Aula 21

Redutibilidade NP

- O problema X é redutível em tempo polinomial (polynomial-time reducible) ao problema Y sse existe uma função f calculável em tempo polinomial tal que:

$$x \in X \iff f(x) \in Y$$

Escrevemos: $X \leq_p Y$

Proposição 4: $Y \in P \wedge X \leq_p Y \Rightarrow X \in P$

Prova:

- Suponhamos que $Y \in P$ e $X \leq_p Y$, há que provar que $X \in P$.
- Seja A o algoritmo q decide Y e f a função q reduz X a Y.

- Temos q construir um algoritmo A' q decide X em tempo polinomial

$A'(x)$:

retorna $A(f(x))$

Completude NP

- Um problema X diz-se NP-difícil sse:
  $\forall Y \in NP.\ Y \leq_p X$

- Um problema X diz-se NP-completo sse:
  - $X \in NP$
  - X é NP-difícil

Proposição 5: Se um problema NP-completo for resolúvel em tempo polinomial então $P = NP$.

Prova:

- Assumindo q existe $X \in P$ tal q X é NP-difícil, temos de provar q $\forall y \in NP.\ y \in P$.
- Tomemos um qualquer $Y \in NP$; como X é NP-difícil, concluímos q existe h, calculável em tempo polinomial, tal q:
  $$y \in Y \iff h(y) \in X$$
- Seja A o algoritmo polinomial q decide X, definimos o algoritmo A' que decide Y em tempo polinomial como se segue:

```
A'(y):
    return A(f(y))
```

Completude NP

- Um problema X diz-se **NP-difícil** sse: $\forall Y \in NP.\ Y \leq_p X$

- Um problema X diz-se **NP-completo** sse:
  - $X \in NP$
  - X é NP-difícil

Proposição 7: $X \in NP \land Y \leq_p X \land Y \in NPC \Rightarrow X \in NPC$

Prova: Há que provar que $\forall Z \in NP.\ Z \leq_p X$

- Tomemos $Z \in NP$.
- Como $Y \in NPC$, temos que $Z \leq_p Y$.
- De $Y \leq_p X$ e $Z \leq_p Y$ segue que $Z \leq_p X$ (pela transitividade de $\leq_p$)

$\blacksquare$

Completude NP

· Um problema X diz-se NP-difícil sse:
$\forall Y \in NP.\ Y \leq_p X$

· Um problema X diz-se NP-completo sse:
· $X \in NP$
· X é NP-difícil

Proposição 7: $X \in NP \wedge Y \leq_p X \wedge Y \in NPC \Rightarrow X \in NPC$

* Como provar que um problema X é NP-completo?

① Provamos que X está em NP (fácil)

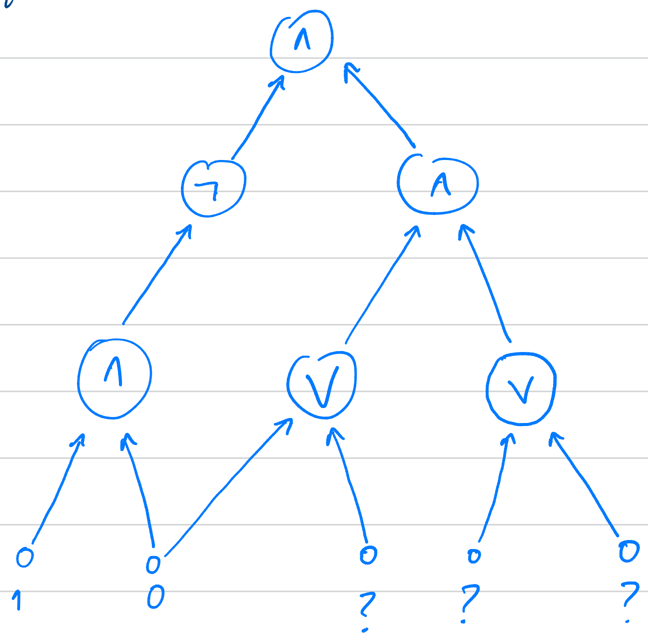
↳ Descobrir o certificado q̄ nos permite verificar X em tempo polinomial

② Seleccionar um problema NP-completo Y e construir uma redução $Y \leq_p X$. (difícil)

O 1º Problema NP-Completo

Circuit-SAT: Dado um circuito combinatório construído com portas And, Or e Not, existem inputs q̄ tornam o output do circuito 1.

Exemplo:



Sim ⇒ 101

O 1º Problema NP-Completo

Circuit-SAT: Dado um circuito combinatório construído com portas And, Or e Not, existem inputs q tornam o output do circuito $\underline{1}$.

Teorema [Cook-Levin] Circuit-SAT é NP-completo.

Esboço de Prova:

- Tome-se $X \in NP$. Da definição de NP segue que existe um algoritmo de verificação A tal que:

$$x \in X \iff \exists y.\ A(x,y)=1$$

- Um algoritmo polinomial pode ser implementado por um circuito combinatório de tamanho polinomial. Seja R esse circuito.

- Fixamos as 1ºs $|x|$ entradas de R com os bits de x. As restantes $|y|$ entradas ficam com ?.

- O circuito R é satisfazível sse $\exists y.\ A(x,y)=1$ (sse $x \in X$).

Road Map

CNF-SAT

3 CNF-SAT

Clique

Vertex-Cover

Set Cover

Shared Stickers

$\boxed{\text{CNF-SAT} \leq_p \text{3-CNF-SAT}}$

Ideia Chave: $C_i \longrightarrow C'_{i_1}, \dots, C'_{i_k}$

- 4 casos a considerar; o número de literais de C_i é:
  - $= 1 \rightarrow$?
  - $= 2 \rightarrow$?
  - $= 3 \rightarrow$ ✓
  - $> 3 \rightarrow$?

Exemplo: $\neg x_1 \vee x_2$

- Uma nova variável y, 2 cláusulas de output

$(\neg x_1 \vee x_2 \vee y)$

$(\neg x_1 \vee x_2 \vee \neg y)$

- Qualquer valoração q não satisfaz $\neg x_1 \vee x_2$ falha uma das cláusulas

$\boxed{\text{CNF-SAT} \leq_p \text{3-CNF-SAT}}$

Ideia Chave: $C_i \longrightarrow C'_{i_1}, \dots, C'_{i_k}$

- 4 casos a considerar; o número de literais de C_i é:
  - $= 1 \rightarrow$?
  - $= 2 \rightarrow$ 1 nova variável + 2 novas cláusulas
  - $= 3 \rightarrow$ ✓
  - $> 3 \rightarrow$?

Exemplo: $\underline{\neg x}$

- Duas novas variáveis y_1 e y_2, 4 cláusulas de output

$(y_1 \vee y_2 \vee \neg x)$
$(y_1 \vee \neg y_2 \vee \neg x)$
$(\neg y_1 \vee y_2 \vee \neg x)$
$(\neg y_1 \vee \neg y_2 \vee \neg x)$

Qualquer valoração q não satisfaça $\neg x$ falha uma das 4 cláusulas.

$\boxed{\text{CNF-SAT} \leq_p \text{3-CNF-SAT}}$

Ideia Chave: $C_i \longrightarrow C'_{i_1}, \dots, C'_{i_k}$

- 4 casos a considerar; o número de literais de C_i é:
  - $= 1 \rightarrow$ 2 novas variáveis, 4 cláusulas de output
  - $= 2 \rightarrow$ 1 nova variável + 2 novas cláusulas
  - $= 3 \rightarrow$ ✓
  - $> 3 \rightarrow$?

- Seja $C_i = \ell$
  - Duas novas variáveis y_1 e y_2, 4 cláusulas de output

$$(y_1 \vee y_2 \vee \ell)$$
$$(y_1 \vee \neg y_2 \vee \ell)$$
$$(\neg y_1 \vee y_2 \vee \ell)$$
$$(\neg y_1 \vee \neg y_2 \vee \ell)$$

$\boxed{\text{CNF-SAT} \leq_p \text{3-CNF-SAT}}$

Ideia Chave : $C_i \longrightarrow C'_{i_1}, \ldots, C'_{i_k}$

- 4 casos a considerar; o número de literais de C_i é:
  - $= 1 \rightarrow$ 2 novas variáveis + 4 novas cláusulas
  - $= 2 \rightarrow$ 1 nova variável + 2 novas cláusulas
  - $= 3 \rightarrow$ ✓
  - $> 3 \rightarrow$?

- Exemplo: $(x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4)$

$$\underbrace{x_1 \vee \neg x_2 \qquad\qquad x_3 \vee \neg x_4}_{\text{só precisamos de satisfazer uma das cláusulas}}$$

$$x_1 \vee \neg x_2 \vee y \;\wedge\; \neg y \vee x_3 \vee \neg x_4$$

↳ O y escolhe a cláusula original q̄ vou satisfazer

$\boxed{\text{CNF-SAT} \leq_p \text{3-CNF-SAT}}$

Ideia Chave: $C_i \longrightarrow C'_{i_1}, \dots, C'_{i_k}$

- 4 casos a considerar; o número de literais de C_i é:
  - $= 1 \rightarrow$ 2 novas variáveis + 4 novas cláusulas
  - $= 2 \rightarrow$ 1 nova variável + 2 novas cláusulas
  - $= 3 \rightarrow$ ✓
  - $> 3 \rightarrow$?

- Exemplo: $(\underline{x_1 \vee \neg x_2} \vee \underline{x_3} \vee \underline{\neg x_4 \vee x_5})$

$$x_1 \vee \neg x_2 \vee y_1 \wedge \neg y_1 \vee x_3 \vee y_2 \wedge \neg y_2 \vee \neg x_4 \vee x_5$$

CNF-SAT $\leq_p$ 3-CNF-SAT

Ideia Chave : $C_i \longrightarrow C'_{i_1}, \ldots, C'_{i_k}$

- 4 casos a considerar, o número de literais de C_i é:
  - $= 1 \longrightarrow$ 2 novas variáveis + 4 novas cláusulas
  - $= 2 \longrightarrow$ 1 nova variável + 2 novas cláusulas
  - $= 3 \longrightarrow$ ✓
  - $> 3 \longrightarrow$?

- Seja $C = l_1 \vee \ldots \vee l_k \quad , \; k \geq 4$

$$
\begin{aligned}
C'_1 &= l_1 \vee l_2 \vee y_1 \\
C'_2 &= \neg y_1 \vee l_3 \vee y_2 \\
&\vdots \\
C'_j &= \neg y_{j-1} \vee l_{j+1} \vee y_j \\
&\vdots \\
C'_{k-3} &= \neg y_{k-4} \vee l_{k-2} \vee y_{k-3} \\
C'_{k-2} &= \neg y_{k-3} \vee l_{k-1} \vee l_{k-2}
\end{aligned}
$$

Proposição: C é satisfazível sse $\bigwedge_{i=1}^{k-2} C'_i$ é satisfazível

$\Updownarrow$

Proposição: ρ satisfaz C sse $\exists \rho' . \; \rho'$ satisfaz $\bigwedge_{i=1}^{k-2} C'_i$

e ρ' estende ρ

CNF-SAT $\leq_p$ 3-CNF-SAT

Proposição: ρ satisfaz C sse existe uma extensão de ρ, ρ', que satisfaz $\bigwedge_{i=1}^{k-2} C_i'$

- $C = \ell_1 \vee \dots \vee \ell_k$, $k \geq 4$

$C_1' = \ell_1 \vee \ell_2 \vee y_1$
$C_2' = \neg y_1 \vee \ell_3 \vee y_2$
$\vdots$
$C_j' = \neg y_{j-1} \vee \ell_{j+1} \vee y_j$
$\vdots$
$C_{k-3}' = \neg y_{k-4} \vee \ell_{k-2} \vee y_{k-3}$
$C_{k-2}' = \neg y_{k-3} \vee \ell_{k-1} \vee \ell_{k-2}$

$\boxed{\Rightarrow}$ Existe $1 \leq i \leq k$ tal que $\rho(\ell_i) = 1$

- Seja $j = \begin{cases} 1 & \text{se } i = 1, 2 \\ k-2 & \text{se } i = k-1, k \\ i-1 & \text{c.c.} \end{cases}$

Concluímos q $\rho'(C_j') = 1$ para qualquer ρ' q estenda ρ

- Temos de "encontrar" a extensão de ρ, ρ', q satisfaz todas as outras cláusulas.

- $\rho'(y_\ell) = \begin{cases} 1 & \text{se } \ell \leq i-2 \\ 0 & \text{se } \ell \geq i-1 \end{cases}$ $\quad \underbrace{\rho'(x_\ell) = \rho(x_\ell) \ \forall x_\ell \in dom(\rho)}_{\rho' \text{ é uma extensão de } \rho}$

$\boxed{\text{CNF-SAT} \leq_p \text{3-CNF-SAT}}$

Proposição: ρ satisfaz C sse ρ satisfaz $\bigwedge_{i=1}^{k-2} C_i'$

- $C = l_1 \vee \dots \vee l_k \quad , \quad k \geq 4$

$C_1' = l_1 \vee l_2 \vee y_1$

$C_2' = \neg y_1 \vee l_3 \vee y_2$

$\vdots$

$C_j' = \neg y_{j-1} \vee l_{j+1} \vee y_j$

$\vdots$

$C_{k-3}' = \neg y_{k-4} \vee l_{k-2} \vee y_{k-3}$

$C_{k-2}' = \neg y_{k-3} \vee l_{k-1} \vee l_{k-2}$

$\boxed{\Leftarrow}$ Suponhamos por contradição que ρ' satisfaz $\bigwedge_{i=1}^{k-2} C_i'$ e ρ não satisfaz C.

- $\forall_{1 \leq i \leq k}.\ \rho'(l_i) = 0$

- Isto significa que $\underbrace{\rho'(y_1)=1, \ \dots \ , \rho'(y_{k-3})=1}_{\text{para satisfazermos as cláusulas } C_1', \dots, C_{k-3}'}$

$\downarrow$

- A cláusula C_{k-2}' não é satisfeita por ρ'.

$\boxed{3\text{-}CNF\text{-}SAT \leq_p Clique}$

Clique: Um clique num grafo não dirigido $G = (V, E)$ é subconjunto $V' \subseteq V$ tal que todos os pares de vértices em V' estão ligados por um arco em E.

V' é um clique de G sse $\forall u, v \in V'.\ (u, v) \in E$

Problema Clique:

$$Clique = \{ \langle G, k \rangle \mid G \text{ é um grafo não dirigido que contém um clique de tamanho } k \}$$

Proposição: O problema Clique é NP-completo.

3-CNF-SAT $\leq_p$ Clique

Proposição: O problema Clique é NP-completo.

- Seja ϕ uma formula proposicional no formato 3-CNF, temos de calcular um grafo $G=(V,E)$ e um inteiro k tal que ϕ é satisfazível sse G tem um Clique de tamanho k.

- **Exemplo:**

$$\phi = \underbrace{(x_1 \vee \neg x_2 \vee \neg x_3)}_{C_1} \wedge \underbrace{(\neg x_1 \vee x_2 \vee x_3)}_{C_2} \wedge \underbrace{(x_1 \vee x_2 \vee x_3)}_{C_3}$$

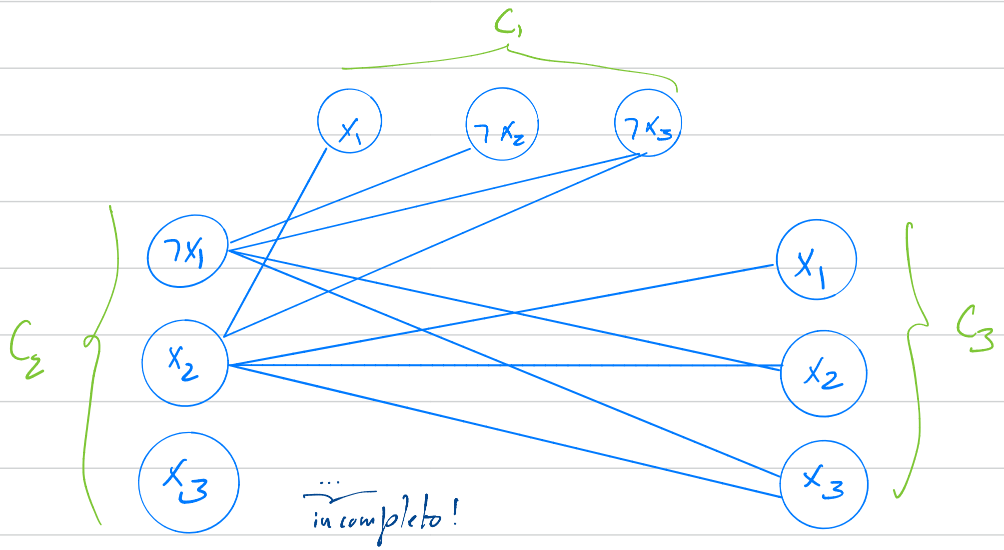
3-CNF-SAT $\leq_p$ Clique

Proposição: O problema Clique é NP-completo.

- Seja ϕ uma fórmula proposicional no formato 3-CNF, temos de calcular um grafo $G=(V,E)$ e um inteiro k tal que ϕ é satisfazível sse G tem um Clique de tamanho k.

- Exemplo:

$$\phi = \underbrace{(x_1 \vee \neg x_2 \vee \neg x_3)}_{C_1} \wedge \underbrace{(\neg x_1 \vee x_2 \vee x_3)}_{C_2} \wedge \underbrace{(x_1 \vee x_2 \vee x_3)}_{C_3}$$



$\boxed{\text{3-CNF-SAT} \leq_p \text{Clique}}$

Proposição: O problema Clique é NP-completo.

- Seja ϕ uma formula proposicional no formato 3-CNF, temos de calcular um grafo $G = (V, E)$ e um inteiro k tal que ϕ é satisfazível sse G tem um Clique de tamanho k.

- Seja $\phi = C_1 \vee \dots \vee C_k$, construímos o seguinte grafo:

$$G_\phi = (V_\phi, E_\phi)$$

$$V_\phi = \{ (l, r) \mid \exists l', l''.\ C_r \equiv (l' \vee l \vee l'') \}$$

$$E_\phi = \{ ((l, r), (l', s)) \mid r \neq s \wedge l' \neq \neg l \wedge l \neq \neg l' \}$$

- Proposição: $\phi = C_1 \vee \dots \vee C_k$ sse $G_\phi = (V_\phi, E_\phi)$ contém um clique de tamanho k.

$\boxed{3\text{-}CNF\text{-}SAT \leq_p Clique}$

Proposição: O problema Clique é NP-completo.

- Seja ϕ uma fórmula proposicional no formato 3-CNF, temos de calcular um grafo $G = (V, E)$ e um inteiro k tal que ϕ é satisfazível sse G tem um Clique de tamanho k.

  - Seja $\phi = C_1 \vee \dots \vee C_k$, construímos o seguinte grafo:

$$G_\phi = (V_\phi, E_\phi)$$

$$V_\phi = \{ (l, r) \mid \exists l', l''.\ C_r \equiv (l' \vee l \vee l'') \}$$

$$E_\phi = \{ ((l, r), (l', s)) \mid r \neq s \wedge l' \neq \neg l \wedge l \neq \neg l' \}$$

- $|V_\phi| = 3 \times k$
- $|E_\phi| = (3 \times k)^2 = 9k^2$

A redução é calculada em tempo polinomial

$\boxed{3\text{-}CNF\text{-}SAT \le_p Clique}$

- Proposição: $\phi = C_1 \wedge \cdots \wedge C_k$ satisfazível sse $G_\phi = (V_\phi, E_\phi)$ contém um clique de tamanho k.

- Seja $\phi = C_1 \wedge \ldots \wedge C_k$, construímos o seguinte grafo:

$$G_\phi = (V_\phi, E_\phi)$$

$$V_\phi = \{ (l, r) \mid \exists l', l''.\ C_r \equiv (l' \vee l \vee l'') \}$$

$$E_\phi = \{ ((l,r),(l',s)) \mid r \neq s \wedge l' \neq \neg l \wedge l \neq \neg l' \}$$

$\boxed{\Rightarrow}$ Em cada cláusula C_i, com $1 \le i \le k$, tem de existir um literal l_i tal que $\rho(l_i) = 1$.

O conjunto

$$\hat{V} = \{ (l_i, i) \mid 1 \le i \le k \}$$

é um clique em G_ϕ.

- Observamos q se $(l_i, i), (l_j, j) \in \hat{V}$ então $((l_i, i), (l_j, j)) \in E_\phi$
  - $i \neq j$
  - $l_i \neq \neg l_j$ e $l_j \neq \neg l_i$

$\boxed{\text{3-CNF-SAT} \leq_p \text{Clique}}$

- Proposição: $\phi = C_1 \wedge \cdots \wedge C_k$ satisfazível sse $G_\phi = (V_\phi, E_\phi)$ contém um clique de tamanho k.

- Seja $\phi = C_1 \wedge \ldots \wedge C_k$, construímos o seguinte grafo:

$$G_\phi = (V_\phi, E_\phi)$$

$$V_\phi = \{ (\ell, r) \mid \exists \ell', \ell''. \ C_r \equiv (\ell' \vee \ell \vee \ell'') \}$$

$$E_\phi = \{ ((\ell, r), (\ell', s)) \mid r \neq s \wedge \ell' \neq \neg \ell \wedge \ell \neq \neg \ell' \}$$

$\boxed{\Leftarrow}$ Seja $\hat{V}$ um clique em G_ϕ de tamanho k.

Escolhemos ρ tal que:

$\forall (l_i, i) \in \hat{V}. \ \rho(l_i) = 1$

$\downarrow$

Por construção, ρ satisfaz ϕ.

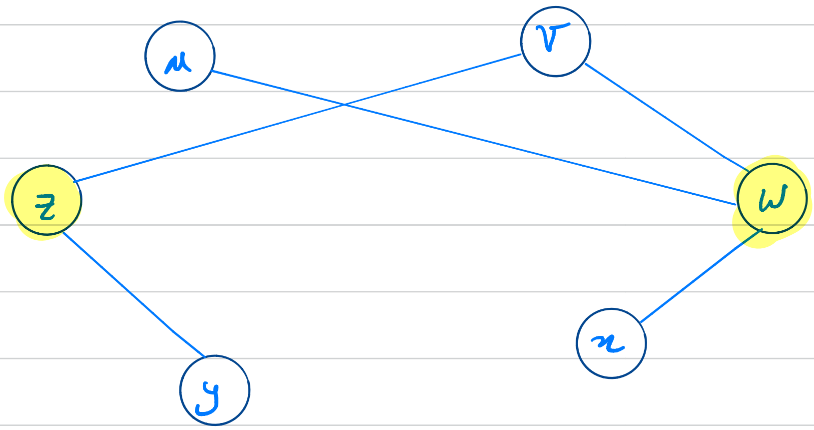
Clique $\leq_p$ Vertex-Cover

Vertex-Cover: V' é uma cobertura de vértices do grafo não dirigido $G=(V,E)$ sse $\forall (u,v) \in E.\ u \in V'$ ou $v \in V'$

Vertex-Cover Problem: Encontrar a cobertura de vértices de cardinalidade mínima

$\Downarrow$ Problema de Decisão

VertexCover $= \{ \langle G, k \rangle \mid G$ é um grafo não dirigido com uma cobertura de vértices de tamanho $k \}$



$\boxed{\text{Clique} \leq_p \text{Vertex-Cover}}$

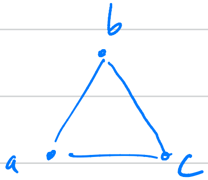
Proposição: $G=(V,E)$ tem um clique de tamanho k <u>sse</u> $\bar{G}=(V,\bar{E})$ tem uma cobertura de vértices de tamanho $|V|-k$.

Onde:

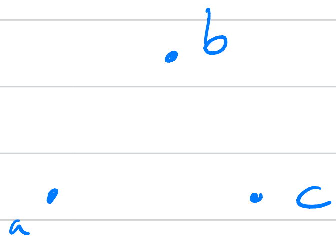
$$\bar{G}=(V,\bar{E}) \qquad \bar{E}=\{(u,v)\in V^2 \mid (u,v)\notin E \wedge u\neq v\}$$

Exemplo:

G:



$\bar{G}$:



Clique $=\{a,b,c\}$

$VC=\emptyset$

Clique $\leq_p$ Vertex-Cover

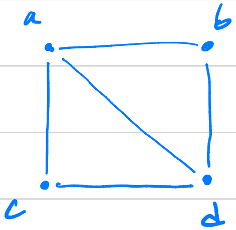
Proposição: $G=(V,E)$ tem um clique de tamanho k <u>sse</u> $\bar{G}=(V,\bar{E})$ tem uma cobertura de vértices de tamanho $|V|-k$.

Onde:

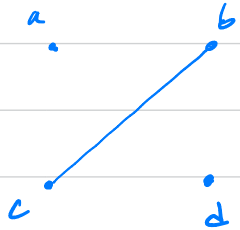
$\bar{G}=(V,\bar{E})$ $\quad \bar{E}=\{(u,v)\in V^2 \mid (u,v)\notin E \wedge u\neq v\}$

Exemplo 2:

G:



$\bar{G}$:



Clique: $\{a,b,d\}$ $\quad$ $VC=\{c\}$

Clique $\leq_p$ Vertex-Cover

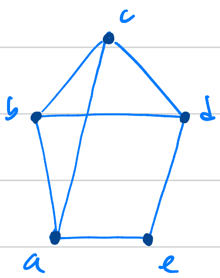
Proposição: $G=(V,E)$ tem um clique de tamanho k **sse** $\bar{G}=(V,\bar{E})$ tem uma cobertura de vértices de tamanho $|V|-k$.

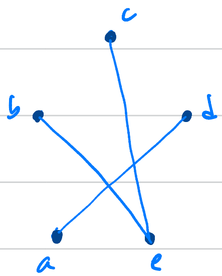
Onde:

$$\bar{G}=(V,\bar{E}) \qquad \bar{E}=\{(u,v)\in V^2 \mid (u,v)\notin E \wedge u\neq v\}$$

Exemplo 3

G:





Clique: $\{a,b,d\}$

$VC=\{a,e\}$

$\boxed{\text{Clique} \leq_P \text{Vertex-Cover}}$

Proposição: $G = (V, E)$ tem um Clique de tamanho k <u>sse</u> $\overline{G} = (V, \overline{E})$ tem uma cobertura de vértices de tamanho $|V| - k$.

Onde:

$$\overline{G} = (V, \overline{E}) \qquad \overline{E} = \{ (u,v) \in V^2 \mid (u,v) \notin E \wedge u \neq v \}$$

Basta provar que:

$\hat{V}$ é um Clique em G sse $V \setminus \hat{V}$ é um VC em $\overline{G}$

$\hat{V}$ é um Clique em G

$$\Leftrightarrow \forall u, v.\ u \in \hat{V} \wedge v \in \hat{V} \Rightarrow (u,v) \in E$$

$$\Leftrightarrow \forall u, v.\ (u,v) \notin E \Rightarrow u \notin \hat{V} \vee v \notin \hat{V}$$

$$\Leftrightarrow \forall u, v.\ (u,v) \in \overline{E} \Rightarrow u \in V \setminus \hat{V} \vee v \in V \setminus \hat{V}$$

$$\Leftrightarrow V \setminus \hat{V} \text{ é uma VC em } \overline{G}$$

Vertex-Cover $\leq_p$ Subset-Cover

- Subset Cover:
  - F é uma família de conjuntos de elementos de um dado conjunto base Ω

    $F = \{X_1, \dots, X_n\}$ $\quad \forall_{1 \leq i \leq n}.\ X_i \subseteq \Omega$

  - $Y \subseteq \Omega$

  - $\text{SubsetCover} = \{ \langle F, Y, k \rangle \mid \exists\, X_1, \dots, X_k \in F.\ X_1 \cup X_2 \cup \dots \cup X_k = Y \}$

Proposição: Subset Cover está em NP

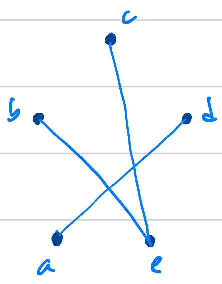
Proposição: Subset Cover é NP-difícil

Vertex-Cover $\leq_p$ Subset-Cover

- Subset Cover:

  - $\text{SubsetCover} = \{ \langle F, Y, k \rangle \mid \exists X_1, \dots, X_k \in F.\ X_1 \cup X_2 \cup \dots \cup X_k = Y \}$

Proposição: Subset Cover é NP-difícil



$X_a = \{(a,d)\}$ $\quad$ $X_d = \{(a,d)\}$

$X_b = \{(b,e)\}$ $\quad$ $X_e = \{(b,e), (c,e)\}$

$X_c = \{(c,e)\}$

$VC = \{a, e\}$

$Y = \{(b,e), (c,e), (a,d)\}$

Vertex-Cover $\leq_p$ Subset-Cover

- Subset Cover:

  - $\text{SubsetCover} = \{ \langle F, Y, k \rangle \mid \exists X_1, \ldots, X_k \in F.\ X_1 \cup X_2 \cup \cdots \cup X_k = Y \}$

Proposição: Subset Cover é NP-difícil

$$\langle G, k \rangle \in VC \iff \langle F_G, E, k \rangle \in \text{SubsetCover}$$

onde
- $G = (V, E)$
- $F_G = \{ E_v \mid v \in V \}$
- $E_v = \{ (u, v) \mid (u, v) \in E \} \cup \{ (v, u) \mid (v, u) \in E \}$

- O resultado segue directamente, observando que:

$$\hat{V} \text{ é uma VC de } G \text{ sse } \bigcup_{u \in \hat{V}} E_u = E$$

Exemplo: Cromos Partilhados

II.d) Seja $C = \{c_1, ..., c_n\}$ uma colecção de cromos e $\mathcal{A} = \{a_1, ..., a_m\}$ um grupo de amigos que coleccionam cromos. Cada membro do grupo detém um subconjunto de C; seja C_i o conjunto de cromos detido por a_i e $\mathcal{C} = \{C_i \mid 1 \leq i \leq m\}$ o conjunto dos conjuntos de cromos de todos os membros do grupo. Os membros do grupo pretendem determinar o mais pequeno conjunto de cromos que contém pelo menos um cromo detido por cada membro do grupo. Formalmente, este problema pode ser modelado através do seguinte problema de decisão:

$$\textbf{SharedStickers} = \{\langle C, \mathcal{C}, k\rangle \mid \exists X \subseteq C. |X| = k \wedge \forall_{1\leq i\leq m}. C_i \cap X \neq \emptyset\}$$

- Mostre q o problema SharedStickers é NP-difícil por redução a partir do problema VCover

Exemplo: Cromos Partilhados

II.d) Seja $C = \{c_1, ..., c_n\}$ uma colecção de cromos e $\mathcal{A} = \{a_1, ..., a_m\}$ um grupo de amigos que coleccionam cromos. Cada membro do grupo detém um subconjunto de C; seja C_i o conjunto de cromos detido por a_i e $\mathcal{C} = \{C_i \mid 1 \leq i \leq m\}$ o conjunto dos conjuntos de cromos de todos os membros do grupo. Os membros do grupo pretendem determinar o mais pequeno conjunto de cromos que contém pelo menos um cromo detido por cada membro do grupo. Formalmente, este problema pode ser modelado através do seguinte problema de decisão:

$$\textbf{SharedStickers} = \{\langle C, \mathcal{C}, k\rangle \mid \exists X \subseteq C. |X| = k \wedge \forall_{1 \leq i \leq m}. C_i \cap X \neq \emptyset\}$$

- Mostre q o problema SharedStickers é NP-difícil por redução a partir do problema VCover

$\langle G, k\rangle \in VCover \Leftrightarrow \langle C, \mathcal{C}, k\rangle \in SharedStickers$

- $C = V$
- $\mathcal{C} = \{ \{u, v\} \mid (u, v) \in E \}$

Complexidade da redução: $O(\underline{V + E})$

Exemplo: Incompatibilidades

II.d) Uma matriz de incompatibilidades é uma matriz quadrada cujas células guardam valores decimais entre 0 e 1. Intuitivamente, dada uma matriz de incompatibilidades M, $n \times n$, a célula M_{ij} guarda a incompatibilidade entre os índices i e j; $M_{ij} = 0$ se i e j são completamente compatíveis e $M_{ij} = 1$ se i e j são completamente incompatíveis. Dado um sub-conjunto de índices $I \subseteq \{1, ..., n\}$, o nível de incompatibilidade do conjunto é dado por: $\sum_{i,j \in I} M_{ij}$. O problema das incompatibilidades define-se formalmente da seguinte maneira:

$$\textbf{Incompat} = \{\langle M, k, v\rangle \mid M \text{ contém um sub-conjunto de índices de tamanho } k \text{ e incompatibilidade igual ou inferior a } v\}$$

- Mostre q o problema Incompat é NP-difícil por redução a partir do Problema ISet.

Problema ISet:

$\langle G, k \rangle \in ISet$ sse G contém um conjunto de vértices independentes de tamanho k

Definição [Conjunto de Vértices Independentes]

X é um conjunto de vértices independentes de G sse $\forall u, v \in X.\ (u,v) \notin E$

Problema ISet

$$\boxed{ISet \leq_p InCompact}$$

- $\langle G, k \rangle \in ISet \iff \langle M_G, k, 0 \rangle \in InCompact$

- $G = (V, E) \qquad V = \{v_1, \dots, v_n\}$

$$M_{ij} = \begin{cases} 0 & \text{se } (v_i, v_j) \notin E \\ 1 & \text{c.c.} \end{cases}$$

- Complexidade da Redução: $O(V^2)$