Algoritmos

Pedro Hokama

Fontes

- [*clrs*] Algoritmos: Teoria e Prática (Terceira Edição) Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest e Clifford Stein.
- [*timr*] Algorithms Illuminated Series, Tim Roughgarden
- Desmistificando Algoritmos, Thomas H. Cormen.
- Algoritmos, Sanjoy Dasgupta, Christos Papadimitriou e Umesh Vazirani
- Stanford Algorithms
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V`
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt5EMI2s2WQBsLsZl7A5HEK6`
- Conjunto de Slides dos Professores Cid C. de Souza, Cândida N. da Silva, Orlando Lee, Pedro J. de Rezende
- Conjunto de Slides do Professores Cid C. de Souza para a disciplina MO420

Qualquer erro é de minha responsabilidade.

3-CNF-SAT

- Uma formula booleana é composta
  - de variáveis booleanas $x_1, x_2, \ldots, x_n$,
  - de conectivos $\neg$ (NOT), $\vee$ (OR), $\wedge$ (AND),
  - Parenteses.
- Uma variável ou sua negação são chamadas **literais**.
- Uma fórmula booleana está na forma normal 3-conjuntiva se
  - está dividido em clausuras
  - cada clausura tem 3 literais conectas por ORs
  - as clausuras estão conectadas por ANDs.

$$(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x1 \vee \neg x3 \vee \neg x4)$$

3-CNF-SAT

3-CNF-SAT

Dado uma fórmula booleana na forma normal 3-conjuntiva, decidir se existe alguma atribuição das variáveis que torna a formula verdadeira.

Teorema

O 3-CNF-SAT é NP-Completo

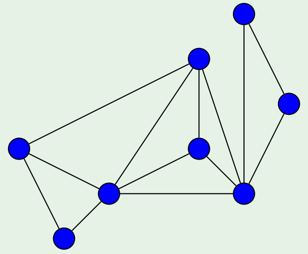
Prova: Veremos depois.

Problema do CLICK

Problema do CLICK

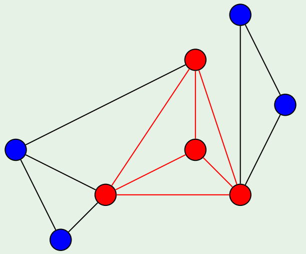
Dado um grafo não orientado $G = (V, E)$, e um inteiro k decidir se existe um subgrafo G' induzido de G que tenha k vértices e seja completo.

Exemplo



Existe uma click de tamanho 4?

Exemplo



Existe uma click de tamanho 4?

O Problema do CLICK

Será que CLICK é NP-Completo?

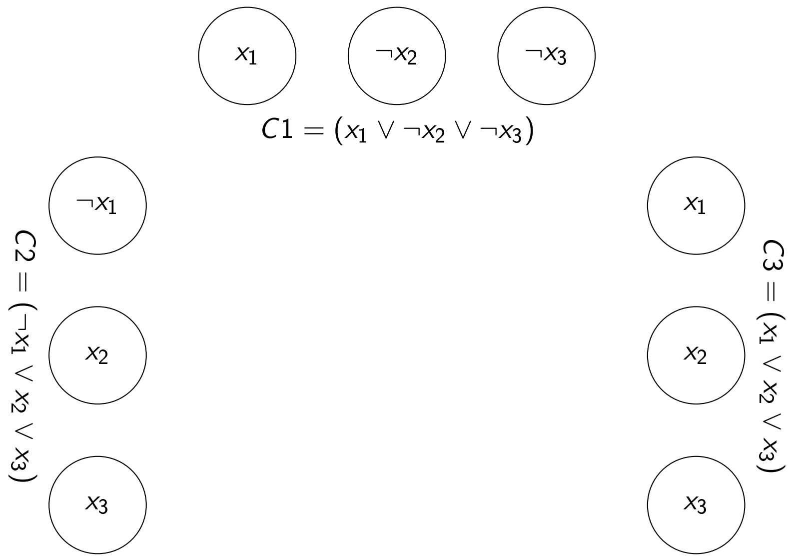
- CLICK $\in$ NP, basta apresentar o conjunto de vértices
- Vamos reduzir o 3-CNF-SAT para CLICK
  - Devemos converter qualquer instancia do 3-CNF-SAT para o problema do CLICK
  - Essa redução deve ter tempo polinomial
  - Aqui vamos mostrar o algoritmo da redução em um exemplo, mas é necessário garantir que funciona para qualquer instância.

