Problemas NP-Completos

Fábio Botler

Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro

Aula passada

- Algoritmos eficientes
- Problemas tratáveis e intratáveis
- Problemas algorítmicos
- Codificações
- Tipos de problemas
- A Classe P
- Problemas aparentemente difíceis

Aula de hoje

- A classe NP
- A questão P=NP
- Complementos de Problemas
- Transformações Polinomiais
- Alguns problemas NP-Completos

A classe NP

A classe NP

Um **certificado** para um problema Π é uma justificativa para a resposta SIM.

clique (G, k)

A classe NP

Um **certificado** para um problema Π é uma justificativa para a resposta SIM.

$$NP \quad = \quad \left\{ \begin{array}{l} \text{problemas de decisão para os quais existe} \\ \text{certificado que pode ser reconhecido por} \\ \text{um algoritmo polinomial} \\ \\ \end{array} \right\}$$

A classe NP

Um **certificado** para um problema Π é uma justificativa para a resposta SIM.

$$NP = \left\{ \begin{array}{l} \text{problemas de decisão para os quais existe} \\ \text{certificado que pode ser reconhecido por} \\ \text{um algoritmo polinomial } \textbf{no tamanho} \\ \textbf{da sua entrada} \end{array} \right\}$$

A classe NP

Um **certificado** para um problema Π é uma justificativa para a resposta SIM.

$$NP = \left\{ \begin{array}{l} \text{problemas de decisão para os quais existe} \\ \text{certificado que pode ser reconhecido por} \\ \text{um algoritmo polinomial } \textbf{no tamanho} \\ \textbf{da sua entrada} \end{array} \right\}$$

DA INSTÂNCIA, NÃO DO CERTIFICADO

▷ O certificado também deve ser polinomial
no tamanho da entrada!

$$n! = n \cdot (n-1) \cdot (n-2) \cdots \cdot 4 \cdot 3 \cdot 2 \cdot 1$$
$$\geq 2 \cdot 2 \quad 2 \quad \cdots \quad 2 \cdot 2 \quad 2 \quad 1 \quad \geq 2^n$$

A classe NP

Um **certificado** para um problema Π é uma justificativa para a resposta SIM.

$$NP \quad = \quad \left\{ \begin{array}{l} \text{problemas de decisão para os quais existe} \\ \text{certificado que pode ser reconhecido por} \\ \text{um algoritmo polinomial } \textbf{no tamanho} \\ \textbf{da sua entrada} \end{array} \right\}$$

▷ O certificado também deve ser polinomial no tamanho da entrada!

Exemplo

Clique, conjunto independente, ciclo Hamiltoniano

Como verificar se um problema Π pertence a NP?

Como verificar se um problema Π pertence a NP?

$\rightarrow$ CERTIFICADO

- $\triangleright$ Definir uma justificativa J conveniente para a resposta SIM

Como verificar se um problema Π pertence a NP?

- $\triangleright$ Definir uma justificativa J conveniente para a resposta SIM
- $\triangleright$ Elaborar um algoritmo R para reconhecer se J está correta.

  O conjunto de dados desse algoritmo são os pares (I, J), onde I é uma instância de Π e J é uma justificativa.

Como verificar se um problema Π pertence a NP?

- $\triangleright$ Definir uma justificativa J conveniente para a resposta SIM
- $\triangleright$ Elaborar um algoritmo R para reconhecer se J está correta.

  O conjunto de dados desse algoritmo são os pares (I, J), onde I é uma instância de Π e J é uma justificativa.

Se R for polinomial no tamanho de I, então Π pertence a NP.

Satisfatibilidade

DADOS:	Uma expressão booleana E na Forma Normal Conjuntiva (FNC)
OBJETIVO:	E é satisfatível?

Satisfatibilidade

DADOS:	Uma expressão booleana E na Forma Normal Conjuntiva (FNC)
OBJETIVO:	E é satisfatível?

Certificado: uma atribuição para cada variável de E

Satisfatibilidade

DADOS:	Uma expressão booleana E na Forma Normal Conjuntiva (FNC)
OBJETIVO:	E é satisfatível?

Certificado:	uma atribuição para cada variável de E
Reconhecimento:	substituir em E cada variável. por seu valor atribuído.

Satisfatibilidade

$(\ \vee\ \underset{T}{\vee}\ ) \wedge (\ \vee\ \underset{T}{\vee}\ ) \wedge \cdots \wedge (\ \underset{T}{}\ )$

DADOS: Uma expressão booleana E
na Forma Normal Conjuntiva (FNC)

OBJETIVO: E é satisfatível?

Certificado: uma atribuição para cada variável de E

Reconhecimento: substituir em E cada variável.
por seu valor atribuído.

se cada cláusula possuir pelo menos uma atribuição 1, então E é satisfatível.

Clique

(G, K)

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma clique de tamanho pelo menos k?

Clique

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma clique de tamanho pelo menos k?

Certificado: um subconjunto de vértices V'

Clique

DADOS: Um grafo G e um inteiro k
OBJETIVO: G possui uma clique
de tamanho pelo menos k?

Certificado: um subconjunto de vértices V'

Reconhecimento: verificar se $|V'| \geq k$, e
se há par $x, y \in V'$ tal que $xy \notin E(G)$.

