受限外部信息源计算的P—NP性质*

吕义忠 刘建斌

(南京大学数学系,南京 210008)

摘要 本文对Oracle图灵机在接受计算中的查询次数加以限制,并且得到结果:存在无穷多个非多项式等价的递归集A, B, A′, B′, A″, B″, A‴, B‴,它们满足性质:$P(A,q)=P(A,q+1)$, $P(B,q)\neq P(B,q+1)$, $P(A',q)=P(A')$, $P(B',q)\neq P(B')$. $NP(A'',q)=NP(A'',q+1)$, $NP(B'',q)\neq NP(B'',q+1)$, $NP(A''',q)=NP(A''')$, $NP(B''',q)\neq NP(B''')$.

关键词 计算复杂性,P—NP问题,递归集,外部信息源.

受限外部信息源的结构复杂类的研究工作在国际上已取得许多重要的成果,如(1)$P=NP\Leftrightarrow\forall\,{}^{}_{A}(P(A)=NP_B(A))$[4]. (2)$\exists\,{}^{}_{S}(S$为稀疏集$\wedge\forall\,{}^{}_{A}(NP(S)\neq NP_B(A)))$. (3)$\exists\,{}^{}_{A}(A$为递归集$\wedge\forall\,{}^{}_{S}(S$为稀疏集$\rightarrow NP_B(A)\neq NP(S)))$.[10] (4)$P=PSPACE\Leftrightarrow\forall\,{}^{}_{A}(P(A)=PQUERY(A))$. (5)$NP=PSPACE\Leftrightarrow\forall\,{}^{}_{A}(NP(A)=NPQUERY(A))$[1]等. 本文则在上述结果的基础上深入讨论了当对Oracle图灵机的查询次数加以限制时,相应的结构复杂类的P—NP性质,并且得到以下结果:

(1)存在递归Oracle A,B,使得$P(A,q)=P(A,q+1)$且$P(B,q)\neq P(B,q+1)$.

(2)存在递归Oracle A,B,使得$P(A,q)=P(A)$且$P(B,q)\neq P(B)$.

(3)存在递归Oracle A,B,使得$NP(A,q)=NP(A,q+1)$且$NP(B,q)\neq NP(B,q+1)$.

(4)存在递归Oracle A,B,使得$NP(A,q)=NP(A)$且$NP(B,q)\neq NP(B)$.

(5)有无穷多个非多项式等价的递归Oracle A,B,使得上述(1)—(4)成立.

1 定 义

我们假定读者熟悉计算复杂性的一些基本概念和符号. 设$\sum=\{0,1\}$,则$\sum^*$表示$\sum$上一切字的集合. 若无特别声明,本文中的集合皆指$\sum^*$的子集. 如果A,B为集合,则$A\leqslant^p_m B$和$A\leqslant^p_T B$分别表示A到B的多一化归和图灵化归,而$A<^p_m B$和$A<^p_T B$则分别表示$A\leqslant^p_m B$但$B\not\leqslant^p_m A$和$A\leqslant^p_T B$但$B\not\leqslant^p_T A$. 对于字$w\in\sum^*$,我们用$|w|$表示w的长度. 对任何

* 本文1991年7月27日收到,1991年12月21日定稿

作者**吕义忠**,57岁,副教授,主要研究领域为数理逻辑,结构复杂性,ULAM问题. **刘建斌**,29岁,1990年硕士毕业于南京大学,主要研究领域为数理逻辑与计算机.

本文通讯联系人:吕义忠,南京 210008,南京大学数学系

集合 A,我们分别用 A^n,$A^{\leqslant n}$ 和 $A^{<n}$ 表示 A 中长度为 n,长度至多为 n,和长度小于 n 的一切字的集合. 我们用 P 和 NP 分别表示确定的和不确定的多项式时间图灵机所接受的集合的类. 一个 Oracle 图灵机(简称 OT 机)是一个多带图灵机,它有一条特殊的询问带和三个特殊状态:Query, Yes 和 No 状态,我们用 L(M,A)表示以 A 为 Oracle 的 T 机 M 所接受的集合. 一个确定的(或不确定的)受限查询的 OT 机是指限制其对 Oracle 的查询次数的确定的(或不确定的)OT 机,更精确地,设给定 Oracle A 及查询函数 q:N→N,则以 A 为 Oracle 的 OT 机 M 受限于 q 而接受 x 当且仅当至少存在 M 对 x 的一个接受计算,使得 M 在此计算中查询 A 的次数≤q(|x|),而集合 L(M,A,q)则表示以 A 为 Oracle,以 q 为查询函数的 OT 机 M 所接受的语言;类似地,我们可定义 P(A,q)={L|存在确定的多项式时间的 OT 机 M,使得 L=L(M,A,q)},NP(A,q)={L|存在不确定的多项式时间的 OT 机 M,使得 L=L(M,A,q)},由这些定义立即得到以下推论:

