1. 2-SAT

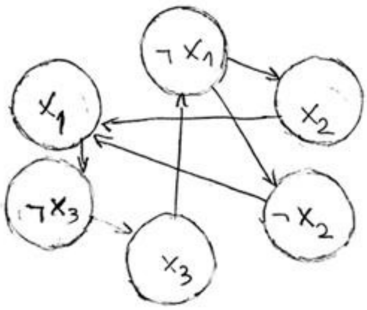
$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_1 \vee x_2) \wedge (x_3 \vee x_3)$$

Prove que 2-SAT $\in P$

Podemos criar um grafo com $2n$ vértices, sendo n o número de variáveis. Cada vértice representa um possível valor desta variável, sendo que cada uma delas pode apresentar como um literal verdadeiro ou falso.

Para cada cláusula de união, um dos literais deve ser sempre verdadeiro, ou seja em um caso $(X \vee Y)$ se x_1 não for verdadeiro, x_2 necessariamente deve ser para que a cláusula seja verdadeira

Partindo deste princípio, cada cláusula pode ser representada por duas arestas: uma de $\neg X$ para Y (negação do primeiro para o segundo) e outra de $\neg Y$ para X (negação da segunda para a primeira)



Se existe a aresta (x, y), existe a cláusula $(\neg x \vee y)$

Se o grafo contem um caminho de X para Y, também deve conter de $\neg Y$ para $\neg X$.

A instância se torna insatisfatível se existir qualquer variável x, tal que:

1. Existe caminho de x para $\neg x$
2. Existe caminho de $\neg X$ para X

Prova por contradição

Suponha que existem caminhos de x para $\neg x$ e de $\neg x$ para x sendo x qualquer variável no grafo, mas também exista uma atribuição satisfatível $\rho(x_1, x_2, \dots x_n)$ para a instância

Caso 1: Em $\rho(x_1, x_2 \dots x_n)$ tal que x = verdadeiro

$$\begin{array}{ccccccc} x & \longrightarrow & A & \longrightarrow & B & \longrightarrow & \neg x \\ V & & V & & F & & F \end{array}$$

Por construção, existe uma aresta de A para B no grafo se existir a cláusula $(\neg A \vee B)$ na instância.
A aresta de A para B indica que se A for verdadeiro, sua negação é falsa, portanto B necessariamente deve ser verdadeiro para a cláusula ser verdadeira.
Como x é verdadeiro, todos os literais do caminho de x para A, incluindo o A, devem ser verdadeiros. Similarmente, todos os caminhos de B para $\neg x$ incluindo B devem ser falsos, pois $\neg x$ = Falso, o que resulta em uma aresta entre A e B, com A = Verdadeiro e B = falso.
A cláusula $(\neg A \vee B)$ seria portanto falsa, contradizendo nossa atribuição de satisfação $\rho(x_1, x_2 \dots x_n)$ para a instancia

Porem, a existência destes caminhos de x para $\neg x$ e/ou de $\neg x$ para x pode ser determinada por algoritmos de busca em grafos, como o Breadth First Search (BFS) ou Depth First Search (DFS). Ambos rodam em tempo polinomial, de complexidade $O(V+E)$, provando que 2-SAT é P.

S T Q Q S S D __/__/__

Questão 2) Para mostrar que HAM-CYCLE é um problema NP-completo iremos fazer a redução do 3-SAT para o HAM-CYCLE da seguinte forma:

Primeiro é importante levar em consideração que para uma expressão 3-SAT que contenha n variáveis, existem 2^n atribuições possíveis para os literais.

Para modelar as 2^n possibilidades, utilizamos um grafo direcionado contendo 2^n diferentes ciclos Hamiltonianos usando o seguinte método:

1 -> Construímos n caminhos $P_1, P_2, \dots, P_n$ correspondente a n variáveis.

Cada caminho P_i deve ter 2K vértices ($v_{i1}, v_{i2} \dots v_{i2K}$) onde K é o número de cláusulas da expressão.

1,a -> Primeiro é necessário criar os nós dos caminhos P_1, P_2, P_3 e P_4.

1,b -> Depois é necessário adicionar os vértices de $v_{i,j-1}$ até v_{ij} em P_i, correspondendo à atribuição $x_i = true$.

Por fim adicionamos os vértices de v_{ij} para $v_{i,j-1}$ em P_i, correspondendo à atribuição $x_i = false$

2 -> No segundo passo todos os caminhos são conectados entre x_i e adicionamos vértices de $v_{i,1}$ e $v_{i,6}$ para $v_{i+1,1}$ e $v_{i+1,6}$.