Clique

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma clique de tamanho pelo menos k?

Certificado:	um subconjunto de vértices V'
Reconhecimento:	verificar se $\vert V'\vert \geq k$, e se há par $x, y \in V'$ tal que $xy \notin E(G)$. se $\vert V'\vert \geq k$ e não houver tal par, então a justificativa está correta.

Cobertura por vértices

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma cobertura por vértices de tamanho no máximo k?

Cobertura por vértices

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma cobertura por vértices de tamanho no máximo k?

Certificado: um subconjunto de vértices V'

Cobertura por vértices

DADOS: Um grafo G e um inteiro k
OBJETIVO: G possui uma cobertura por vértices de tamanho no máximo k?

Certificado: um subconjunto de vértices V'

Reconhecimento: verificar se $|V'| \leq k$, e
se há aresta $xy \in E(G)$ tal que $x, y \notin V'$.

Cobertura por vértices

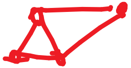
DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma cobertura por vértices de tamanho no máximo k?

Certificado:	um subconjunto de vértices V'
Reconhecimento:	verificar se $\|V'\| \leq k$, e se há aresta $xy \in E(G)$ tal que $x, y \notin V'$. se $\|V'\| \leq k$ e não houver tal aresta, então a justificativa está correta.

Clique máxima

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	uma clique máxima de G tem tamanho k?

Clique máxima



DADOS: Um grafo G e um inteiro k

OBJETIVO: uma clique máxima de G tem tamanho k?

Certificado: um conjunto S com todas as cliques maximais de G

Clique máxima

DADOS: Um grafo G e um inteiro k
OBJETIVO: uma clique máxima de G tem tamanho k?

Certificado: um conjunto S com todas as cliques maximais de G

Reconhecimento: comprovar se S tem todas as cliques maximais de G, e verificar se o tamanho da maior clique é k

Clique máxima

DADOS: Um grafo G e um inteiro k
OBJETIVO: uma clique máxima de G tem tamanho k?

Certificado: um conjunto S com todas as cliques maximais de G

Reconhecimento: comprovar se S tem todas as cliques maximais de G, e verificar se o tamanho da maior clique é k

se ambas respostas forem afirmativas, então a justificativa está correta.

PROBLEMA:
É POLINOMIAL?

Caminho mínimo

DADOS:	Um grafo G, dois vértices $x, y \in V(G)$, e um inteiro k
OBJETIVO:	G possui um caminho ligando x a y com comprimento no máximo k?

Caminho mínimo

DADOS:	Um grafo G, dois vértices $x, y \in V(G)$, e um inteiro k
OBJETIVO:	G possui um caminho ligando x a y com comprimento no máximo k?

Certificado:

Caminho mínimo

DADOS: Um grafo G, dois vértices $x, y \in V(G)$,
e um inteiro k

OBJETIVO: G possui um caminho ligando x a y
com comprimento no máximo k?

Certificado:

Reconhecimento: executar uma busca em largura começando em x
e guardando, para cada vértice z,
sua distância $d(z)$ até x.

Caminho mínimo

DADOS:	Um grafo G, dois vértices $x, y \in V(G)$, e um inteiro k
OBJETIVO:	G possui um caminho ligando x a y com comprimento no máximo k?

Certificado:

Reconhecimento: executar uma busca em largura começando em x e guardando, para cada vértice z, sua distância $d(z)$ até x.

se $d(y) \leq k$, então a justificativa está correta.

Wolfram MathWorld™ the web's most extensive mathematics resource

Built with Mathematica Technology

AKS Primality Test

In August 2002, M. Agrawal and colleagues announced a deterministic algorithm for determining if a number is prime that runs in polynomial time (Agrawal *et al.* 2004). While this had long been believed possible (Wagon 1991), no one had previously been able to produce an explicit polynomial time deterministic algorithm (although probabilistic algorithms were known that seem to run in polynomial time). This test is now known as the Agrawal-Kayal-Saxena primality test, cyclotomic AKS test, or AKS primality test.

Annals of Mathematics, **160** (2004), 781–793

PRIMES is in P

By Manindra Agrawal, Neeraj Kayal, and Nitin Saxena*

Abstract

We present an unconditional deterministic polynomial-time algorithm that determines whether an input number is prime or composite.

Primalidade

DADOS:	Um inteiro n
OBJETIVO:	n é primo?

Primalidade

DADOS:	Um inteiro n
OBJETIVO:	n é primo?

Certificado:

Primalidade

DADOS: Um inteiro n
OBJETIVO: n é primo?

Certificado:

Reconhecimento: executar o AKS.

Primalidade

DADOS: Um inteiro n
OBJETIVO: n é primo?

Certificado:

Reconhecimento: executar o AKS.

se o AKS retornar SIM,
então a justificativa está correta.

Qual a relação entre as classes P e NP?

Qual a relação entre as classes P e NP?

Proposição

$P \subseteq NP$

Qual a relação entre as classes P e NP?

Proposição

$P \subseteq NP$

Proof.

Seja Π um problema em P, e A um algoritmo polinomial que decide Π.

