
第 $O(m)$ 部分和で十分

計算可能ならば連続

定理

- 計算可能な実関数は連続
- 多項式時間計算可能な関数 $f: [0, 1] \rightarrow \mathbf{R}$ は多項式の**連続度**すなわち次を満たす多項式 $p: \mathbf{N} \rightarrow \mathbf{N}$ をもつ

$$|t - t'| < 2^{-p(m)} \Rightarrow |f(t) - f(t')| < 2^{-m}$$

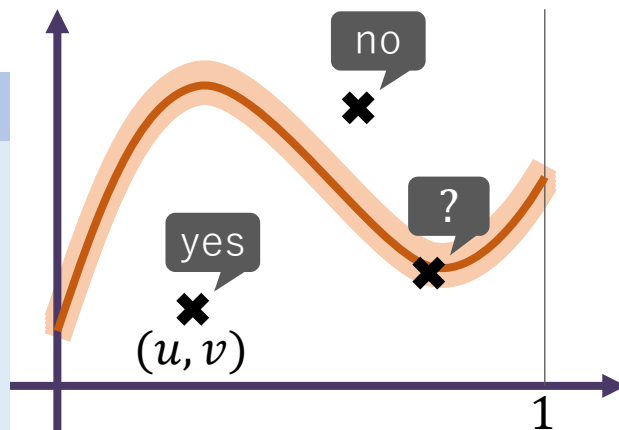




計算可能ならば連続

定理

- 計算可能な実関数は連続
- 多項式時間計算可能な函数 $f: [0, 1] \rightarrow \mathbf{R}$ は多項式の連続度すなわち次を満す多項式 $p: \mathbf{N} \rightarrow \mathbf{N}$ をもつ
$$|t - t'| < 2^{-p(m)} \Rightarrow |f(t) - f(t')| < 2^{-m}$$



定理

実函数 $f: [0, 1] \rightarrow \mathbf{R}$ が多項式時間 iff

- f は多項式の連続度をもち
- 次の約束つき判定問題 $(\text{Below}_f, \text{Above}_f)$ が (通常の一型の) $\mathbf{P}$ に属する

$$\text{Below}_f = \{ (u, v, 0^n) : v < f(u) - 2^{-n} \}$$

$$\text{Above}_f = \{ (u, v, 0^n) : v > f(u) + 2^{-n} \}$$



実函数へのどのような操作が 多項式時間を保つか？

最大値をとる

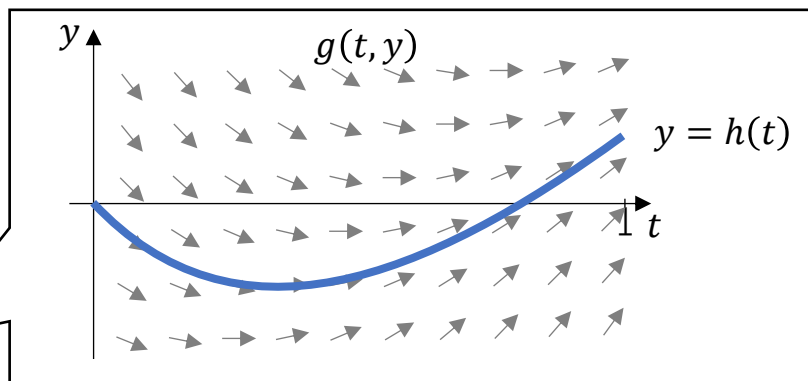
$$h(t) = \max_{0 \leq \tau \leq t} g(\tau)$$

積分する

$$h(0) = 0, \quad h'(t) = g(t)$$

微分方程式を解く

$$h(0) = 0, \quad h'(t) = g(t, h(t))$$



g が多項式時間ならば h も多項式時間といえるか？

上記三つの操作については これがそれぞれ **P = NP**

P = PP

P = PSPACE に同値

BPP の「 $\geq 2/3$ 対 $\leq 1/3$ 」の代りに
「 $> 1/2$ 対 $\leq 1/2$ 」にしたもの



定理 [Fri84]

