LISTA 8 - ANALISE

Patrícia Rodrigues

December 15, 2023

70

3. Uma f ormula booleana C sobre um conjunto X de vari aveis booleanas (n ao necessariamente em CNF) e uma tautologia se toda atribui c ao a X satisfaz C. O problema tautologia consiste em, dado X e C, decidir se C e ou n ao uma tautologia. Prove que o problema tautologia est a em coNP.

Na lógica proposicional, uma tautologia é uma fórmula proposicional que é verdadeira para todas as possíveis valorações de suas variáveis proposicionais.

Para provar que o problema da tautologia está em co-NP, consideramos o complemento do problema, ou seja, o problema de decidir se uma fórmula booleana não é uma tautologia.

Dado um conjunto X de variáveis booleanas e uma fórmula booleana C, o complemento do problema da tautologia pergunta se existe alguma valoração para as variáveis em X tal que a fórmula C seja falsa. Portanto, precisamos mostrar que podemos eficientemente verificar se uma dada valoração é um contraexemplo para a não tautologia de C.

Verificação do Contraexemplo:

Dada uma valoração específica para as variáveis em X, podemos eficientemente avaliar a fórmula C. Se a fórmula C é verdadeira para essa valoração, então não é um contraexemplo, pois a fórmula é verdadeira para esses valores. Se a fórmula C é falsa, então a valoração é um contraexemplo, indicando que a fórmula C não é uma tautologia.

Eficiência em Tempo Polinomial:

O algoritmo é eficiente em tempo polinomial porque a avaliação da fórmula C pode ser realizada em um tempo proporcional ao número de operações na fórmula. Assim, o tempo de execução do algoritmo é limitado por um polinômio no tamanho da entrada.

Conclusão:

Portanto, o problema da tautologia está em co-NP, pois podemos eficientemente verificar a não tautologia de uma fórmula booleana fornecendo um contraexemplo.

Como é o verificador? Quando ele devolve S ou N?

Por que o verificador satisfaz as condições do slide 53 da aula 16?

6. Seja G = (V, E) um grafo. Um conjunto S V e independente se quaisquer dois v ertices de S n ao s ao adjacentes. Ou seja, n ao h a nenhuma aresta do grafo com as duas pontas em S. O problema IS consiste no seguinte: dado um grafo G e um inteiro k 0, existe um conjunto independente em G com k v ertices? Mostre que o problema IS e NP-completo. N ao pode usar o fato que o problema Clique e NP-completo, mas pode se inspirar na prova deste teorema.

Resposta: Para demonstrar que o problema IS pertence à classe NP, é necessário mostrar a existência de um algoritmo não determinístico capaz de verificar uma solução proposta em tempo polinomial. Em outras palavras, a verificação da validade de uma solução deve ser realizada em tempo polinomial.

Um conjunto S é considerado independente em um grafo G = (V, E) se não existirem arestas conectando quaisquer dois vértices em S, ou seja, não há adjacência entre eles. Para determinar a independência de um conjunto, é suficiente percorrer todos os vértices e verificar a presença ou ausência de adjacência entre eles, o que pode ser feito em tempo polinomial. Assim, o problema IS é classificado como NP.

O problema SAT, demonstrado como NP-completo por Stephen Cook em 1971, é utilizado como referência para estabelecer a NP-completude de outros problemas. Ao reduzir 3-SAT para IS, é possível mostrar a NP-completude do problema IS, uma vez que, se X é um problema NP-completo e X pode ser polinomialmente reduzido a Y, então Y também é NP-completo.

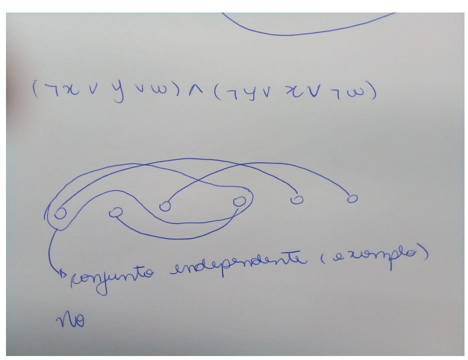
A redução de 3-SAT para IS pode ser ilustrada considerando a expressão lógica:

(¬xyw)(¬yz¬w)

Adicionamos arestas entre variáveis cuja existência simultânea é contraditória, representando o oposto do processo utilizado na redução para o problema Clique. Essa abordagem é necessária devido à restrição de não utilizar o problema Clique na demonstração.