Qual a relação entre as classes P e NP?

Proposição

$P \subseteq NP$

Proof.

Seja Π um problema em P, e A um algoritmo polinomial que decide Π.

Certificado:

Qual a relação entre as classes P e NP?

Proposição

$P \subseteq NP$

Proof.

Seja Π um problema em P, e A um algoritmo polinomial que decide Π.

Certificado:

Reconhecimento: executar A.

Qual a relação entre as classes P e NP?

Proposição

$P \subseteq NP$

Proof.

Seja Π um problema em P, e A um algoritmo polinomial que decide Π.

Certificado:

Reconhecimento: executar A.

se A retornar SIM,
então a justificativa está correta.

$\square$

$$NP \subseteq P?$$

Classes de Problemas

$$P = \left\{ \begin{array}{l} \text{problemas de decisão que podem ser} \\ \text{decididos por um algoritmo polinomial} \end{array} \right\}$$

$$NP = \left\{ \begin{array}{l} \text{problemas de decisão para os quais existe} \\ \text{certificado que pode ser reconhecido por} \\ \text{um algoritmo polinomial no tamanho} \\ \text{de sua entrada} \end{array} \right\}$$

2^{n} $\quad 4^{n}$

$2^{p(n)}$ $\quad 2^{n^5+n^2+3}$ $\quad \binom{n}{k} \leq \underbrace{2^{n}}_{|\text{SUBCONJS DE } V|}$

Proposição

Seja $\Pi \in NP$. Existe algoritmo exponencial que decide Π.

OBS: SE $\Pi \in NP$, O CERTIFICADO TAMBÉM TEM TAMANHO POLINOMIAL.

Complementos de Problemas

Certificados

Um **certificado** para um problema Π é uma justificativa para a resposta SIM.

Um **co-certificado** para um problema Π é uma justificativa para a resposta NÃO.

Classes de Problemas

$$NP = \left\{ \begin{array}{l} \text{problemas de decisão para os quais existe} \\ \text{certificado que pode ser reconhecido por} \\ \text{um algoritmo polinomial no tamanho} \\ \text{da sua entrada} \end{array} \right\}$$

$$\text{Co-}NP = \left\{ \begin{array}{l} \text{problemas de decisão para os quais existe} \\ \textbf{co}\text{-certificado que pode ser reconhecido por} \\ \text{um algoritmo polinomial no tamanho} \\ \text{da sua entrada} \end{array} \right\}$$

Definição

Dado um problema de decisão Π, o problema $\overline{\Pi}$ é o problema tal que $\Pi(I) = SIM$ se e somente se $\overline{\Pi}(I) = NÃO$.

Satisfatibilidade

DADOS:	Uma expressão booleana E na Forma Normal Conjuntiva (FNC)
OBJETIVO:	decidir se E é satisfatível.

Satisfatibilidade

DADOS:	Uma expressão booleana E na Forma Normal Conjuntiva (FNC)
OBJETIVO:	decidir se E é satisfatível.

$\overline{\texttt{Satisfatibilidade}}$

DADOS:	Uma expressão booleana E na Forma Normal Conjuntiva (FNC)
OBJETIVO:	decidir se E **não** é satisfatível.

Clique

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	decidir se G possui uma clique de tamanho pelo menos k.

Clique

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	decidir se G possui uma clique de tamanho pelo menos k.

$\overline{\texttt{Clique}}$

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	decidir se G **não** possui uma clique de tamanho pelo menos k.

Cobertura por vértices

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	decidir se G possui uma cobertura por vértices de tamanho no máximo k.

Cobertura por vértices

CERTIFICADO

DADOS: Um grafo G e um inteiro k
OBJETIVO: decidir se G possui uma cobertura por vértices de tamanho no máximo k.

`Cobertura por vértices`

DADOS: Um grafo G e um inteiro k
OBJETIVO: decidir se G **não** possui uma cobertura por vértices de tamanho no máximo k.

CO-CERTIFICADO

Proposição

$Co\text{-}NP = \{\overline{\Pi} \colon \Pi \in NP\}$.

Proposição

Co-NP $= \{\overline{\Pi} \colon \Pi \in NP\}$.

Proposição

$\Pi \in NP$ *se e somente se* $\overline{\Pi} \in$ *Co-NP*.

Proposição

$Co\text{-}NP = \{\overline{\Pi} \colon \Pi \in NP\}$.

Proposição

$\Pi \in NP$ *se e somente se* $\overline{\Pi} \in Co\text{-}NP$.

Proposição

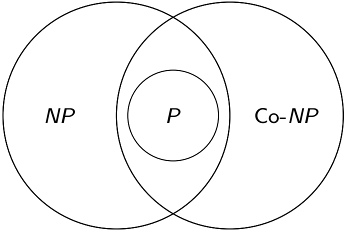
$P \subseteq Co\text{-}NP$.

co-cert. $= \emptyset$

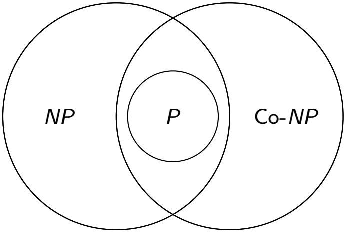
Alg. Rec. Poly $\rightsquigarrow A_\pi$

Qual a relação entre as classes P e NP?