もし $g: [0, 1] \rightarrow \mathbf{R}$ が多項式時間なら
 $h(0) = 0, \quad h'(t) = g(t)$
で定まる $h: [0, 1] \rightarrow \mathbf{R}$ も多項式時間

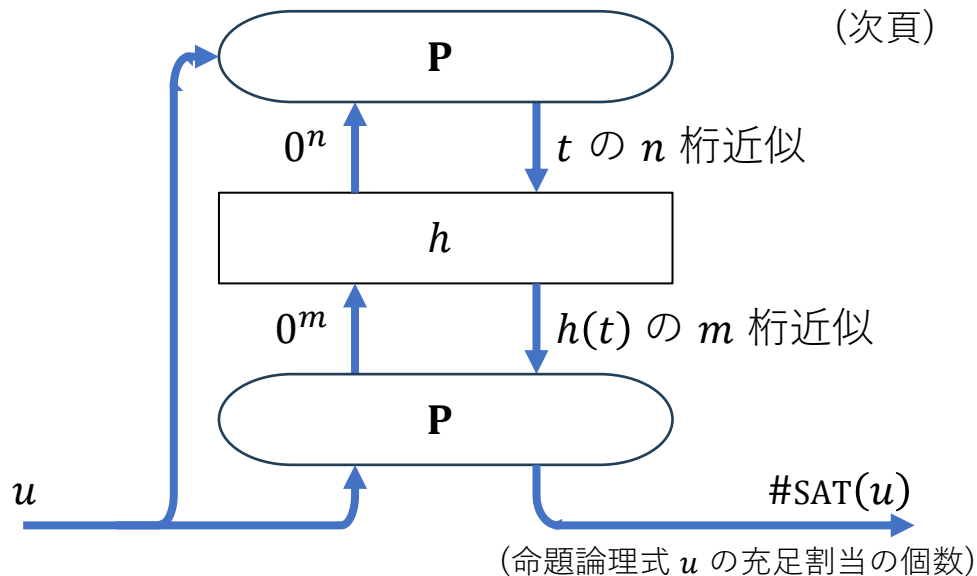
$\Leftrightarrow \mathbf{P} = \mathbf{PP}$

$\Leftarrow$

$h(t)$ の近似値は 適切な精度で
「 $u < t$ で $(u, v, 0^n) \in \text{Below}_g$ となる
 (u, v) の個数を数える」ことで得られる
 $\mathbf{P} = \mathbf{PP}$ ならばそれが多項式時間

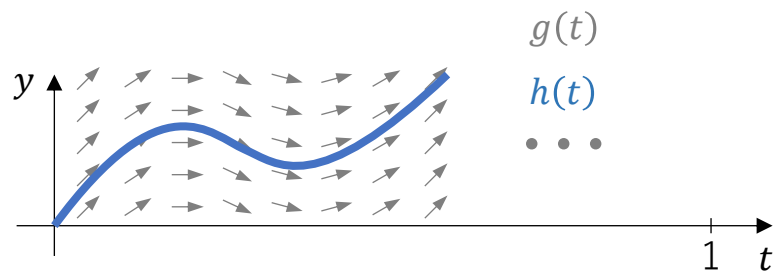
$\Rightarrow$

g が多項式時間だが
対応する h が次の意味で $\#P$ 困難なものを作る
(次頁)





