Universidade Federal de Roraima

Departamento de Ciência da Computação

DISCIPLINA: DCC606

LISTA 2 - Prazo de Entrega: 30/05/2019 até às 23:55h

ALUNO(A): Larissa Santos Silva NOTA:____________

ATENÇÃO: Descrever as soluções com o máximo de detalhes possível, no caso de programas (escritos em C ou C++), inclusive a forma como os testes foram feitos. Todos os artefatos (relatório, código fonte de programas, e outros) gerados para este trabalho devem ser adicionados em um repositório no site github.com com os seus pontos extras da disciplina.

1) Descreva o que é a NP-Completude.

Pertence à classe NP-Completo se: primeiro, pertence à classe NP; e em segundo, todos os problemas da classe NP puderem ser redutíveis a ele. Os problemas NP-Completos tem a interessante propriedade de que, se um problema NP-Completo tiver solução determinística polinomial, qualquer outro problema NP também tem. (Teoria da Complexidade, 2011)

Esta propriedade é posta a partir do conceito de redução polinomial. O fato de não estar provado que P=NP e acreditar-se que P ≠ NP torna a identificação de um problema na classe NP-Completa uma questão na prática muito importante, pois determina o não conhecimento de solução eficiente para o problema. Há problemas que estão numa classe “meio nebulosa” como o de decidir se duas expressões regulares são equivalentes e o de decidir se uma palavra é gerada por uma certa gramática sensível ao contexto. Esses problemas são polinomialmente reduzíveis de problemas NP-Completos, entretanto não se conhece algoritmo não determinístico de tempo polinomial que os resolva, e portanto não se sabe se pertencem ou não a P. Esses problemas são chamados NP-Difíceis. (Teoria da Complexidade, 2011)

2) Apresente 5 problemas provados ser NP-Completo, com suas respectivas referências.

- Caixeiro Viajante e NP-Completo

Um ciclo hamiltoniano em um grafo é um ciclo simples (que passa por todos os vértices uma única vez). O problema de encontrar um ciclo hamiltoniano (PCH) e um caso especial do problema do caixeiro viajante (PCV), no qual cada aresta (u, v) possui distancia c_{uv} . (Teoria da Complexidade, 2011)

O PCH foi um dos primeiros problemas a ser provado ser NP-Completo. Vamos mostrar que o PCV ∝ CH. Primeiro, mostraremos que o problema está em NP. Dada uma instancia do PCV, esta deve ser verificada em tempo polinomial. Usando a sequência de n vértices do circuito, o algoritmo de verificação confirma que a sequência contém cada vértice exatamente uma vez, totaliza os custos de arestas e verifica se a soma é no máximo k. Esse processo certamente pode ser feito em tempo polinomial. (Teoria da Complexidade, 2011)

Para provar que problema do caixeiro viajante (PCV) e NP-difícil, apresentamos uma redução polinomial do problema de encontrar um ciclo hamiltoniano (PCH) para o PCV. Seja G = (V, E) uma instância do PCH. Construímos uma instancia (em tempo polinomial) do PCV formalizando o grafo completo G′ = (V, E′), onde E′ = {(i, j) : i, j ∈ V e i ≠ j} onde a função de custo c é. (Teoria da Complexidade, 2011)

$$c(i,j) = \begin{cases} 0 \text{ se } (i,j) \in E, \\ 1 \text{ se } (i,j) \neq E. \end{cases}$$

- Soma de subconjunto:

Dados números naturais p1,..., pn e c, encontrar um subconjunto K de {1,..., n} tal que a soma dos pk para k em K seja igual a c (ou constatar que um tal K não existe). (Bartlett, s.d.)

- Caminho curto:

Dados vértices r e s de um grafo e um número d, encontrar um caminho de r as no grafo que tenha comprimento menor que d (ou constatar que não existe tal caminho). (Bartlett, s.d.)

- Clique grande:

Dado um grafo e um número k, encontrar uma clique com mais que k vértices (ou constatar que tal clique não existe). (Bartlett, s.d.)

- Ciclo longo:

Dado um grafo e um número k, encontrar um ciclo de comprimento maior que k (ou constatar que o grafo não tem tal ciclo).

3) Apresente um lauda sobre o artigo:

S. Cook, The complexity of theorem-proving procedures, Proceedings of the 3rd Symposium on the Theory of Computing, ACM, pp.151-158. 1971.

Disponivel em: https://dl.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=805047

O artigo de Stephen Cook tem como objetivo prova que qualquer problema de reconhecimento resolvido em tempo polinomial pela máquina não-determinista de Turing pode ser ‘reduzida’ a um problema de determinar se uma formula é uma tautologia ou não.

Sendo assim “reduzido” significa, grosso modo, que o primeiro problema pode ser resolvido deterministicamente em tempo polinomial, desde que haja um oráculo disponível para resolver o segundo. A partir desta noção de são definidos graus de dificuldade redutíveis, polinomiais, e é mostrado que o problema de determinar tautologia tem o mesmo grau polinomial que o problema de determinar se o primeiro de dois dado gráficos é isomorfo a um sub grafo do segundo. A seguir as definições e os teoremas provados.

Teorema 1. Se um conjunto S de cadeias é aceito por alguma máquina de Turing não determinista dentro do tempo polinomial, então S é redutível para P {DNF tautologias}.

Isso ocorre porque cada conjunto, ou seu complemento, é aceito em tempo polinomial por alguma máquina de Turing não-determinística.

Teorema 2. Os conjuntos seguintes são redutíveis para P uns aos outros em pares (e portanto cada um tem o mesmo grau de dificuldade polinomial): tautologias, DNF tautologias, D3 e sub gráficos.

Bibliografia

Bartlett, A. A. (s.d.). *Complexidade: problemas NP-completos*. Fonte: https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html#hamilton

Cook, S. A. (1971). The Complexity of Theorem-Proving Procedures. *Proceedings of the third annual ACM symposium on Theory of computing*, 151-158.

Loreto, A. B. (Janeiro de 2006). Análise da Complexidade Computacional de Problemas de Estatística Descritiva com Entradas Intervalares. p. 93. Fonte: http://repositorio.unisc.br:8080/jspui/bitstream/11624/584/1/AlineBrumLoreto.pdf

Teoria da Complexidade. (25 de Novembro de 2011). p. 5.