コンピュータサイエンス第２

南出 靖彦

第 6 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

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 $w_i = 111, 10, 0$

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

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

⇒ 最短の解：長さ 44

例 2 ：

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

⇒ 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.rb`

入力: 自然数 x
```
x = 入力 x
a = 0; b = 0
while a != x
    b = b + 1
    a = a + 2
end
```
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

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

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

- $\bot$: 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

Ruby のインタープリタ: ruby

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

- 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：

$$\mathrm{ruby}(\overline{d.rb}, 6) = d.rb(6) = 3$$

$$\mathrm{ruby}(\overline{d.rb}, 7) = d.rb(7) = \bot$$

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

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

- $Q_0(\overline{Q_0})$ 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．

文字列の置換問題（semi-Thue system の語の問題）

入力 文字列の有限個の置換ルール（特定の文字列を特定の文字列に書き換える）および開始文字列と終了文字列。

出力 置換ルールを何回か用いて開始文字列から終了文字列へ書き換える事ができるか否か。

置換問題の具体例

ルール：$(\mathrm{aab} \to \mathrm{bab})$, $(\mathrm{abb} \to \mathrm{bb})$, $(\mathrm{ba} \to \mathrm{bab})$.
開始文字列：aaabbb,
終了文字列：bbbb.

これに対して

$$\mathrm{aa}\underline{\mathrm{abb}}\mathrm{b} \to \underline{\mathrm{aab}}\mathrm{bb} \to \mathrm{b}\underline{\mathrm{abb}}\mathrm{b} \to \mathrm{bbbb}$$

は
と書き換えられるので正解は Yes。

計算不可能性の証明

もしも文字列置換問題を解くアルゴリズムが存在したら，それを使うことでプログラム停止性判定問題を解くことができてしまうことを示す。

- これを「プログラム停止性判定問題は文字列置換問題に還元可能である」と言う

計算不可能性の証明

- コンピュータの動作は，メモリ上に書かれた 0,1 文字列の書き換えと見なすことができる
- この書き換えは有限個の規則（これを $R_1, R_2, \ldots, R_n$ とする）で記述される。
- 停止性判定問題の入力として与えられた P と x から，
  - 「P を x で実行開始した場合のメモリの初期内容の 0,1 文字列」（これを α とする）
  - 「それが止まった場合の最終的なメモリ内容の 0,1 列」（これを β とする）

  を作ることができる。
- $\langle R_1, R_2, \ldots, R_n, \alpha, \beta \rangle$ を文字列置換問題を解くアルゴリズムに渡して書き換え可能かどうかを計算する.
- 停止性判定問題の答が得られる。