Problemas NP-Completos

Fábio Botler

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

Aula passada

- A classe NP
- A questão P=NP
- Complementos de Problemas

Aula de hoje

- 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.

A classe NP

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

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

Qual a relação entre as classes P e NP?

Qual a relação entre as classes P e NP?

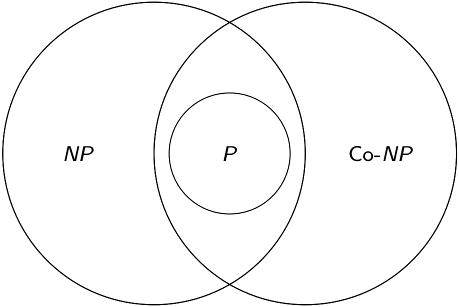
- $P \subseteq NP$

Qual a relação entre as classes P e NP?

- $\triangleright$ $P \subseteq NP$
- $\triangleright$ $NP \subseteq P$?

Qual a relação entre as classes P e NP?

- $\triangleright$ $P \subseteq NP$
- $\triangleright$ $NP \subseteq P$?



$P \times NP$

$P \times NP$

- Como resolver esse problema?

$P \times NP$

- $\triangleright$ Como resolver esse problema?
- $\triangleright$ 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*

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?

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

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

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?

- $\triangleright$ 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 $|V(G)| - k$.

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 $|V(G)| - k$.
- ▷ 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 $\propto$

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* $\Pi_1 \in P$*.*

A relação $\propto$

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

A relação $\propto$

- $\triangleright$ A redução f transforma instância genérica de Π_1 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 $\propto$

- $\triangleright$ A redução f transforma instância genérica de Π_1 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**".

A relação $\propto$

- $\triangleright$ A redução f transforma instância genérica de Π_1 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.

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:* $\Pi \in NPC$

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* $\Pi' \in NP$

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,

Definição

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

(i) $\Pi \in NP$

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

- ▷ *notação:* $\Pi \in NPC$

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

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

- ▷ 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* $\Pi' \in NP$

▷ *notação:* $\Pi \in NPC$

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

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

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

▷ 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.

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

Lema

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

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.

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!

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!

Teorema (Cook–Levin, 1971)

Satisfatibilidade ($\texttt{SAT}$) 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

`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.

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.

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

`Clique` é NP-Completo.

Proof.

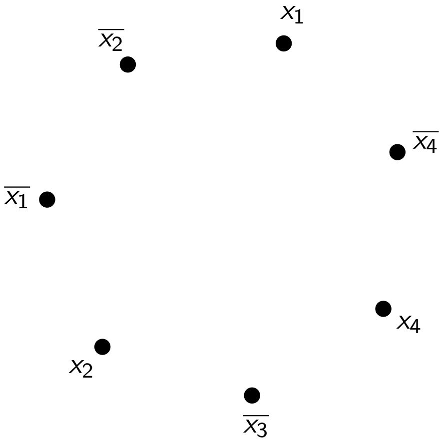
- ▷ Existe redução f de `SAT` para `Clique`.
- ▷ dois literais de cláusulas **distintas** são **compatíveis** se eles podem assumir 1 simultaneamente.
- ▷ Seja $E = C_1 \wedge \cdots \wedge C_p$ uma expressão booleana com cláusulas C_i.
- ▷ Seja G o grafo que possui um vértice v_i para cada literal x_i 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