3-CNF-SAT $\leq_p$ CLICK

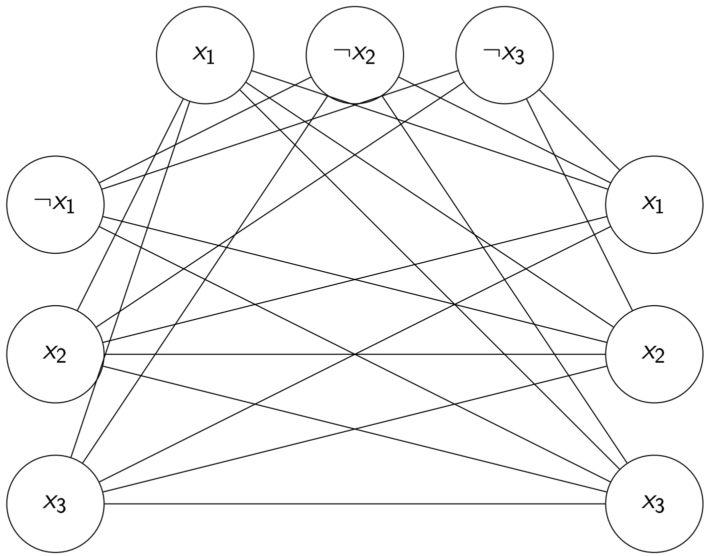
Considere uma formula booleana com k clausulas na forma normal 3-conjuntiva.

$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

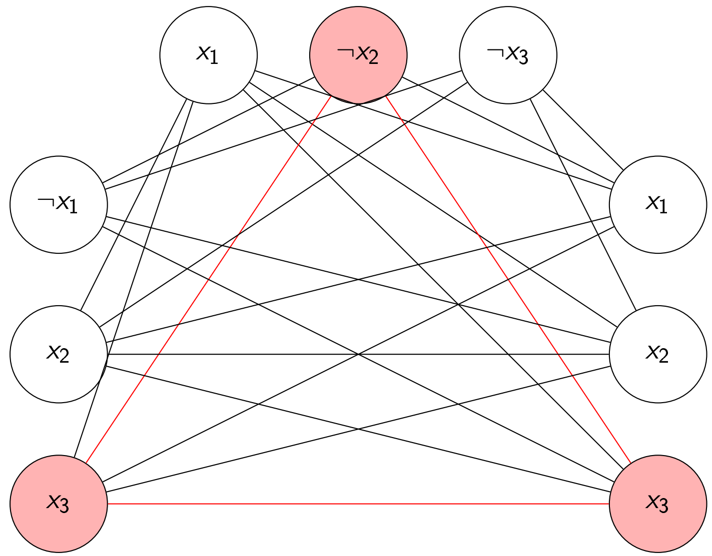
$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$



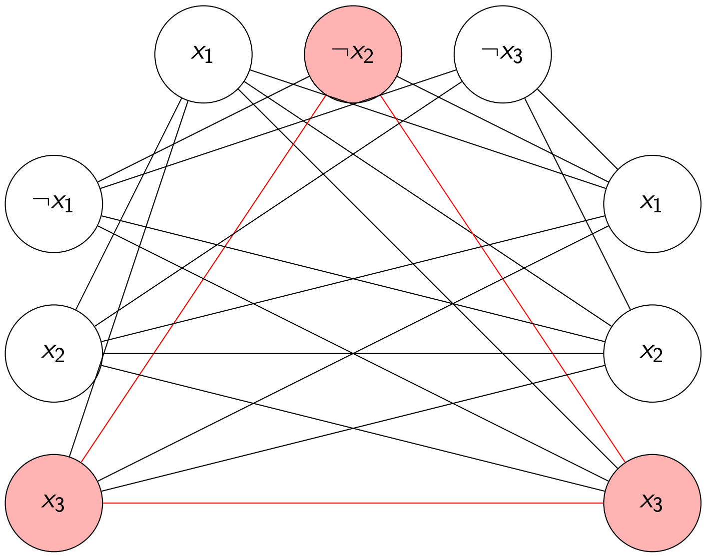
$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$



$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$



$(x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_3)$



A questão P vs NP

- Seria $P = NP$?
- Bastaria mostrar 1 algoritmo com tempo de execução polinomial para 1 problema NP-completo.
- Conjectura-se (fortemente) que $P \neq NP$.
- Mas também não foi provado.