#SAT を積分に帰着



作りたい状況

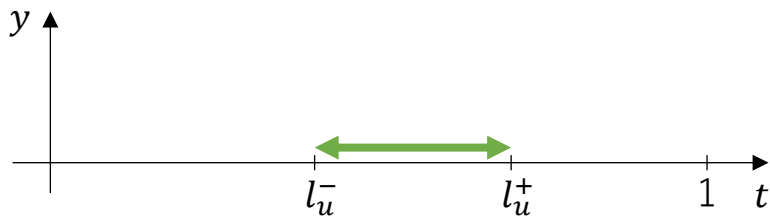
- ▶ g は単純 (多項式時間)
- ▶ h は #SAT 相当の情報をもつ



#SAT を積分に帰着

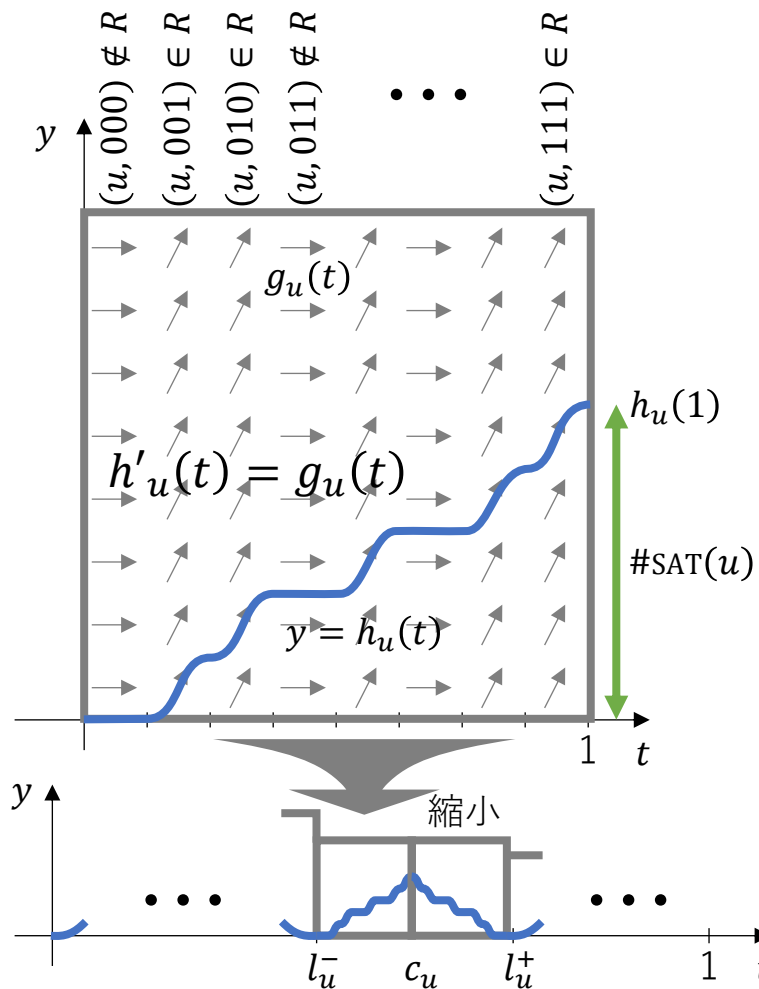
$\#SAT(u)$ は
 $(u, v) \in R$ なる v の個数 ($R \in \mathbf{P}$)

各 $u \in \Sigma^*$ に区間 $[l_u^-, l_u^+]$ を割当て





#SAT を積分に帰着



$\#SAT(u)$ は
 $(u, v) \in R$ なる v の個数 ($R \in \mathbf{P}$)

各 $u \in \Sigma^*$ に区間 $[l_u^-, l_u^+]$ を割当て
 g_u (を縮小したもの) を置く

$\#SAT(u)$ は $h_u(1)$ を見れば
(すなわち $h(c_u)$ を見れば) わかる



微分方程式の計算量

定理 [Ko83 / K10]

もしリプシッツ連続な $g: [0, 1]^2 \rightarrow \mathbf{R}$ が多項式時間ならば
で定まる $h: [0, 1] \rightarrow \mathbf{R}$ も多項式時間

