Melhores momentos

AULA PASSADA

Verificador

Um **verificador** é uma máquina de Turing que sempre pára.

A **linguagem de um verificador** V é

$$L_V = \{w : V \text{ aceita } \langle w, c\rangle \text{ para alguma cadeia } c\}$$

Um **verificador polinomial** consome tempo polinomial no comprimento de w.

A cadeia c é chamada de **prova** ou **certificado** de pertinência de w em L.

V = Artur
c = certificado que Merlin fornece a Artur.

As classes P e NP

P é classe de linguagens decidíveis em tempo polinomial por uma máquina de Turing determinística:

NP é a classe de linguagens que têm um verificador polinomial.

Exemplos de linguagens

CAM $= \{\langle G, s, t\rangle :$ G é um grafo orientado que possui um caminho do nó s ao nó $t\}$

PRI-MES $= \{\langle a, b\rangle :$ a e b são primos entre si$\}$

MDC $= \{\langle a, b, d\rangle :$ mdc$(a, b) = d\}$

CASAMENTO $= \{\langle G\rangle :$ G é um grafo bipartido que possui um emparelhamento perfeito$\}$

COMPOSTO $= \{\langle k\rangle : k = i \times j,\ i, j > 1\}$

CAMHAM $= \{\langle G, s, t\rangle :$ G é um grafo orientado que possui um caminho hamiltoniano de s a $t\}$

Problemas associados

CAM $=$ Dados G, s, t, decidir se existe um caminho de s a t

PRI-MES $=$ Dados a e b, decidir se mdc$(a, b) = 1$

MDC $=$ Dados a, b, d, decidir se mdc$(a, b) = d$

CASAMENTO $=$ Dado um grafo bipartido G, decidir se G tem um emparelhamento perfeito

COMPOSTO $=$ Dado k, decidir se existem $i, j > 1$ tais que $k = i \times j$

CAMHAM $=$ Dado G, s, t, decidir se existe um caminho hamiltoniano de s a $t\}$

Certificados

Certificado para CAM: caminho de s a t

Certificado para PRI-MES: inteiros x, y tais que $ax + by = 1$

Certificado para MDC: inteiros x, y tais que $ax + by = d$

Certificado para CASAMENTO: emparelhamento perfeito

Certificado para COMPOSTO: um divisor de k maior que 1

Certificado para CAMHAM: caminho hamiltoniano de s a t

Conclusões

- CAM está em NP (e também em P)
- PRI-MES está em NP (e também em P)
- MDC está em NP (e também em P)
- CASAMENTO está em NP (e também em P)
- COMPOSTO está em NP (e também em P)
- CAMHAM está em NP (**não se sabe** se está P)

AULA 8

A classe NP

MS 7.3

$\mathbf{P} \subseteq \mathbf{NP}$

Seja L uma linguagem em P.

Existe uma maquina de Turing determinística M que consome tempo polinomial e decide L.

Contruíremos um verificador polinomial V para L.

$V =$ "com entrada w, c

1. Rode M com entrada w.
2. Se M aceita w *aceite*, caso contrário, *rejeite*."

Claramente a linguagem de V é L e V consome tempo polinomial no compromento da cadeia w. ■

Complemento linguagens

O **complemento** de uma linguagem L sobre o alfabeto Σ, denotada por $\overline{L}$ é o conjunto das cadeias em Σ^* que não estão em L:

$$\overline{L} = \Sigma^* \setminus L.$$

É conveniente considerarmos o complemento de uma linguagem L que codifica um problema como sendo o conjunto das cadeias que codificam entradas válidas para o problema e que não pertencem a L.

Exemplos

$\overline{\text{CASAMENTO}} = \{\langle G \rangle : G$ é um grafo bipartido que **não** possui um emparelhamento perfeito$\}$

$\overline{\text{CAMHAM}} = \{\langle G, s, t \rangle : G$ é um grafo orientado que **não** possui um caminho hamiltoniano de s a $t\}$

$$\overline{\text{COMPOSTO}} = \{\langle k \rangle : k \neq i \times j,\ \ i, j > 1\}$$

$= \{\langle p \rangle : p$ é um número primo$\}$

$= \text{PRIME}$

$\overline{\text{CASAMENTO}}$

Certificado para não existência de um emparelhamento perfeito em um grafo bipartido (AULA 1):

$S \subseteq X$ tal que $|S| > |\text{vizinhos}(S)|$.

