Lista 6 de CAL – Vinícius Takeo F. K.

Questão 1: Provar que SUBSET-SUM pertence a NP

Para provar que SUBSET-SUM pertence a NP, precisamos provar duas coisas:

1. A existência de um algoritmo de verificação de tempo polinomial, isto é, dado uma instancia de SUBSET-SUM chamada S, onde S = um conjunto de valores numéricos e um t, onde t corresponde a soma de elementos pertencente a S', deve-se verificar que t é uma soma valida de um subconjunto interno S' de S em tempo polinomial.
2. Reduzir 3CNF-SAT (um problema NP já bastante estudado da literatura) para SUBSET-SUM, utilizando o algoritmo apresentado no livro base da disciplina.

Para provar a etapa 1, apresenta-se um algoritmo de tempo polinomial descrito no pseudocódigo abaixo:

```
verificaSum(instancia,chave)

se chave == 0
        fim do algoritmo, retorne true

para cada elemento da instancia:
        retorno = false;
        se chave for maior que o elemento:
                retorno = chamar recursivamente subtraindo o valor do primeiro elemento
na chave e retirando o elemento da lista.
        se retorno:
                retorne true
retorne false
```

A complexidade do algoritmo pode ser calculada da seguinte forma: a complexidade base do algoritmo é O(1), pois ele itera sob os elementos do conjunto. Como cada chamada recursiva custa O(n-1), pois sempre irá haver um elemento a menos no conjunto, senão a chamada recursiva nem ocorre. Ao final a relação de recorrência é T(n) = T(n-1) + 1, cuja complexidade é O(n).

Já para a etapa seguinte, basta seguir os passos do algoritmo apresentado no livro referência da disciplina. A seguinte tabela é construída, para a instância f = C1 ^ C2 ^ C3 ^ C4 do problema 3CNF-SAT, onde C1 = (x1 ∨ ¬x2 ∨ ¬x3), C2 = (¬x1 ∨ ¬x2 ∨ ¬x3), C3 = (¬x1 ∨ ¬x2 ∨ x3) e C4 = (x1 ∨ x2 ∨ x3).

	X1	X2	X3	C1	C2	C3	C4
V1							
V1'							
V2							
V2'							
V3							

V3’							
S1							
S1’							
S2							
S2’							
S3							
S3’							
S4							
S4’							
t	1	1	1	4	4	4	4

Lembrando que Vi e Vi’ representam uma variável i, onde Vi’ representa ¬Xi = T e Vi representa Xi = T. Para isso, para cada dupla Vi e Vi’ vamos colocar um 1 na tabela, porque cada variável Xi pode assumir ou T ou F:

	X1	X2	X3	C1	C2	C3	C4
V1	1						
V1’	1						
V2		1					
V2’		1					
V3			1				
V3’			1				
S1							
S1’							
S2							
S2’							
S3							
S3’							
S4							
S4’							
t	1	1	1	4	4	4	4

Como apenas uma delas será selecionada, temos que t será 1 em cada uma das colunas, já que t representa a soma dos elementos.

Para o passo seguinte, precisamos voltar a fórmula e marcar em cada uma das colunas Ci os valores em que as variáveis Vi e Vi’ aparecem:

c	X1	X2	X3	C1	C2	C3	C4
V1	1			1			1
V1’	1				1	1	
V2		1					1
V2’		1		1	1	1	
V3			1			1	1
V3’			1	1	1		
S1							
S1’							
S2							
S2’							

S3							
S3’							
S4							
S4’							
t	1	1	1	4	4	4	4

Como para cada termo da 3CNF-SAT é possível que 2 elementos sejam falsos e 1 positivo, ou melhor, basta um ser positivo, podemos preencher a parte verde do quadro, distribuindo atribuições com o 1 e atribuindo 2 a sua variável linha, isto é, se S1 for 1, S1’ seja 2:

	X1	X2	X3	C1	C2	C3	C4
V1	1			1			1
V1’	1				1	1	
V2		1					1
V2’		1		1	1	1	
V3			1			1	1
V3’			1	1	1		
S1				1			
S1’				2			
S2					1		
S2’					2		
S3						1	
S3’						2	
S4							1
S4’							2
t	1	1	1	4	4	4	4

A partir dessa última etapa, temos os números do nosso conjunto S, tais números são os números descritos nas linhas da tabela: {2,1,20,10,200,100,2000,1000,11100,10011,101110,100001,1000110,1001001}.