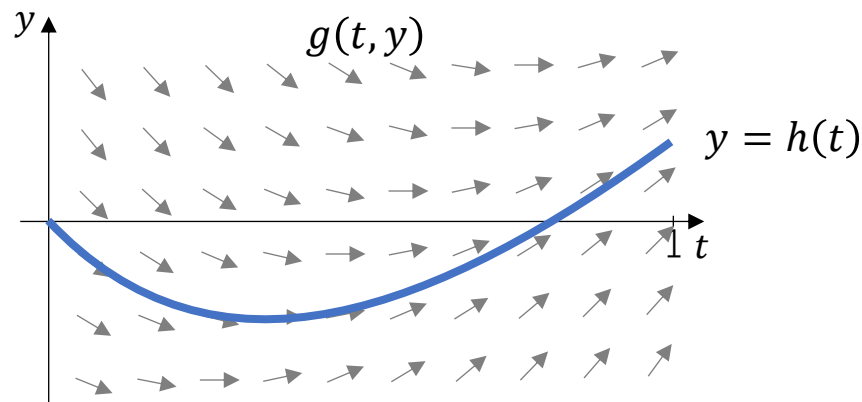
$$h(0) = 0, \quad h'(t) = g(t, h(t))$$

$$\Leftrightarrow \mathbf{P} = \mathbf{PSPACE}$$

定数 L が存在して

$$|g(t, y_0) - g(t, y_1)| \leq L|y_0 - y_1|$$

このとき解 h がただ一つ存在
することが知られている
(コーシー・リプシッツの定理)



⇐ オイラー法など単純な数値計算で
必要な精度を調べるとわかる

⇒ 次頁～

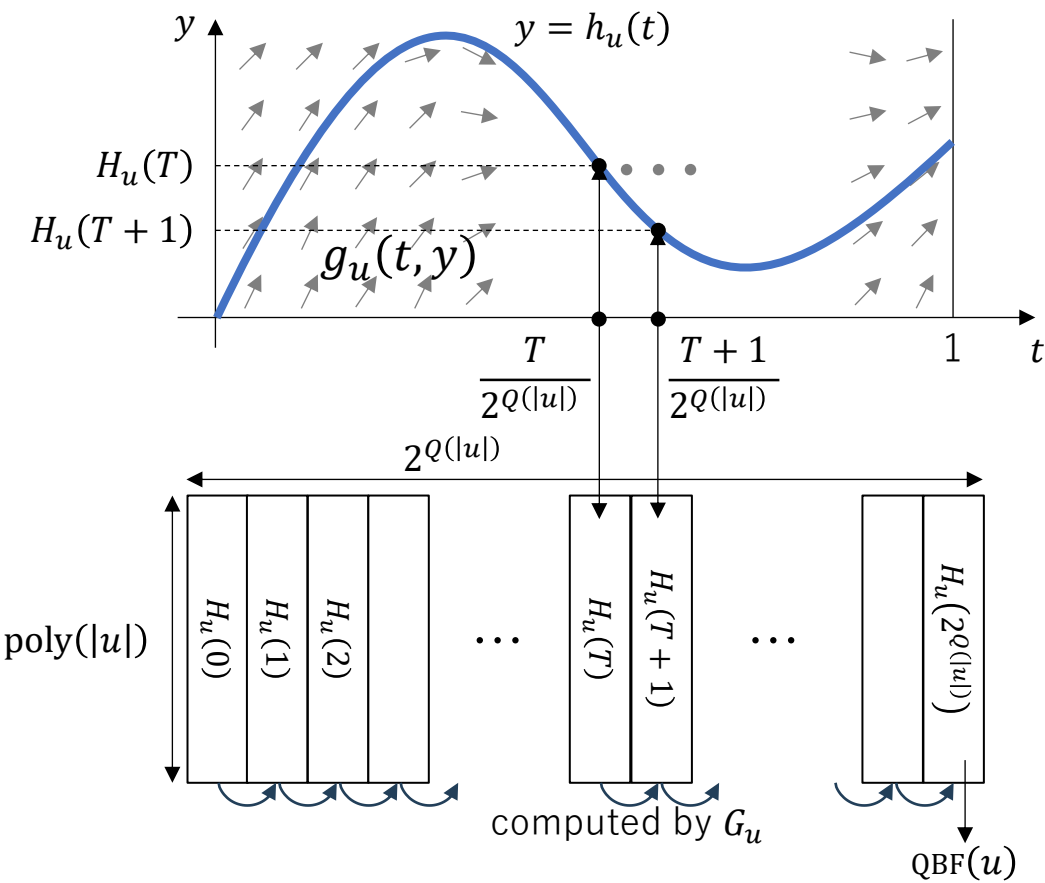
[Ko83] K. Ko. On the computational complexity of ordinary differential equations. Inform. Contr. 58, 1983.

