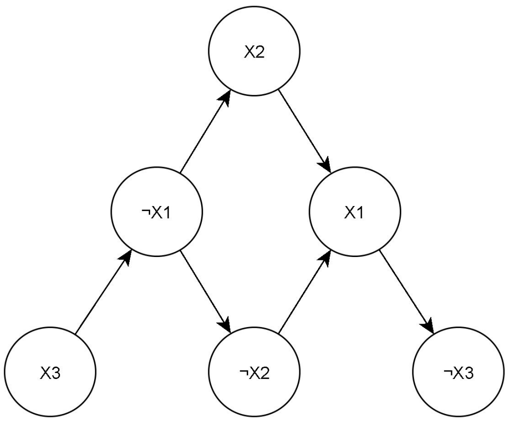
Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Andre Garrido Damaceno - 15/0117531
Patrick Vitas Reguera Beal - 15/0143672
Projeto e Análise de Algoritmos
Prof. Flavio L. C. de Moura

# Prova 3
# NP-completude

**Questão 1)** 2-SAT: $(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge x_3$

A forma conjuntiva normal do 2-SAT a partir da expressão acima, é definida pela seguinte regra : para cada item $(A \vee B)$ da equação, em que $A$ e $B$ são literais, criam-se as arestas $\overline{A}B$ e $\overline{B}A$. Clarificando, no primeiro termo, o vértice $\overline{A}$ é direcionado a $B$, formando a aresta direcionada ( $\overline{A}$, $B$ ). O mesmo vale para o segundo termo. Aplicando esta regra ao 2-SAT dado, obtemos:




Afirmação: Por indução, podemos analisar que, se o grafo G contém um caminho de $x_i$ para $x_j$, existe também o caminho de $\overline{x_j}$ para $\overline{x_i}$.

Prova: Pela definição de construção do 2-SAT, se há uma aresta ( $x_1$ , $x_2$ ) em um digrafo, então também existe uma aresta ( $\overline{x_2}$ , $\overline{x_1}$ ). Dessa forma, dado o caminho ( $x_i$ -> $x_{(i+1)}$ -> … -> $x_j$ ), então também existem as arestas ( $x_i$ , $x_{(i+1)}$ ), ( $x_{(i+1)}$ , $x_{(i+2)}$ ), …, ( $x_{(i+n)}$ , $x_j$ ) que formam o caminho. De forma análoga, por indução, existem também as arestas ($\overline{x_j}$ , $\overline{x_{(i+n)}}$ ), ( $\overline{x_{(i+n-1)}}$ , $\overline{x_{(i+n-2)}}$ ), …, ( $\overline{x_{(i+1)}}$ , $\overline{x_i}$ ) e portanto o caminho inverso existe e é dado por ( $\overline{x_j}$ -> $\overline{x_{(i+n)}}$ -> … -> $\overline{x_i}$ ) .

Afirmação: Um grafo G contendo um caminho $(x_i , \ldots , \overline{x_i})$ não satisfaz a condição 2-CNP.

Prova: Qualquer escolha de valores para os N vértices de G não resultará em 1.

Usando como base a instância de 2-SAT, temos:

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

Realizando a primeira operação de interseção, temos:

$$(x_1 \wedge \overline{x_1}) \ \vee (x_1 \wedge \ \overline{x_3}) \ \vee \ \ldots$$

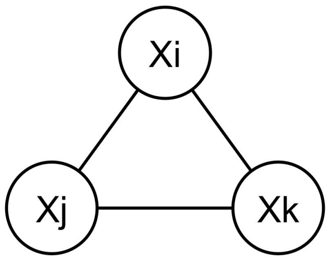
Com isso, vemos que o primeiro termo representa um caminho entre $x_1$ e $\overline{x_1}$ , mostrando a condição insatisfatória.

Portanto, basta achar um caminho de ( $x_i$ , … , $\overline{x_i}$ ) para mostrar a condição insatisfatória presente em um digrafo G fruto de um problema 2-SAT. Ao utilizarmos um algoritmo de busca em profundidade ou largura (que possuem complexidade temporal polinomial, como o DFS e o BFS), podemos detectar a indeterminação presente nestes dígrafos.

Logo, o 2-SAT acima pertence a classe de problemas que podem ser determinados em tempo polinomial, ou seja, 2-SAT $\in$ P.

**Questão 2)**

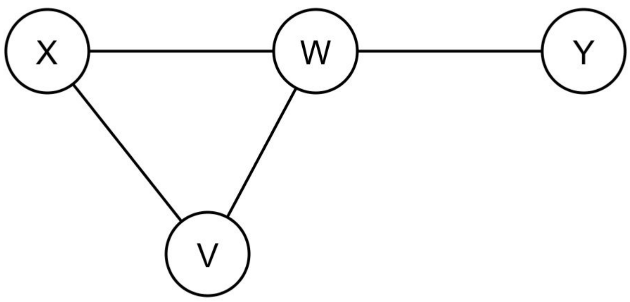
Para provar que um problema de HAM-CYCLE pertence a classe de problemas NP-Completo, por definição, devemos provar que este mesmo problema pertence a classe de problemas NP e também NP-Difícil.
Para provar que um problema HAM-CYCLE $\in$ NP, podemos analisar um Grafo genérico $G$ representado por um HAM-CYCLE, como o abaixo.




Obtendo este Grafo, vemos que para provar seu HAM-CYCLE, basta utilizar um algoritmo de busca em largura ou profundidade, checando se a respectiva aresta já foi visitada. Esta solução possui custo de verificação em tempo proporcional a $O(V+E)$, sendo então um custo polinomial e portanto HAM-CYCLE $\in$ NP.

