コンピュータサイエンス第２

南出 靖彦

第７回：後半

計算可能性理論，計算量理論

- 計算可能性理論
  - 判定不能問題：（プログラムで）計算できない問題
- 計算量理論：問題の難しさ
  - P: 多項式時間で解ける問題

1900 年	ヒルベルトの 23 問題の中の第 10 問題： ディオファントス方程式の整数解存在判定方法を求めよ。
1930 年代	チューリング機械や他の方法による「計算可能性」の定義， 停止性判定問題の計算不可能性証明。
1940 年代	プログラム内蔵方式コンピュータ登場。 （「万能チューリング機械」を実際に作ったものに相当）
1970 年	マチャセビッチ他による第 10 問題の否定的解決 （計算不可能性の証明）。
2000 年	クレイ数学研究所のミレニアム懸賞問題（賞金 100 万ドル） P $\neq$ NP 予想。

問題とは

問題 ：入力に対して Yes/No を問う形式に限る。

問題 Q が 計算可能（判定可能） である

- Q を解くアルゴリズムが存在することであり，正確には次のようなプログラム A が存在することである。
  - 問題 Q へのどんな入力 x についても
  - x を入力データとしてプログラム A を実行開始すると
  - 有限時間の計算の後に必ず正しい答 (Yes/No) を出力する。

ただし，プログラムは「理想的な実行環境」で実行するとする

- どんなに大きなデータやどんなに長いプログラムも扱える
- 実行途中で必要なメモリはいくらでも供給される

問題全体

- プログラムの停止性判定問題
- ディオファントス方程式の整数解存在判定問題
- 文字列の置換問題, ポストの対応問題

計算可能問題

有限時間で計算可能な問題

- プレスバーガー算術命題の真偽判定問題

NP 問題

一般に指数関数的に増加する有限の候補中の解の存在を問い，解が示されれば確認は多項式時間で計算可能な問題

- ハミルトン閉路問題, 巡回セールスマン問題, 部分和問題

P 問題　　（この境界線が未確定：**P**≠**NP** 予想）

多項式時間で計算可能な問題

- オイラー閉路問題

計算の限界：判定不能問題

- プログラムの停止性判定問題
- ディオファントス方程式の整数解存在判定問題
- 文字列の置換問題
- ポストの対応問題
- ...

【停止性判定問題】

入力 プログラム P とそれへの入力データ x。

出力 理想的な実行環境において P を入力 x で実行開始すると有限ステップで停止するか否か。

【ディオファントス方程式の整数解存在判定問題】

入力 ディオファントス方程式（整係数多変数高次不定方程式）E

- 例: $X^5 + Y^5 = Z^5$

出力 E が整数解を持つか否か。

ポストの対応問題

入力 二つの文字列のリスト

$$v_1, v_2, \ldots, v_k$$

$$w_1, w_2, \ldots, w_k$$

出力 以下を満たす整数列 $i_1, i_2, \ldots, i_m$ $(m \geq 1)$ が存在するか

$$v_{i_1} v_{i_2} \cdots v_{i_m} = w_{i_1} w_{i_2} \cdots w_{i_m}$$

例

- $v_i = 1, 10111, 10$ と $w_i = 111, 10, 0$

$$v_2 v_1 v_1 v_3 = 101111110 = w_2 w_1 w_1 w_3$$

ポストの対応問題

問題

1. b, a, ca, abc と ca, ab, a, c
2. $10, 011, 101$ と $101, 11, 011$

ポストの対応問題

問題

1. b, a, ca, abc と ca, ab, a, c
2. $100, 0, 1$ と $1, 100, 00$

答え

1.

```
a    b    ca   a    abc
ab   ca   a    ab   c
```

```
abcaaabc
```

2.

```
100  1   100  100  1   0    0
1    00  1    1    00  100  100
```

```
1001100100100
```

本当に難しい？

例 1 ：

- 0, 01, 1
- 1, 0, 101

例 2 ：

- 10, 0, 001
- 0, 001, 1

本当に難しい？

例 1 ：

- 0, 01, 1
- 1, 0, 101

$\Rightarrow$ 最短の解：長さ 44

例 2 ：

- 10, 0, 001
- 0, 001, 1

$\Rightarrow$ 解があるか分かっていない (2001 年)

プログラムの停止性

プログラム例: d.py

入力: 自然数 x

```
x = 入力 x
a = 0
b = 0
while a != x:
    b = b + 1
    a = a + 2
```

b を出力し停止

- 偶数 x を対して，$x/2$ を計算
- 奇数に対しては，止まらない.

