

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

CIC 117536 - Projeto e Análise de Algoritmos

Terceira Prova

Turma: B

NP-completude

Prof. Flávio L. C. de Moura

6 de dezembro de 2018

Alex Souza - 15/0115474

Rafael Fernandes - 14/0030395

1. **(2.5 pontos)** O problema 2-SAT tem como instâncias as fórmulas lógicas formadas por conjunções de disjunções de até dois literais, onde um literal é uma variável booleana ou a negação de uma variável booleana. Por exemplo, a expressão a seguir é uma instância de 2-SAT:

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

Prove que $2\text{-SAT} \in \text{P}$.

Solução.

Afirmção: $2\text{-SAT} \in \text{P}$.

Provaremos que $2\text{-SAT} \in P$ construindo um algoritmo que pode ser executado, no pior caso, em tempo polinomial.

Algoritmo: Seja ϕ uma instância do problema 2-SAT.

1. Escolha uma variável v em ϕ .
2. Atribua o valor verdade de v para TRUE.
3. Remova de ϕ todas as cláusulas da forma $(v \vee w)$.
4. Para cada cláusula da forma $(\bar{v} \vee w)$ em ϕ , atribua o valor verdade de w para TRUE e remova a cláusula $(\bar{v} \vee w)$ de ϕ .
5. Continue atribuindo valores verdades para as variáveis e removendo as cláusulas que são satisfatíveis com atribuição de TRUE, como valor verdade, até que:
 - (a) Não haja mais variáveis que foram atribuídas valores verdade.
 - (b) Acontece um problema do valor verdade de uma variável ser atribuída para TRUE e FALSE.

No caso (a), ϕ foi reduzida para uma fórmula 2-SAT menor. Se ϕ estiver vazio, conclua que ϕ é satisfatível e pare. Caso contrário, execute o algoritmo na fórmula menor.

No caso (b), prossiga o algoritmo:

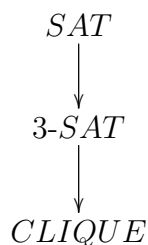
6. Atribua o valor verdade de v para FALSE.
7. Remova de ϕ todas as cláusulas da forma $(\bar{v} \vee w)$.
8. Para cada cláusula da forma $(v \vee w)$ em ϕ , atribua o valor verdade de w para TRUE e remova a cláusula $(v \vee w)$ de ϕ .
9. Continue atribuindo valores verdades para as variáveis e removendo as cláusulas que são satisfatíveis com atribuição de TRUE, como valor verdade, até que:
 - (a) Não haja mais variáveis que foram atribuídas valores verdade.
 - (b) Acontece um problema do valor verdade de uma variável ser atribuída para TRUE e FALSE.

No caso (a), ϕ foi reduzida para uma fórmula 2-SAT menor. Se ϕ estiver vazio, conclua que ϕ é satisfatível e pare. Caso contrário, execute o algoritmo na fórmula menor.

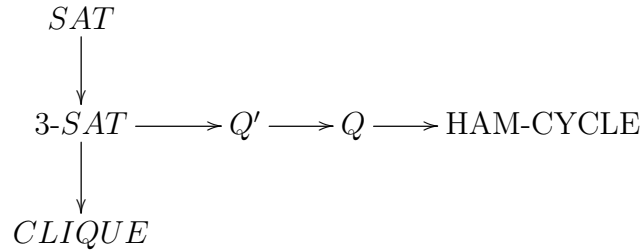
No caso (b), conclua que ϕ não é satisfatível e pare.

Este algoritmo pode ser executado em tempo $O(m^2)$, onde m = número de cláusulas, dado que uma estrutura de dados apropriada está sendo usada para armazenar as cláusulas. Logo, 2-SAT \in P.

