João Marcelo Nunes Chaves 15/0132085 e João Victor Poletti 15/0132425

Flávio L. C. de Moura

Projeto e Análise de Algoritmos

6 de dezembro de 2018

Prova 3 - Projeto e Análise de Algoritmos

1 -

Pela definição de SAT (problema de satisfatibilidade booliana) temos que uma instância Φ de SAT está na forma normal conjuntiva (CNF), desse modo é uma conjunção de uma disjunção de literais (ou seja, conjunção de cláusulas). Essa afirmação pode ser observada a partir da instância abaixo:

$$\Phi = (x1 \lor \neg x2) \land (\neg x1 \lor x2) \land (\neg x1 \lor \neg x2) \land (x1 \lor \neg x3)$$

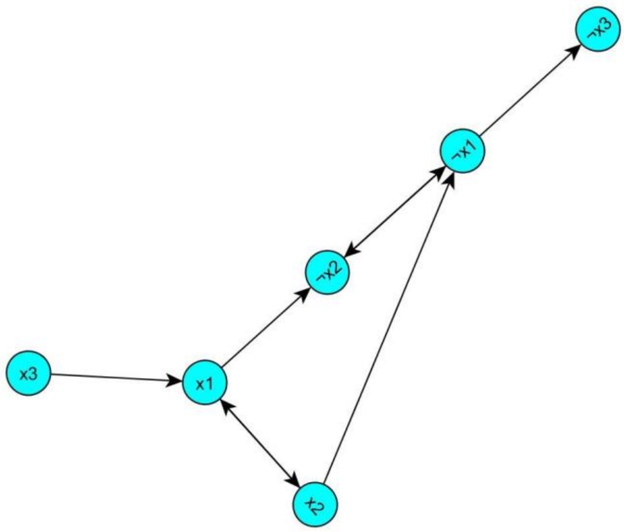
Para provarmos que 2-SAT pertence a P, temos que provar que ele pode ser resolvido em tempo polinomial.

Considere G(V, E) onde V é o conjunto de vértices e E o conjunto de arestas. Ademais, considere que uma disjunção da forma (x1 ∨ x2) corresponde a relação: `

¬x1 -> x2, ¬x2 -> x1.

Além disso, para garantir que uma cláusula de Φ seja satisfatível para a solução da instância devemos garantir que ela retorne 1 (é verdadeira). Para tal, faz-se necessário respeitar a ideia de que um literal xn dessa cláusula não alcance o seu literal negado (¬xn). Caso isso acontece, essa cláusula será falsa (x1 -> ¬x1, por exemplo) e, portanto, fará com que a instância Φ seja falsa. Desse modo, Φ é insatisfatível.

Sendo assim, a instância Φ pode ser descrita pelo grafo abaixo, é importante observar que há vértices xn que alcançam, sim, o seu literal negado (¬xn):



Portanto, pode-se observar que o grafo acima é insatisfatível.

Por fim, sabendo dos fatos acima, podemos realizar uma busca em profundidade ou em largura que possuem complexidade O(V+E) que é polinomial. Portanto, podemos dizer que SAT-2 é pertencente a P, pois é resolvido em tempo polinomial.

2 -

Para provar que HAM-CYCLE é NP-completo, temos que mostrar que HAM-CYCLE é NP-difícil, bem como NP.

Primeiramente iremos provar que HAM-CYCLE é NP-difícil. Para tal, levamos em consideração que CLICK é NP-completo, logo é NP-difícil. No entanto, devemos antes provar que VERTEX-COVER é, também, NP-difícil.

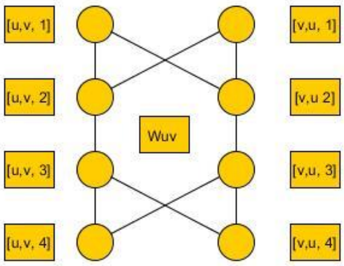
Um VERTEX-COVER do grafo G é um conjunto de vértices que cobre todos as arestas de E, ou seja, a partir desse conjunto e de suas respectivas arestas pode-se chegar em todos os outros vértices de G. É válido ressaltar que as respectivas arestas desse subconjunto de V não pertencem a E. Esse conjunto de vértices é descrito por seu tamanho e é dado pela quantidade de vértices que ele possui. A partir do grafo G, cria-se um grafo G' = (V, E'), de tal forma que E' não contém nenhuma aresta de E (Complemento de E). Agora, considere um 'k' inteiro que corresponda ao tamanho de V', ou seja, |V'| = k. A partir dessa ideia, basta verificar que os vértices de V' alcançam todos os outros vértices de V - V' com suas arestas. Em outras palavras, uma aresta (*u,v*) pertencente a E em que *u* pertencente a V' ou *v* pertence a V'. Isso pode ser verificado em tempo polinomial.

Por outro lado, devemos provar também que VERTEX-COVER é NP-difícil. Para isso, devemos provar $CLIQUE <= pVERTEX-COVER$. Para tal, vamos considerar que G', definido anteriormente, corresponde ao complemento de G. Além disso, é possível calcular G' a partir de G em tempo polinomial. Por fim, consideramos que o grafo G tem um CLIQUE de 'k' se o grafo G' tem um VERTEX-COVER de |V|-k. Em outras palavras, V - V' é um CLIQUE de tamanho |V| - |V'| = k.