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

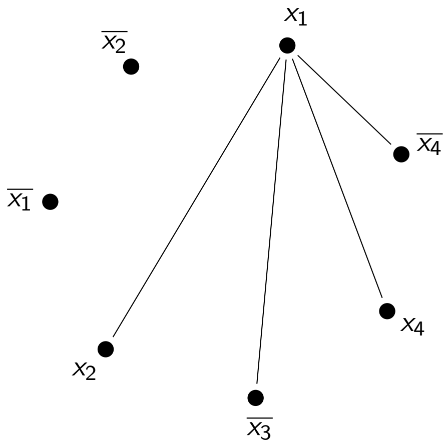
Exemplo

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



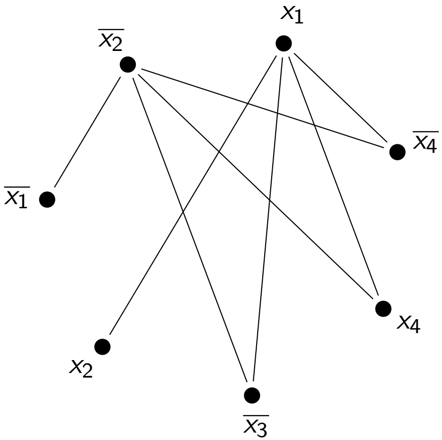
Exemplo

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



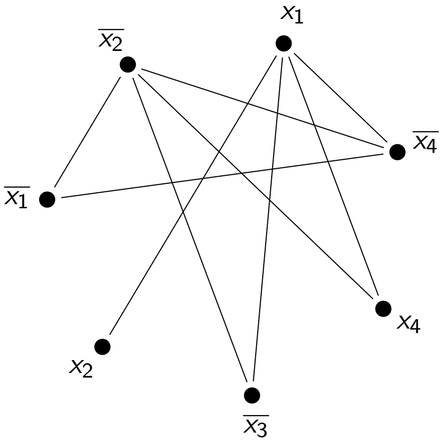
Exemplo

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



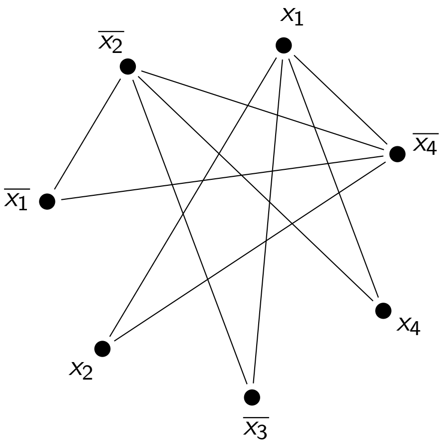
Exemplo

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



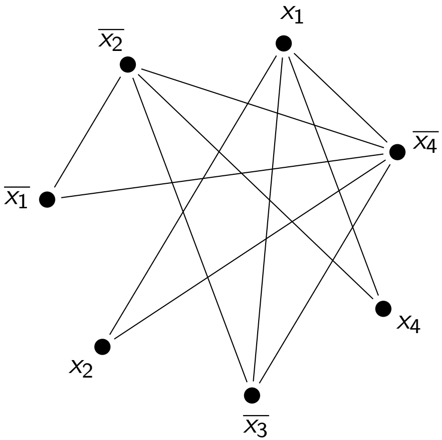
Exemplo

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



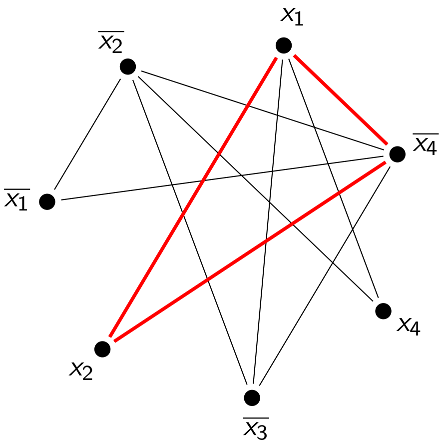
Exemplo

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



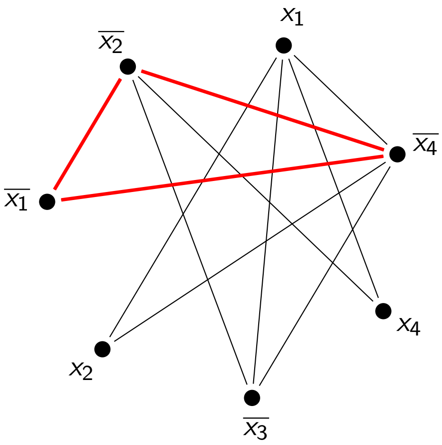
Exemplo

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



Exemplo

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

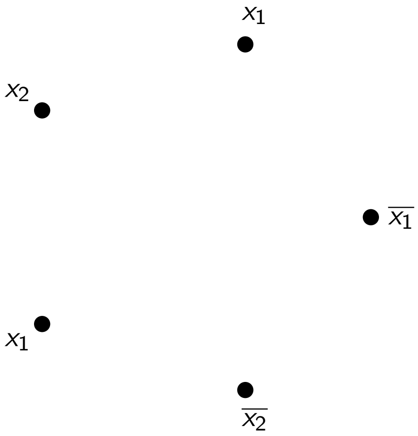


Exemplo

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

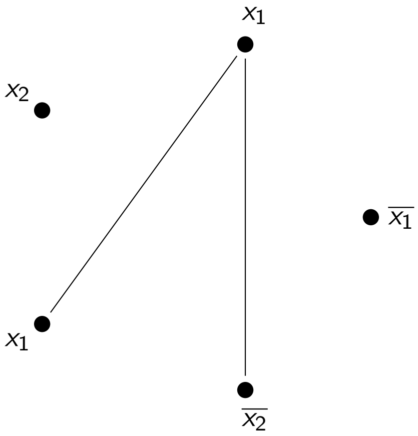
Exemplo

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



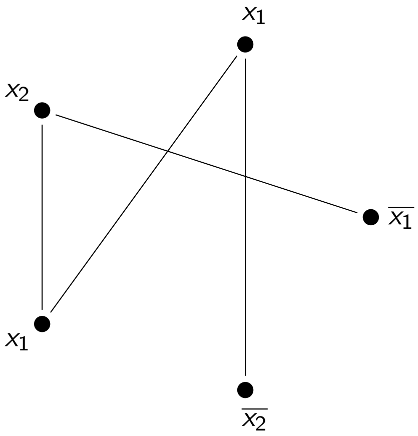
Exemplo

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



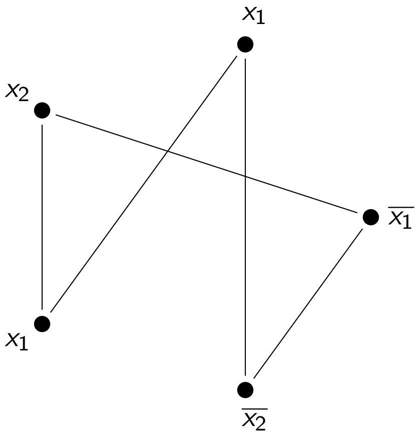
Exemplo

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



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