プログラムが行う計算を関数として表現

$$d.py(6) = 3$$

$$d.py(7) = \bot$$

- $\bot$: 止まらない

インタープリタ

インタープリタ：プログラムを実行するプログラム

Python のインタープリタ: python

$$\mathrm{python}(\overline{P}, x) = P(x)$$

- P: プログラム
- $\overline{P}$: プログラムを表現する記号列
- x: プログラム P に対する入力

例：

$$\mathrm{python}(\overline{d.py}, 6) = d.py(6) = 3$$

$$\mathrm{python}(\overline{d.py}, 7) = d.py(7) = \bot$$

停止性判定問題

停止性判定問題：プログラム P と入力 x が与えられた時，入力 x に対する P の実行が停止するかを判定する問題

- 停止性判定問題は，計算 (判定) 不能である．(アラン チューリング, 1936 年)

以下の計算を行うプログラム halt は存在しない

$$halt(\overline{P}, x) = \begin{cases} 0 & P(x) = \bot \text{ の時} \\ 1 & P(x) \neq \bot \text{ の時} \end{cases}$$

計算不可能性の証明

背理法で証明：プログラム P_0 が halt を計算すると仮定して，矛盾を導く．

プログラム P_0 から以下の計算を行う Q_0 を作る．

1. 入力 z に対して，$P_0(z, z)$ を計算する．
2. $P_0(z, z)$ の出力に応じて以下のどちらかを行う．
   - 2.1 1 のとき，無限ループに入るようにする．
   - 2.2 0 のとき，停止する．

$Q_0(\overline{Q_0})$ を計算すると矛盾がでてくる

計算不可能性の証明

- $Q_0(\overline{Q_0})$ が止まる場合

$$
\begin{aligned}
Q_0(\overline{Q_0}) \neq \perp \quad &\Rightarrow \quad \mathrm{halt}(\overline{Q_0}, \overline{Q_0}) = 1 \\
&\Rightarrow \quad P_0(\overline{Q_0}, \overline{Q_0}) = 1 \\
&\Rightarrow \quad Q_0(\overline{Q_0}) \text{ は無限ループする} \\
&\quad\ \ (Q_0 \text{ の作り方から})
\end{aligned}
$$

- $Q_0(\overline{Q_0})$ が止まらない場合

$$
\begin{aligned}
Q_0(\overline{Q_0}) = \perp \quad &\Rightarrow \quad \mathrm{halt}(\overline{Q_0}, \overline{Q_0}) = 0 \\
&\Rightarrow \quad P_0(\overline{Q_0}, \overline{Q_0}) = 0 \\
&\Rightarrow \quad Q_0(\overline{Q_0}) \neq \perp
\end{aligned}
$$

Q_0:

1. 入力 z に対して，$P_0(z, z)$ を計算する．
2. $P_0(z, z)$ の出力に応じて以下のどちらかを行う．
   - 2.1 1 のとき．無限ループに入るようにする．
   - 2.2 0 のとき．停止する．

問題全体

- プログラムの停止性判定問題
- ディオファントス方程式の整数解存在判定問題
- 文字列の置換問題, ポストの対応問題

計算可能問題

有限時間で計算可能な問題

- プレスバーガー算術命題の真偽判定問題

NP 問題

一般に指数関数的に増加する有限の候補中の解の存在を問い，解が示されれば確認は多項式時間で計算可能な問題

- ハミルトン閉路問題, 巡回セールスマン問題, 部分和問題

P 問題　　（この境界線が未確定：**P**≠**NP** 予想）

多項式時間で計算可能な問題

- オイラー閉路問題

P 問題

入力サイズの多項式オーダーの時間で答が求まる問題を **P** 問題と呼ぶ。

- P は Polynomial（多項式）の頭文字。
- 正確には，問題 Ψ が P 問題であるとは次のようなプログラム A が存在することである。

- 多項式 $f(x) = c_n x^n + c_{n-1} x^{(n-1)} + \cdots + c_1 x + x_0$ が存在して，
- 問題 Ψ に対するどんな入力 x に対しても，
- x を入力データとして A を実行開始すると
- $f(|x|)$ ステップ以内に停止して正しい答（Yes/No）を出力する。
  ただし $|x|$ は x の大きさ（文字列としての長さ）。