Qual a relação entre as classes P e NP?



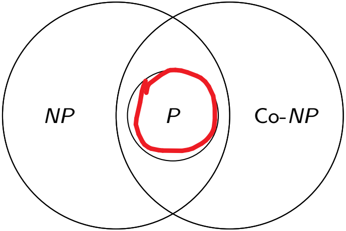
Qual a relação entre as classes P e NP?



Problemas em aberto:

- $P = NP$?
- $NP = \text{Co-}NP$?
- $P = NP \cap \text{Co-}NP$?

Qual a relação entre as classes P e NP?



$\bar{\Pi} \in P$

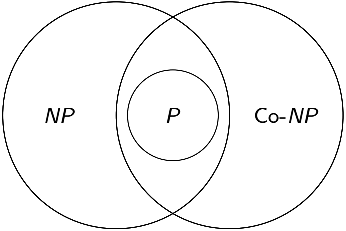
$\Rightarrow \Pi \in P$

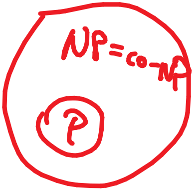
Problemas em aberto:

- $P = NP$?
- $NP =$ Co-NP?
- $P = NP \cap$ Co-NP?

Se $P = NP$, então $NP =$ Co-NP e $P = NP \cap$ Co-NP.

Qual a relação entre as classes P e NP?





Problemas em aberto:

- $P = NP$?
- $NP =$ Co-NP?
- $P = NP \cap$ Co-NP?

Se $P = NP$, então $NP =$ Co-NP e $P = NP \cap$ Co-NP.

Se $NP =$ Co-NP, então $P = NP$?

$P \times NP$

$P \times NP$

$NP \subseteq P$?

- ▷ Como resolver esse problema?

$P \times NP$

- Como resolver esse problema?

- É possível transformar um algoritmo polinomial para reconhecer certificados em um algoritmo polinomial que decide o problema?

$P \times NP$

- Como resolver esse problema?

- É possível transformar um algoritmo polinomial para reconhecer certificados em um algoritmo polinomial que decide o problema?

- Há algum problema mais difícil em NP?

Transformações polinomiais

Transformações polinomiais

Como decidir se um problema é mais difícil que outro?

Transformações polinomiais

Como decidir se um problema é mais difícil que outro?

Definição

Uma **transformação polinomial** *ou* **redução** *é um processo f que transforma instâncias de um problema Π_1 em instâncias um problema Π_2 tal que:*

1. $\Pi_1(I) = SIM$ *se e somente se* $\Pi_2(f(I)) = SIM$
2. f *pode ser computada em tempo polinomial*

Transformações polinomiais

Como decidir se um problema é mais difícil que outro?

Definição

Uma **transformação polinomial** *ou* **redução** *é um processo f que transforma instâncias de um problema Π_1 em instâncias um problema Π_2 tal que:*

1. $\Pi_1(I) = SIM$ *se e somente se* $\Pi_2(f(I)) = SIM$
2. f *pode ser computada em tempo polinomial*

▷ notação: $\Pi_1 \propto \Pi_2$

Π_1: `Clique`

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma clique de tamanho pelo menos k?

Π_2: `Conjunto Independente de Vértices`

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui um conjunto independente de vértices de tamanho pelo menos k?

Π_1: `Clique`

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma clique de tamanho pelo menos k?

Π_2: `Conjunto Independente de Vértices`

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui um conjunto independente de vértices de tamanho pelo menos k?

- $\triangleright$ Seja $f_{12}\big((G,k)\big) = (\overline{G}, k)$.
- $\triangleright$ G possui uma clique de tamanho pelo menos k se e somente se $\overline{G}$ possui um conjunto independente de tamanho pelo menos k.

Π_1: `Clique`

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma clique de tamanho pelo menos k?

Π_2: `Conjunto Independente de Vértices`

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui um conjunto independente de vértices de tamanho pelo menos k?

- $\triangleright$ Seja $f_{12}\big((G,k)\big) = (\overline{G},k)$.
- $\triangleright$ G possui uma clique de tamanho pelo menos k se e somente se $\overline{G}$ possui um conjunto independente de tamanho pelo menos k.
- $\triangleright$ logo $\Pi_1\big((G,k)\big) = \Pi_2\big((\overline{G},k)\big) = \Pi_2\Big(f_{12}\big((G,k)\big)\Big)$.

Π_1: `Clique`

DADOS: Um grafo G e um inteiro k

OBJETIVO: G possui uma clique de tamanho pelo menos k?

Π_2: `Conjunto Independente de Vértices`

DADOS: Um grafo G e um inteiro k

OBJETIVO: G possui um conjunto independente de vértices de tamanho pelo menos k?

- $\triangleright$ Seja $f_{12}\big((G,k)\big) = (\overline{G},k)$.
- $\triangleright$ G possui uma clique de tamanho pelo menos k se e somente se $\overline{G}$ possui um conjunto independente de tamanho pelo menos k.
- $\triangleright$ logo $\Pi_1\big((G,k)\big) = \Pi_2\big((\overline{G},k)\big) = \Pi_2\Big(f_{12}\big((G,k)\big)\Big)$.
- $\triangleright$ $\Pi_1 \propto \Pi_2$

Π_2: `Conjunto Independente de Vértices`

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui um conjunto independente de vértices de tamanho pelo menos k?