[K10] A. Kawamura. Lipschitz continuous ordinary differential equations are polynomial-space complete. Comput. Complexity 19, 2010.



証明概要 (1/3): PSPACE 計算を微分方程式で模倣できるか？

先程と同じく 次のような部品 g_u を作りたい $h_u(1) \Leftrightarrow u \in \text{QBF}$.



微分方程式

$$h'_u(t) = g_u(t, h(t)) \quad (t \in [0, 1])$$

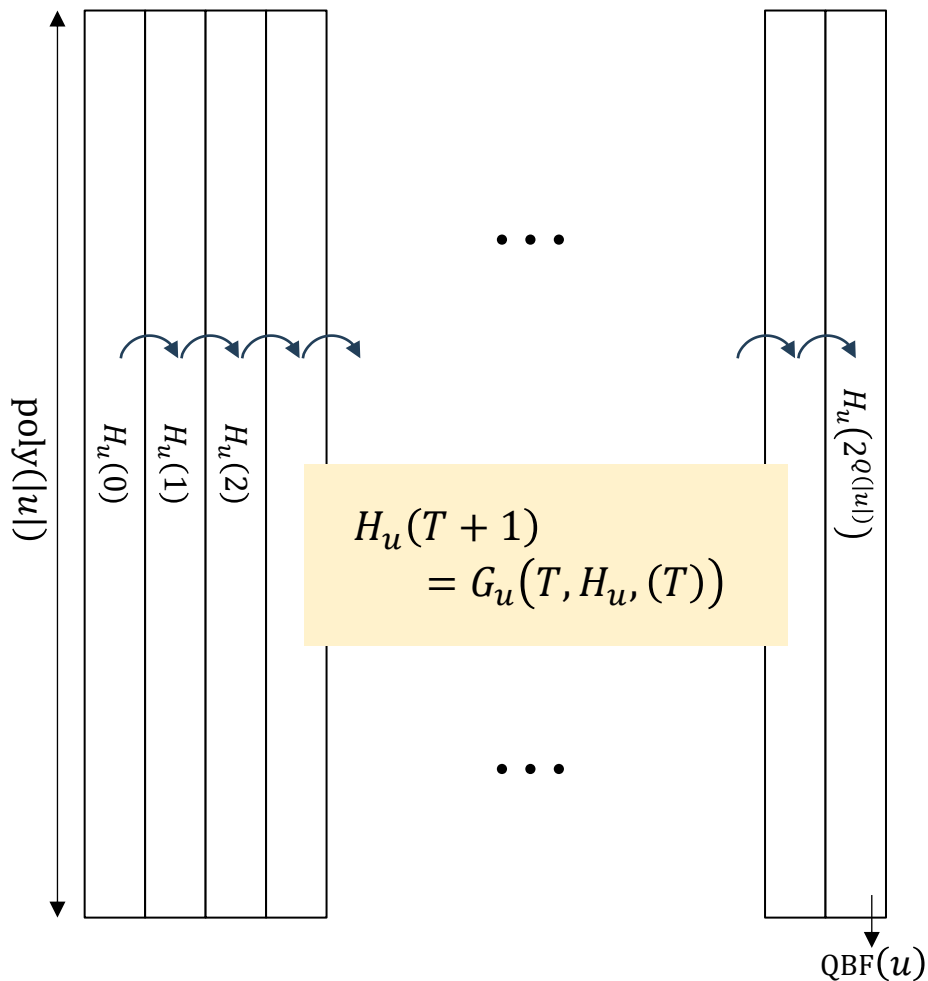
PSPACE 機械の遷移

$$H_u(T + 1) = G_u(T, H_u(T)) \quad (T = 0, 1, \dots, 2^{Q(|u|)} - 1)$$

Q は多項式

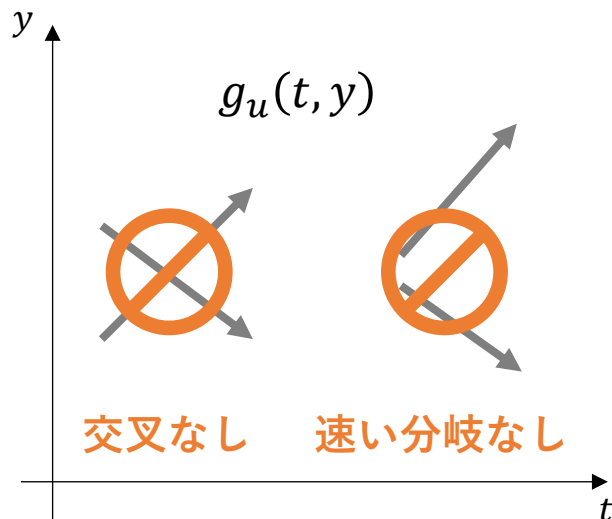


証明概要 (2/3)