Teorema de Hall: G tem um emparelhamento perfeito se e somente se

$$|S| \leq |\text{vizinhos}(S)|, \quad \text{para todo } S \subseteq X.$$

Um verificador polinomial V recebe $\langle G \rangle, \langle S \rangle$ e verifica se S é um "conjunto de Hall".

Conclusão: $\overline{\text{CASAMENTO}}$ possui um verificador polinomial

$\overline{\text{CAMHAM}}$

Certificado para pertinência de $\langle G, s, t\rangle$ em $\overline{\text{CAMHAM}}$ é uma . . . lista de todos os caminhos de s a t em G. . . não serve . . .

Não se conhece verificador polinomial que recebe $\langle G, s, t\rangle, c$ e que verifica que $\langle G, s, t\rangle$ está em $\overline{\text{CAMHAM}}$.

Não se sabe $\overline{\text{CAMHAM}}$ está em P.

Certificados

Certificado para $\overline{\text{PRI-MES}}$: número $d > 1$ que divida a e b

Certificado para MDC: inteiros x, y tais que $ax + by = d$

Certificado para PRI-MES: inteiros x, y tais que $ax + by = 1$

Certificado para CASAMENTO: emparelhamento perfeito

Certificado para $\overline{\text{CASAMENTO}}$: um “conjunto de Hall”

Certificado para COMPOSTO: um divisor de k maior que 1

Certificado para CAMHAM: caminho hamiltoniano de s a t

Certificado para CAM: caminho de s a t

Certificado para $\overline{\text{CAMHAM}}$: **não** se conhece

A classe NP

NP é a classe de linguagens que tem um verificador polinomial.

Exemplos:

- CAM está em NP (e também em P)
- $\overline{\text{CAM}}$ está em NP (e também em P)
- PRI-MES está em NP (e também em P)
- $\overline{\text{PRI-MES}}$ está em NP (e também em P)
- CASAMENTO está em NP (e também em P)
- $\overline{\text{CASAMENTO}}$ está em NP (e também em P)