推论 1. 对任何递归集 A,B 和任何自然数函数 $q_1,q_2:N\to N$, 均有

(1) $P\subseteq P(A,q_1)\subseteq P(A)$ 且 $NP\subseteq NP(A,q_1)\subseteq NP(A)$.

(2)如果 $q_1\leqslant q_2$,则 $P(A,q_1)\subseteq P(A,q_2)$ 且 $NP(A,q_1)\subseteq NP(A,q_2)$.

(3)如果 $A\leqslant_m^p B$,则 $P(A,q)\subseteq P(B,q)$ 且 $NP(A,q)\subseteq NP(B,q)$

(4)如果 $A\equiv_m^p B$,则 $P(A,q)=P(B,q)$ 且 $NP(A,q)=NP(B,q)$.

对任何集合 A,B,通常 A⊕B={0x|x∈A}∪{1x|x∈B},且显然有以下性质:

推论 2. 对任何集合 A,B,C 均有

(1) $A\leqslant_m^p A\oplus B$ 且 $B\leqslant_m^p A\oplus B$.

(2) 若 $A\leqslant_m^p C$ 且 $B\leqslant_m^p C$,则 $A\oplus B\leqslant_m^p C$.

还有一些较特殊的概念和符号将在需要时再给出它们的定义.

2 结　果

定理 1. 存在递归 Oracle A,使得 P(A,q)=P(A,q+1)=P(A)且 NP(A,q)=NP(A,q+1)=NP(A).

证明:取 A∈P 则 P=P(A)且 NP=NP(A),故由推论 1 即得证.

不失一般性,以下假定查询函数 q:N→N 为多项式可计算函数且满足∀ n∈N(q(n)≤p(n)),其中 p 为多项式.

定理 2. 存在递归 Oracle B, 使得 P(B,q)≠P(B,q+1).

证明:对任何递归 Oracle B 和常数 c,我们定义

$L(B)=\{o^n|n\geqslant c$ 且 $\sum^n$ 中按字典次序排列的首 q(n)+1 个字均∈B},则以下算法证明了,对任何递归集 B,L(B)∈P(B,q+1):

输入 x;

n:=|x|;

检查 n≥c 和 $x=o^n$,若否,则拒绝;

For i:=1 to q(n)+1 do

　$y:=\sum^n$ 中第 i 个字;

询问 y∈B 吗? 若否,则拒绝;END;

接受;End.

现在,设 C 为充分大的自然数,使得当 n≥C 时,均有 q(n)+1≤2^n. 又设 M_i 为一切限制查询的多项式时间 p_i 的 OT 机的能行枚举. 不失一般性,可假定 $p_i(n) \leqslant p_{i+1}(n)$ 及 p_i 为不减多项式. 我们如下构造递归集 B,使得 L(B)∉P(B,q):

第 0 步: $B_0 := \Phi; t_0 := 0$;

第 i 步: $n_i := max(t_{i-1}+1, c); x_i := 0^{n_i}$;

$t_i := p_i(n_i)$;命如下修剪 M_i 对输入 x_i 的计算树:若该树中含有询问句"y∈B?"且 $|y| < n_i$,则当 $y \in B_{i-1}$ 时便剪去"No"分枝下的整个子树,而当 $y \notin B_{i-1}$ 时便剪去"Yes"分枝下的整个子树,显然,修剪后的该树中可能出现的查询字 y 必满足 $|y| \geqslant n_i$,现在,分两种情形构造 B_i:

(1)若计算树中存在接受路径,则固定 π 为其中一条接受路径,且设

$C_0 = \{y \mid$ 在 π 中 M_i 询问了 y 且回答"Yes"$\}$,

$C_1 = \{y \mid$ 在 π 中 M_i 询问了 y 且回答"No"$\}$.