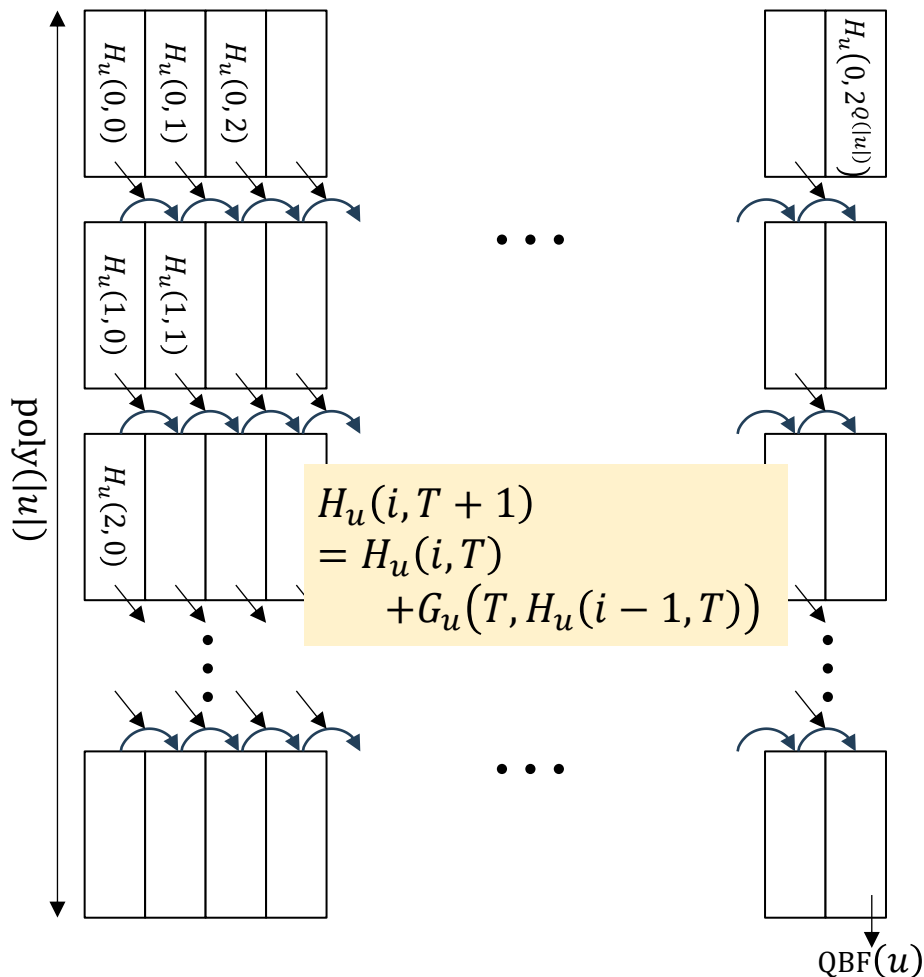
しかし

滑らかな微分方程式は
一般の **PSPACE** 計算に
相当する漸化式を
そのまま模倣することはできない





証明概要 (2/3)



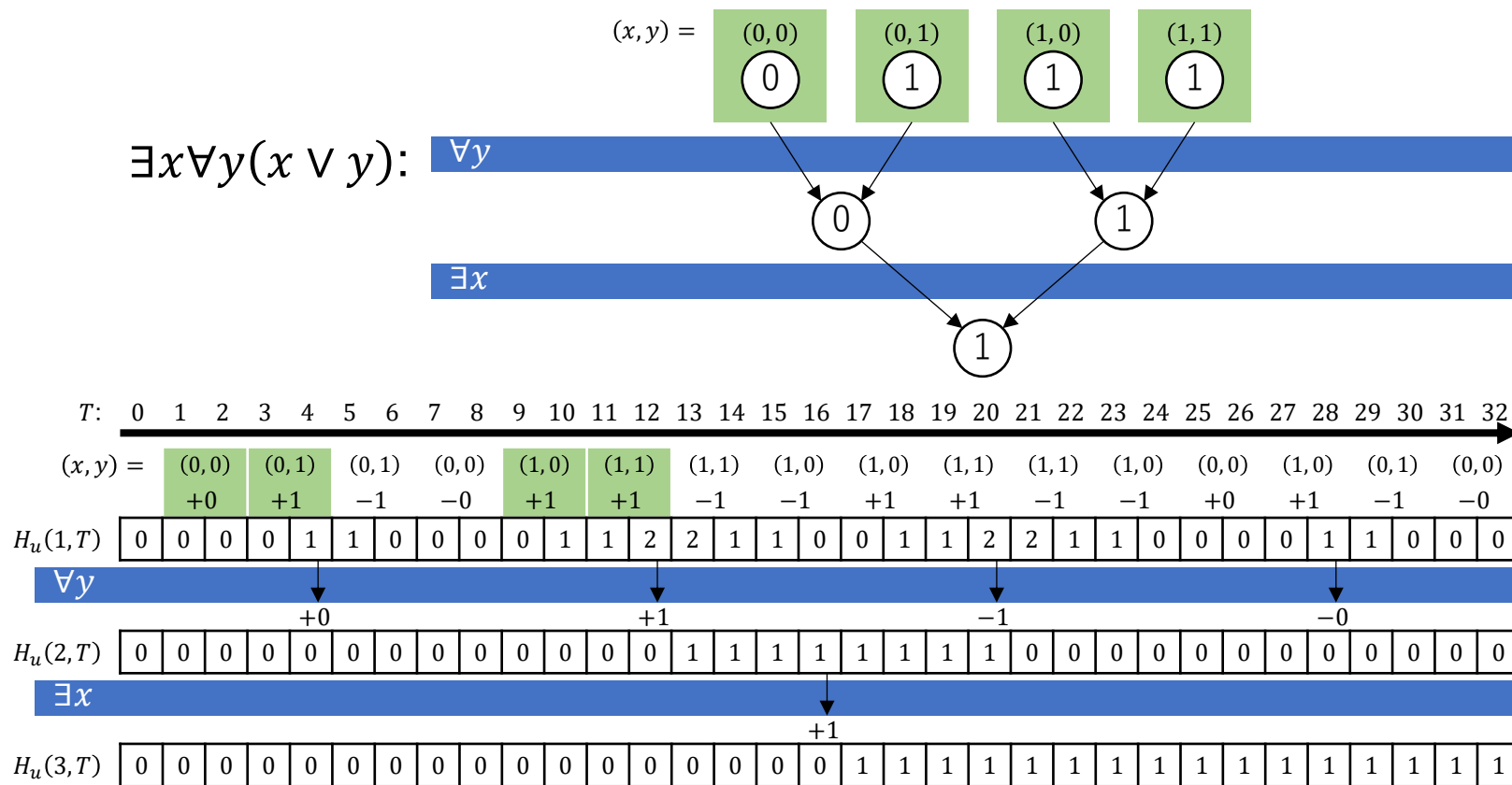
滑らかな微分方程式は
一般の **PSPACE** 計算に
相当する漸化式を
そのまま模倣することはできない