Π_3: `Cobertura por vértices`

DADOS:	Um grafo G e um inteiro k
OBJETIVO:	G possui uma cobertura por vértices de tamanho no máximo k?

Π_2: `Conjunto Independente de Vértices`

DADOS: Um grafo G e um inteiro k

OBJETIVO: G possui um conjunto independente de vértices de tamanho pelo menos k?

Π_3: `Cobertura por vértices`

DADOS: Um grafo G e um inteiro k

OBJETIVO: G possui uma cobertura por vértices de tamanho no máximo k?

- ▷ Seja $f_{23}\big((G,k)\big) = (G, |V(G)| - k)$.
- ▷ G possui um conjunto independente de tamanho pelo menos k se e somente se G possui uma cobertura por vértices de tamanho no máximo $|V(G)| - k$.

Π_2: Conjunto Independente de Vértices

DADOS: Um grafo G e um inteiro k

OBJETIVO: G possui um conjunto independente de vértices de tamanho pelo menos k?

Π_3: Cobertura por vértices

DADOS: Um grafo G e um inteiro k

OBJETIVO: G possui uma cobertura por vértices de tamanho no máximo k?

- $\triangleright$ Seja $f_{23}\big((G,k)\big) = (G, |V(G)| - k)$.
- $\triangleright$ G possui um conjunto independente de tamanho pelo menos k se e somente se G possui uma cobertura por vértices de tamanho no máximo $|V(G)| - k$.
- $\triangleright$ logo $\Pi_1\big((G,k)\big) = \Pi_2\big((G, |V(G)| - k)\big) = \Pi_2\big(f_{23}\big((G,k)\big)\big)$.

Π_2: Conjunto Independente de Vértices

DADOS: Um grafo G e um inteiro k

OBJETIVO: G possui um conjunto independente de vértices de tamanho pelo menos k?

Π_3: Cobertura por vértices

DADOS: Um grafo G e um inteiro k

OBJETIVO: G possui uma cobertura por vértices de tamanho no máximo k?

- $\triangleright$ Seja $f_{23}\big((G, k)\big) = (G, |V(G)| - k)$.
- $\triangleright$ G possui um conjunto independente de tamanho pelo menos k se e somente se G possui uma cobertura por vértices de tamanho no máximo $|V(G)| - k$.
- $\triangleright$ logo $\Pi_1\big((G, k)\big) = \Pi_2\big((G, |V(G)| - k)\big) = \Pi_2\big(f_{23}\big((G, k)\big)\big)$.
- $\triangleright$ $\Pi_2 \propto \Pi_3$

A relação $\propto$

Observação (Transitividade)

Se $\Pi_1 \propto \Pi_2$ *e* $\Pi_2 \propto \Pi_3$, *então* $\Pi_1 \propto \Pi_3$.

A relação $\propto$

$$f_{23}(f_{12}(G,k)) = (\bar{G}, n-k)$$

Observação (Transitividade)

Se $\Pi_1 \propto \Pi_2$ *e* $\Pi_2 \propto \Pi_3$, *então* $\Pi_1 \propto \Pi_3$.

Lema

Sejam $\Pi_1, \Pi_2 \in NP$ *dois problemas tais que* $\Pi_1 \propto \Pi_2$.
Se $\Pi_2 \in P$, *então* $\Pi_1 \in P$.

$\Pi_a \xrightarrow[\propto]{f_{ab}} \Pi_b$ A_b → SIM / NÃO

$$A_b(f_{ab}(I)) = \Delta_a$$

A relação $\propto$

- A redução f transforma instância genérica de Π_1 em uma instância particular de Π_2.

A relação $\propto$

- $\triangleright$ A redução f transforma instância genérica de Π_1 em uma instância particular de Π_2.

- $\triangleright$ Se $\Pi_1 \propto \Pi_2$, então "Π_2 é **pelo menos tão difícil quanto** Π_1".

A relação $\propto$

- A redução f transforma instância genérica de Π_1 em uma instância particular de Π_2.

- Se $\Pi_1 \propto \Pi_2$, então
  "Π_2 é **pelo menos tão difícil quanto** Π_1".

- Se $\Pi_1 \propto \Pi_2$ e $\Pi_2 \propto \Pi_1$, então
  "Π_2 e Π_1 são **equivalentes**".

A relação $\propto$

- $\triangleright$ A redução f transforma instância genérica de Π_1 em uma instância particular de Π_2.
- $\triangleright$ Se $\Pi_1 \propto \Pi_2$, então "Π_2 é **pelo menos tão difícil quanto** Π_1".
- $\triangleright$ Se $\Pi_1 \propto \Pi_2$ e $\Pi_2 \propto \Pi_1$, então "Π_2 e Π_1 são **equivalentes**".
- $\triangleright$ $\propto$ divide os problemas em *NP* em classes de equivalência.