Agora, para provar que um problema HAM-CYCLE $\in$ NP-Difícil, inicialmente precisaremos partir de uma solução já existente em SAT, 3-SAT ou CLIQUE, pois as três classes de problemas estão dentro de problemas NP-Completos.

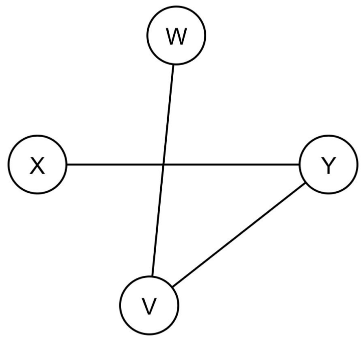
Considere o Grafo $G(V,E)$ abaixo:



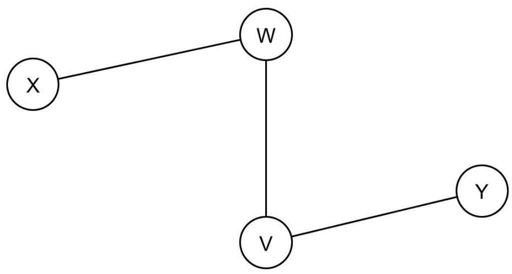

O problema de cobertura de vértices consiste em um subconjunto $V'$ de $V$, para um grafo $G(V,E)$, em que $V'$ corresponde aos vértices que cubram as respectivas arestas incidentes de $G$ e não possui arestas em $E$.

Considere um número arbitrário $J$, em que $J = V'$ do grafo $G$, para confirmar se $J = V'$, basta percorrer o subconjunto $V'$ e verificar se todas as novas arestas ali presentes cobrem $V$. Para isso, pode-se utilizar um algoritmo de busca em largura ou profundidade, com complexidade $O(V + E)$. Portanto, cobertura de vértices $\in$ NP.

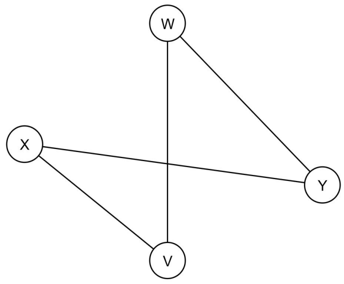
O grafo em questão contém o 3-CLIQUE $(x, v, w)$. Ao analisarmos o problema de cobertura de vértices deste grafo, obtemos:




Com este grafo resultante, a solução do problema de cobertura de vértices torna-se polinomial. Adicionalmente, o problema de cobertura de vértice possui custo de verificação em tempo proporcional a $O(V + E)$. Portanto, o problema da cobertura de vértices é provado NP-Completo por transferência por CLIQUE.
Considerando agora o seguinte Grafo:




Ao analisarmos o problema de cobertura de vértices para este Grafo, vemos que $V' = \{V, W\}$, e obtemos o seguinte grafo complementar:




Podemos notar que este subgrafo também é um HAM-CYCLE,  já que existe um ciclo que visita todos os vértices uma única vez. Logo está provado que HAM-CYCLE $\in$ NP-Difícil. Juntando esta prova com a inicial do problema, de que HAM-CYCLE $\in$ NP, está provado que HAM-CYCLE $\in$ NP-Completo.

**Questão 4)**

Partindo da seguinte instância FNCX-SAT como exemplo:

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

pode-se provar que FNCX-SAT é satisfazível em tempo polinomial ao reduzir a equação usando a seguinte lógica:

Seguindo a definição, um literal em aritmética MOD 2 pode ser escrito como:

$$\overline{x_i} = x_i + 1$$

A operação lógica de XOR é equivalente a aritmética em módulo 2.
Logo, ao representar a instância acima em MOD 2, podemos obter as seguintes expressões:

$$(x_1 + x_2 + (x_3 + 1) + x_4) \wedge (x_2 + x_3) \textbf{ MOD 2}$$

Assim, para a expressão acima ser satisfazível, a seguinte relação deve se manter:

$$(x_1 + x_2 + (x_3 + 1) + x_4) = 1$$
$$(x_2 + x_3) = 1$$

Como pode ser visto, o FNCX-SAT pode ser representado como um sistema de equações lineares do tipo MOD 2.

Utilizando a propriedade de operação lógica, temos :

$$(x_i \oplus x_j) \ = \ (x_j \oplus x_i)$$

e também que

$$(x_i \oplus x_i) \ = \ 0$$

Assim, podemos montar a matriz equivalente ao sistema acima:

$$\begin{bmatrix} x1 & x2 & (x3+1) & x4 \\ & x2 & x3 & \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

A matriz em questão pode ser resolvida utilizando a técnica de Redução Gaussiana. A execução do algoritmo de eliminação gaussiana executa os seguintes tipos de operações elementares:

- Executa a troca de duas linhas
- Multiplica a linha por um número não nulo
- Adiciona o múltiplo de uma linha na outra.

Efetivamente sendo calculado apenas a parte superior da matriz triangular (como explicado acima, elementos $A_{i\,j}$ , em que i = j, possuem o valor nulo, e os elementos $A_{i\,j}$ são equivalentes aos $A_{ji}$ .

Dessa forma, sabendo que são necessárias $\frac{n*(n+1)}{2}$ divisões, $\frac{2*n^3 + 3*n^2 - 5*n}{6}$ multiplicações e $\frac{2*n^3 + 3*n^2 - 5*n}{6}$ subtrações. Vemos que é são necessárias aproximadamente $\frac{2*n^3}{3}$ operações.

Portanto, vemos que a solução pode ser encontrada em tempo $O(n^3)$ , então FNCX-SAT $\in$ P.