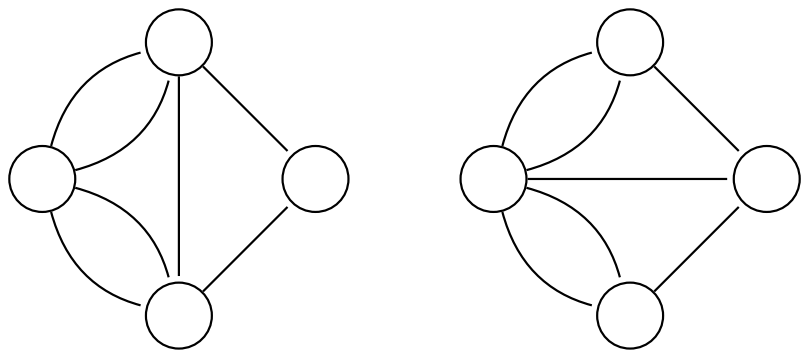
P 問題の例

【オイラー閉路問題】

入力　グラフ G

出力　G にオイラー閉路が存在するか否か

- すべての辺をちょうど 1 回ずつ通って出発点に戻る経路（出発点に戻る一筆書き）



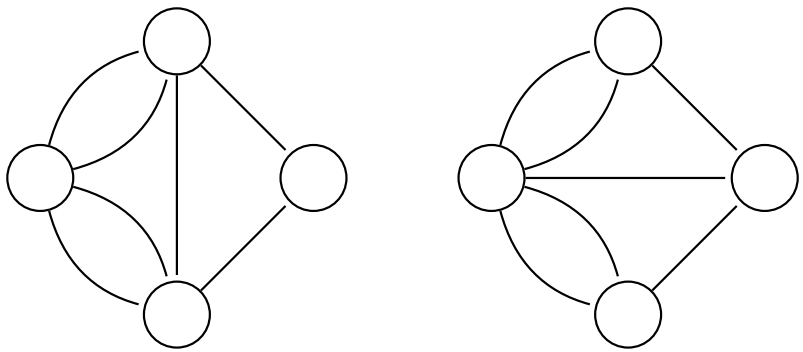
P 問題の例

【オイラー閉路問題】

入力　グラフ G

出力　G にオイラー閉路が存在するか否か

- すべての辺をちょうど 1 回ずつ通って出発点に戻る経路（出発点に戻る一筆書き）



グラフがオイラー閉路を持つ

$\Updownarrow$

全ての頂点の次数が偶数

- 次数：各頂点から出ている辺の本数

$\Rightarrow$ すべての頂点の次数が偶数か確認すれば良い.