Inicialmente, considere um grafo G = (V, E) com um ciclo Hamiltoniano, podemos visitar todos os vértices utilizando um algoritmo de busca em grafo que possua O(V+E), dessa forma, saberemos quantas vezes um vértice foi visitado. Algoritmos de busca em largura (BFS) e busca em profundidade (DFS) visitam o vértice apenas uma vez, visto que os nós são marcados para que não sejam mais percorridos. Logo, teremos a resposta para HAM-CYCLE que será em tempo polinomial. Portanto HAM-CYCLE é pertencente a NP.

Como já provamos que HAM-CYCLE é NP, basta provar que HAM-CYCLE pertence à classe NP-difícil. Para essa prova, iremos construir um grafo G'' = (V', E') de tal forma que G = (*V, E)* tem um VERTEX-COVER de tamanho k se, e somente se, G' tenha um ciclo hamiltoniano. Isso é feito de tal forma que reduzimos o VERTEX-COVER para o ciclo hamiltoniano.

Em seguida, utilizamos um *widget,* o qual é um pedaço do grafo J que reforça certas propriedades. Uma aresta *(u, v)* do grafo G corresponde ao *widget Wuv* do grafo G'' criado na redução. Além disso, cada vértice do *widget* é representado por [*u, v, i*], onde i limitado superiormente pelo número de vértices do grafo G. Ou seja, se considerarmos que i = 4, então |V| = 4. Desse modo, teremos o *widget* da seguinte forma para esse caso hipotético:



Esse *widget* determinará a forma com que o ciclo hamiltoniano de G' deve atravessar as arestas de *Wuv.* Isso é feito considerando que apenas os vértices do topo ([u, v, 1], [v, u, 1]) e os de baixo ([u, v, 4], [v, u, 4]) são capazes de se conectar com vértices fora de *Wuv*. Assim, o ciclo deverá entrar em *Wuv* por algum desses 4 vértices (vértices extremos ou de borda). É válido assinalar, também, que se o ciclo entrar pelo lado esquerdo de *Wuv*, ele deve percorrer *Wuv* de tal forma que translade pelo lado oposto que entrou e consiga sair de *Wuv* pelo lado esquerdo.

Além dos vértices pertencentes à *Wuv,* temos os vértices seletores (que se encontram fora de *Wuv*), que se conectam com os vértices extremos (de borda). Os seletores são considerados espaços vazios que podem ser preenchidos para representar qualquer vértice. Logo, são conectados ao início e ao fim de cada cadeia. Mas claro, para um grafo G' podemos ter vários *widgtes*, os quais vão se conectar entre si por seus respectivos vértices de borda. Logo, G’ terá vários *widgtes* conectados entre si e vários outros vértices fora desses *widgets.*

Por fim, a ideia de se determinar o caminho a percorrer dentro de cada *widget*, conectando *widgets* entre si, bem como interligando vértices fora de qualquer *widgets* à esses vértices de extremo, garantimos o ciclo hamiltoniano: caminho fechado que visita cada vértice uma única vez.

4 -

Uma fórmula FNCX-SAT pode ser descrita como:

$(x1 \oplus x2 \oplus x3...xn) \wedge (x1 \oplus x2 \oplus \neg x3...xn)$

Para realizar questão, temos que provar que FNCX-SAT é satisfatível em tempo polinomial. Para isso, iniciamos mostrando as seguintes operações possíveis em módulo 2:

$$\begin{cases} 0+0 \pmod 2 & =0 \\ 0+1 \pmod 2 & =1 \\ 1+0 \pmod 2 & =1 \\ 1+1 \pmod 2 & =0 \end{cases}$$

É possível perceber que as operações possíveis se equivalem aos resultados do operador XOR, quando utilizado com dois literais:

Entradas		Saída
A	**B**	**S**
0	0	0
0	1	1
1	0	1
1	1	0

Utilizando a seguinte fórmula:

$(x1 \oplus x2 \oplus x3) \wedge (x2 \oplus \neg x3)$

Temos que ela é satisfatível apenas se para cada cláusula:

1. $(x1 \oplus x2 \oplus x3) \equiv 1$
2. $(x2 \oplus \neg x3) \equiv 1$

Ou seja, sabendo que as operações de módulo 2 são equivalente ao operador XOR, podemos reescrever operações XOR para somas de literais, como feito a seguir para as equações 1 e 2.

1. $(x1 + x2 + x3) = 1$

Pela tabela:

x3	¬x3	x3 + 1
0	1	1
1	0	0

Podemos observar que os valores de ¬x3 são equivalentes aos valores de x3+1 utilizando as operações de módulo 2.

Desse modo:

2. $(x2 + (1 + x3)) = 1$

Dessa forma temos um sistema de equações lineares passíveis para a resolução. Podemos utilizar o método da eliminação Gaussiana para realizar esse passo.

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

Uma possível solução para a matriz acima é:

x1 = 1;

x2 = 0;

x3 = 0;

Desse modo, teremos cerca de $2n^3/3$ operações para obter um resultado possível. Sabendo disso, temos que a eliminação Gaussiana é $O(n^3)$, já que o número de operações é da ordem $n3$. Sabendo que $n3$ é polinomial, temos que a eliminação Gaussiana para solucionar o problema FNCX-SAT pertence à classe P e por isso FNCX-SAT é, também, pertencente a classe P.