2. **(2.5 pontos)** Em aula, assumimos que SAT é um problema NP-completo (Teorema de Cook-Levin), e a partir deste fato mostramos que 3-SAT e CLIQUE também são problemas NP-completos. As reduções foram feitas de acordo com o seguinte diagrama:



Um ciclo Hamiltoniano é um ciclo simples que visita cada vértice de um grafo exatamente uma vez. Considere o problema de decisão HAM-CYCLE que pergunta se um dado grafo (não-dirigido) G possui um ciclo Hamiltoniano. Mostre que HAM-CYCLE é um problema NP-completo. Sua solução deve ser construída a partir de SAT, 3-SAT ou CLIQUE. Caso, você não veja como reduzir diretamente HAM-CYCLE a partir destes, mas sabe como fazê-lo a partir de um certo problema Q então inicialmente mostre que Q é NP-completo a partir de SAT, 3-SAT ou CLIQUE, e assim por diante. Digamos que você não saiba como mostrar que Q é NP-completo diretamente a partir de SAT, 3-SAT ou CLIQUE, mas você sabe como fazê-lo a partir de outro problema Q' , e também sabe como mostrar que Q' é NP-completo a partir de 3-SAT, por exemplo. Então o diagrama correspondente à sua solução seria:



E todas as reduções (de 3-SAT para Q' , de Q' para Q e de Q para HAM-CYCLE) devem ser detalhadas na sua solução.

Solução.

Para provarmos que HAM-CYCLE é NP-Completo, primeiro mostraremos que DIRECTED-HAM-CYCLE é NP-Completo, utilizando-se do fato que 3-SAT é NP-Completo e que $3-SAT \leq_P DIRECTED-HAM-CYCLE$.

Afirmção: DIRECTED-HAM-CYCLE é NP-Completo.

Para que DIRECTED-HAM-CYCLE seja NP-Completo, é necessário mostrar que este problema é NP e NP-Difícil.

Afirmção: DIRECTED-HAM-CYCLE \in NP.

Podemos escrever um algoritmo verificador para qualquer possível solução do problema DIRECTED-HAM-CYCLE.

Algoritmo:

1. Seja $H = (v_1, v_2, \dots, v_n)$ o conjunto de vértices de um grafo G dirigido.
2. Para cada vértice v_i e v_{i+1} pertencente a H , verifique se existe a aresta (v_i, v_{i+1}) em $G.E$. Caso exista aresta para todo v_i e v_{i+1} e v_1 for igual a v_n , então H forma um ciclo Hamiltoniano em G .
3. Caso contrário, H não forma um ciclo Hamiltoniano em G .

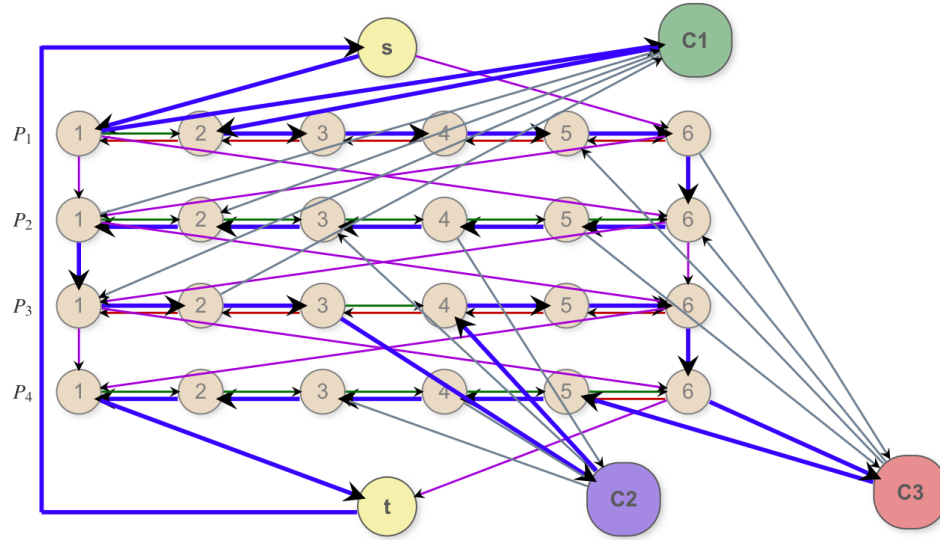
Esse algoritmo pode ser executado em tempo polinomial. Logo, podemos concluir que $DIRECTED-HAM-CYCLE \in NP$.