Problemas NP-Completos

Definição

Um problema Π *é dito* **NP-Completo** *se*

(i) $\Pi \in NP$

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

Problemas NP-Completos

Definição

Um problema Π *é dito* **NP-Completo** *se*

(i) $\Pi \in NP$

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

$\triangleright$ *notação:* $\Pi \in NPC$

Problemas NP-Completos

Definição

Um problema Π *é dito* **NP-Completo** *se*

(i) $\Pi \in NP$

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

$\triangleright$ *notação:* $\Pi \in NPC$

Um problema Π *é dito* **NP-Difícil** *se*

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

Problemas NP-Completos

Definição

Um problema Π *é dito* **NP-Completo** *se*

(i) $\Pi \in NP$

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

$\triangleright$ *notação:* $\Pi \in NPC$

Um problema Π *é dito* **NP-Difícil** *se*

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

$\triangleright$ Se um problema NP-Completo está em P,

Problemas NP-Completos

Definição

Um problema Π *é dito* **NP-Completo** *se*

(i) $\Pi \in NP$

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

$\triangleright$ *notação:* $\Pi \in NPC$

Um problema Π *é dito* **NP-Difícil** *se*

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

$\triangleright$ Se um problema NP-Completo está em P, então $NP \subseteq P$.

Problemas NP-Completos

Definição

Um problema Π *é dito* **NP-Completo** *se*

(i) $\Pi \in NP$

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

$\triangleright$ *notação:* $\Pi \in NPC$

Um problema Π *é dito* **NP-Difícil** *se*

(ii) $\Pi' \propto \Pi$ *para todo* $\Pi' \in NP$

$\triangleright$ Se um problema NP-Completo está em P, então $NP \subseteq P$.

$\triangleright$ Os problemas NP-Completos são os problemas mais difíceis em NP.

Como mostramos que um problema Π é NP-Completo?

Como mostramos que um problema Π é NP-Completo?

(i) $\Pi \in NP$

Como mostramos que um problema Π é NP-Completo?

(i) $\Pi \in NP$

(ii) construir para todo $\Pi' \in NP$ uma redução f de Π' para Π.

Como mostramos que um problema Π é NP-Completo?

(i) $\Pi \in NP$

(ii) construir para todo $\Pi' \in NP$ uma redução f de Π' para Π.

$\triangleright$ se apenas (ii) vale, então Π é NP-Difícil.

Como mostramos que um problema Π é NP-Completo?

(i) $\Pi \in NP$

(ii) construir para todo $\Pi' \in NP$ uma redução f de Π' para Π.

▷ se apenas (ii) vale, então Π é NP-Difícil.

Lema

Sejam $\Pi_1, \Pi_2 \in NP$. *Se* Π_1 *é NP-Completo e* $\Pi_1 \propto \Pi_2$,

Como mostramos que um problema Π é NP-Completo?

(i) $\Pi \in NP$

(ii) construir para todo $\Pi' \in NP$ uma redução f de Π' para Π.

▷ se apenas (ii) vale, então Π é NP-Difícil.

Lema

$\pi' \leq$

Sejam $\Pi_1, \Pi_2 \in NP$. Se Π_1 é NP-Completo e $\Pi_1 \propto \Pi_2$, então Π_2 é NP-Completo.

Como mostramos que um problema Π é NP-Completo?

(i) $\Pi \in NP$

(ii) construir para todo $\Pi' \in NP$ uma redução f de Π' para Π.

- $\triangleright$ se apenas (ii) vale, então Π é NP-Difícil.

Lema

Sejam $\Pi_1, \Pi_2 \in NP$. Se Π_1 é NP-Completo e $\Pi_1 \propto \Pi_2$, então Π_2 é NP-Completo.

Agora temos duas formas de mostrar que um problema é NP-Completo!

Como mostramos que um problema Π é NP-Completo?

(i) $\Pi \in NP$

(ii) construir para todo $\Pi' \in NP$ uma redução f de Π' para Π.

▷ se apenas (ii) vale, então Π é NP-Difícil.

Lema

Sejam $\Pi_1, \Pi_2 \in NP$. Se Π_1 é NP-Completo e $\Pi_1 \propto \Pi_2$, então Π_2 é NP-Completo.

Agora temos duas formas de mostrar que um problema é NP-Completo!

Teorema (Cook–Levin, 1971)

Satisfatibilidade ($\mathtt{SAT}$) é NP-Completo.

The Complexity of Theorem-Proving Procedures

Stephen A. Cook

University of Toronto

КРАТКИЕ СООБЩЕНИЯ

УДК 519.14

УНИВЕРСАЛЬНЫЕ ЗАДАЧИ ПЕРЕБОРА

Л. А. Левин

В статье рассматривается несколько известных массовых задач «переборного типа» и доказывается, что эти задачи можно решать лишь за такое время, за которое можно решать вообще любые задачи указанного типа.

Teorema

`Clique` é NP-Completo.

Teorema

Clique é NP-Completo.

Proof.

- ▷ Existe redução f de SAT para Clique.

Teorema

$\mathtt{Clique}$ é NP-Completo.

Proof.

- $\triangleright$ Existe redução f de `SAT` para `Clique`.
- $\triangleright$ dois literais de cláusulas **distintas** são **compatíveis** se eles podem assumir 1 simultaneamente.

