Complexidade Computacional na Geometria Algébrica

Thomas Michael Bartlett

UNICAMP

th.m.bartlett@gmail.com

25 de novembro de 2016

Overview

1. Fundamentos da Complexidade Computacional
   - Definição de Máquina de Turing
   - MT Determinística e a classe P
   - MT Não-determinística e a classe NP
   - Problema NP-Completo e teorema de Cook
   - NP-Difícil
   - Outras Classes de Complexidade
2. Exemplos NP-Completo na Algebra e Geometria
   - 3-SAT
   - TRAVELING SALESMAN
   - QUADRATIC DIOPHANTINE EQUATIONS
   - ALGEBRAIC EQUATIONS OVER GF[2]
   - DECODING LINEAR CODES
   - GROBNER BASIS

Definição de Máquina de Turing - 1

Uma *Máquina de Turing* é um par de fita infinita e um cabeçote com controle de estados finitos:

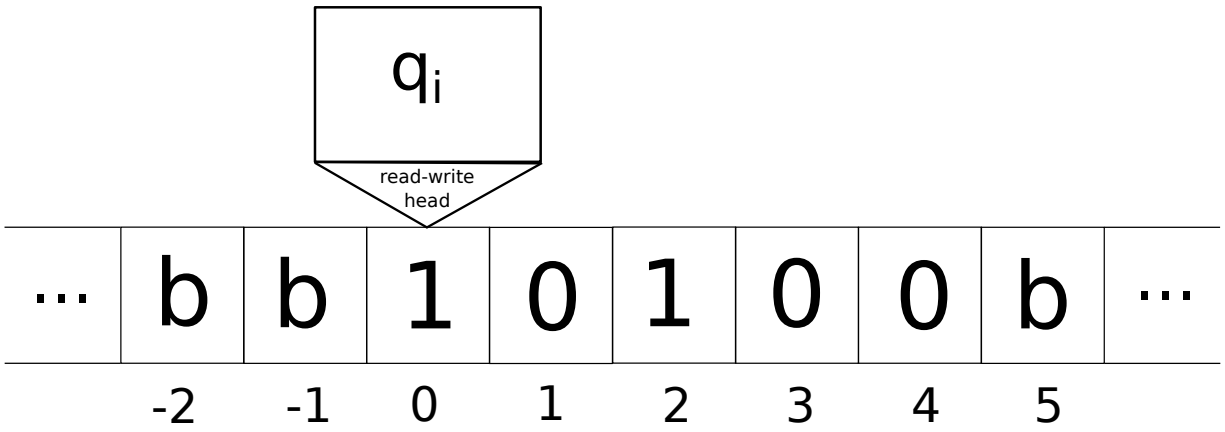

Figura : Equipamentos da Máquina de Turing

Definição de Máquina de Turing - 2

Def.: Um *programa determinístico* para a MT é a tupla $M = (\mathbf{\Gamma}, \mathbf{Q}, \delta)$

- Um conjunto finito $\mathbf{\Gamma} = \{b, \gamma_1, ..., \gamma_n\}$ de *símbolos*;
- Um conjunto finito $\mathbf{Q} = \{q_Y, q_N, q_0, q_1, ..., q_d\}$ de *estados*
- Uma função $\delta : (\mathbf{Q} - \{q_Y, q_N\}) \times \mathbf{\Gamma} \to \mathbf{Q} \times \mathbf{\Gamma} \times \{-1, +1\}$, chamada *função transição*;

Exemplo de programa - 1

Seja o programa que reconhece se os dois símbolos mais a direita são 0:

- $\mathbf{\Gamma} = \{b, 0, 1\}$;
- $\mathbf{Q} = \{q_Y, q_N, q_0, q_1, q_2, q_3\}$;
- δ a seguir

q	0	1	b
q_0	$(q_0, 1, +1)$	$(q_0, 1, +1)$	$(q_1, 1, -1)$
q_1	$(q_2, b, -1)$	$(q_3, b, -1)$	$(q_N, b, -1)$
q_2	$(q_Y, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$
q_3	$(q_N, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$

