

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

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-SAT  $\in$  P.

**Solução.**

A abordagem será propor um algoritmo que decide a satisfatibilidade de 2-SAT em tempo polinomial.

Estamos considerando fórmulas no formato CNF (*Conjunctive normal form*) que são compostas por cláusulas que contém apenas 2 termos e

queremos apontar a existência (ou não) de alguma combinação de valores tal que a CNF seja verdadeira (todas cláusulas sejam verdadeiras).

Caso alguma cláusula contenha apenas um termo, como expresso no exemplo do enunciado, pode-se repetir a variável única daquela cláusula sem perda de generalidade de forma a satisfazer a existência de exatamente dois literais por cláusula (Afinal a repetição de uma variável em uma disjunção não alterará o resultado lógico).

Agora, seja  $G(V,E)$  um dígrafo cujo conjunto de vértices é igual a 2 vezes o número de variáveis expressas na CNF (Para cada  $p \in CNF$ , é adicionado também um vértice para  $\neg p$  no gráfico  $G$ ). Esse grafo representará a CNF através de implicações lógicas.

A criação das arestas de  $G$  é realizada a partir da equivalência lógica

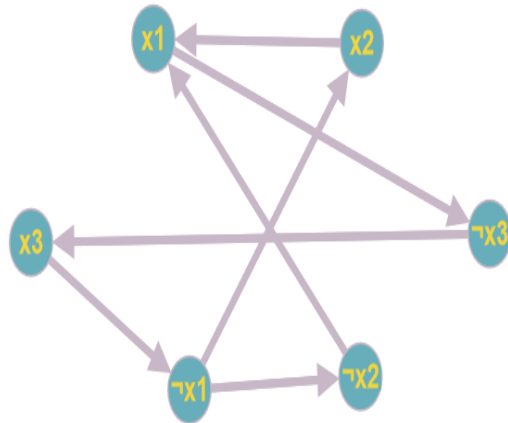
$$(p \vee q) \iff (\neg p \implies q) \wedge (\neg q \implies p)$$

onde para cada cláusula  $(p \vee q) \in CNF$ , são criadas duas arestas no gráfico  $G$ : uma ligando o vértice  $\neg p$  ao vértice  $q$  e a outra ligando o vértice  $\neg q$  ao vértice  $p$ .

Para ilustrar, a seguinte CNF (Equivalente à dada no enunciado)

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

gerará o seguinte dígrafo  $G$ :



Agora, analisemos o caso particular em que  $\exists(p, \neg p), (\neg p, p) \in E$  para algum  $p \in V$ :

- A coexistência dessas duas arestas geram uma contradição no sentido de que não é possível que ambas implicações sejam satisfeitas independente do valor de  $p$ , conforme evidenciado pela tabela verdade abaixo.

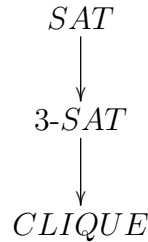
$p$	$\neg p$	$(p \implies \neg p)$	$(\neg p \implies p)$
0	1	1	0
1	0	0	1

- Como  $G$  é um dígrafo que representa a CNF através de implicações lógicas, a existência de um caminho do vértice  $p$  ao vértice  $\neg p$  possui o mesmo significado da existência de uma aresta  $(p, \neg p)$ , pela transitividade da operação de implicação.
- Assim, a coexistência dos caminhos de  $(p, \neg p)$  e  $(\neg p, p)$  no grafo  $G$  também representa uma contradição lógica, representando que a CNF é insatisfatível, nesse caso.
- Por outro lado, caso apenas um desses caminhos (ou nenhum deles) exista, a CNF é dita satisfatível, pelas tabelas verdade da operação de implicação.