Millennium Problems

Yang–Mills and Mass Gap

Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.

Riemann Hypothesis

The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.

P vs NP Problem

If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

Navier–Stokes Equation

This is the equation which governs the flow of fluids such as water and air. However, there is no proof for the most basic questions one can ask: do solutions exist, and are they unique? Why ask for a proof? Because a proof gives not only certitude, but also understanding.

Hodge Conjecture

The answer to this conjecture determines how much of the topology of the solution set of a system of algebraic equations can be defined in terms of further algebraic equations. The Hodge conjecture is known in certain special cases, e.g., when the solution set has dimension less than four. But in dimension four it is unknown.

Poincaré Conjecture

In 1904 the French mathematician Henri Poincaré asked if the three dimensional sphere is characterized as the unique simply connected three manifold. This question, the Poincaré conjecture, was a special case of Thurston's geometrization conjecture. Perelman's proof tells us that every three manifold is built from a set of standard pieces, each with one of eight well-understood geometries.

Birch and Swinnerton-Dyer Conjecture

Supported by much experimental evidence, this conjecture relates the number of points on an elliptic curve mod p to the rank of the group of rational points. Elliptic curves, defined by cubic equations in two variables, are fundamental mathematical objects that arise in many areas: Wiles' proof of the Fermat Conjecture, factorization of numbers into primes, and cryptography, to name three.

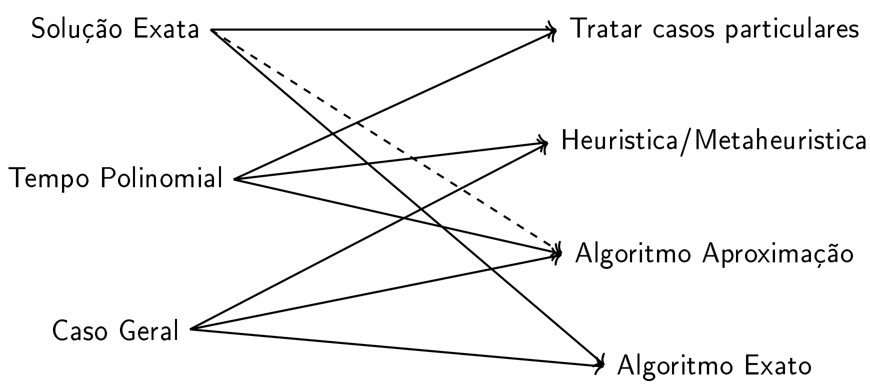
O nome NP

- O nome NP deriva de Non-deterministic Polynomial Time
- São problemas que podem ser resolvidos em tempo polinomial em uma máquina de Turing não determinística.
- Informalmente, considere uma máquina que conseguisse testar todas as soluções simultaneamente.

Problemas Intratáveis

- O que podemos fazer se um problema é NP-completo? (sem ser desistir)
- Na verdade muita coisa pode ser feita, e é disso que trataremos nessa disciplina.

Se $P \neq NP$ não conseguiremos para um problema NP-Difícil:



Casos Especiais

- Muitas vezes um problema geral pode ser NP-completo mas alguns casos especiais podem ser mais fáceis de resolver
  1. Encontrar um conjunto independente máximo é NP-completo. Mas no grafo caminho é fácil.
  2. O problema da Mochila binária é NP-completo, mas o da mochila fracionária é fácil.
  3. Na mochila binária, se a capacidade for polinomial no número de itens o problema também é fácil
  4. Satisfabilidade de fórmulas booleanas é NP-completo, mas o 2-CNF-SAT é fácil

Heurísticas

- Podemos abrir mão de encontrar a solução ótima, e tentar encontrar uma solução boa o bastante.
- Normalmente heurísticas são rápidas.
- Geralmente não possuem garantias do quão próximas estão da solução ótima

Algoritmos Aproximados

- Podemos abrir mão de encontrar a solução ótima, e tentar encontrar uma solução boa o bastante.
- Executam em tempo polinomial.
- Possuem garantias do quão próximas estão da solução ótima.
- Por exemplo: A solução de um algoritmo aproximado vai estar no máximo a um fator α da solução ótima.

