Nome: Nicolle Beatrice Asquino

TRABALHO – REDUÇÃO

COMPLEXIDADE DE ALGORITMOS

ENUNCIADO:

Selecione dois problemas NPs, os quais chamaremos de problema A e problema B. Prove que B também é um problema NP. Apresente todos os passos necessários para a prova, mostrando como é feita a redução e a transformação da instância de A para B e de B para A.

1. SELECIONE DOIS PROBLEMAS NPS

Problema A: 3-SAT – dada uma fórmula booleana (usando E, OU e negação), decidir se existe uma forma de atribuir verdadeiro/falso às variáveis para que a fórmula seja verdadeira.

Problema B: CLIQUE – dado um grafo G = (V, E) e inteiro k, decidir se existe um subgrafo completo (clique) de tamanho k.

2. PROVE QUE B TAMBÉM É UM PROBLEMA NP

Dado um conjunto S ⊆ V de vértices com | S | = k, o verificador checa se todo par $u, v \in S$ tem aresta $(u, v) \in E$. Há (*2 k*) pares; checar cada par é $O(1)$ se o grafo está em representação adequada (matriz de adjacência ou tabela hash). Assim verificação em tempo *O(k^2)* <= polinomial. Portanto B ∈ NP.

3. REDUÇÃO POLINOMIAL: TRANSFORMAR INSTÂNCIA A EM B.

Se soubermos resolver CLIQUE, também conseguimos resolver 3-SAT.

Passo a passo:

i. Pegue a fórmula booleana φ = C1 ∧ C2 ∧ . . . ∧ Cn
onde cada Ci é uma cláusula com até 3 literais (por exemplo, x1, ~x2)

ii. Para cada literal de cada cláusula, crie um vértice no grafo.

    a. Exemplo: se a fórmula tem 3 cláusulas com 3 literais cada → o grafo terá 9 vértices.

iii. Conecte dois vértices com uma aresta se eles vêm de cláusulas diferentes, e não são contraditórios (ou seja, não são x e ~x).

iv. Defina k = número de cláusulas.

Ou seja:

Se a fórmula é satisfeita, significa que existe uma escolha de literais que podem ser verdadeiros ao mesmo tempo. Esses literais correspondem a vértices que se conectam entre si, formando um clique de tamanho m.

Se o grafo tem um clique de tamanho m, significa que há um literal verdadeiro por cláusula e que eles não se contradizem.

Exemplo:

- D = (a V b) ^ (~a V c)

- Cláusulas:
  - C1 = (a V b)
  - C2 = (~a V c)
- Vértices
  - V1a, V1b, V2~a, V2c
- Conecte vértices de cláusulas diferentes que não são opostos:
  - V1a conecta com V2c
  - V1b conecta com V2~a e com V2c
- K = 2
  - Como existe a clique {v1b, V2c}, a fórmula é satisfazível com b = verdadeiro, c = verdadeiro.

4. REDUÇÃO POLINOMIAL: TRANSFORMAR INSTÂNCIA B EM A.

Agora temos o caminho contrário: se soubermos resolver 3-SAT, também conseguimos resolver CLIQUE.

Passo a passo:

Dado um grafo G = (V, E) o e um número k:

i. Crie variáveis booleanas $x$iv, onde:
   a. i é a “posição” dentro do clique (de 1 até k);
   b. v é o vértice do grafo.
   c. Exemplo: x23 → “na posição 2 do clique escolhi o vértice 3”.

ii. Crie cláusulas que garantem que:
   a. Cada posição tem algum vértice escolhido. Ex.: (xi1 V xi2 V . . . V $x$in)
   b. Uma posição não tem dois vértices ao mesmo tempo. Ex.: (~$x$iv, V ~$x$iv’)
   c. Um mesmo vértice não aparece em duas posições diferentes. Ex.: (~$x$iv V ~$x$jv)
   d. E, se dois vértices não têm aresta, eles não podem aparecer juntos. Ex.: (~$x$iu V ~$x$jv) quando (u,v) não é aresta.

iii. O conjunto dessas cláusulas forma uma fórmula em CNF, que é verdadeira se existe uma clique.

iv. Podemos converter para 3-SAT com técnicas padrão, adicionando variáveis auxiliares.

Ou seja
Se o grafo tem uma clique de tamanho k, é possível preencher a fórmula de modo que todas as cláusulas sejam verdadeiras.

Se a fórmula for satisfeita, isso representa exatamente uma escolha de k vértices que estão todos conectados.

Exemplo:

Seja grafo G com V = {v1, v2, v3, v4} e arestas E = {(v1, v2), (v2, v3), (v1, v3), (v3, v4)} (triângulo entre 1, 2, 3 e uma aresta 3 - 4), e k = 3.

- Variáveis: $x11, x12, x13, x14$; $x21...$; $x34$.
- Cláusulas exemplo:
  - Para i = 1: (x11 V x12 V x13 V x14) → reduziremos a 3-CNF com variáveis auxiliares.
  - Não-aresta, por exemplo (v1, v4) não existe, então para qualquer i != j teremos (~xi1 V ~xj4).
- Essa fórmula tem solução (ex.: escolher v1, v2, v3 nas posições 1, 2, 3) porque {v1, v2, v3} é clique de tamanho 3.

5. CONCLUSÃO

As duas transformações entre os problemas existem e são muito conhecidas na área de complexidade computacional:

- **3-SAT → CLIQUE (A → B) :**
  Isso significa que qualquer fórmula booleana (no formato 3-CNF) pode ser transformada, em tempo polinomial, em um grafo que representa o mesmo problema.
  Ou seja, se conseguirmos resolver o problema da clique, também conseguimos resolver o 3-SAT.
- **CLIQUE → 3-SAT (B → A) :**
  Também é possível transformar um grafo em uma fórmula booleana equivalente, de forma eficiente.
  Assim, se tivermos um algoritmo que resolve 3-SAT, ele também resolverá CLIQUE.