现在, $B_i := B_{i-1} \cup C_0$.

(2)若计算树中不存在任何接受路径,则设 π 为恒回答"Yes"的那条路径,且又设

$$C = \{y \mid \text{在 } \pi \text{ 中 } M_i \text{ 询问了 } y\}$$

再设 z_k 为 $\sum^{n_i}$ 中的第 k 个字,则

$$B_i := B_{i-1} \cup C \cup \{z_1, \cdots, z_{q(n_i)+1}\}$$

最后,我们有 $B = \bigcup_{i \in N} B_i$.

由以上构造显然可见 B 是递归集,现在,首先证明 B_i 满足以下需求:

$$R'_i: x_i \in L(B_i) \Leftrightarrow x_i \notin L(M_i, B_i, q)$$

对于(1),B_i 中长度为 n_i 的字均在 C_0 中,又由于查询次数≤$q(n_i)$,故 $|C_0| \leqslant q(n_i)$,从而,在 $\sum^{n_i}$ 中不可能有 $q(n_i)+1$ 个不同的字出现在 B_i 中,因此,$x_i = O^{n_i} \notin L(B_i)$. 另一方面,$M_i$ 存在受限接受 x_i 的路径 π,故 $x_i \in L(M_i, B_i, q)$. 因此,需求 R'_i 被满足.

对于(2),因为 $z_1, \cdots, z_{q(n_i)+1}$ 为 $\sum^{n_i}$ 中首 $q(n_i)+1$ 个字且均在 B_i 中,故 $x_i = O^{n_i} \in L(B_i)$. 另一方面,M_i 对 x_i 无任何接受路径,故 $x_i \notin L(M_i B_i, q)$. 因此,需求 R'_i 也被满足.

其次,由于 $B^{\leqslant t_i} = B_i$,又因为 $x_i = O^{n_i} \in L(B)$ 与否及 M_i^B 在 x_i 上的受限计算都只涉及长度≤t_i 的字,故由 B_i 满足 R'_i 显然可得 B 满足以下需求:

$$R_i: x_i \in L(B) \Leftrightarrow x_i \notin L(M_i, B, q)$$

现在,我们可以肯定 $L(B) \notin P(B,q)$. 设否,则必有 M_k 使得 $L(B) = L(M_k, B, q)$,但是,由需求 R_i 知当 $i=k$ 时应有 $L(B) \neq L(M_k, B, q)$,这个矛盾便证明了上述结论,从而定理得证.

推论 3. 存在递归集 B,使得 $P(B,q) \neq P(B)$.

证明:由推论 1,$P(B,q) \subseteq P(B,q+1) \subseteq P(B)$ 及上述定理即得.

定理 3. 存在递归集 B,使得 $NP(B,q) \neq NP(B,q+1)$.

证明:对 q,p,c 的假定同定理 2,现设

$L(B) = \{o^n \mid n \geqslant c$ 且至少存在 $q(n)+1$ 个长为 n 的字 $\in B\}$,则以下算法证明了,对任何递归集 B,$L(B) \in NP(B,q+1)$:

输入 x;

$n_i := |x|$;

检查 n≥c 及 $x=O^n$,若否,则拒绝;

For i:=1 to q(n)+1 do

不确定地猜一个长为 n 的字 y_i

检查 $y_i\neq y_1\wedge y_i\neq y_2\wedge\cdots\wedge y_i\neq y_{i-1}$;

$y_i\in B$;

若否,则拒绝;End;

接受;End

现在如下构造递归集 B,使得 L(B)∉NP(B,q):

第 0 步:$B_0:=\Phi;t_0:=0$;

第 i 步:$n_i:=max(t_{i-1}+1,C);x_i:=O^{n_i};t_i:=p_i(n_i)$;

今如下修剪不确定的 M_i 对输入 x_i 的计算树:若该树中含询问"$y\in B$?"且 $|y|<n_i$,则当 $y\in B_{i-1}$ 时便剪去"No"分枝下的子树,而当 $y\notin B_{i-1}$ 时便剪去"Yes"分枝下的子树. 显然,修剪后的计算树中的任何查询字 y 皆满足 $|y|\geqslant n_i$. 现在,分两种情况构造 B_i:

(1)若计算树中存在接受路径,则固定 π 为其中的一条接受路径且设

$C_0=\{y|$在 π 中 M_i 询问了 y 且回答"Yes"$\}$,

$C_1=\{y|$在 π 中 M_i 询问了 y 且回答"No"$\}$.

