Tópicos em Complexidade Computacional

Oráculos, Relativização e o Teorema de Baker-Gill-Solovay

Professor Murilo V. G. da Silva

Departamento de Informática

Universidade Federal do Paraná

07/07/2022

Máquinas de Turing com Oráculos

Máquina de Turing com Oráculo (MTO's)

Seja O uma linguagem. Uma MT com oráculo para O é uma MT com as modificações:

- A MT tem uma fita extra, conhecida como fita de oráculo.
- Três estados extras, q_{QUERY}, q_{SIM} e $q_{\text{NÃO}}$.
- Durante a computação, quando a máquina atinge o estado q_{QUERY}, o próximo passo computacional depende da string w presente na fita de oráculo:
  - Se $w \in O$, então o próximo estado é q_{YES}.
  - Se $w \notin O$, então o próximo estado é q_{NO}.

Pergunta: Essas máquinas tem mais poder computacional do que MTs tradicionais?

Resposta: Sim. Considere o caso em que O é uma linguagem indecidível, por exemplo.

- Tipicamente estamos interessados em oráculos "poderosos"
- A definição de MTNOs é similar
- Escrevemos M^L para MTOs com oráculo para linguagem L

Máquinas de Turing com Oráculos

Máquinas Polinomiais com Oráculo

Seja O uma linguagem.

- P^O: Classe das linguagens decididas por MTs polinomiais com oráculo para O.
- NP^O: Classe das linguagens decididas por MTNs polinomiais com oráculo para O.

Exemplo: Digamos que $O = L_{\mathrm{SAT}}$

- Quais problemas estão contidos em $\mathrm{P}^{L_{\mathrm{SAT}}}$?
  - Todo problema em NP
  - Mas também todo problem em co-NP!

    Podemos decidir $\overline{L_{\mathrm{SAT}}}$ usando o oráculo para L_{SAT} e inverter a resposta!

Exemplo: Digamos que $O \in \mathrm{P}$ qualquer

- Neste caso $\mathrm{P}^O = \mathrm{P}$

Oráculo com poder exponencial

Seja $L_{\text{XC}} = \{\langle M, x, 1^n\rangle \ : \ M(x) = 1 \text{ em } 2^n \text{ passos}\}$.

Teorema 7.1

$$\text{P}^{L_{\text{XC}}} = \text{NP}^{L_{\text{XC}}} = \text{EXP}$$

Prova:

- A estratégia é mostar que $\text{EXP} \subseteq \text{P}^{L_{\text{XC}}} \subseteq \text{NP}^{L_{\text{XC}}} \subseteq \text{EXP}$
- $\text{P}^{L_{\text{XC}}} \subseteq \text{NP}^{L_{\text{XC}}}$, pois MTNO's podem simular MTO's
- Basta mostrar agora (1) $\text{EXP} \subseteq \text{P}^{L_{\text{XC}}}$ e (2) $\text{NP}^{L_{\text{XC}}} \subseteq \text{EXP}$

  (1): Seja $L \in \text{EXP}$. Seja M_L a MT de tempo $2^{p(n)}$ que decide L
  A seguinte MTNO polinomial $N^{L_{\text{XC}}}$ decide L:

  - Dado x, basta escrever $\langle M_L, x, 1^{p(|x|)}\rangle$ na fita de oráculo

  (2): Seja uma MTNO polinomial $A^{L_{xc}}$

  - A árvore de computação de $A^{L_{xc}}$ tem tamanho $2^{p(n)}$
  - Tempo para simular uma única consulta da MTNO com uma MT: $2^{q(n)}$
  - Potencialmente cada nó da árvore é consulta para oráculo
  - Entretanto, note que $2^{p(n)} \cdot 2^{q(n)} = 2^{p(n)+q(n)}$

  Portanto uma MT exponencial pode simular $A^{L_{xc}}$ em tempo $2^{p(n)+q(n)}$

Relativização

Sabemos que $\mathrm{P} \neq \mathrm{EXP}$

- Isso é corolário do Teorema da Hierarquia de Tempo
- Em particular, corolário de $\mathrm{DTIME}(n^c) \neq \mathrm{DTIME}(2^n)$

Seja O um oráculo qualquer.

- Seria verdade que $\mathrm{P}^O \neq \mathrm{EXP}^O$?
- Respota: Sim! Prova abaixo:

Corolário (de T. 5.1)

Para qualquer linguagem L, temos $\mathrm{P}^L \neq \mathrm{EXP}^L$

Prova:

- Seja L uma linguagem
- Repita **exatamente** a mesma da prova do T. 5.1, exceto que,
  - Cada MT M que aparece na prova, é trocada pela MTO M^L (inc. a MT universal)
- Note que o argumento funciona por que tratamos MTs como "caixas pretas"
- Tais provas (i.e., usando "somente diagonalização") são ditas relativizantes.
- O Corolário acima vale para quaisquer classes separadas por provas relativizantes

Relativização

Sejam $\mathcal{C}$ e $\mathcal{D}$ classes de complexidade com relação desconhecida, mas tal que

- $\exists A$ tal que $\mathcal{C}^A = \mathcal{D}^A$
- $\exists B$ tal que $\mathcal{C}^B \neq \mathcal{D}^B$

Digamos que alguém apresente uma demonstração relativizante de que $\mathcal{C} \neq \mathcal{D}$

- Conclusão: A demonstração está errada!
- Se a demonstração relativiza, podemos derivar também que $\forall A,\ \mathcal{C}^A \neq \mathcal{D}^A$

Digamos que alguém apresente uma demonstração relativizante de que $\mathcal{C} = \mathcal{D}$

- Conclusão: A demonstração está errada!
- Se a demonstração relativiza, podemos derivar também que $\forall B,\ \mathcal{C}^B = \mathcal{D}^B$

Teorema 7.2 [Baker, Gill, Solovay (1975)

$\exists A$ tal que $\mathrm{P}^A = \mathrm{NP}^A$ e $\exists B$ tal que $\mathrm{P}^B \neq \mathrm{NP}^B$.

Prova: Aula que vem.

Teorema de Baker-Gill-Solovay

Na prova do teorema usaremos o seguinte:

Linguagem unária para tamanhos

Seja L uma linguagem. A *linguagem unária para tamanhos* de L é:

$$U_L = \{1^n \ : \ \exists x \in L \text{ tal que } |x| = n\}$$

Afirmação (A1)

Seja L uma linguagem e U_L a linguagem unária para tamanhos correspondente. Então

$$U_L \in \mathrm{NP}^L$$

Prova de (A1): Segue a MTNO M^L polinomial para U_L:

- Dada entrada x, a máquina M^L faz:
- Rejeita se x tem bits em 0
- "Advinha" string de tamanho $|x|$ e escreve na fita de óraculo
- aceita se oráculo responde "sim"

Teorema de Baker-Gill-Solovay

Teorema 7.2 [Baker, Gill, Solovay (1975)

$\exists A$ tal que $\mathrm{P}^A = \mathrm{NP}^A$ e $\exists B$ tal que $\mathrm{P}^B \neq \mathrm{NP}^B$.

Prova:

- Pelo Teo. 7.1, para $A = L_{\mathrm{XC}}$ temos $\mathrm{P}^A = \mathrm{NP}^A$.
- Vamos construir uma linguagem B de maneira que:
  - $U_B \notin \mathrm{P}^B$.
  - Como sabemos de (A1) que $U_B \in \mathrm{NP}^B$.
  - Então podemos concluir que para tal B, temos $\mathrm{P}^B \neq \mathrm{NP}^B$
- No próximo slide segue a **construção de B tal que $U_B \notin \mathbf{P}^B$**
- Antes da construção:
  - Note: independente de L, a string que descreve a MTO M^L é sempre a mesma
  - Para $i = 1, 2, ...$, considere a enumeração de cada MTO M_i^B que para

Teorema de Baker-Gill-Solovay

Prova (cont.): Construindo uma linguagem B tal que $U_B \notin \mathrm{P}^B$:

- Começamos fazendo $B = \emptyset$ e vamos inserir strings em B estágio por estágio:
  - No **estágio** i da construção, as strings w inseridas em B garantem que
    - M_i^B não decide U_B em $\frac{2^n}{10}$ ou menos (para n grande o suficiente)
  - Cada estágio **estabelece o status** de uma quantidade finita de strings (i.e., estabelece que certas strings pertencem ou não pertencem à B)
  - A construção não estabelece sistematicamente status de **toda string** de $\{0,1\}^*$
    - podemos, se quisermos, dizer que strings cujo status não foi estabelecido ao final da construção, por definição não estão em B

Teorema de Baker-Gill-Solovay

Prova (cont.): Construindo uma linguagem B tal que $U_B \notin \mathrm{P}^B$:

- **Estágio i**: Seja $n = n_i$ maior que o tamanho de toda string de status estabelecido
  - Execute $\frac{2^n}{10}$ passos de $M_i^B(1^n)$ e considere cada consulta ao oráculo por y
    - Se y é uma string com status já estabelecido, continue a execução a partir do estado adequado (i.e., q_{YES} se $y \in B$ ou q_{NO} se $y \notin B$)
    - Se y é uma string com status não estabelecido, estabeleça $y \notin B$ e continue a execução a partir de q_{NO}
  - Ao final de $\frac{2^n}{10}$ (ou menos):
    - Se $M_i^B(1^n) = 1$

      Estabeleça que strings (restantes) de tamanho n não estão em B

      Note: até agora fizemos apenas $y \notin B$ para strings sem status

      Antes do estágio i, strings de tamanho n não tiveram status estabelecido

      Portanto $\forall x$ com $|x| = n$, temos $x \notin B$. Então $1^n \notin U_B$

      $\therefore$ M_i^B não decide U_B em tempo $\frac{2^n}{10}$
    - Se $M_i^B(1^n) = 0$

      Escolha um string y de tamanho n sem status e estabeleça $y \in B$

      Note: existe tal string, pois rodamos apenas $\frac{2^n}{10}$ passos

      E portanto $1^n \in U_B$ $\therefore$ M_i^B não decide U_B em tempo $\frac{2^n}{10}$
  - Execute até final estabelecendo o status das strings consultadas
  - Note: strings com status **estabelecido neste estágio** (de qualquer tamanho), não foram consultadas por $M_1^B, ..., M_{i-1}^B$, portanto tais MTOs continuam falhando

Teorema de Baker-Gill-Solovay

Prova (cont.): Provando que $U_B \notin \mathrm{P}^B$ (para B que construimos):

- Suponha que $U_B \in \mathrm{P}^B$.
  - Seja M^B a MTO que decide U_B em $p(n)$ passos
    - Note: Existem infinitas representações M_i^B para essa MTO
    - Considere M_i^B tal que $\frac{2^{n_i}}{10} > p(n_i)$
  - M^B decide qualque string de tamanho n_i em menos de $\frac{2^{n_i}}{10}$ passos
  - Seja $x = 1^{n_i}$
  - Pela construção de B, temos $M^B(x) = 1 \Leftrightarrow x \notin U_B$. Contradição.