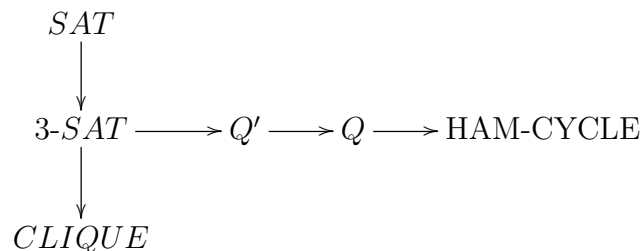
Portanto, observamos que o problema 2-SAT se reduz ao problema de verificar a existência, em um grafo  $G$ , de caminhos da forma  $(p, \neg p)$  e  $(\neg p, p)$ ,  $\forall p \in V$ , e isso pode ser realizado por algoritmos simples como Busca em Largura ou Busca em Profundidade, que rodam em tempo polinomial.

Logo, 2-SAT é decidível em tempo polinomial ( $2\text{-SAT} \in P$ ).

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 simple 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 provar que HAM-CYCLE é NP-completo, deve-se mostrar que:

- (a) HAM-CYCLE está em NP;
- (b)  $\forall A \in NP$ ,  $A$  é redutível em tempo polinomial a HAM-CYCLE.

Para (a), devemos mostrar que é possível verificar uma solução para HAM-CYCLE em tempo polinomial.

Assim, suponha uma solução proposta que retorne uma lista com os vértices na ordem que aparecem no caminho hamiltoniano. Para verificar essa solução, basta verificar se todos os vértices citados estão de fato no grafo, checar se todos eles estão conectados com o próximo vértice da lista, e confirmar se o último vértice está conectado ao primeiro. Note que essa verificação pode ser realizada em tempo polinomial. Logo, HAM-CYCLE  $\in$  NP.

Para (b), mostramos que existe uma redução de 3-SAT para HAM-CYCLE. Isto é, mostramos que através de uma solução para HAM-CYCLE pode-se resolver 3-SAT. Para computar essa redução, deve-se elaborar uma forma de converter fórmulas 3CNF em grafos em que a satisfatibilidade da CNF corresponde à existência de um ciclo hamiltoniano.

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.** Escreva aqui sua solução.

4. **(2.5 pontos)** Uma fórmula booleana em *forma normal conjuntiva com disjunção exclusiva (FNEX)* 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.**

Mostraremos que FNCX-SAT está em  $P$  propondo um algoritmo que decide FNCX-SAT e roda em tempo polinomial.

Apesar de similar às CFNs visto anteriormente, agora as conjunções operam sobre cláusulas que contém operações lógicas XOR. Para ilustrar a diferença, segue a seguinte fórmula na FNCX:

$$(a \oplus b \oplus c) \wedge (b \oplus d \oplus \neg c)$$

No entanto, o interessante está no fato que a operação  $\oplus$  é equivalente à soma em álgebra abstrata, o que nos permite realizar a transformação da fórmula FNCX em um sistema de equações lineares módulo 2. Assim, pela fórmula acima deriva-se o seguinte sistema de equações:

$$a + b + c \equiv 1 \pmod{2}$$

$$b + d + (1 + c) \equiv 1 \pmod{2}$$

Assim, o que antes era uma fórmula de satisfatibilidade lógica, agora é um sistema linear. Para apurar a existência de soluções, pode-se utilizar diversos mecanismos de álgebra linear que nos permite computar esse sistema sem grandes problemas, como o algoritmo de escalonamento (Eliminação de Gauss), por exemplo. Nesse caso particular utilizando a Eliminação de Gauss, são realizadas sucessivas operações elementares, como:

- (a) trocar duas linhas;
- (b) multiplicar os elementos de uma linha por uma constante
- (c) substituir uma linha pela sua soma com um múltiplo de outra

Observe que todas essas operações podem ser realizadas em tempo polinomial.

Logo, a Eliminação de Gauss, que é a execução sucessiva dessas operações elementares, também roda em tempo polinomial.

Dessa forma, é possível decidir FNCX-SAT em tempo polinomial através da conversão da fórmula lógica em um sistema linear módulo 2 seguido da utilização do método do escalonamento para resolver o sistema.  
(FNCX-SAT  $\in$  P)