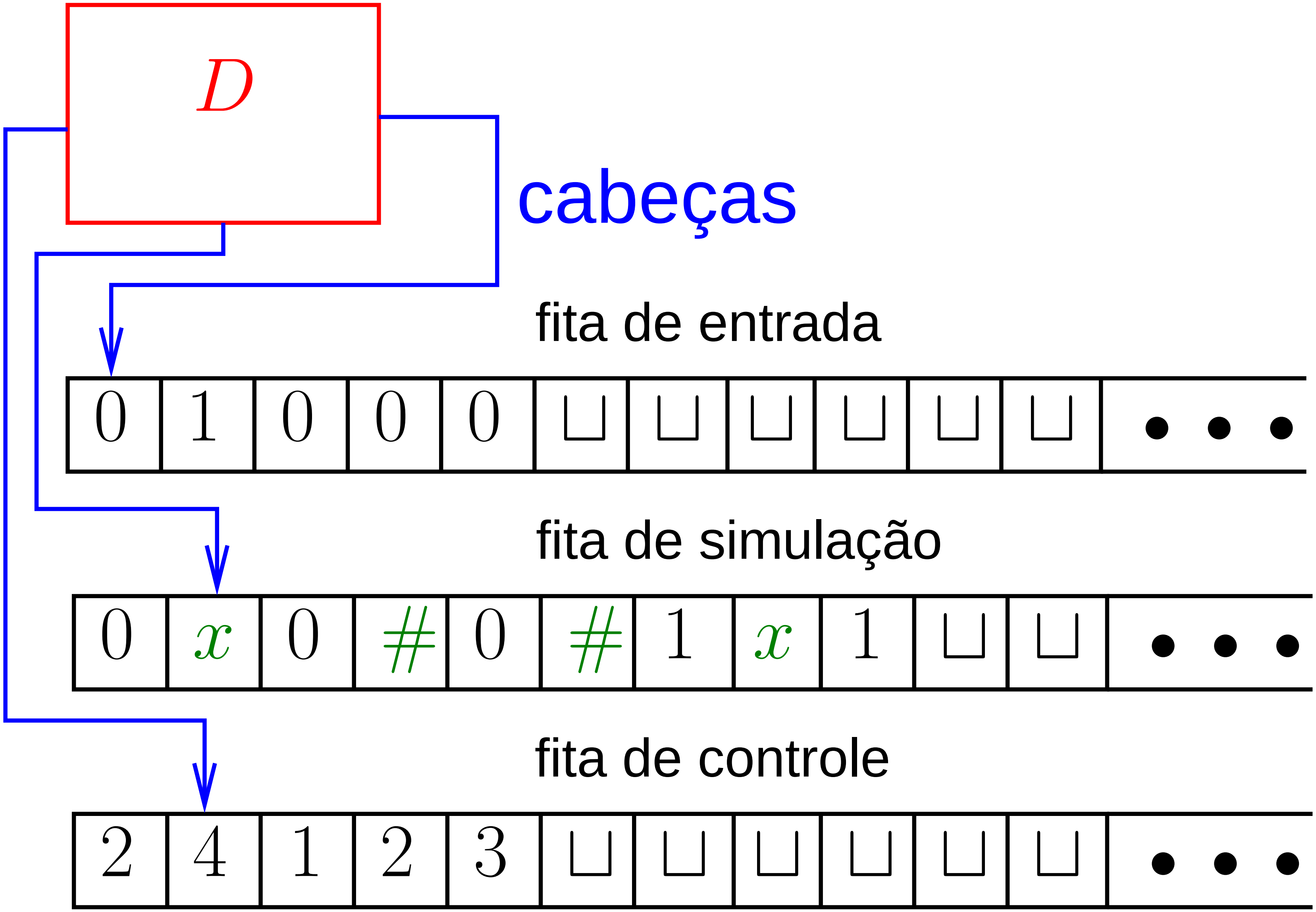
A classe NP

NP é a classe de linguagens que tem um verificador polinomial.

Mais exemplos:

- COMPOSTO está em NP (e também em P)
- PRIME está em NP, Vaughan Pratt, 1975
  (e também em P, *AKS*, 2002)
- CAMHAM está em NP (**não se sabe** se está P)
- $\overline{\text{CAMHAM}}$ **não se sabe** se está NP e tão pouco em P

Não-determinismo por determinismo



CAMHAM e não-determinismo

CAMHAM pode ser decidida por um MT não-determinística em tempo polinomial. N_1 decide CAMHAM.

$N_1 =$ "com entrada $\langle G, s, t\rangle$

1. Escreva uma lista de m números, $v_1, v_2, \ldots, v_m$ onde m é o número de nós de G. Cada número da lista é um número entre 1 e m.
2. Verifique se existe repetições na lista. Se existe, *rejeite*.
3. Verifique se $s \neq v_1$ ou $t \neq v_m$. Se for o caso, *rejeite*.
4. Para cada i entre 1 e $m-1$, verifique se $v_i v_{i+1}$ é um arco de G. Se forem todos arcos, *aceite*, senão *rejeite*."

Algoritmo não-determinístico

Recebe dois nós s e t de um grafo $G = (N, A)$ e **decide** se existe um caminho de hamiltoniano s a t.

CAMHAM-ND (N, E, s, t)

para cada i em N **faça**
  $T \leftarrow T \cup \{i\}$

$u \leftarrow s \quad T \leftarrow T - \{s\}$
enquanto existe $uv \in A$ com $v \in T$ **faça**
  escolha v tal que $uv \in A$ com $v \in T$
  $u \leftarrow v$
  $T \leftarrow T - \{v\}$

se $t = u$ **e** $T = \emptyset$
  então existe o caminho $\quad \triangleright$ aceite $\langle G, s, t \rangle$
  senão não existe o caminho $\quad \triangleright$ rejeite $\langle G, s, t \rangle$

NP e não-determinísmo

Teorema. Uma linguagem L está em NP se e somente se alguma máquina de Turing não-determinística que roda em tempo polinomial a reconhece.

Prova:
($\Rightarrow$) Seja V verificador polinomial de L.
Vamos construir MT não-determinística N que decide L em tempo polinomial. Suponha que V roda em tempo $\leq n^k$.

$N =$ "com entrada w de comprimento n:

1. Chute uma cadeia c de comprimento $\leq n^k$.
2. Simule V com entrada $\langle w, c \rangle$.
3. Se V aceita, então *aceite*, senão *rejeite*."

Seja N uma MT não determinística que decide L.

Vamos construir um verificado polinomial V para L.

$V =$ "com entrada w, c

1. Simule N com entrada w, tratando cada símbolo de c com uma descrição da escolha não-determinística a ser feita em cada passo.
2. Se essa computação de N aceita, então *aceite*, senão *rejeite*."

c é codificação do ramo da computação não-determinística que aceita w.

■

NP e não-determinísmo

NTIME$(t(n)) = \{L : L$ é decidida por uma MT
não-determinística em tempo $\mathrm{O}(t(n))\}$

Corolário. NP $= \cup_k$NTIME(n^k).

NP = Nondeterministic Polynomial time