现在,$B_i:=B_{i-1}\cup C_0$.

(2)若计算树中不存在任何接受路径,则固定 π 为恒回答"Yes"的那条路径,且设

$C=\{y|$在 π 中 M_i 询问了 $y\}$,又设

z_k 为 $\sum^{n_i}$ 中的第 k 个字,则

$B_i:=B_{i-1}\cup C\cup\{z_1,\cdots,z_{q(n_i)+1}\}$.

最后,$B=\bigcup_{i\in N}B_i$

类似于定理 2 的证明,容易证明所构造的 B 是递归的且满足以下需求:

$$R_i:x_i\in L(B)\Leftrightarrow x_i\notin L(M_i,B,q)$$

并且由此即可证明 $L(B)\notin NP(B,q)$,从而定理得证.

推论 4. 存在递归集 B,使得 $NP(B,q)=NP(B)$.

证明:由推论 1,$NP(B,q)\subseteq NP(B,q+1)\subseteq NP(B)$ 及上述定理即得.

现在要问,任给集合 C,是否恒存在集合 A 和 B,使得 $C<_m^p A$ 且 $C<_m^p B$ 且 $P(A,q)=P(A)$ 且 $P(B,q)\neq P(B)$ 呢?类似地,也可对 NP 问同样的问题. 为了讨论这些问题,我们需要一些引理:

引理 1. 设 $D(A)=\{O^j|x|O^j|M_i^A$ 在 j 步内接受 $x\}$,则对任何集合 A,均有:$D(A)$ 是 $P(A)$ 完全的且对任何 y,$y\in D(A)\Leftrightarrow y\in D(A^{<|y|})$.

证明:以下算法证明了 D(A)∈P(A):

输入 y;

检查 $y=O^j|x|O^j$;

模似 M_i 在 x 上的 j 步计算;

若 M_i 接受,则算法接受,否则,拒绝;End

现在,再证 $D(A)$ 是 $P(A)-hard$ 的:设任给集合 $L\in P(A)$,则有确定的多项式时间界为 p_i 的 OT 机 M_i,使得 $L=L(M_i,A)$. 又设 $f(x)=O^j|x|O^{p_i(|x|)}$,则 f 显然为多项式时间可计算函数. 由于

$$x\in L\Leftrightarrow M_i^A\text{ 接受 }x\Leftrightarrow M_i^A\text{ 在 }p_i(|x|)\text{ 步内接受 }x\Leftrightarrow O^j|x|O^{p_i(|x|)}\in D(A)\Leftrightarrow f(x)\in D(A).$$

故 $L\leqslant_m^p D(A)$，即 $D(A)$是 $P(A)-hard$ 的，从而即得 $D(A)$是 $P(A)$完全的.

最后，我们证明，$y\in D(A)\Leftrightarrow y\in D(A^{<|y|})$：

设 $y\in D(A)$，则必有 i,j,x 使得 $y=O^i|x|O^j$ 且 M_i^A 在 j 步内接受 x，由于 M_i^A 至多走 j 步，故它所产生的一切字均$\leqslant|x|+j<|y|$，当然，查询字也必$<|y|$，故 $y\in D(A^{<|y|})$. 同理，由 $y\notin D(A)$显然可得 $y\notin D(A^{<|y|})$.

引理 2. 对任何集合 A，总存在递归于 A 的集合 B，使得 $B\not\leqslant_T^p A$ 且 $P(A\oplus B,1)=P(A\oplus B)$.

