オラクルとP対NP問題

ごたんとう（@mathonigori）

2019年6月13日

概要

$\mathrm{P}^A = \mathrm{NP}^A, \mathrm{P}^B \neq \mathrm{NP}^B$ をみたすオラクル A, B の存在から，P 対 NP 問題がオラクルの有無に影響されない方法では解決できないと分かる．ここではその証明を書く．$\mathrm{P}^A = \mathrm{NP}^A$ となる A の存在を示すのは簡単だが，$\mathrm{P}^B \neq \mathrm{NP}^B$ をみたす B が存在することの証明はテクニカルで面白い．

定理 1. $\mathrm{P}^A = \mathrm{NP}^A$ をみたすオラクル A が存在する．

証明 . A として PSPACE 完全な言語をとる．A は PSPACE 完全なので多項式時間で PSPACE 問題を A へ帰着でき，$\mathrm{PSPACE} \subseteq \mathrm{P}^A$ が成立する．また，$\mathrm{P}^A \subseteq \mathrm{NP}^A$ が定義よりわかる．さらに，オラクル A への問い合わせに対する答えを計算する非決定性多項式領域 *Turing* 機械が存在することと $\mathrm{NP} \subseteq \mathrm{NPSPACE}$ より $\mathrm{NP}^A \subseteq \mathrm{NPSPACE}$ である．また，*Savitch* の定理より $\mathrm{NPSPACE} \subseteq \mathrm{PSPACE}$．以上より，

$$\mathrm{PSPACE} \subseteq \mathrm{P}^A \subseteq \mathrm{NP}^A \subseteq \mathrm{NPSPACE} \subseteq \mathrm{PSPACE}$$

であり，$\mathrm{P}^A = \mathrm{NP}^A$ と分かる． □

定理 2. $\mathrm{P}^B \neq \mathrm{NP}^B$ をみたすオラクル B が存在する．

証明 . $M_1^?, M_2^?, \dots$ をすべてのオラクル付き *Turing* 機械が現れる列であって，任意の $i \in \mathbb{N}$ について $M_i^?$ と同じ機械が無限回現れるものとする．そのような列が存在することは，オラクル付き *Turing* 機械を符号化して自然数と対応させられることからわかる．以下では帰納的に B_n, X_n を定義し，最終的に $B = \bigcup B_n, L = \{0^n | \exists x \in B \, s.t. |x| = n\}$ と定めることで $L \in \mathrm{NP}^B \setminus \mathrm{P}^B$ であることを示す．

(1) $n = 0$ のとき

　$B_0 = X_0 = \emptyset$ と定める．

(2) $n = i - 1$ まで B_n, X_n が定まっているとき

　まず，$X_i = X_i - 1$ としておく．$M_i^{B_{i-1}}$ に入力として 0^i を与えて，$i^{\log i}$ ステップまで動作させる．動作の中でオラクルへ文字列 x が B_{i-1} に入っているかの問い合わせがあるとき，$|x| \geq i$ であれば X_i を $X_i \cup \{x\}$ とする．

　<u>① $i^{\log i}$ ステップ以内に $M_i^{B_{i-1}}$ が 0^i を拒否した場合</u>

$B_i := B_{i-1} \cup \{x \in \{0,1\}^* | \ |x| = i, x \notin X_i\}$ と定義する．$\displaystyle\sum_{j=1}^{i} j^{\log j} < 2^i$ より $\{x \in \{0,1\}^* | \ |x| = i, x \notin X_i\} \neq \emptyset$ をみたすので，あとで定義する L が $0^i \in L$ をみたすようになる．

② $i^{\log i}$ ステップ以内に $M_i^{B_{i-1}}$ が 0^i を受理した場合

$B_i := B_{i-1}$ と定義する．この定め方によって，あとで定義する L が $0^i \notin L$ をみたすようになる．

③ $i^{\log i}$ ステップ以内に $M_i^{B_{i-1}}$ が停止しなかった場合

$B_i := B_{i-1}$ と定義する．

$(1),(2)$ から帰納的に B_n, X_n が定義された．ここで，

$$B := \bigcup B_n,\ L := \{0^n | \ \exists x \in B\ s.t.\ |x| = n\}$$

と定義しよう．すると，多項式時間 $p(n)$ で動く任意の M_i^B について，$L(M_i^B) \neq L$ が言える．まず，$n = i$ での定義を与えるときに①か②の場合であったならば，入力を 0^i としたときを考えることで $L(M_i^B) \neq L$ が分かる．では，③の場合ではどうだろうか．この場合は，

$$\exists I \in \mathbb{N}\ s.t\ I^{\log I} > p(I) \text{ かつ } M_I^B = M_i^B$$

であることから，$n = I$ で①か②の場合の定義をすることになり，同じ機械 M_I^B について $L(M_I^B) \neq L$ と言える．よって任意の多項式時間オラクル *Turing* 機械が L を判定しないと分かり,$L \notin \mathrm{P}^B$ ．また，非決定的にある長さの文字列が A の元であるか問い合わせをすることができることから $L \in \mathrm{NP}^B$ が言える．以上から $\mathrm{P}^B \neq \mathrm{NP}^B$ が示された． □

定理 1 と定理 2 より，オラクルの有無が影響しない手法では P 対 NP 問題を解くことができないとわかる．

参考文献

[1] Christos H. Papadimitriou,"Computational Complexity. Addison-Wesley",pp.339 - 343, 1994.

[2] Michael Sipser, "Introduction to the Theory of Computation", PWS Publishing, pp.318 - 321, 1997.