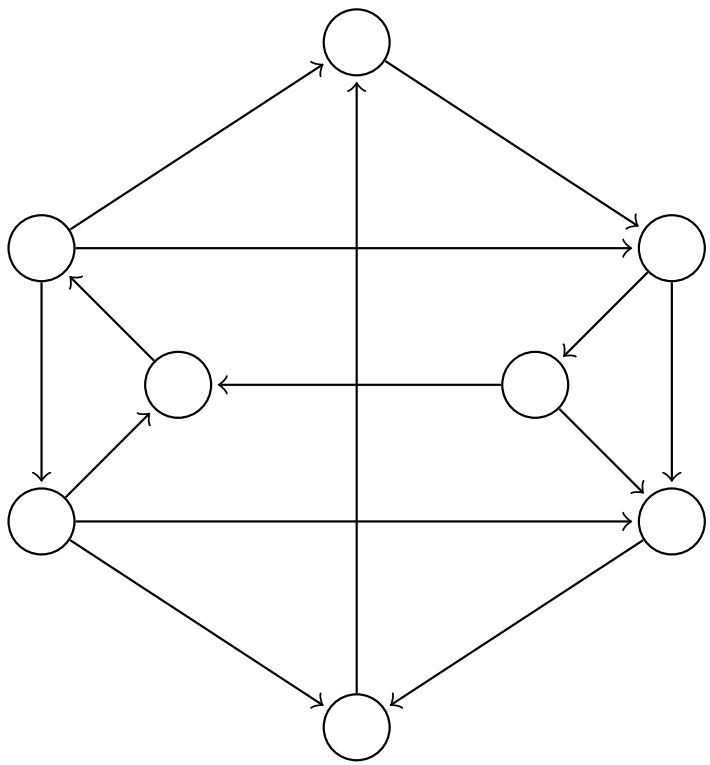
多項式時間で解けないと考えられている問題

【ハミルトン閉路問題】

入力　グラフ G

出力　G にハミルトン閉路が存在するか否か

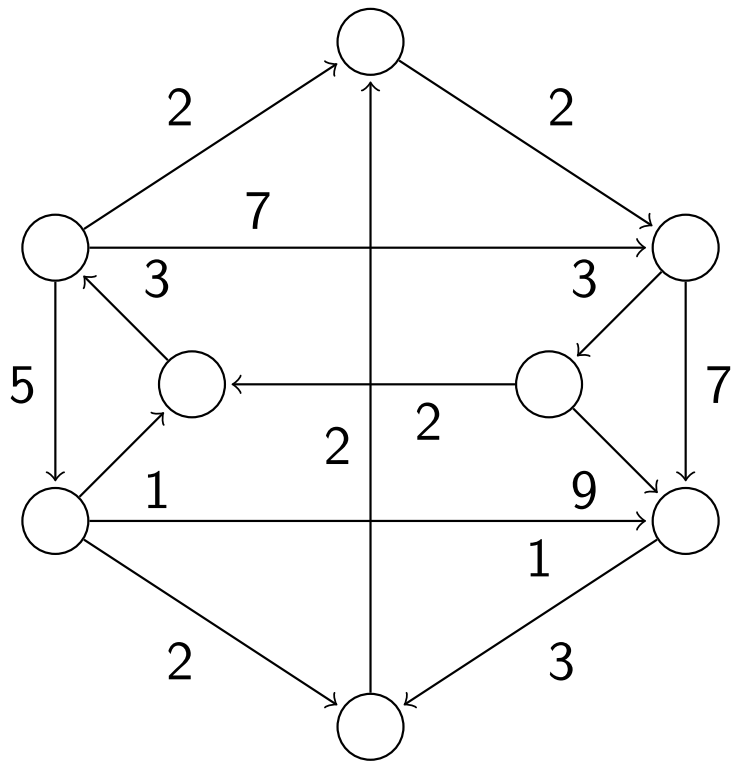
- すべての頂点をちょうど 1 回ずつ通って出発点に戻る経路



【巡回セールスマン問題】

入力 n 個の都市間の移動に要するコスト（時間または費用）の一覧（$\frac{n(n-1)}{2}$ 個の自然数），および使用可能なコストを表す自然数 c

出力 すべての都市をちょうど 1 回ずつ訪れて出発点に戻ってきてその移動に要するコストの総和が c 以下になる経路が存在するか否か



【部分和問題】

入力 自然数 m, および自然数 $a_1, a_2, \ldots, a_n$

出力 $\{a_1, a_2, \ldots, a_n\}$ の部分集合で全要素の和が m になるものが存在するか否か

NP 問題

問題 Ψ が **NP** 問題であるとは次の条件を満たすプログラム A が存在することである。

- NP は Non-deterministic（非決定性） Polynomial の頭文字。

1. A は非決定的な文（「ふたつの文のうちどちらかを実行せよ」という命令）を含んでいてもよい。
   - したがって同じ入力で何回も実行すると，そのたびに実行結果が異なってもよい。
2. 多項式 $f(x)$ が存在して，問題 Ψ に対するどんな入力 x に対しても，x を入力データとして A を実行開始すると，$f(|x|)$ ステップ以内に停止して答を出力する。
3. その出力された答は次の条件を満たす。
   - 入力 x に対する正解が Yes の場合：Yes を出力する可能性がゼロでない，つまり非決定的な部分でうまい選択をすれば Yes を出力することができる。
   - 入力 x に対する正解が No の場合：絶対に No を出力する，つまり非決定的な部分でどんな選択をしても No を出力する。

部分和問題は NP 問題

部分和問題を解く非決定性 Python もどきプログラム

```
print("和の目標を入力してください")
goal = int(input())
print("複数の自然数を空白で区切って入力してください")
a = list(map(int, input().split()))
n = len(a)
# 以上で入力自然数が goal と a[0],a[1],...,a[n-1] に入る
sum = 0
for i in range(n):
  次のどちらかを実行
    sum = sum + a[i]
  または
    sum = sum
# 以上で a[0],...,a[n-1] から非決定的に選んだひとつの部分集合の和が sum に
入る
if (sum == goal):
    print("Yes")
else:
    print("No")
```

NP 問題（の中で難しい問題）はたいてい次の形をしている：

- 条件を満たす組み合わせが存在するか否かを Yes/No で答える
- 組み合わせは有限個なので全検索すれば正解が答えられる
- 組み合わせ総数は入力サイズに対して一般に指数関数的に増えるので，全検索の方法は多項式時間でできない。
- ただし，組み合わせがひとつだけ与えられた場合に「それが条件を満たしているか？」という計算は多項式時間でできる。

P≠NP予想

- P 問題は定義上 NP 問題でもある。
- P≠NP予想：逆（「NP 問題は P 問題である」）は成り立たない.

P≠NPだとすると

- NP 問題の中の難しい問題は，指数関数的な時間をかけて全検索（に近いこと）をするしか解法がない

計算可能 & NP より難しい

【プレスバーガー算術命題の真偽判定問題】

入力 プレスバーガー算術命題 A

- 以下を組み合わせて作られる命題
  - 自然数定数，自然数変数，$+$，$=$，$<$，$\neq$
  - かつ，または，ならば，$\forall$，$\exists$
- 掛け算は使えない

出力 A が真であるか偽であるか

例：プレスバーガー算術命題

$$\forall n(\exists k(n = k + k \vee n = k + k + 1))$$

- すべての自然数は，偶数であるか奇数である．

アンケート