证明：令 $\mathscr{L}_{A\oplus B}=\{x|1xO^{|x|}\in A\oplus B\}=\{x|xO^{|x|}\in B\}$易见 $\mathscr{L}_{A\oplus B}\in P(A\oplus B,1)$. 下面使用对角线方法的编码技巧构造递归于 A 的集合 B，使得下列需求满足：

$R_{n,0}$：$\exists\ x_n(x_n\in B\Leftrightarrow M_n(A)$拒绝 $x_n)$ $n=1,2\cdots$

$R_{i,1}$：$\forall\ x(|x|=i\rightarrow(x\in D(A\oplus B)\Leftrightarrow x\in\mathscr{L}_{A\oplus B}))i=1,2,\cdots$

其中 $M_n(n=1,2,\cdots)$是确定性多项式时间 OTM 的一个枚举. 构造过程如下：

Stage 0：$B(0)=\Phi$，$t(0)=0$

Stage n：设 $B(n-1)$和 $t(n-1)$已有，且对任何 $y\in B(n-1)$，都有$|y|\leqslant t(n-1)$，取 $m=2t(n-1)+1$，$t(n)=p_n(m)$，$x(t(n-1))=B(n-1)$. 下面再分成 $t(n)-t(n-1)$个子步：子步 $t(n-1)+1\cdots$，子步 $m,\cdots$，子步 $t(n)$. 在子步 $j(j=t(n-1)+1,\cdots,t(n))$，如果：

① j 是奇数且 $j\neq m$，则 $x(j)=x(j-1)$.

② $j=m$，则要在这一步试图满足需求 $R_{n,0}$：令 $x_n=o^m$，若 $M_n(A)$接受 x_n，则 $x(m)=x(m-1)$，否则 $x(m)=x(m-1)\cup\{x_n\}$.

③ j 是偶数，设 $j=2i$，这时要试图满足需求 $R_{i,1}$：对每一个长为 i 的串 x，确定 x 是否在 $D(A\oplus x(j-1))$中，若 $x\in D(A\oplus x(j-1))$，则 $Yx=\{xo^{|x|}\}$；否则 $Yx=\Phi$，令 $x(j)=x(j-1)\cup(\cup_{|x|=i}Yx)$.

以上 $t(n)-t(n-1)$个子步完成后，令 $B(n)=x(t(n))$.

$B=\cup_{n\geqslant 0}B(n)$. 由构造过程易见：$B$ 是递归于 A 的，下证 B 满足引理中的条件.

① $t(n)$单调上升，且当 $n\rightarrow\infty$时 $t(n)\rightarrow\infty$，这是因为 $t(n)=p_n(m)>m=2t(n-1)+1>t(n-1)$，而 $t(n)$皆是自然数，故当 $n\rightarrow\infty$时 $t(n)\rightarrow\infty$.

② 需求 $R_{n,0}(n=1,2,\cdots)$被满足，由构造过程知：

$M_n(A)$拒绝 $x_n\Rightarrow x_n\in x(m)\Rightarrow x_n\in B(n)\Rightarrow x_n\in B$

$M_n(A)$接受 $x_n\Rightarrow x_n\notin x(m)\Rightarrow x_n\notin x(t(n))=B(n)$(因为在子步 $m+1,\cdots,t(n)$中，往 $B(n)$中放入的字长度为偶数，而$|x_n|=m$ 为奇数，故 x_n 不可能在子步 $m+1,\cdots,t(n)$中被放入 B 中)

又因为在 *Stage* n 之后，往 B 中放入的字的长度皆$>t(n)$，而$|x_n|=m\leqslant t(n)$，故 $x_n\notin B(n)\Rightarrow x_n\notin B$.

所以 $M_n(A)$拒绝 $x_n\Leftrightarrow x_n\in B$，即：需求 $R_{n,0}$被满足.

③ 需求 $R_{i,1}(i=1,2,\cdots)$被满足.

由①知，存在自然数 n，使得 $t(n-1)<2i\leqslant t(n)$，由构造过程，在 *Stage* n 的子步 $2i$ 中，对任何长度为 i 的串 x，$x\in D(A\oplus x(2i-1))\Leftrightarrow XO^{|x|}\in x(2i)$. 而在 $x\in D(A\oplus B)$的计算过程中所涉及到的字的长度皆$<|x|=i$(引理 1)，且$(A\oplus B)^{<i}=A^{<i-1}\oplus B^{<i-1}=A^{<i-1}\oplus x_{(3)}^{\leqslant i-1}$