Algoritmos Exatos

- Busca a solução ótima.
- Desiste do tempo polinomial.
- A ideia é reduzir ao máximo a complexidade, e aumentar ao máximo o tamanho das instâncias que conseguimos resolver.
  - Branch-and-Bound
  - Programação Linear Inteira
  - Programação por restrições

Satisfazibilidade de Fórmulas

- Uma fórmula booleana ϕ é composta de:
  - n variáveis booleanas: $x_1, x_2, \ldots, x_n$;
  - m conectivos booleanos: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
  - parênteses
- Ex. $\phi = ((x_1 \rightarrow x_2) \vee \neg((\neg x_1 \leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$

Satisfazibilidade de Fórmulas

Problema da Satisfazibilidade de Fórmulas - SAT

Dada uma fórmula booleana, decidir se existe uma atribuição das variáveis booleanas que faz com que ela seja avaliada como 1 (verdadeira).

- No exemplo ϕ é satisfazível com a atribuição $< x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1 >$
- Um algoritmo ingênuo que testa todas as possibilidades é inviável pois existem 2^n atribuições diferentes.

Teorema

SAT é NP-completo

Lema

SAT $\in$ NP

Lema

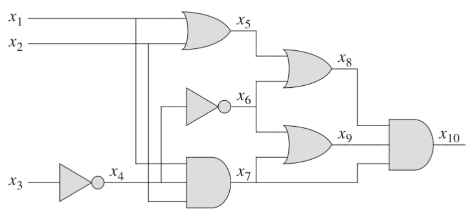
SAT $\in$ NP-Difícil

Lema

SAT $\in$ NP

- Para mostrar que SAT $\in$ NP, mostramos que um certificado consiste em uma atribuição satisfatória das variáveis. Esse certificado pode ser verificado substituindo cada variável pelo valor dessa atribuição e avaliando a expressão, que pode ser feito em tempo polinomial. Se o resultado for 1 o algoritmo verificou que a formula é satisfazível.

- Para mostrar que SAT $\in$ NP-Difícil mostraremos que CIRCUIT-SAT $\leq_p$ SAT.
- Considere um circuito C qualquer. Para cada fio x_i no circuito, a fórmula ϕ vai ter uma variável x_i, expressamos cada porta lógica como uma cláusula na formula booleana que representa o seu comportamento. No fim fazemos a conjunção das cláusulas.



$$
\begin{aligned}
\phi = x_{10} & \wedge (x_4 \leftrightarrow \neg x_3) \\
& \wedge (x_5 \leftrightarrow (x_1 \vee x_2)) \\
& \wedge (x_6 \leftrightarrow \neg x_4) \\
& \wedge (x_7 \leftrightarrow (x_1 \wedge x_2 \wedge x_4)) \\
& \wedge (x_8 \leftrightarrow (x_5 \vee x_6)) \\
& \wedge (x_9 \leftrightarrow (x_6 \vee x_7)) \\
& \wedge (x_{10} \leftrightarrow (x_7 \wedge x_8 \wedge x_9)) \,.
\end{aligned}
$$

Lema

C é satisfazível se e somente se ϕ é satisfazível.

- Esse algoritmo de redução executa em tempo polinomial.
- ($\rightarrow$) Se C tem uma atribuição que satisfaz, cada fio tem um valor bem definido e a saída do circuito é 1.
- Portanto se atribuirmos os valores de cada fio para as respectivas variáveis, a formula também terá valor 1.
- ($\leftarrow$) Se uma atribuição faz ϕ se verdadeira. Podemos atribuir o valor de cada fio com o valor das suas respectivas variáveis. E C terá saída igual a 1.
- Portanto mostramos que CIRCUIT-SAT $\leq_p$ SAT. $\square$

Satisfazibilidade 3-CNF

- Uma formula está na forma normal conjuntiva (*conjunctive normal form* - CNF) se é expressa como uma conjunção (ANDs) de clausulas, e cada clausula como disjunções (ORs) de uma ou mais literais.
- Uma fórmula booleana está na forma normal 3-conjuntiva (3-CNF) se está na forma normal conjuntiva e cada cláusula tem exatamente três literais distintas.
- Por exemplo:

$$(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

3-CNF-SAT

Dada uma fórmula booleana na forma normal 3-conjuntiva, decidir se existe uma atribuição das variáveis booleanas que faz com que ela seja avaliada como 1 (verdadeira).

Teorema

3-CNF-SAT é NP-completo

Lema

3-CNF-SAT $\in$ NP

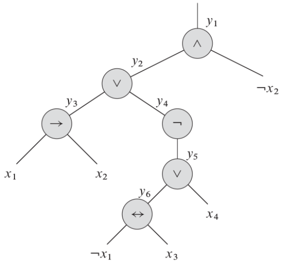
Prova igual à prova que SAT $\in$ NP.