Exemplo de programa - 2

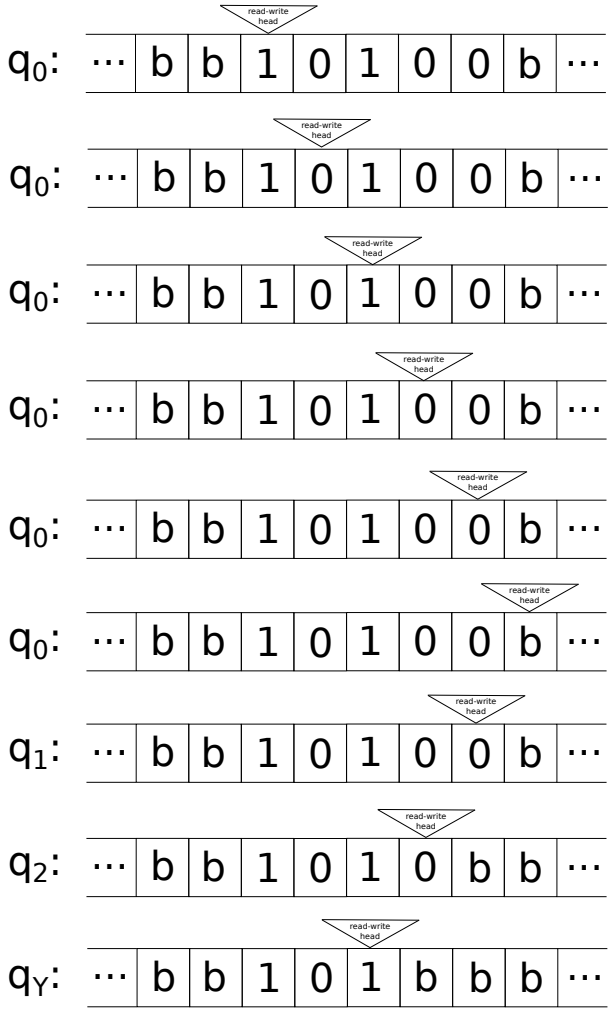

Figura : Execução do programa exemplo

Problemas de decisão - 1

Def.: Dizemos que um programa M *aceita* $x \in \mathbf{\Gamma}^*$ se o programa termina em q_Y quando o input é x. E dizemos que a linguagem L_M reconhecida pelo programa M é

$$L_M = \{x \in \mathbf{\Gamma}^* : M \text{ aceita } x\}$$

Exemplo:

$L_M = \{x \in \{0,1\}^* : \text{ os dois simbolos mais a direita de } x \text{ são } 0\}$

Def.: Seja um problema de decisão Π numa codificação **e** e uma instância **I**, a sua linguagem associada é

$$L[\Pi, \mathbf{e}] = \{x \in \mathbf{\Gamma}^* : \mathbf{\Gamma} \text{ é o alfabeto de } \mathbf{e} \text{ e}$$
$$\text{x é uma codificação em } \mathbf{e} \text{ de uma instância } \mathbf{I}\}$$

Problemas de decisão - 2

Exemplo: Numa codificação binária $\mathbf{\Gamma} = \{b, 0, 1\}$:

Instância: Um número inteiro positivo N
Questão: Existe um número m tal que $N = 4m$?

$L[\Pi, \mathbf{e}] = \{N \in \mathbf{\Gamma}^* : \text{ existe } m \in \mathbf{\Gamma}^* \text{ tal que } N = 4m\}$

Def.: Dizemos que o programa M resolve um problema de decisão Π numa codificação $\mathbf{e}$ se $L_M = L[\Pi, \mathbf{e}]$.

Exemplo: No exemplos acima temos que $L_M = L[\Pi, \mathbf{e}]$.

MT Determinística e a classe P

Def.: Seja um programa determinístico M, sua função complexidade temporal $T_M(n)$ é definida por :

$$T_M(n) = \max\{m : \text{ existe um } x \in \mathbf{\Gamma}^* \text{ , com } |x| = n, \text{ tal que } M \text{ termina depois de } m \text{ iterações de } \delta\}$$

Exemplo: A função complexidade temporal de M é $T_M(n) = n + 2$.