$=(A\oplus x(2i-1))^{<i}$. 所以

$$x\in D(A\oplus B)\Leftrightarrow x\in D((A\oplus B)^{<i})\Leftrightarrow x\in D((A\oplus x(2i-1))^{<i})$$
$$\Leftrightarrow x\in D(A\oplus x(2i-1))\Leftrightarrow XO^{|x|}\in x(2i)$$

又因为在子步 $2i+1,\cdots,t(n)$ 中,往 $B(n)$ 中放入的字的长度皆$>2i$,故 $XO^{|x|}\in x(2i)\Leftrightarrow xo^{|x|}\in B(n)$,在 *Stage* n 之后往 B 中放入的字的长度皆$>t(n)$. 而 $|xo^{|x|}|=2i\leqslant t(n)$,故 $xo^{|x|}\in B(n)\Leftrightarrow xo^{|x|}\in B$

所以 $x\in D(A\oplus B)\Leftrightarrow xo^{|x|}\in B\Leftrightarrow x\in\mathscr{L}_{A\oplus B}$

所以需求 $R_{i,1}$ 被满足.

④ $B\not\leqslant_T^p A$.

反设 $B\leqslant_T^p A$,则有 M_k,使得 $B=\mathscr{L}(M_k,A)$,即对任何 x,$x\in B\Leftrightarrow M_k(A)$ 接受 x,而需求 $R_{n,0}$ 对任何 n 皆满足,故对 $n=k$ 也满足,从而存在 x_k,使得 $x_k\in B\Leftrightarrow M_k(A)$ 拒绝 x_k,矛盾.

⑤ $P(A\oplus B,1)=P(A\oplus B)$.

设 $\mathscr{L}\in P(A\oplus B)$,由引理 1,$D(A\oplus B)$ 是 $P(A\oplus B)$ 完全的,故有多项式时间可计算函数 f,使得 $x\in\mathscr{L}\Leftrightarrow f(x)\in D(A\oplus B)$,而需求 $R_{i,1}$ 对任何 i 皆满足,故 $x\in D(A\oplus B)\Leftrightarrow x\in\mathscr{L}_{A\oplus B}$

所以 $x\in\mathscr{L}\Leftrightarrow f(x)\in D(A\oplus B)\Leftrightarrow f(x)\in\mathscr{L}_{A\oplus B}$

而 $\mathscr{L}_{A\oplus B}\in P(A\oplus B,1)$,所以 $\mathscr{L}\in P(A\oplus B,1)$

所以 $P(A\oplus B)\subseteq P(A\oplus B,1)$,又因为 $P(A\oplus B,1)\subseteq P(A\oplus B)$

所以 $P(A\oplus B,1)=P(A\oplus B)$ □

定理 3. 设 c 是任一集合,则存在递归于 c 的集合 A,使得 $c<_m^p A$ 并且 $P(A,q)=P(A)$.

证明:由引理 2,存在递归于 c 的集合 D,使得 $D\not\leqslant_T^p C$ 且 $P(C\oplus D,1)=P(C\oplus D)$. 取 $A=C\oplus D$,则:

① A 递归于 C,这是因为 D 递归于 C.

② $C<_m^p A$.

因为 $C\leqslant_m^p C\oplus D$,而另一方面,$D\not\leqslant_T^p C$,故 $C\oplus D\not\leqslant_m^p C$(因为若 $C\oplus D\leqslant_m^p C$,则 $D\leqslant_m^p C$,从而 $D\leqslant_T^p C$. 矛盾)

所以 $C<_m^p C\oplus D$,即 $C<_m^p A$.

③ $P(A,q)=P(A)$.

因为 $P(A,1)\subseteq P(A,q)\subseteq P(A)$

而 $P(A,1)=P(C\oplus D,1)=P(C\oplus D)=P(A)$,

所以 $P(A,q)=P(A)$. □

引理 3. 设 A 为一个集合,则存在递归于 A 的集合 B,使得 $B\not\leqslant_T^p A$,并且 $P(A\oplus B,q)\neq P(A\oplus B,q+1)$.

证明:设 $M_n(n=1,2,\cdots)$ 是确定性多项式时间 *OTM* 的一个枚举,$M'_n(n=1,2,\cdots)$ 是限制查询确定性多项式时间 *OTM* 的一个枚举,M_n 和 M'_n 的运行时间皆$\leqslant p_n$.

根据对 q 的假定,存在多项式 p,使得:$q(n)\leqslant p(n)$,故有 $N_1>0$,当 $n>N_1$ 时 $q(n)+1<2^n$. 令 $\mathscr{L}_{A\oplus B}=\{o^n\mid n\geqslant N_1$ 且长度为 n 的前 $q(n)+1$ 个字皆$\in B\}$,易见 $\mathscr{L}_{A\oplus B}\in P(A\oplus B,q+1)$. 算法见下图.

输入 x

$n:=|x|$

检查 $n\geqslant N_1$ 且 $x=o^n$，若不满足，则拒绝；

for $i=1$ *to* $q(n)+1$ *do*

$y:=$长度为 n 的第 i 个字；

$z:=1y$；

查询 $z\in$？*oracle* 集，若回答 No，则拒绝

end for

接受.

end

要使本引理成立，仅需构造递归于 A 的集合 B，使得下列需求：

$R_{n,0}$：$\exists\ x_{n,0}(x_{n,0}\in B\Leftrightarrow M_n(A)$拒绝 $x_{n,0})$

$R_{n,1}$：$\exists\ x_{n,1}(x_{n,1}\in\mathscr{L}_{A\oplus B}\Leftrightarrow x_{n,1}\notin\mathscr{L}(M'_n,A\oplus B,q))$对任何 n 皆成立就够了. 其构造过程如下：

stage 0：$B(0)=\Phi,t(o)=0$

stage n：设 $B(n-1)$ 和 $t(n-1)$ 已有，且对任何 $y\in B(n-1)$，都有 $|y|\leqslant t(n-1)$，取 $m=max(t(n-1)+1,N_1)$，$t(n)=p_n(m)$，$x_{n,0}=1^m,x_{n,1}=o^m$.

如果 $M_n(A)$ 拒绝 $x_{n,0}$，则令 $c_0=\{x_{n,0}\}$；否则 $c_0=\Phi$.

现在考虑 M'_n 在 $x_{n,1}$ 上生成的 *oracle* 树. 对该树作下列修剪：

① 对查询“$oy\in$？ $A\oplus B$”，当 $y\in A$ 时剪去 *No* 子树，当 $y\notin A$ 时剪去 *Yes* 子树；

② 对查询“$1y\in$？ $A\oplus B$”($|y|<m$)，则当 $y\in B(n-1)$时剪去 *No* 子树；$y\notin B(n-1)$时剪去 *Yes* 子树；

③ 对查询“$1^{M+1}\in$？ $A\oplus B$”，则 $c_0=\{x_{n,0}\}$时剪去 *No* 子树；$c_0=\Phi$ 时剪去 *Yes* 子树.

进行过以上修剪之后 oracle 树中的查询仅与 B 中长度$\geqslant m$ 的字有关. 下面分两种情况进行讨论：

case 1. 树中有一个接受路径 π.

令 $c_1=\{y|y$ is queried and answered Yes in path $\pi\}$.

$$B(n)=B(n-1)\cup c_0\cup c_1.$$

显见，① $x_{n,0}\in B(n)\Leftrightarrow M_n(A)$拒绝 $x_{n,0}$.

② 由于 $q(m)+1<2^m$，故 $x_{n,0}=1^m$ 不是长为 m 的前 $q(m)+1$ 个字，所以 $B(n)$中长为 m 的前 $q(m)+1$ 个字皆在 c_1 中，由于$|c_1|\leqslant q(m)$，故 $x_{n,1}\notin\mathscr{L}\ A\oplus B(n)$. 但 M'_n 在输入 $x_{n,1}$ 上以 $A\oplus B(n)$为 *oracle* 时对应于路径 π，故 $x_{n,1}\in\mathscr{L}(M'_n,A\oplus B(n),q)$.

所以集合 $B(n)$和 $x_{n,0},x_{n,1}$使需求 $R_{n,0},R_{n,1}$满足.

case 2. 树中无接受路径.

设 π 是对所有查询回答皆是 *Yes* 的那条路径，令 $c_1=\{y|y$ is queried in path $\pi\}$，$B(n)=B(n-1)\cup c_0\cup\{z_1,z_2,\cdots,z_{q(m)+1}\}$，其中 z_i 是长度为 m 的第 i 个字($i=1,\cdots,q(m)+1$). 则：

①$x_{n,0}\in B(n)\Leftrightarrow M_n(A)$拒绝 $x_{n,0}$.

②$x_{n,1}\in\mathscr{L}(A\oplus B(n))$. 而 M'_n 在输入 $x_{n,1}$上以 $A\oplus B(n)$为 *oracle* 时对应于路径 π，故它

拒绝 $x_{n,1}$.

所以集合 $B(n)$和 $x_{n,0}$、$x_{n,1}$使需求 $R_{n,0}$、$R_{n,1}$满足.

令 $B=\bigcup_{n\geqslant 0}B(n)$. 由构造过程易知，$B$ 递归于 A，且 $B^{\leqslant t(n)}=B(n)$，而 $Mn(A)$在输入 $x_{n,0}$ 上的计算与 B 无关，$x_{n,0}\in B$、$x_{n,1}\in\mathscr{L}_{A\oplus B}$以及 $M'_n(A\oplus B)$在输入 $x_{n,1}$上的计算都仅涉及到长度$\leqslant t(n)$的串，故：$x_{n,0}$和集合 B 使得需求 $R_{n,0}$满足，$x_{n,1}$和集合 B 使得需求 $R_{n,1}$满足.

所以本引理成立. □

定理 4. 对任何集合 C，总存在递归于 C 的集合 B，使得 $C<_m^p B$ 并且 $p(B,q)\neq P(B,q+1)$.

证明：由引理 3 可知，存在递归于 C 的集合 D，使得 $D\not\leqslant_m^p C$ 且 $P(C\oplus D,q)\neq P(C\oplus D,q+1)$. 取 $B=C\oplus D$ 即可. □

需要指出的是，定理 3 和定理 4 对于 NP 的情形也同样成立，证明过程类似，就不再列出了.

参考文献

1 Balcazer J, Book R, Schning U. On bounded query machines. Theoret. Comput. Sci., 1985;**40**:237—243.

2 Balcazar J L, Diaz J, Gabarro J. Structure complexity. 1988(**1**), 1990(**2**).

3 Baker J, Gill J, Solovey R. Relativizations of the P=NP question. SIAM J. Computing, 1975;**4(4)**:431—442.

4 Book R, Long T, Selman A. Quantitative relativizations of complexity classes. SIAM J. Comput., 1984;**13**:461—487.

5 Ashor K C, Dexter C K, Larry J S. Alternation. JACM, 1981;**18(1)**:114—133.

6 Hans, Heller. Relativized polynomial hierarchies extending two levels. Math. System Theory, 1984;**17(2)**:71—84.

7 Kadin J. $P^{NP}[\log n]$ and sparse turing—complete sets for NP. Proc. of 2nd conference on Structure in Complexity Theory, 1987:33—40.

8 Ker—I Ko. Constructing oracles by lower bound techniques for circuits. A Lecture at the International Symposium on Combinatorial Optimization Held at the Nankai Institute of Math., Tianjing, August 1988.

9 Ladner R E, Lynch N A. Relativization of questioins about log—space computability. 1976;**10(1)**:19—32.

10 Timothy J L. On restricting the size of oracles compared with restricting access to oracles. SIAM. J. Comput., 1984;**14**:585—597.

11 Klaus W W. Bounded query computations. Proc. of 3rd Conference on Structure in Complexity Theory, 1988:260—277.

THE P—NP PROPERTIES OF BOUNDED QUERY COMPUTATIONS

Lü Yizhong and Liu Jianbin

(*Department of Mathematics, Nanjing University, Nanjing* 210008)

Abstract　This paper restricts oracle Turing machine to query its oracle in an accepting computation, and obtains result: there exist infinite number of recursive sets A, B, A',

B′, A″, B″, A‴, B‴, which are not$\equiv_m^p$ equivalent satisfying properties: $P(A,q)=P(A,q+1)$, $P(B,q)\neq P(B,q+1)$ $P(a',q)=P(A')$, $P(B',q)\neq P(B')$, $NP(A'',q)=NP(A'',q+1)$, $NP(B'',q)\neq(B'',q+1)$, $NP(A''',q)=NP(A''')$, $NP(B''',q)\neq NP(B''')$.

Key words Computational complexity, P—NP question, recursive set, oracle.