Teorema

$\mathtt{Clique}$ é NP-Completo.

Proof.

- ▷ Existe redução f de `SAT` para `Clique`.
- ▷ dois literais de cláusulas **distintas** são **compatíveis** se eles podem assumir 1 simultaneamente.
- ▷ Seja $E = C_1 \wedge \cdots \wedge C_p$ uma expressão booleana com cláusulas C_i.

Teorema

$\mathtt{Clique}$ é NP-Completo.

Proof.

- ▷ Existe redução f de `SAT` para `Clique`.
- ▷ dois literais de cláusulas **distintas** são **compatíveis** se eles podem assumir 1 simultaneamente.
- ▷ Seja $E = C_1 \wedge \cdots \wedge C_p$ uma expressão booleana com cláusulas C_i.
- ▷ Seja G o grafo que possui um vértice v_i para cada literal x_i de cada cláusula (se uma variável está em duas cláusulas diferentes, criamos dois vértices distintos) e tal que $v_i v_j$ se x_i e x_j são compatíveis.

Teorema

$\mathtt{Clique}$ é NP-Completo.

Proof.

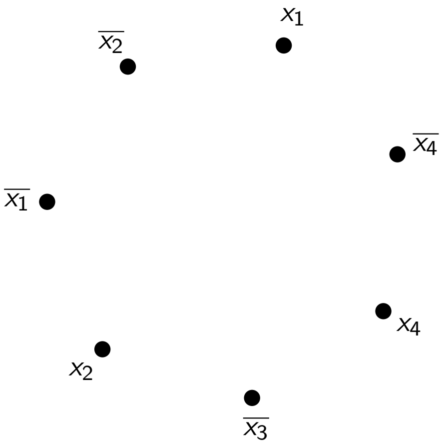
- ▷ Existe redução f de `SAT` para `Clique`.
- ▷ dois literais de cláusulas **distintas** são **compatíveis** se eles podem assumir 1 simultaneamente.
- ▷ Seja $E = C_1 \wedge \cdots \wedge C_p$ uma expressão booleana com cláusulas C_i.
- ▷ Seja G o grafo que possui um vértice v_i para cada literal x_i de cada cláusula (se uma variável está em duas cláusulas diferentes, criamos dois vértices distintos) e tal que $v_i v_j$ se x_i e x_j são compatíveis.
- ▷ G possui uma clique de tamanho p se e somente se E é satisfatível □

Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$

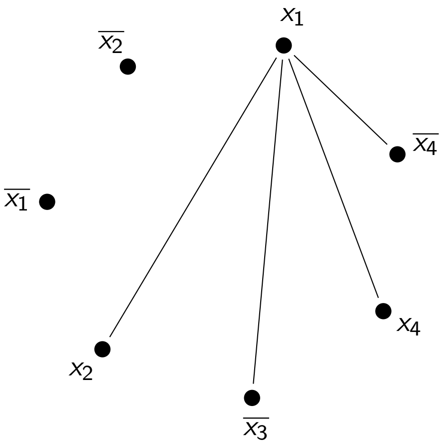
Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$



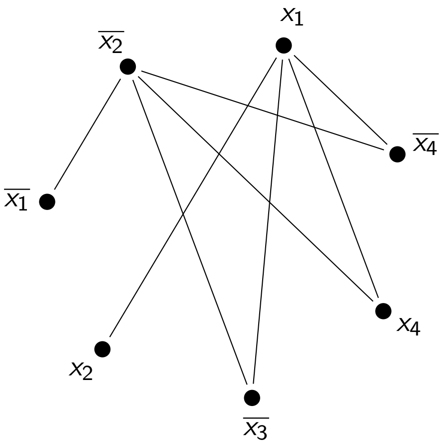
Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$



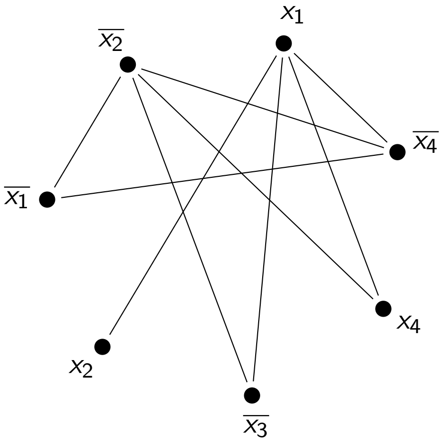
Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$



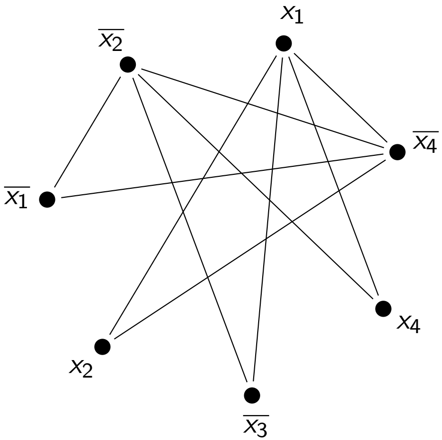
Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$