Def.: M é um *programa determinístico de tempo polinomial (PDTP)* se existe um polinômio p tal que $T_M(n) \leq p(n), \forall n \geq 0$.
Def.: A *classe de linguagens determinísticas de tempo polinomial* $\mathbf{P}$ é

$$\mathbf{P} = \{L : \text{ existe um PDTP } M \text{ tal que } L_M = L\}.$$

Exemplo: $[\Pi, \mathbf{e}] \in \mathbf{P}$.

MT Não-determinística e a classe NP - 1

Def.: Um *programa não-determinístico (PND)* é uma tupla $M = (\mathbf{\Gamma}, \mathbf{Q}, \delta)$, onde δ é uma relação, $\delta \subset ((\mathbf{Q} - \{q_Y, q_N\}) \times \mathbf{\Gamma}) \times (\mathbf{Q} \times \mathbf{\Gamma} \times \{-1, +1\})$.

Em um programa não determinístico é possível, testar todas as possíveis soluções S de modo a testar a validade de questão com a instância $I \in \mathbf{\Gamma}^*$.

Def.: Seja um PND M, sua função complexidade temporal $T_M(n)$ é definida por :

$$T_M(n) = \max\{\{1\} \cup \{m : \text{ existe um } x \in L_M \text{ , com } |x| = n, \text{ tal que } M \text{ termina depois de } m \text{ iterações de } \delta\}\}$$

Nota.: A complexidade temporal de um PND M é, dado um input com tamanho n, determinar em quanto tempo uma solução é verificada.

MT Não-determinística e a classe NP - 2

Def.: M é um *programa não-determinístico de tempo polinomial (PNDTP)* se existe um polinômio p tal que $T_M(n) \leq p(n), \forall n \geq 0$.
Def.: A *classe de linguagens não-determinísticas de tempo polinomial* **NP** é

$$\mathbf{NP} = \{L : \text{ existe um PNDTP } M \text{ tal que } L_M = L\}.$$

Exemplo: $\mathbf{P} \subset \mathbf{NP}$
Teo.: Se $\Pi \in \mathbf{NP}$, então existe um polinômio p tal que Π é resolvido por um programa determinístico em tempo $O(2^{p(n)})$.

MT Não-determinística e a classe NP - 3

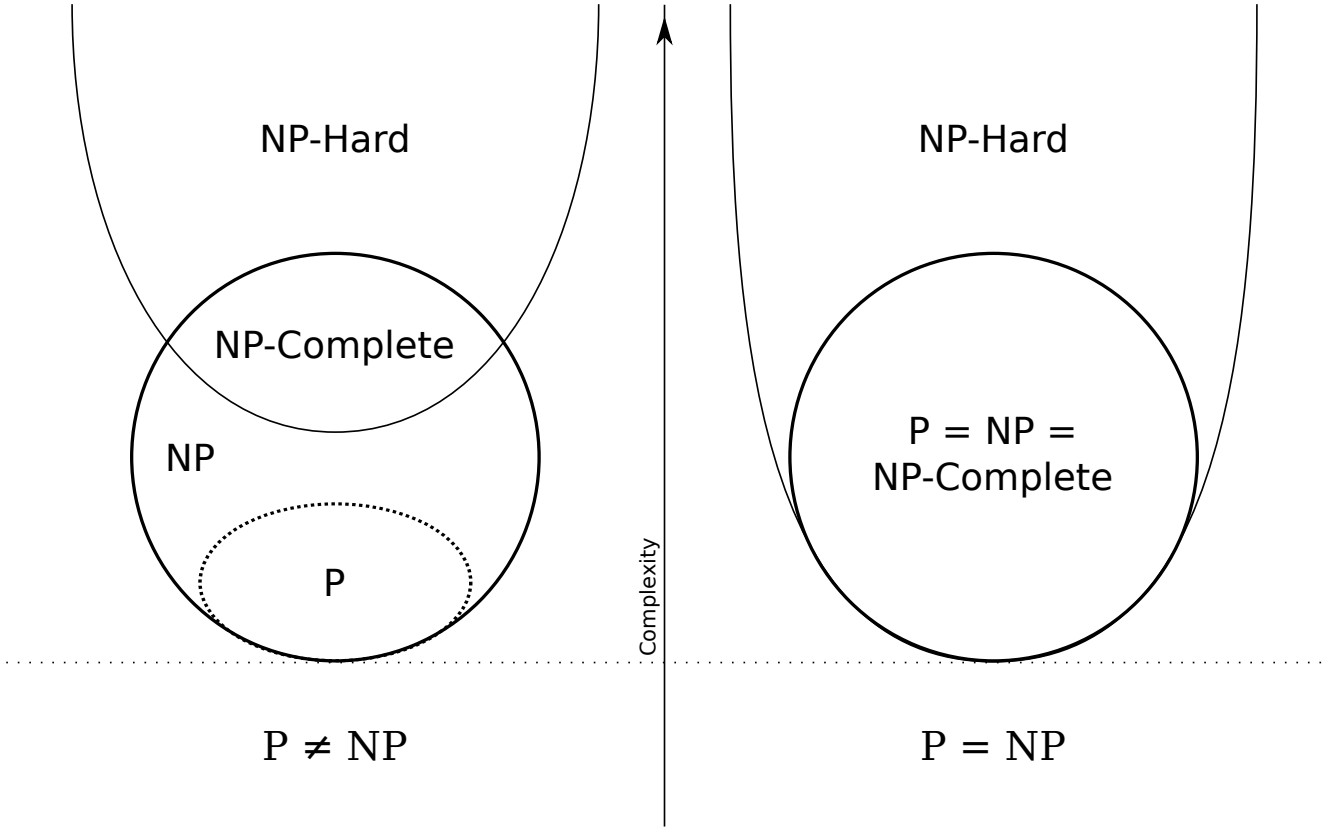

Figura : Dependências de classes

Problema NP-Completo e teorema de Cook - 1

Pergunta: $\mathbf{P} \neq \mathbf{NP}$?

Def.: Uma linguagem $L_1 \subset \mathbf{\Gamma}_1^*$ transforma polinomialmente em $L_2 \subset \mathbf{\Gamma}_2^*$ ($L_1 \propto L_2$), se existe uma função $f : \mathbf{\Gamma}_1^* \to \mathbf{\Gamma}_2^*$ tal que:

- existe um PDTP que computa f;
- $\forall x \in \mathbf{\Gamma}_1^*$, $x \in L_1 \iff x \in L_2$.

Lema: Se $L_1 \propto L_2$, então $L_2 \in \mathbf{P}$ implica $L_1 \in \mathbf{P}$.
Lema: Se $L_1 \propto L_2$ e $L_2 \propto L_3$, então $L_1 \propto L_3$

Problema NP-Completo e teorema de Cook - 2

Def.: Dizemos que $L \in \mathbf{NP}$ é *NP-Completo* se $L' \in \mathbf{NP} \Rightarrow L' \propto L$.

Observação: Seja $L \in \mathbf{NP}$ NP-Completo, se $L \in \mathbf{P}$ então $\mathbf{NP} \subset \mathbf{P}$.

Lema: Sejam $L_1, L_2 \in \mathbf{NP}$, L_1 é NP-Completo e $L_1 \propto L_2$ então L_2 é NP-Completo.
Observação: Para provar que uma linguagem L é NP-Completa, deve-se:

- provar que $L \in \mathbf{NP}$;
- provar que uma linguagem L_0 conhecidamente NP-Completa transforma polinomialmente para L, $L_0 \propto L$.

Problema NP-Completo e teorema de Cook - 3

O primeiro problema NP-Completo historicamente é o seguinte: Seja $U = \{u_1, u_2, ..., u_m\}$ um conjunto de variáveis booleanas e uma valoração de verdade $t : U \to \{V, F\}$. Uma fórmula booleana f pertencente a Algebra de Boole é satisfazível se existe t tal que $t(f) = V$.

Problema de satisfazibilidade booleana (SAT):
Instância: U um conjunto de variáveis booleanas e f uma fórmula booleana;
Questão: Existe uma valoração t tal que $t(f) = V$?

Teo.: SAT é NP-Completo.

NP-Difícil

Def.: Um problema L é dito NP-Difícil se $L' \in \mathbf{NP}$ então $L' \propto L$.

Nota: NP-Completo $\Rightarrow$ NP-Difícil, mas não vale a volta.

Exemplo: Em breve...

Outras Classes de Complexidade

$$\mathbf{P} \subset \mathbf{NP} \subset \mathbf{PSPACE} \subset \mathbf{EXPTIME} \subset \mathbf{NEXPTIME} \subset \mathbf{EXPSPACE}$$

Há um zoológico de classes de complexidade! (`https://complexityzoo.uwaterloo.ca/Complexity_Zoo`)

3-SAT

Def.: Uma fórmula 3-booleana é $f = c_1 \wedge c_2 \wedge \cdots \wedge c_m$ com
$|c_i|$ = Número de variáveis em$c_i = 3$.

Problema de 3-satisfazibilidade booleana (3-SAT)
Instância: U um conjunto de variáveis booleanas e f uma fórmula 3-booleana;
Questão: Existe uma valoração t tal que $t(f) = V$?

TRAVELING SALESMAN

Problema de caixeiro viajante:
Instância: $C = \{c_1, c_2, ..., c_m\}$ cidades e uma distância $d(c_i, c_j) \in \mathbf{Z}^+$ e uma cota $B \in \mathbf{Z}^+$;
Questão: Existe um tour $T = (c_{\pi(1)}, c_{\pi(2)}, ..., c_{\pi(m)})$ tal que

$$\sum_{i=1}^{m} d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(m)}, c_{\pi(1)}) \leq B \ ?$$

QUADRATIC DIOPHANTINE EQUATIONS

Equação Quadrática Diofantina:
Instância: Sejam $a, b, c \in \mathbf{Z}^+$;
Questão: Existe $x, y \in \mathbf{Z}^+$ tal que $ax^2 + by = c$?

ALGEBRAIC EQUATIONS OVER GF[2]

Equação Algébrica em $\mathbf{F}_2$:
Instância: Sejam $\{P_i(x_1, x_2, ..., x_n) : 1 \leq i \leq m\}$ polinômios em $\mathbf{F}_2$;
Questão: Existe $u_1, u_2, ..., u_n \in \{0, 1\}$ tal que $P_i(u_1, u_2, ..., u_n) = 0, \forall i$?

DECODING LINEAR CODES

Decodificação de Códigos Lineares:
Instância: H uma matriz $m \times n$ em $\mathbf{F}_2$, $s \in \mathbf{F}_2^m$ e $t \in \mathbf{Z}^+$;
Questão: Existe um $x \in \mathbf{F}_2^n$ tal que $xH^T = s$ com $wt(x) \leq t$?

GROBNER BASIS

- Sejam $f_0, f_1, ..., f_m \in K[x_1, ..., x_n]$, decidir se $f_0 \in \langle f_1, ..., f_m \rangle$ é EXPSPACE-Completo;
- Sejam $f_1, ..., f_m \in K[x_1, ..., x_n]$ tal que o conjunto solução é 0-dimensional, então calcular sua base de Grobner é PSPACE;
- Se $f_1, ..., f_m$ são homogêneos então calcular sua base de Grobner é NP.

References


Bardet, Magali. "On the complexity of a Gröbner basis algorithm." Algorithms Seminar, 2002–2004. 2005.


Michael R. Garey and David S. Johnson. 1979. Computers and Intractability: A Guide to the Theory of Np-Completeness. W. H. Freeman & Co., New York, NY, USA.


Pellikaan,Ruud. Coding Theory- MasterMath 2MMC30 Slides. (www.win.tue.nl/ ruudp/courses/2MMC30/week13-2.coding-theory.pdf)

The End