Ao realizar essa redução, estabelecemos uma correspondência entre instâncias de 3-SAT e instâncias de IS, demonstrando assim a NP-completude do problema IS.

Teremos um grafo da seguinte maneira. Veja na figura abaixo



Nesse contexto, os elementos $\neg$x, w e $\neg$y constituem um conjunto independente. Assim, evidenciei a capacidade de reduzir um problema já estabelecido como NP-completo para IS, demonstrando, consequentemente, que IS também pertence à classe NP-completo.

0

9. Mostre que o problema abaixo e NP-completo. Problema mochila: Dado um nu mero W 0, um nu mero V 0, um nu mero inteiro n ao negativo n, uma colec ao de nu meros n ao negativos w1,...,wn, e uma cole c ao de nu meros n ao negativos v1,...,vn, decidir se existe um subconjunto S de 1,...,n tal que XwiW e XviV. iS iS Pode assumir que o problema Partic a o (Exerc ıcio 8) e NP-completo.

Resposta:

Se conseguimos realizar uma redução de 3SAT (um problema NP-completo) para SubsetSum e, subsequentemente, reduzir SubsetSum para o problema da mochila, isso implica que o problema da mochila é NP-completo. Demonstrarei essa NP-COMPLETUDE realizando uma redução de 3-SAT para o problema SUBSET e, em seguida, reduzindo para o problema da mochila.

3SAT para SUBSET SUM

3-SAT para subsetsum

Seja a fórmula $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$

	x_1	x_2	x_3	c_1	c_2
y_1	1	0	0	1	0
z_1	1	0	0	0	1
y_2	0	1	0	1	1
z_2	0	1	0	0	0
y_3	0	0	1	0	0
z_3	0	0	1	1	1
s_1	0	0	0	1	0
t_1	0	0	0	1	0
s_2	0	0	0	0	1
				0	1
t_2	0	0	0		
k	1	1	1	3	3

y_i = ocorrências positivas de x_i

z_i=ocorrências negativas

*a lista de números pode ser construída em tempo polinomial

e aí ?

Nada foi provado mesmo assim

SUBSETSUM para MOCHILA

Subset sum para mochila

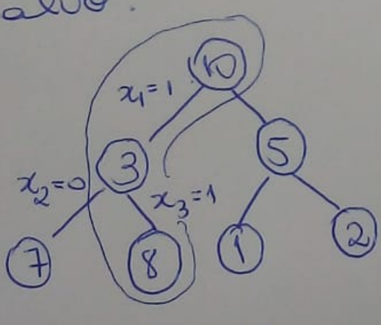
SUBSET: Backtra

$\left(\sum_{i=1}^{k} x_i \cdot w_i\right) + w_k \leq$ alvo ? ⇝ que ?? ⇝ BackTra ?? Esse não é o subset sum

$x_i = [0,1]$. x no caso funcionaria como um vetor binário. Se $x_k = \begin{cases} 1, & \text{se } \sum_{i=1}^{k} x_i \cdot w_i + w_k \leq \text{alvo} \\ 0, & \text{e.c} \end{cases}$

Caso, em algum momento do algoritmo $\sum_{i=1}^{k} x_i \cdot w_i =$ alvo $\Rightarrow$ achou um subconjunto que soma o alvo, mas caso isso nunca aconteça, ou seja, $\sum_{i=1}^{k} x_i w_i <$ alvo ou $\sum_{i=1}^{k} x_i w_i >$ alvo, em todos os caso, então o algoritimo retorna que não existe um subconjunto que de a soma alvo

Árvore que ilustra essa busca pela soma alvo:



SOMA alvo: 21

$x =$

1	0	1	

1 2 3 ...

$w =$

10	3	7	8

x_1 x_2 x_3

como?

Quem são os inputs? Como prova que dá certo?

Para transformar em problema da mochila, basta alteramos a condição para

$$\left(\sum_{i=1}^{k} x_i w_i\right) + w_k \leq W \quad \text{e} \quad \sum_{i=1}^{k} v_i \geq V$$

Portanto se $3SAT \geq_p$ SUBSET SUM $\geq_p$ MOCHILA $\Rightarrow$ MOCHILA $\in$ NP-completo