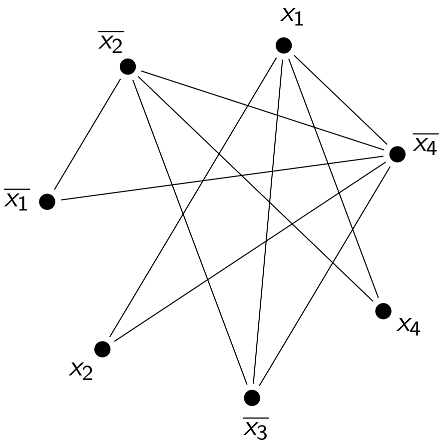
Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$



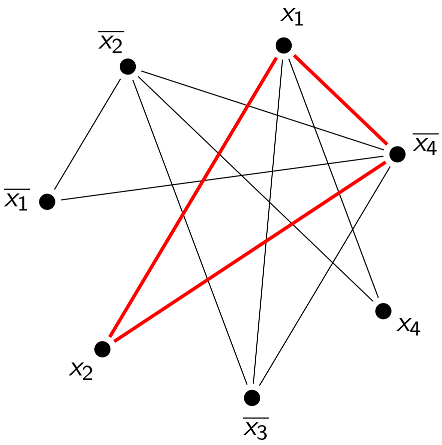
Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$



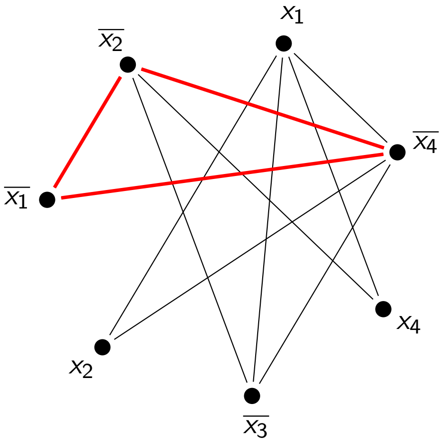
Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$



Exemplo

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_4})$$

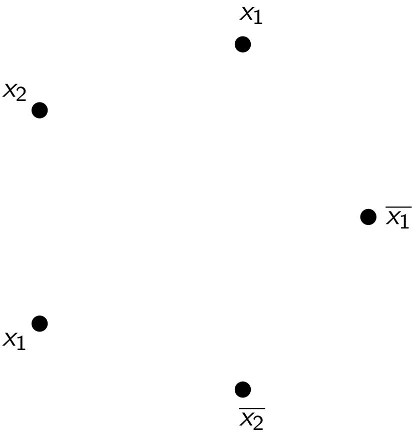


Exemplo

$$(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1})$$

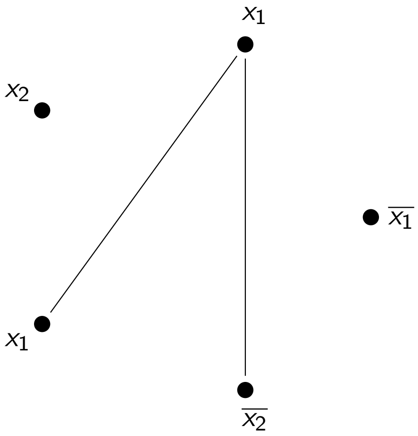
Exemplo

$$(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1})$$



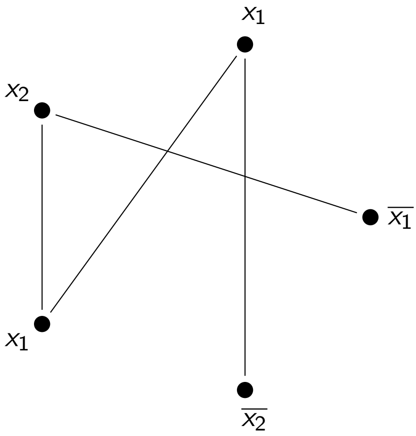
Exemplo

$$(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1})$$



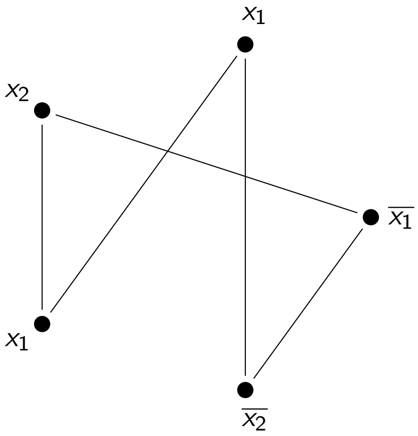
Exemplo

$$(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1})$$



Exemplo

$$(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1})$$



Teorema

- *Conjunto Independente* é NP-Completo.
- *Cobertura por Vértices* é NP-Completo.
- *Circuito Hamiltoniano* é NP-Completo.
- *Coloração de Vértices* é NP-Completo.
- *Coloração de Arestas* é NP-Completo.
- *Cobertura por Conjuntos* é NP-Completo.
- *Clique Máxima* é NP-Difícil.

21 problemas NP-completos de Karp

REDUCIBILITY AMONG COMBINATORIAL PROBLEMS[†]

Richard M. Karp

University of California at Berkeley

Problemas NP-Completos

Fábio Botler

Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro