Tópicos em Complexidade Computacional

O Teorema de Cook-Levin

Professor Murilo V. G. da Silva

Departamento de Informática

Universidade Federal do Paraná

07/07/2022

Teorema de Cook-Levin

Teorema 8.1 [Cook(1971), Levin (1973)]

SAT é NP-completo.

Prova:

- Pelo Ex. 3.1, $\mathrm{SAT} \in \mathrm{NP}$.
- "Resta" mostrar que $\forall L \in \mathrm{NP}$, $L \leq_p \mathrm{SAT}$.

Queremos uma redução $R_L(x) = \phi_x$ que faça o seguinte:

- $x \in L \quad \Rightarrow \quad \phi_x \in \mathrm{SAT}$
- $x \notin L \quad \Rightarrow \quad \phi_x \notin \mathrm{SAT}$

Teorema de Cook-Levin – Preliminares

Aquecimento: Sejam x e y strigs binárias de mesmo tamanho:

- $x = x_1 x_2 \ldots x_n$
- $y = y_1 y_2 \ldots y_n$

Igualdade de strings: O problema de **"testar se $x = y$"** é equivalente a satisfatibilidade de:

$$(x_1 \vee \overline{y_1}) \wedge (\overline{x_1} \vee y_1) \wedge (x_2 \vee \overline{y_2}) \wedge (\overline{x_2} \vee y_2) \wedge \ldots \wedge (x_n \vee \overline{y_n}) \wedge (\overline{x_n} \vee y_n)$$

Equivalência de fórmulas: Escrevemos $\phi_1 = \phi_2$ se $(\phi_1 \vee \overline{\phi_2}) \wedge (\overline{\phi_1} \vee \phi_2)$

Afirmação (A1) (Universalidade de AND,OR e NOT): $\forall$ função booleana $f : \{0,1\}^k \to \{0,1\}$

- Existe uma fórmula CNF ϕ de n variáveis de tamanho $k \cdot 2^k$
- De maneira que $\phi(u) = f(u)$ para todo $u \in \{0,1\}^k$
- Tamanho de ϕ: número de símbolos $\wedge/\vee$ na fórmula

Prova:

- Ver *Claim 2.13* do livro: prova construtiva.

Teorema de Cook-Levin

Prova:

Seja $L \in \mathrm{NP}$.

- Existe uma MT polinomial M_L e polinômio $p(n)$ tal que $\forall x$.
- $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ tal que $M_L(x,u) = 1$

- LEMBRANDO: Queremos uma redução $R_L(x) = \phi_x$ que faça o seguinte:
  - $x \in L \quad \Leftrightarrow \quad \phi_x \in \mathrm{SAT}$
- Ou seja $\phi_x \in \mathrm{SAT} \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ tal que $M_L(x,u) = 1$

- Note: $M_L(x, \ )$ pode ser vista como uma função
  - $u \rightarrow M_L(x,u)$ (i.e., a função computada pela MT com x "fixo")
  - Uma função $f : \{0,1\}^{p(|x|)} \rightarrow \{0,1\}$
- Considere a fórmula CNF ϕ_x equivalente a função $M_L(x,u)$
  - ϕ_x valoração que à satisfaz $\Leftrightarrow$ existe valor que faz $M_L(x,u) = 1$
- Com isso $\phi_x \in \mathrm{SAT} \Leftrightarrow$ existe valor que faz $M_L(x,u) = 1$
- Ou seja, $\phi_x \in \mathrm{SAT} \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ tal que $M(x,u) = 1$

Ou seja, contrução da prova de (A1) é a redução. Qual seria o problema?
Tamanho exponencial: $p(|x|)2^{|x|}$

Teorema de Cook-Levin

- Vamos ter que apresentar uma redução R_L melhor
  - Levando em consideração que M_L é polinomial e que *computação é local*
  - Não vamos usar M_L com *caixa preta*.

Podemos assumir que $M_L = (Q, \Gamma, \delta)$ é imparcial e tem duas fitas

- Quando rodamos $M_L(x, u)$, às vezes escreveremos $y = xu$

- Na redução (que é um algoritmo), uma das rotinas será computar o seguinte:
  - Qual é a posição das cabeças de leitura de $M_L(x, \ )$ (independente de u) no passo i?
  - Sabemos que não depende de u. Mas qual é a posição?
  - Rodamos $M_L(x, 1^{p(|x|)})$
- Seja $\langle a, b, q \rangle$ uma fotografia da execução de M_L $\qquad a, b \in \Gamma, \ q \in Q$
- Seja $c = |\langle a, b, q \rangle|$

Teorema de Cook-Levin

Ideia central: Dados M_L, x e sequêcia de fotografias z_1, ..., $z_{q(|x|+p(|x|))}$

- Como conseguimos saber se $\exists u$ que satizfaz $M(x, u) = 1$?
- Olhando corretude de $z_i's$: Basta olhar z_{i-1}, $z_{\text{PREV(i)}}$ e $y_{\text{INPUTPOS}(i)}$

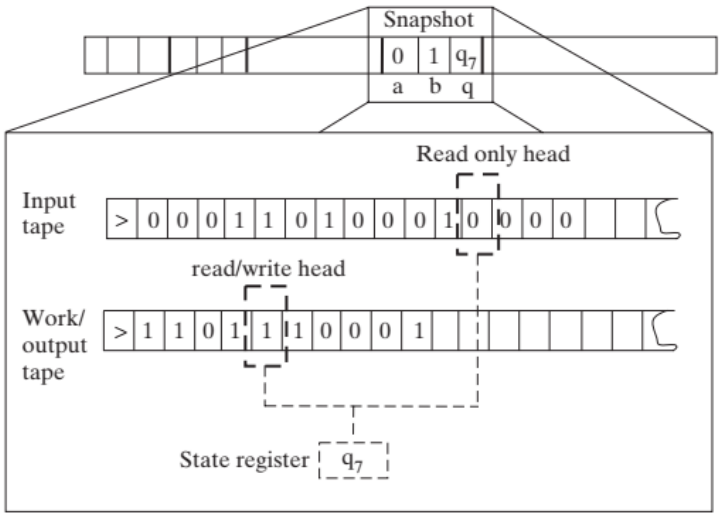

Figura: Barak & Arora, pág. 48

- z_{i-1}: contem o estado anterior que M_L se encontrava
- $y_{\text{INPUTPOS}(i)}$: contém o símbolo na fita 1
- $z_{\text{PREV(i)}}$: contém o símbolo na fita 2

Teorema de Cook-Levin

Ideia central: Dados M_L, x e sequêcia de fotografias z_1, ..., $z_{q(|x|+p(|x|))}$

- Como conseguimos saber se $\exists u$ que satizfaz $M(x, u) = 1$?
- Olhando corretude de $z_i's$: Basta olhar z_{i-1}, $z_{\text{PREV(i)}}$ e $y_{\text{INPUTPOS}(i)}$

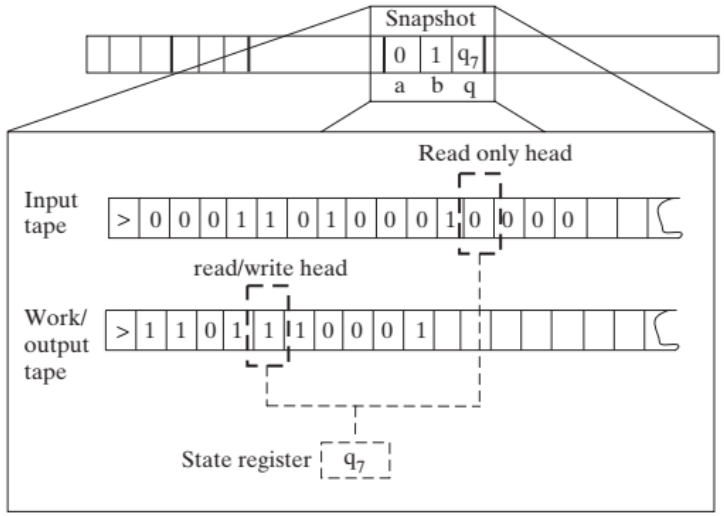

Figura: Barak & Arora, pág. 48

Considere a função booleana $F : \{0,1\}^{2c+1} \to \{0,1\}^c$

- $F(\ z_{i-1},\ y_{\text{INPUTPOS}(i)},\ z_{\text{PREV(i)}}) = z_i$
- Importante: Função de tamanho constante (não depende de y)

Teorema de Cook-Levin

Ideia central: Dados M_L, x a partir da sequêcia de fotografias z_1, ..., $z_{|x|+p(|x|)}$

- Conseguimos saber se $\exists u$ apenas fotografias e computando $z_{\text{PREV(i)}}$ e $y_{\text{INPUTPOS}(i)}$

Lembrando que $F : \{0,1\}^{2c+1} \to \{0,1\}^c$ é a função

- $F(\ z_{i-1},\ y_{\text{INPUTPOS}(i)},\ z_{\text{PREV(i)}}) = z_i$

Redução de L para SAT:

- Lembrando que $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ tal que $M_L(x,u) = 1$
- Lembrando que $x \in L \Leftrightarrow \exists y \in \{0,1\}^{|x|+p(|x|)}$ e $z_1, ..., z_{q(|x|+p(|x|))}$ satisfazendo:
  (1) primeiros $|x|$ bits de y são iguais a x
  (2) z_1 é igual string de fotografia inicial
  (3) $\forall z_i$, temos $F(\ z_{i-1},\ y_{\text{INPUTPOS}(i)},\ z_{\text{PREV(i)}}) = z_i$
  (4) $z_{q(|x|+p(|x|))}$ é igual fotografia de aceitação
- Redução produz ϕ_x satisfazendo existência de y e fotografias
  - Compute $\text{INPUTPOS}(i)$, $z_{\text{PREV(i)}}$ rodando M_L
  - Compute fórmula CNF para (1), (2), (3) e (4)

Teorema de Cook-Levin

Na prova do Teorema de Cook-Levin, mostramos que

- Existe uma Redução de Karp de cada linguagem de $L \in \mathrm{NP}$ para SAT
- Ou seja, uma redução $R_L(x) = \phi_x$ que faz o seguinte:
  - $x \in L \iff \phi_x \in \mathrm{SAT}$

Mas é possível fazer melhor:

- Mostrar que $\forall L \in \mathrm{NP}$, existe redução mais "forte" de L para SAT.

Redução de Levin: $\forall L \in \mathrm{NP}$, a redução $R_L(x,c) = (\phi_x, v)$ também mapeia certificados.

- Se $V_L(x, c) = 1 \iff V_{\mathrm{SAT}}(\phi_x, v) = 1$.

Observação: fazendo $c = \epsilon$ a redução produz $v = \epsilon$, ou seja, é redução de Karp

Redução Parcimoniosa: Redução de Levin com a seguinte propriedade:

- Induz bijeção entre certificados e valorações.
- Dado certificado é possível computar valoração e vice-versa

Problemas de Busca

Seja $L \in \mathrm{NP}$ e V_L seu verificador. Considere o seguinte problema

Problema de Busca associado a L: Dado uma string de entrada x qualquer,

- Se $x \in L$, encontrar certificado c tal que $V_L(x, c) = 1$
- Se $x \notin L$, retornar $\mathrm{NÃO}$.

Chamamos, às vezes, de *"versão de busca"* do problema L.

Fato: Se $\mathrm{P} \neq \mathrm{NP} \implies$ "versão de busca" pelo menos tão difícil quanto "versão de decisão"

Teorema 8.2: Se $\mathrm{P} = \mathrm{NP}$, então existe um algoritmo polinomial para a versão de busca de L.

Idéia da prova:

- Caso particular de SAT: Seja M_{SAT} a MT polinomial para SAT
  - Use a M_{SAT} para decidir se ϕ é satisfazível com $x_1 = 0$ ou $x_1 = 0$
  - Se falhar para ambos, retorne $\mathrm{NÃO}$
  - Senão escolha um caso em que a fórmula é satifeita, "simplifique" a fórmula
  - Resolva fórmula simplificada
- Linguagem L qualquer:
  - Use algoritmo (redução) R_L para obter ϕ_x a partir de x
  - Obternha valoração para ϕ_x (ou retorne $\mathrm{NÃO}$)
  - Pela redução parsimoniosa, obtenha certificado equivalente