Nosso t, já é definido no começo do problema: t = 1114444.

Agora, vamos definir uma designação válida para as variáveis: x1 = T, x2 = F e x3 = T. Tendo em vista essa definição, podemos agora selecionar as linhas que participarão de S’. Olharemos para cada uma das variáveis Vi e Vi’, se o valor for tal como na nossa definição, selecionaremos a linha. As linhas escolhidas são V1 (X1=T), V2’ (¬X2=T) e V3 (X3=T).

Note que as colunas X1,X2 e X3, possuem apenas um valor de 1 selecionado, logo, a soma é 1 em t para cada uma das colunas.

Para selecionarmos as linhas de Si e Si’ a serem escolhidas, precisamos encontrar somas igual a 4 para cada coluna de Ci:

	X1	X2	X3	C1	C2	C3	C4
V1	1			1			1
V1’	1				1	1	
V2		1					1
V2’		1		1	1	1	
V3			1			1	1
V3’			1	1	1		
S1				1			

S1’				2			
S2					1		
S2’					2		
S3						1	
S3’						2	
S4							1
S4’							2
t	1	1	1	4	4	4	4

Note que a soma em C1 é 2, e precisamos de 4. Logo, iremos selecionar a linha S1’, a única com valor igual a nessa coluna. Repetiremos o processo para as demais linhas. As selecionadas serão pintadas de vermelho.

	X1	X2	X3	C1	C2	C3	C4
V1	1			1			1
V1’	1				1	1	
V2		1					1
V2’		1		1	1	1	
V3			1			1	1
V3’			1	1	1		
S1				1			
S1’				2			
S2					1		
S2’					2		
S3						1	
S3’						2	
S4							1
S4’							2
t	1	1	1	4	4	4	4

Com as linhas selecionadas, temos o conjunto S’, cuja soma é t:

S’ = {1001001,101110,10011,2000,100,200,20,2}.

Com isso, reduzimos o problema de 3CNF-SAT a uma instância de SUBSET-SUM.

Questão 2: Instancia do 3CNF-SAT não satisfazível

Seja a fórmula:

(x1 V x2 V x3) ^ (~x1 V x2 V x3) ^ (x1 ^ ~x2 ^ x3) ^ (x1 V x2 V ~x3) ^ (~x1 V ~x2 V ~x3), não é possível essa fórmula ser verdadeira nunca, visto para qualquer valor de x1,x2 e x3, sempre haverá um componente que será falso, como são separados por and’s, não há um conjunto de valores que satisfaz essa fórmula.

Questão 3: Reduzir a instância de 3CNF-SAT a uma instancia de SUBSET-SUM

Utilizando a transformação apresentada no livro:

	x1	x2	x3	C1	C2	C3	C4
v1	1	0	0	1	0	0	0
v1'	1	0	0	0	1	1	1
v2	0	1	0	1	1	0	0
v2'	0	1	0	0	0	1	1
v3	0	0	1	1	1	1	0
v3'	0	0	1	0	0	0	1
s1	0	0	0	1	0	0	0
s1'	0	0	0	2	0	0	0
s2	0	0	0	0	1	0	0
s2'	0	0	0	0	2	0	0
s3	0	0	0	0	0	1	0
s3'	0	0	0	0	0	2	0
s4	0	0	0	0	0	0	1
s4'	0	0	0	0	0	0	2
t	1	1	1	4	4	4	4

O objetivo é decidir se S = {1001000,1000111,101100,100011,11110,10001,1000,2000,100,200,10,20,1,2} possui um subconjunto cuja soma é 1114444.

Questão 4: Atribuição que torna a fórmula verdadeira

Não há uma atribuição que torna a fórmula verdadeira. Ao buscar todas as combinações em uma tabela verdade, tem-se:

x1	x2	x3	C1	C2	C3	C4	Res
1	1	1	1	1	1	0	0
1	1	0	1	1	0	0	0
1	0	1	1	1	1	0	0
1	0	0	1	0	1	0	0
0	1	1	1	1	1	0	0
0	1	0	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	1	1	1	0

Logo, se o problema original não possui uma resposta, ao reduzi-lo a outro é esperado que esse outro também não possua.