Lema

3-CNF-SAT $\in$ NP-Difícil

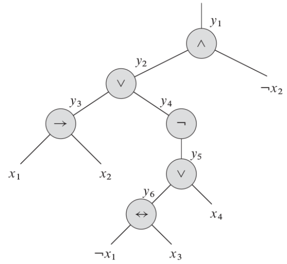
- Para mostrar que 3-CNF-SAT $\in$ NP-Difícil mostraremos que SAT $\leq_p$ 3-CNF-SAT.
- Considere ϕ uma formula booleana qualquer.
- Primeiramente construímos uma árvore de análise para ϕ em que cada literal é uma folha e os conectivos são nós internos. Essa construção sempre é possível (usando a associatividade podemos colocar parenteses para explicitar uma ordenação)

- Por exemplo $\phi = ((x_1 \to x_2) \vee \neg((\neg x_1 \leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$.



- Depois dessa construção, introduzimos uma variável para cada aresta que sobe dos nós internos.
- A seguir escrevemos cláusulas para cada nó interno. E fazemos a conjunção dessas cláusulas e criamos uma formula ϕ'.

- Por exemplo $\phi = ((x_1 \to x_2) \vee \neg((\neg x_1 \leftrightarrow x_3) \vee x_4)) \wedge \neg x_2$.



- No exemplo ao lado a formula obtida seria:

$$\begin{aligned}
\phi' \;=\; y_1 & \wedge (y_1 \leftrightarrow (y_2 \wedge \neg x_2)) \\
& \wedge (y_2 \leftrightarrow (y_3 \vee y_4)) \\
& \wedge (y_3 \leftrightarrow (x_1 \to x_2)) \\
& \wedge (y_4 \leftrightarrow \neg y_5) \\
& \wedge (y_5 \leftrightarrow (y_6 \vee x_4)) \\
& \wedge (y_6 \leftrightarrow (\neg x_1 \leftrightarrow x_3)) \,.
\end{aligned}$$

- Dessa forma obtemos clausulas que tem no máximo 3 literais. Mas ainda não está na forma normal 3-conjuntiva. Então podemos escrever a tabela verdade de cada clausula, e encontrar os valores que tornam ela FALSA. Por exemplo a cláusula $\phi'_1 = y_1 \leftrightarrow (y_2 \wedge \neg x_2))$

y_1	y_2	x_2	$(y_1 \leftrightarrow (y_2 \wedge \neg x_2))$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	1

- Analisando as linhas da tabela que tornam a cláusula FALSA obtemos:

$(y_1 \wedge y_2 \wedge x_2) \vee (y_1 \wedge \neg y_2 \wedge x_2) \vee (y_1 \wedge \neg y_2 \wedge \neg x_2) \vee (\neg y_1 \wedge y_2 \wedge \neg x_2)$

- Agora aplicamos a lei de DeMorgan

$\phi''_1 = (\neg y_1 \vee \neg y_2 \vee \neg x_2) \wedge (\neg y_1 \vee y_2 \vee \neg x_2) \wedge (\neg y_1 \vee y_2 \vee x_2) \wedge (y_1 \vee \neg y_2 \vee x_2)$

- Caso a cláusula resultante só tenha 2 literais $(l_1 \vee l_2)$, então incluímos uma literal auxiliar e a seguinte formula $(l_1 \vee l_2 \vee p) \wedge (l_1 \vee l_2 \vee \neg p)$
- No caso de uma cláusula com uma única literal l, incluímos 2 literais auxiliares $(l \vee p \vee q) \wedge (l \vee p \vee \neg q) \wedge (l \vee \neg p \vee q) \wedge (l \vee \neg p \vee \neg q)$.
- Seja ϕ''' a fórmula em 3-CNF-SAT resultante.

Lema

ϕ''' é satisfazível se e somente se ϕ é satisfazível.

- Como todas as transformações preservam o valor algébrico da formula, tanto ϕ quanto ϕ''' são equivalentes.
- A redução é calculada em tempo polinomial. Construir ϕ' a partir de ϕ insere no máximo uma variável e uma cláusula por conectivo.
- Construir ϕ'' a partir de ϕ' introduz no máximo oito cláusulas para cada cláusula. E a construção de ϕ''' no máximo multiplica por 4 o número de cláusulas.
- Portanto mostramos uma redução de tempo polinomial de SAT para 3-CNF-SAT e concluímos a prova. $\square$

CLICK

Problema do CLICK

Dado um grafo não orientado $G = (V, E)$, e um inteiro k decidir se existe um subgrafo G' induzido de G que tenha k vértices e seja completo.