___/___/___ S T Q Q S S D

3 -> Neste passo adicionamos dois nós: um nó correspondente à origem (S) e outro correspondente ao destino (t).

4 -> No quarto passo os nós origem (S) e destino (t) são conectados ao grafo de maneira que dois vértices saiam de S para $v_{1,1}$ e $v_{1,6}$ e um vértice saia de $v_{4,1}$ para t e outro de $v_{4,6}$ para t.

5 -> Neste passo é adicionado um vértice saindo do destino (t) para a origem (S).

6 -> No sexto passo adicionamos nós correspondentes às cláusulas da expressão que utilizaremos.

3-CNF: $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee x_4) \wedge (x_1 \vee \bar{x}_2 \vee x_4)$

7 -> Neste passo conectamos as cláusulas aos caminhos da seguinte forma:

Se a cláusula C_j contém a variável x_i

a -> Conectamos C_j em $v_{i,2j-1}$ e $v_{i,2j}$

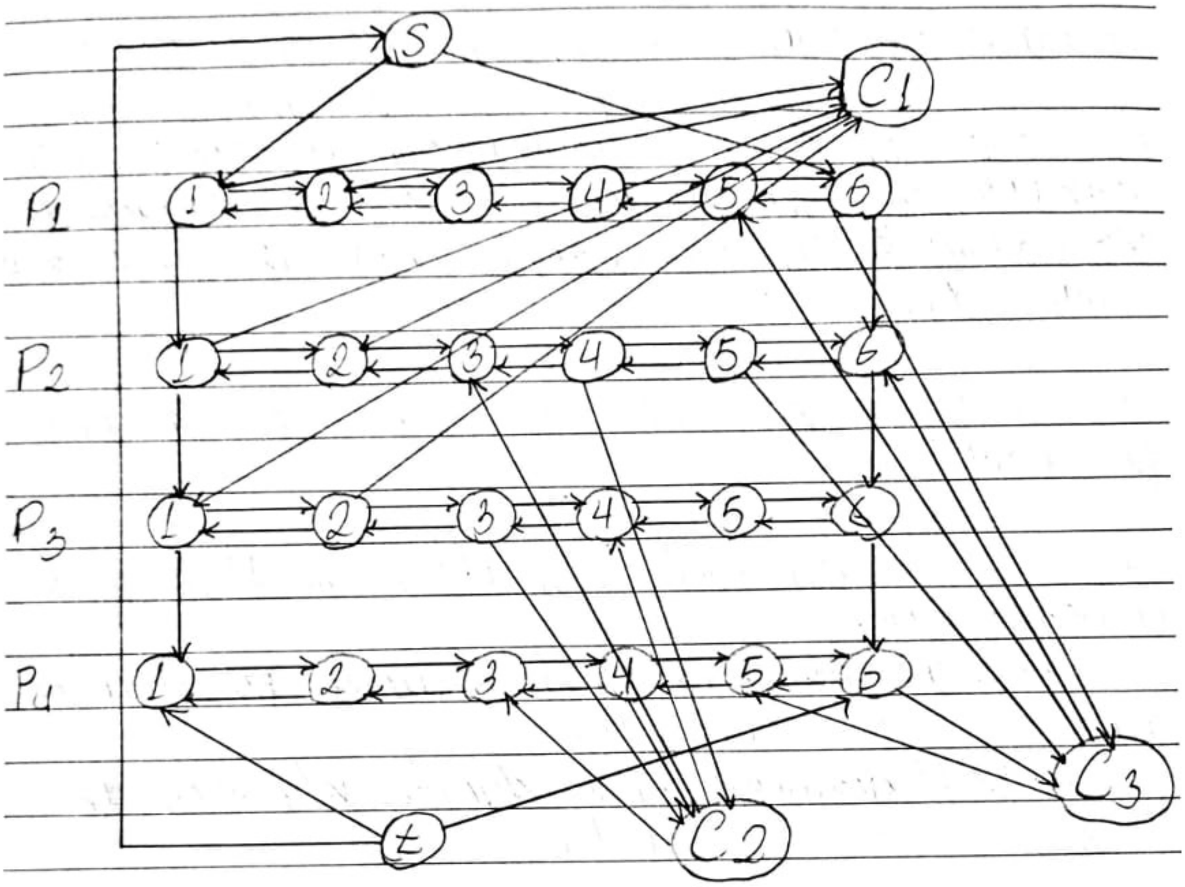
b -> A direção da conexão do caminho $C_j, v_{i,2j-1}$ e $v_{i,2j}$ deve ser:

b.1 -> direita para esquerda se C_j contém x_i

b.2 -> Esquerda p/ direita se C_j contém $\bar{x}_i$

spiral

Dessa maneira teremos o seguinte grafo:



Com relação ao grafo, é importante saber que:

• Qualquer ciclo Hamiltoniano no grafo construído atravessa o caminho P_i tanto da esquerda p/ direita quanto da direita p/ esquerda. Isso ocorre porque qualquer caminho que entra em um vértice $v_{i,j}$ tem saída de $v_{i,j+1}$ tanto imediatamente quanto por um nó de cláusula entre eles, para que a propriedade hamiltoniana seja mantida. Similarmente, todos os caminhos

___/___/___ S T Q Q S S D

que entram por $v_{i,j-1}$ tem saída por $v_{i,j}$

• Uma vez que cada caminho Pi pode ser atravessado em 2 formas possíveis e temos n caminhos p/ cada um dos n literais, podem existir 2^n ciclos hamiltonianos no grafo. Cada um dos 2^n ciclos correspondem à uma atribuição particular das variáveis $x_1, x_2, \dots x_n$

• Este grafo pode ser construido na em tempo polinomial.

• Se existe um ciclo Hamiltoniano H no grafo G, vemos que:

→ Se H atravessa Pi da esquerda p/ direita devemos atribuir $x_i = true$.

→ Se H atravessa Pi da direita p/ esquerda devemos atribuir $x_i = false$

• Desde que H visite cada nó Cj, pelo menos um caminho Pi foi caminhado na direção da direita relativa ao nó Cj

→ Esta atribuição satisfaz o 3CNF.

• Se existe uma atribuição que satisfaz o 3CNF:

→ Selecionar o caminho que atravessa Pi da esquerda p/ direita se $x_i = true$, ou

spiral

S T Q Q S S D   __/__/__

da direita p/ esquerda se x_i = false.
→ Incluir as cláusulas assim que possível.
→ Conectar a origem S p/ P1, Pn até o destino t e Pi em Pi+1 p/ manter a continuidade do caminho.
→ Conectar t em S p/ completar o ciclo.
→ Já que a atribuição é tal que todas as cláusulas são satisfeitas, todos os nó-cláusulas são incluídos no caminho.
Os Pi vértices, vértice S e t são todos incluídos e como o caminho é unidirecional nenhum vértice é visitado duas vezes

Com isso, o caminho obtido é um ciclo Hamiltoniano.

Assim sendo, a seguinte afirmação satisfaz a expressão $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee x_4) \wedge (x_1 \vee \bar{x}_2 \vee x_4)$:

x_1 = True, x_2 = false, x_3 = True, x_4 = false

Com isso, o ciclo Hamiltoniano é NP-completo.

4.

Uma cláusula XOR é uma série de operações sobre um grupo de literais e/ou constantes booleanas.

A clausula Xor está na forma padrão quando todos os literais se apresentam na fase positiva.

Por exemplo, uma clausula $(x_1 \oplus x_2 \oplus \neg x_3)$ pode ser escrita na maneira padrão como $(x_1 \oplus x_2 \oplus x_3 \oplus 1)$, o que significa que $x_1 \oplus x_2 \oplus x_3 = 0$, para que XOR seja verdadeiro.

Cada cláusula XOR consiste em n variáveis e uma conjunção de 2^{n-1} cláusulas com n literais cada. A cláusula antes apresentada $(x_1 \oplus x_2 \oplus \neg x_3)$ pode ser interpretada com a CNF (forma normal conjuntiva) equivalente

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

Um conjunto de m cláusulas XOR sobre n variáveis pode ser considerado um sistema de m equações lineares. Portanto, podemos representá-las em formato matricial como $Ax = b$, onde A é uma matriz $m \times n$ e b um vetor de valores constantes booleanos.

$Ax = b$ é alternativamente apresentado como uma única matriz de booleanos, quando se concatena A e b. $M = [A|b]$

Exemplo:

$c_1: (x_1 \oplus x_2)$; $c_2: (x_2 \oplus x_4)$; $c_3 (x_1 \oplus \neg x_2 \oplus \neg x_3)$

corresponde a matriz

$$M \to \begin{array}{c|ccccc} & x_1 & x_2 & x_3 & x_4 & b \\ \hline c_1 & 1 & 1 & 0 & 0 & 0 \\ c_2 & 0 & 1 & 0 & 1 & 0 \\ c_3 & 1 & 1 & 1 & 0 & 1 \end{array}$$

A matriz $M = [A|b]$ pode ser reduzida por eliminação Gaussiana ou de Gauss-Jordan, removendo linearmente equações dependentes, operando em tempo polinomial. Logo, FNCX-SAT $\in P$