しかし
上層が下層の増減にのみ影響する
という限定された漸化式をなら
模倣できる



証明概要 (3/3)

この制限された漸化式であっても
PSPACE 完全問題である QBF を解くことができる





微分方程式の計算量

$$h(0) = 0, \quad h'(t) = g(t, h(t))$$

g が多項式時間のとき h は？



仮定	計算量の上界	計算量の下界
なし	—	解がみな計算不能であり得る [PR79]
解 h が唯一	計算可能	計算量は任意に大きくなり得る [Mil70]
g が「局所リプシッツ」	EXPSPACE [Ko 1992]	EXPSPACE 困難であり得る [Kaw10]
g がリプシッツ連続	PSPACE [Ko83]	PSPACE 困難 [Kaw10]
g が C^1 級		PSPACE 困難 [KORZ15]
g が C^k 級		CH 困難 [KORZ15]
g が解析的	P [Mül87]	—

[Ko92] K. Ko. On the computational complexity of integral equations. *Annals of Pure and Applied Logic* 58, 201–228, 1992.

[KORZ15] A. Kawamura, H. Ota, C. Rösnick and Martin Ziegler. Computational complexity of smooth differential equations. *Logical Methods in Computer Science*, 10, 2014.

[Mil70] W. Miller. Recursive function theory and numerical analysis. *Journal of Computer and System Sciences* 4, 465–472, 1970.

[Mül87] N. Müller. Uniform computational complexity of Taylor series. *Automata, Languages and Programming*, 435–444, 1987.

[PR79] M. B. Pour-El and I. Richards. A computable ordinary differential equation which possesses no computable solution. *Annals of Mathematical Logic* 17, 61–90, 1979



微分方程式の解の存在と計算量

$$h(0) = 0, \quad h'(t) = g(t, h(t))$$

コーシー・ペアノの定理

任意の g に対し解 h は
(原点の近くで) 存在

コーシー・リプシッツの定理

リプシッツ連続な g に対し
解 h が存在

コーシー・コワレフスカヤの定理

解析的な g に対し
な解 h が存在

様々な数学的現象の
データ操作としての側面を理解

コーシー・ペアノの定理(の 不成立) @計算可能性 [PR79]

$g \mapsto h$ は表現 δ_C の下で
計算不可能

数値計算の問題の難しさを
計算量により分類

$g \mapsto h$ は表現 δ_{CL} の下で多項
式空間

コーシー・コワレフスカヤの定理@P

$g \mapsto h$ は表現 $\delta_{analytic}$ の
下で多項式時間



まとめ 第三日 帰着と完全問題

- 帰着により問題の難しさを比べる
- $○○$ 完全 = 「 $○○$ なうちで最難」の問題
- 多様なジャンルの $○○$ 完全問題
見た目は違えど困難さの核心は同じ
- **PSPACE** 完全な問題
「到達性の判定」「二者間のゲーム」等の要素からくる困難さ
- **NP** 完全な問題
「指数個の候補から何かを見つけ出す」ことの困難さ