Afirmção: DIRECTED-HAM-CYCLE \in NP-Difícil.

Mostraremos que DIRECTED-HAM-CYCLE \in NP-Difícil provando que 3-SAT \leq_P DIRECTED-HAM-CYCLE.

Dada uma instância de 3-SAT com n variáveis x_i e k cláusulas C_j , construímos um digrafo $G = (V, E)$ que tenha 2^n ciclos Hamiltonianos.

- (a) Construa n caminhos P_1, P_2, \dots, P_n correspondentes a n variáveis. Cada caminho P_i deve conter $2k$ vértices $(v_{i,1}, v_{i,2}, \dots, v_{i,2k})$.
- (b) Adicione esses vértices ao grafo G .
- (c) Adicione as arestas $(v_{i,j-1}, v_{i,j})$ no caminho P_i do grafo G , que corresponderá a $x_i = \text{TRUE}$.
- (d) Adicione as arestas $(v_{i,j}, v_{i,j-1})$ no caminho P_i do grafo G , que corresponderá a $x_i = \text{FALSE}$.
- (e) Adicione as arestas $(v_{i,1}, v_{i+1,1}), (v_{i,1}, v_{i+1,2k}), (v_{i,2k}, v_{i+1,1}), (v_{i,2k}, v_{i+1,2k})$ ao grafo G .
- (f) Adicione os vértices s de origem e t de destino ao grafo G .
- (g) Adicione as arestas $(s, v_{i,1})$ e $(s, v_{i,2k})$ ao grafo G .
- (h) Adicione as arestas $(v_{n,1}, t)$ e $(v_{n,2k}, t)$ ao grafo G .
- (i) Adicione a aresta (t, s) ao grafo G .
- (j) Adicione os vértices C_i ao grafo G .
- (k) Adicione arestas dos vértices C_j para os vértices dos caminhos P_i :
 - i. Se a cláusula C_j contém a variável x_i então adicione as arestas $(v_{i,2j-1}, C_j)$ e $(C_j, v_{i,2j})$ ao grafo G .
 - ii. Se a cláusula C_j contém a variável $\overline{x_i}$ então adicione as arestas $(C_j, v_{i,2j-1})$ e $(v_{i,2j}, C_j)$ ao grafo G .



Qualquer ciclo Hamiltoniano no grafo G atravessa os caminhos P_i ou da direita para a esquerda ou da esquerda para a direita. Isso acontece por conta de que cada caminho P_i entra por um vértice $v_{i,j}$ e sai ou por $v_{i,j+1}$ ou por um vértice C_j a fim de manter a propriedade do ciclo Hamiltoniano. Analogamente, qualquer caminho P_i pode entrar por um vértice $v_{i,j-1}$ e sai ou por um vértice $v_{i,j}$ ou por um vértice C_j .

Desde que cada caminho P_i pode atravessar G de 2 diferentes formas e nós temos n caminhos mapeando n variáveis, existe um total de 2^n ciclos Hamiltonianos no grafo $G - \{C_1, C_2, \dots, C_k\}$. Cada um desses ciclos Hamiltonianos corresponde a uma combinação de atribuições para as variáveis x_1, x_2, \dots, x_n .

Esse grafo pode ser construído em tempo polinomial.

Conclusão:

1. Se existe um caminho Hamiltoniano H no grafo G .
 - (a) H atravessa o caminho P_i da esquerda para a direita, então atribua TRUE à variável x_i .
 - (b) H atravessa o caminho P_i da direita para a esquerda, então atribua FALSE à variável x_i .

Então a atribuição obtida satisfaz a fórmula 3-CNF.

2. Se existe uma atribuição que satisfaz a fórmula 3-CNF.
 - (a) Escolha o caminho que atravessa P_i da esquerda para a direita se x_i tiver o valor TRUE ou da direita para a esquerda se x_i tiver o valor FALSE. Inclua os vértices C_i sempre que possível.
 - (b) Conecte o vértice s a P_1 , t a P_n e o caminho P_i à $P_i + 1$ apropriadamente de forma a manter a continuidade do caminho e
 - (c) Conecte t a s para completar o ciclo.

Então o caminho obtido é um ciclo Hamiltoniano.

Dessa forma, podemos afirmar que **DIRECTED-HAM-CYCLE** \in **NP-Difícil**.

E, como **DIRECTED-HAM-CYCLE** \in NP, logo **DIRECTED-HAM-CYCLE** é **NP-Completo**.

Afirmção: HAM-CYCLE é NP-Completo.

Para que HAM-CYCLE seja NP-Completo, é necessário mostrar que este problema é NP e NP-Difícil.

Afirmção: HAM-CYCLE \in NP.

Podemos escrever um algoritmo verificador para qualquer possível solução do problema HAM-CYCLE.

Algoritmo:

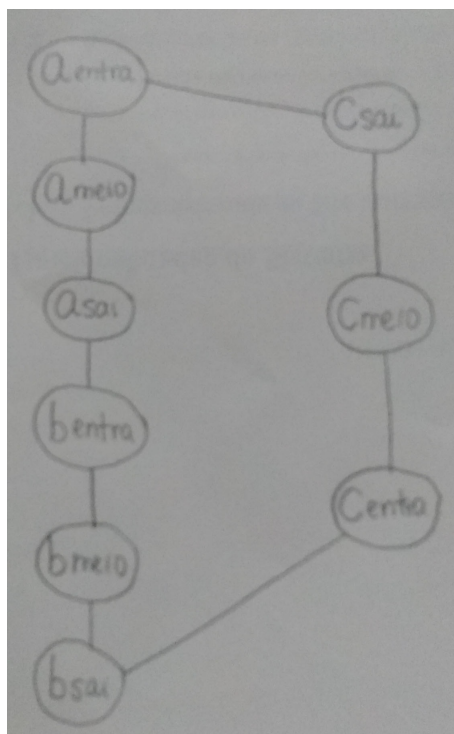
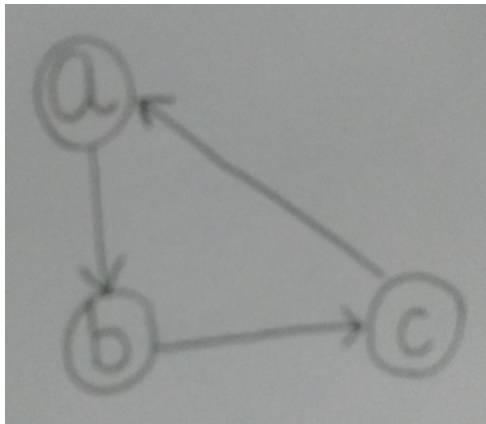
1. Seja $H = (v_1, v_2, \dots, v_n)$ o conjunto de vértices de um grafo G não dirigido.
2. Para cada vértice v_i e v_{i+1} pertencente a H , verifique se existe a aresta (v_i, v_{i+1}) em $G.E$. Caso exista aresta para todo v_i e v_{i+1} e v_1 for igual a v_n , então H forma um ciclo Hamiltoniano em G .
3. Caso contrário, H não forma um ciclo Hamiltoniano em G .

Esse algoritmo pode ser executado em tempo polinomial. Logo, podemos concluir que **HAM-CYCLE** \in NP.

Afirmção: HAM-CYCLE \in NP-Difícil.

Mostraremos que $\text{HAM-CYCLE} \in \text{NP-Difícil}$ provando que $\text{DIRECTED-HAM-CYCLE} \leq_P \text{HAM-CYCLE}$.

Dado um grafo dirigido $G = (V, E)$, construímos um grafo G' com $3n$ vértices, onde $n = |V|$.



Temos que mostrar que G possui um ciclo Hamiltoniano se e somente se G' também possui.

1. Prova \Rightarrow

Suponha que G tenha um ciclo Hamiltoniano H dirigido, como consequência direta, G' tem um ciclo Hamiltoniano não dirigido.

2. Prova \Leftarrow

Suponha que G' tenha um ciclo Hamiltoniano não dirigido H' . H' deve percorrer os vértices em G' em uma das duas ordens:

(a) $\dots v_{sai}, v_{entra}, v_{meio}, v_{sai}, v_{entra}, v_{meio}, v_{sai}, v_{entra}, v_{meio} \dots$

(b) $\dots v_{entra}, v_{sai}, v_{meio}, v_{entra}, v_{sai}, v_{meio}, v_{entra}, v_{sai}, v_{meio} \dots$

Os vértices v_{meio} em H' formam o ciclo Hamiltoniano H em G , ou o reverso de um.

Dessa forma, podemos afirmar que **HAM-CYCLE** \in **NP-Difícil**.

E, como **HAM-CYCLE** \in **NP**, logo **HAM-CYCLE** é **NP-Completo**.

3. **(2.5 pontos)** Considere o seguinte jogo em um grafo (não-dirigido) G , que inicialmente contém 0 ou mais bolas de gude em seus vértices: um movimento deste jogo consiste em remover duas bolas de gude de um vértice $v \in G$, e adicionar uma bola a algum vértice adjacente de v . Agora, considere o seguinte problema: Dado um grafo G , e uma função $p(v)$ que retorna o número de bolas de gude no vértice v , existe uma sequência de movimentos que remove todas as bolas de G , exceto uma? Mostre que este problema é NP-completo. A mesma observação feita no exercício anterior vale aqui: a prova deve ser feita a partir de problemas que provamos serem NP-completos, e reduções intermediárias, caso existam, devem ser incluídas na solução.

Solução.

BOLA-DE-GUDE é o problema de decisão que pergunta se um dado grafo (não dirigido) $G = (V, E)$, possui uma sequência de movimentos $play(u, v), u, v \in V$, que remove todas as bolas de gude no vértice v , exceto uma.

Definimos a função $play(u, v), u, v \in V$ como uma função que visita o vértice u , retira duas bolas de gude deste, caso $p(u) \geq 2$ e transfere uma bola de gude para o vértice adjacente.

Afirmção: BOLA-DE-GUDE é NP-Completo.

Para que BOLA-DE-GUDE seja NP-Completo, é necessário mostrar que este problema é NP e NP-Difícil.

Afirmção: BOLA-DE-GUDE \in NP.

Podemos escrever um algoritmo verificador para qualquer possível solução do problema BOLA-DE-GUDE.

Algoritmo:

1. Seja $G = (V, E)$ um grafo não dirigido com um número arbitrário, não negativo e não nulo de bolas de gude em seus vértices.
2. Seja $S = (play(u_i, v_i), \dots, play(u_i, v_i)), u_i, v_i \in G.V$ uma sequência de movimentos a serem executados no grafo G .
3. Para cada movimento $play(u_i, v_i)$, remova 2 bolas de gude do vértice u_i e adicione uma bola de gude ao vértice v_i .
4. Ao final da sequência S , verifique se $\sum_{i=1}^n p(v_i) = 1$. Caso seja, então a sequência S zera todos os vértices, exceto um.
5. Caso contrário, a sequência S não é uma solução do problema.

Esse algoritmo pode ser executado em tempo polinomial. Logo, podemos concluir que BOLA-DE-GUDE \in NP.

Afirmção: BOLA-DE-GUDE \in NP-Difícil.

Mostraremos que BOLA-DE-GUDE \in NP-Difícil provando que HAM-CYCLE \leq_P BOLA-DE-GUDE.

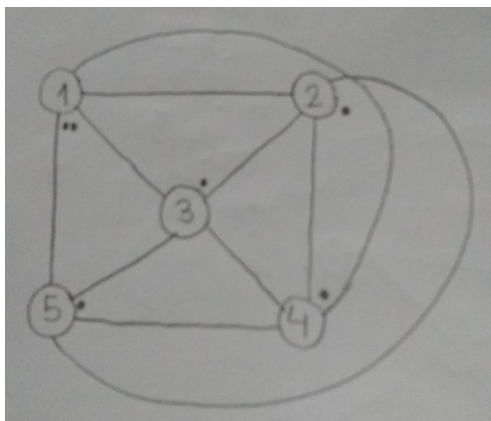
Dado um grafo $G = (V, E)$ que contenha um ciclo hamiltoniano H e o conjunto de vértices $V = \{v_1, v_2, \dots, v_n\}$ como abaixo:

Construímos um grafo $G' = (V, E')$, tal que $E' = E \cup \overline{E}$, onde \overline{E} são as arestas que não está contido em E .

- (a) Atribua a quantidade de bolas de gude de um vértice arbitrário $v \in V$ para 2.
- (b) Para os demais vértices $u \in V - \{v\}$, atribua a quantidade de bolas de gude para 1.

Este grafo pode ser construído em tempo polinomial.

No exemplo:



Aplica-se a sequência de movimentos: $play(1, 3)$, $play(3, 2)$, $play(2, 4)$, $play(4, 5)$, $play(5, 1)$, marcando os vértices que passamos.

Conclusão:

1. Se existe uma sequência de movimentos em G' que zere todas as bolas de gude dos vértices, exceto em um.

Então o caminho de vértices formado pelos movimentos define um ciclo Hamiltoniano em G .

2. Se existe um ciclo Hamiltoniano H em G .

Então a sequência de movimentos $play(u, v)$, onde $u, v \in (u, v)$ e $(u, v) \in H$, zera todas as bolas de gude dos vértices de H , exceto um.

Dessa forma, podemos afirmar que **BOLA-DE-GUDE** \in **NP-Difícil**.

E, como **BOLA-DE-GUDE** \in **NP**, logo **BOLA-DE-GUDE** é **NP-Completo**.

4. **(2.5 pontos)** Uma fórmula booleana em *forma normal conjuntiva com disjunção exclusiva (FNCX)* é uma conjunção de diversas cláusulas, e cada cláusula é uma disjunção exclusiva (XOR) de diversos literais. Lembre-se que a disjunção exclusiva é dada por:

a	b	$a \oplus b$
V	V	F
V	F	V
F	V	V
F	F	F

O problema FNCX-SAT pergunta se uma dada fórmula em FNCX é satisfatível. Mostre que o problema FNCX-SAT está em P , ou então que FNCX-SAT é NP-completo. No último caso, a mesma observação feita nos dois exercícios anteriores vale aqui: a prova deve ser feita a partir de problemas que provamos serem NP-completos, e reduções intermediárias, caso existam, devem ser incluídas na solução.

Solução.

Sabemos que:

$$\begin{aligned}
 x_1 \oplus x_2 &= \\
 &= (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2}) \\
 &= (\overline{x_1} \vee \overline{x_2}) \wedge (x_1 \vee x_2)
 \end{aligned}$$

Sendo assim, sabemos que toda disjunção exclusiva de dois literais pode ser convertida para uma conjunção de disjunções. Portanto, podemos utilizar o seguinte algoritmo para resolver FNCX-SAT:

1. Transforme, em pares, os literais que são submetidos a disjunção exclusiva para duas cláusulas como demonstrado acima,
2. Execute o algoritmo explicitado na questão 1 para a instância de 2-SAT obtida após a transformação.

Assim, provamos que o problema FNCX-SAT está em P , dado que a transformação e o algoritmo da questão 1 pode ser executado em tempo polinomial.