1. Mostre que P é fechada sobre a complementação. Sua prova também implica que NP é fechada sobre complementação?

1) Para mostrar que P é fechada sobre complementação, vamos criar uma MT M que decide em tempo polinomial, dado uma MT Comp:

Comp: "Sobre a entrada w, faça:

1. Rode M sobre w
2. Se M aceitar, rejeite. Se M rejeitar, aceite"

- Dado que a classe NP é um conjunto de problemas que são verificados em tempo polinomial, basta achar apenas um problema para provar o pertencimento, mas negar uma solução não prova que para **todos** os problemas em NP são fechados para complementação.

2. Seja $CONEXO = \{\langle G\rangle | G \textit{ é um grafo não direcionado conexo}\}$. Esta linguagem está em P? Justifique.

2- $A = \{\langle G\rangle | G$ é um grafo não direcionado conexo$\}$

n = nº de nós
E = nº de arestas

MT M que decide A:

M = "sobre a entrada $\langle G\rangle$:

1- Selecione o 1º nó de G e marque $\rightarrow O(1)$ 5

2- Repita o seguinte estágio até que nenhum novo nó seja marcado: $\rightarrow O(n)$

3- Para cada nó em G, marque-o, se ele estiver ligado por uma aresta a um nó que já esteja marcado. $\rightarrow O(E)$

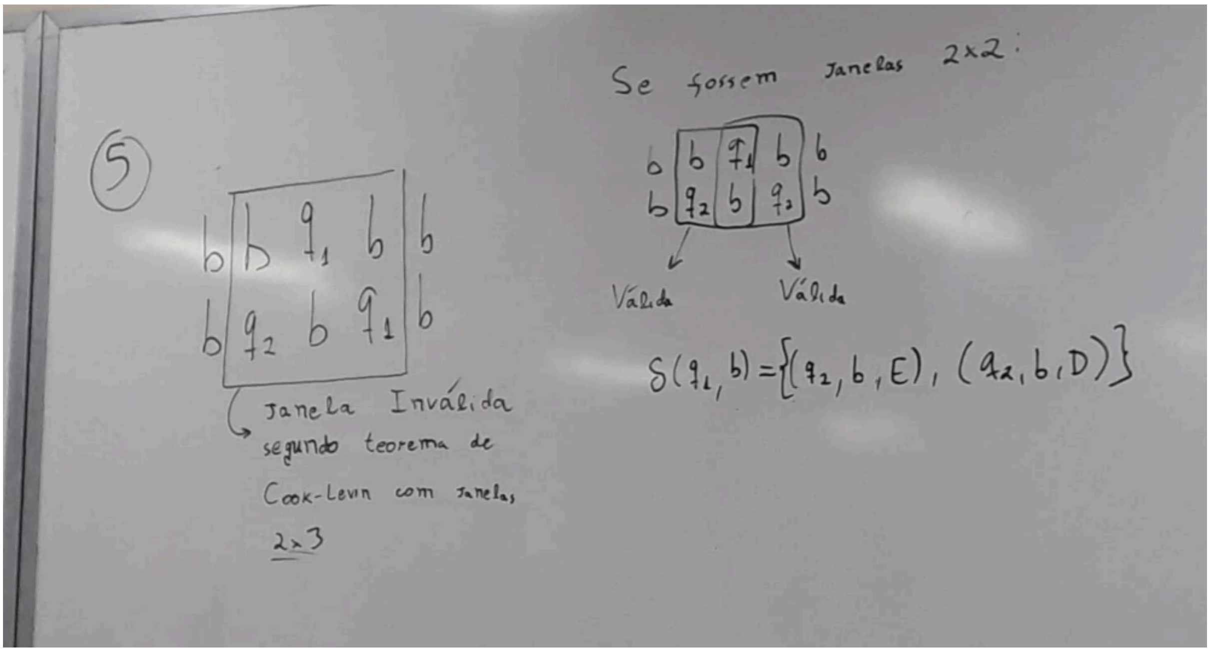
$O(n) \text{ e } O(E) \Rightarrow O(n \times E)$

4- Faça uma varredura em todos os nós de G p/ determinar se eles estão todos marcados. $\rightarrow O(n)$

5- Se sim, aceite, do contrário, rejeite" $\rightarrow O(1)$

$$O(1) + O(n \times E) + O(n) + O(1) \Rightarrow O(n \times E)$$

5. Na prova do teorema de Cook-Levin, uma janela é um retângulo 2 × 3 de células. Mostre que a prova falharia se, em vez disso, tivéssemos usado janelas 2 × 2.



Aqui, a janela 2×3 evidencia que, de acordo com Cook-Levin, a configuração é ilegal, pois contém dois estados na mesma linha. No entanto, ao utilizar janelas 2×2, essas configurações parecem legais. Isso representa um problema, pois as janelas 2×2 induzem ao erro de acreditar que o tableau está seguindo corretamente as transições da Máquina de Turing, quando, na realidade, isso não ocorre.

4. Mostre que se P = NP, existe um algoritmo de tempo polinomial que recebe um grafo não direcionado como entrada e encontra um maior clique contido no grafo.

Comment

Step 2 of 2

The following algorithm will find the largest clique in the graph:

1. Let n be the no. of nodes in the given graph G.

i be the variable which runs from 1 ton.

2. Using the polynomial time algorithm form clique, check whether there exist a clique of size i.

3. Output the Large i for which a clique exists.

To find the maximum clique, we start with i, the maximum clique size.

Remove one node and see if there is still a clique of size i.

If not, restore that node and remove another node.

If so, respect the process until we are left with a graph of i nodes, which must be a clique.

- This algorithm will take almost n trials to find which node to remove and at most n nodes to be removed.

Then the total running time is polynomial.

Comment