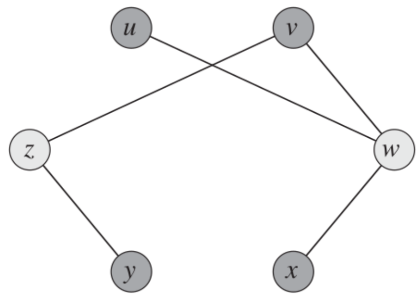
Teorema

CLICK é NP-completo

- Provado com 3-CNF-SAT $\leq_p$ CLICK

Cobertura por Vértices

- Dado um grafo não direcionado $G = (V, E)$ uma cobertura por vértices de G é um subconjunto $V' \subseteq V$ tal que para toda aresta $(u, v) \in E$, pelo menos um entre u e v deve estar em V'.
- Dizemos que um vértice em V' cobre todas as arestas adjacentes a ele. E em uma cobertura por vértices todas as arestas devem ser cobertas.
- O **tamanho** de uma cobertura por vértices é o número de vértices que contêm.



Problema da Cobertura por Vértices - VERTEX-COVER

Dado um grafo não orientado $G = (V, E)$, e um inteiro k decidir se existe uma cobertura por vértices $V' \subseteq V$ de tamanho k.

Teorema

VERTEX-COVER é NP-completo

Lema

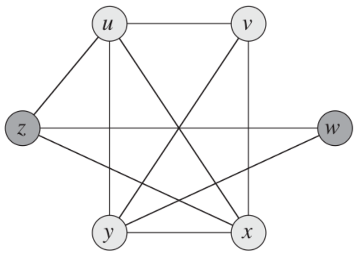
VERTEX-COVER $\in$ NP

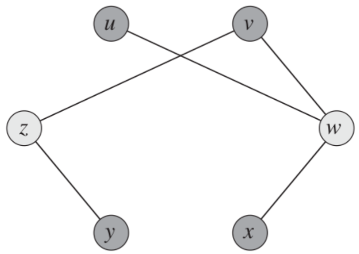
- Utilizando a própria cobertura V' como certificado, podemos verificar facilmente, em tempo polinomial, se $|V'| = k$ e se toda aresta (u, v), u ou v está em V'.

Lema

VERTEX-COVER $\in$ NP-Difícil

- Vamos mostrar que CLICK $\leq_p$ VERTEX-COVER.
- Definição: O **Complemento** de um grafo $G = (V, E)$ denotado por $\overline{G} = (V, \overline{E})$, em que, $\overline{E} = \{(u, v) : u, v \in V, u \neq v \text{ e } (u, v) \notin E\}$. Ou seja, o complemento de G é um grafo com os mesmos vértices e exatamente as arestas que não estão em G.
- A redução consiste em dada uma entrada $< G, k >$ do problema da CLICK. Calcular o complemento $\overline{G}$, o que pode ser feito em tempo polinomial.
- E então usar $\overline{G}$ como entrada para o problema do VERTEX-COVER, procurando uma cobertura de tamanho $|V| - k$.





Lema

G tem uma CLICK de tamanho k, se e somente se, $\overline{G}$ tem uma cobertura por vértices de tamanho $|V| - k$

- ($\rightarrow$) Se G tem uma CLICK C com k vértices, então $V \setminus C$ é uma cobertura em $\overline{G}$.
- Seja (u, v) qualquer aresta em $\overline{E}$, então (u, v) não está em G e portanto u e v não podem estar em C simultaneamente.
- Logo, pelo menos um deles está em $V \setminus C$ e portanto (u, v) está coberta.
- ($\leftarrow$) Se $\overline{G}$ tem uma cobertura por vértices $V' \subseteq V$ em que $|V'| = |V| - k$, então para todo $u, v \in V$ se $(u, v) \in \overline{E}$ então ou $u \in V'$ ou $v \in V'$ ou ambos. Pela contrapositiva se nenhum dos dois está em V' significa que (u, v) está em E e portanto $V - V'$ é uma CLICK, e tem tamanho $|V| - |V'| = k$
- Portanto mostramos um redução CLICK $\leq_p$ VERTEX-COVER. $\square$