Aula 30: Problemas P, NP, NP-árduos, e NP-completos

Uma breve introdução

David Déharbe
Programa de Pós-graduação em Sistemas e Computação
Universidade Federal do Rio Grande do Norte
Centro de Ciências Exatas e da Terra
Departamento de Informática e Matemática Aplicada

Download me from `http://DavidDeharbe.github.io`

Plano

Introdução

Arcabouço teórico

Arcabouço teórico

A classe de problemas $\mathcal{P}$

A classe de problemas $\mathcal{NP}$

NP-completeza
- Redução
- Definição e exemplo

Referência: Cormen, cap 36.

PPgSC

Introdução

- Considere os seguintes problemas:
  - Determinar se um grafo $G = (V, E)$ é Hamiltoniano: $\exists$ uma sequência de arestas que visita cada vértice exatamente uma vez, e volta ao vértice inicial
  - Dado um circuito combinacional, formado por portas lógicas (*and*, *or*, *not*), com n entradas e uma única saída, determine se existe uma combinação de valores na entrada tal que a saída é 1.
  - Dado um conjunto de números $V_1, V_2, \ldots V_n$, e um valor N, determine se existe um subconjunto cuja soma é V.
- Para nenhum desses problemas conhece-se um algoritmo de complexidade polinomial
- Se existe um algoritmo de complexidade polinomial para um, então existe um algoritmo polinomial para os demais.
- Qual é a teoria envolvida nesses resultados surpreendentes?
- Classes de problemas $\mathcal{P}$, $\mathcal{NP}$; problemas $\mathcal{NP}$-completos; problemas $\mathcal{NP}$-árduos.

PPgSC

Princípios

- É considerado tratável um problema para o qual existe um algoritmo com complexidade polinomial ($O(n^k)$)
  - na prática k é geralmente "pequeno"
- É considerado intratável um problema para o qual só são conhecidos algoritmos com complexidade não polinomial ($O(k^n)$)
- Motivação
  - abordagens heurísticas
  - consciência desta barreira teórical (provável)

PPgSC

Arcabouço teórico

- problemas de decisão
  - saída: sim / não
  - exemplo (circuito combinacional): existe uma valoração das entradas tal que a saída é 1.
- instâncias de problemas são codificadas, digamos em binário
  - (prática: entrada é uma sequência de bits)
  - complexidade é função do tamanho desta codificação
- um conjunto de instâncias
  - $\equiv$ um conjunto de palavras binárias
  - $\equiv$ uma linguagem sobre $\{0, 1\}$
- a cada problema de decisão P corresponde uma linguagem L
- decidir uma instância I de P
  - $\equiv$ decidir se uma palavra pertence à linguagem

PPgSC

Problemas de decisão

- E se o problema não for de decisão?
- Geralmente existe um problema de decisão relacionado
- Exemplo

  original Qual o tamanho do menor caminho entre dois vértices u e v de um grafo sem pesos?

  decisão Existe um caminho de u até v passando por exatamente k arestas?

PPgSC

Decisão e aceitação

Definição (Decisão)

O algoritmo A *decide* L quando, dado uma palavra x, ou A aceita x, ou A rejeita x: A termina e retorna 1 ou 0.

Definição (Decisão em tempo polinomial)

O algoritmo A *decide* L em tempo polinomial quando existe k tal que, dado x uma palavra de tamanho n, A decide se $x \in L$ em $O(n^k)$.

Definição (Aceitação)

O algoritmo A *aceita* L quando, dado uma palavra $x \in L$, A aceita x. Se $x \notin L$, ou A rejeita x, ou A não termina.

Definição (Aceitação em tempo polinomial)

O algoritmo A *aceita* L em tempo polinomial quando existe k tal que, dado $x \in L$ uma palavra de tamanho n, A aceita x em $O(n^k)$.

Decisão e aceitação

Definição (Decisão)

O algoritmo A *decide* L quando, dado uma palavra x, ou A aceita x, ou A rejeita x: A termina e retorna 1 ou 0.

Definição (Decisão em tempo polinomial)

O algoritmo A *decide* L em tempo polinomial quando existe k tal que, dado x uma palavra de tamanho n, A decide se $x \in L$ em $O(n^k)$.

Definição (Aceitação)

O algoritmo A *aceita* L quando, dado uma palavra $x \in L$, A aceita x. Se $x \notin L$, ou A rejeita x, ou A não termina.

Definição (Aceitação em tempo polinomial)

O algoritmo A *aceita* L em tempo polinomial quando existe k tal que, dado $x \in L$ uma palavra de tamanho n, A aceita x em $O(n^k)$.

- Decidir é mais "difícil" que aceitar?

A classe de problemas $\mathcal{P}$

Definição (A classe de problemas $\mathcal{P}$)

A classe de problemas de decisão polinomiais é $\mathcal{P} = \{L \subseteq \{0,1\}^* \text{ existe um algoritmo que decide } L \text{ em tempo polinomial}\}$.

Teorema

A classe de problemas $\mathcal{P}$ é a classe de problemas que são aceitos em tempo polinomial.

PPgSC

Verificação

Definição (Verificação)

O algoritmo A *verifica* L quando, dado uma palavra $x \in L$, e um certificado y, $A(x, y) = 1$.

O algoritmo de verificação confere se uma instância x pertence à linguagem com base uma evidência y.

Definição (A classe de problemas $\mathcal{NP}$)

A classe de problemas $\mathcal{NP}$ é a classe de problemas L tais que existe um algoritmo de verificação polinomial.

Definição (A classe de problemas co-$\mathcal{NP}$)

A classe de problemas co-$\mathcal{NP}$ é a classe de problemas L tais que existe um algoritmo de verificação polinomial para $\bar{L}$.

PPgSC

Relação entre classes de problemas

- $\mathcal{P} \subseteq \mathcal{NP}$, $\mathcal{P} \subseteq$ co-$\mathcal{NP}$
- $\mathcal{P} = \mathcal{NP}$?
- $\mathcal{NP} =$ co-$\mathcal{NP}$?
- $\mathcal{P} = \mathcal{NP} \cap$ co-$\mathcal{NP}$?

PPgSC

Redução entre problemas

Definição (Redução entre problemas)

Um problema (linguagem) L_1 é redutível polinomialmente em um problema L_2 se existe uma função f calculável em tempo polinomial tal que $\forall x \in \{0,1\}^* \cdot x \in L_1 \iff f(x) \in L_2$.

- Solucionar L_1 não é mais difícil que solucionar L_2.
- Notação: $L_1 \leq_P L_2$

PPgSC

Redução e $\mathcal{P}$

Lema

Seja $L_1, L_2 \subseteq \{0,1\}^*$, *tais que* $L_1 \leq_P L_2$. *Então* $L_2 \in \mathcal{P} \Rightarrow L_1 \in \mathcal{P}$.

$\mathcal{NP}$-completeza

Definição ($\mathcal{NP}$-árduo)

$L \subseteq \{0,1\}^*$ é $\mathcal{NP}$-árduo se, para cada $L' \in \mathcal{NP}$, $L' \leq_P L$.

Definição ($\mathcal{NP}$-completo)

$L \subseteq \{0,1\}^*$ é $\mathcal{NP}$-completo se

1. $L \in \mathcal{NP}$, e
2. $L' \leq_P L$, para cada $L' \in \mathcal{NP}$ (L é NP-árduo)

- O problema SAT de determinar se um dado circuito lógico pode ter a sua saída setada a um é um problema $\mathcal{NP}$-completo.
- Há centenas de outros problemas computacionais que foram mostrados como send $\mathcal{NP}$-completos.

PPgSC