TEORIA DA COMPUTAÇÃO

Prova 2 - 28 de junho de 2012 - Prof. Marcus Ramos

Questão 1 (1 ponto) - Explique, com as suas próprias palavras, porque o problema do pertencimento (determinar se uma cadeia pertence ou não à uma linguagem) é decidível no caso das linguagens sensíveis ao contexto (aquelas que são aceitas por Máquinas de Turing com fita limitada).

Ele é decidível pois existe um número máximo de configurações que a Máquina de Turing com fita limitada pode assumir para uma certa cadeia de entrada. Assim, o algoritmo que decide a linguagem pode contar quantas configurações diferentes a máquina assume e, se essa quantidade for superior ao máximo, isso indica que existe um loop e consequentemente a cadeia não pertence à linguagem.

Questão 2 (2 pontos) - Em relação ao PCP:

a) Em que consiste o problema?

   Decidir se dois conjuntos de cadeias, que constituem listas chamadas de A e B, cujos elementos estão relacionados um-a-um e são identificados por números inteiros crescentes, possuem uma "solução". A solução, nesse caso, é uma seqüência de números inteiros, com eventuais repetições, de tal forma que as cadeias obtidas pela concatenação das respectivas cadeias da lista A coincide com a concatenação das correspondentes cadeias da lista B.

b) Qual a sua importância?

   Ele é usado para demonstrar a indecidibilidade de vários problemas relacionados com gramáticas e livres de contexto. Além disso, trata-se de um problema de análise combinatória que não envolve linguagens formais ou autômatos.

c) Qual a estratégia usada para demonstrar a sua indecidibilidade?

   Por redução da Linguagem Universal (L_u) para uma versão modificada do PCP (MPCP), e depois por redução de MPCP para PCP.

d) Dê exemplo de três problemas relacionados com linguagens e gramáticas livres de contexto que são demonstrados indecidíveis por redução a partir de PCP.

   (i) Determinar se uma gramática livre de contexto é ambígua, (ii) determinar se duas gramáticas livres de contexto geram a mesma linguagem; (iii) determinar se uma gramática livre de contexto gera todas as cadeias sobre o seu alfabeto de entrada.

Questão 3 (2 pontos) - Cite:

a) Duas maneiras diferentes para se provar que um problema pertence à classe P;

   (i) Apresentando uma Máquina de Turing determinística de tempo polinomial que soluciona esse problema ou (ii) apresentando uma redução de tempo polinomial do mesmo para um problema pertencente à classe *P*.

b) Duas maneiras diferentes (inclusive das duas acima) para se provar que um problema pertence à classe NP.

(i) Apresentando uma Máquina de Turing não-determinística de tempo polinomial que soluciona o problema ou (ii) apresentando um verificador de tempo polinomial para o problema.

Questão 4 (1 ponto) - Faça uma comparação entre técnica da redução usada na Teoria da Decidibilidade com a versão dessa técnica usada na Teoria da Complexidade, assim como os resultados que podem ser obtidos em cada caso.

Na decidibilidade não existe qualquer restrição em relação ao tipo de redução que é feita. Exige-se apenas que $w \in L_1 \Leftrightarrow w \in L_2$. Na complexidade, exige-se que, além disso, a redução possa ser feita em tempo polinomial em função do tamanho cadeia de entrada. No primeiro caso, ela pode ser usada para estabelecer a decidibilidade ou indecidibilidade da linguagem em questão; no segundo caso ela é usada para determinar se a linguagem pertence à classe P, mas não se presta para provar que a linguagem não pertence à essa mesma classe.

Questão 5 (1 ponto) - Existe alguma estratégia para provar que um problema pertence à $NP - P$? Em caso afirmativo, qual seria ela? Em caso negativo, quais são os melhores indícios que isso possa ser verdade?

Não se conhece tal estratégia, pois isso constitui a essência da questão *PxNP*, ainda não resolvida depois de mais de 40 anos de pesquisas. O melhor indício de que uma linguagem provavelmente não pertence à classe *P* é quando se pode caracterizá-la como sendo *NP*-completa.

Questão 6 (1 ponto) - Obter formas normais para os seguintes termos:

a) $(\lambda xy.xyy)uv$
b) $(\lambda xy.yx)(uv)zw$

A resposta está nos slides usados em sala de aula.

Questão 7 (1 ponto) - Avaliar no Cálculo Lambda, passo-a-passo, a expressão lógica $\overline{and}\ \overline{true}\ \left(\overline{or}\ \overline{false}\ (\overline{not}\ \overline{false})\right)$, mostrando o resultado final da mesma. Lembrar que:

$\overline{true} := \lambda x.\lambda y.x$
$\overline{false} := \lambda x.\lambda y.y$
$\overline{and} := \lambda x.\lambda y.xyx$
$\overline{or} := \lambda x.\lambda y.xxy$
$\overline{not} := \lambda x.\lambda y.\lambda z.xzy$

$\overline{and}\ \overline{true}\ \left(\overline{or}\ \overline{false}\ (\overline{not}\ \overline{false})\right) \rhd_\beta$
$\overline{and}\ \overline{true}\ (\overline{or}\ \overline{false}\ \overline{true}) \rhd_\beta$
$\overline{and}\ \overline{true}\ (\overline{true}) \rhd_\beta$
$\overline{true}$

Questão 8 (1 ponto) - O valor da expressão $f(n) = \sum_{i=1}^{n} i$ pode ser definido de maneira recursiva como abaixo:

$f(1) = 1$
$f(n) = f(n-1) + n$ para todo $n > 1$.

Pede-se:

a) Uma equação lambda que representa essa definição recursiva;

$$xy \ =_\beta \ \overline{if}\ \left(\overline{zero}\ (\overline{sub}\ y\ \overline{1})\right)\overline{1}\ \left(\overline{add}\ y\ \left(x\ (\overline{sub}\ y\ \overline{1})\right)\right)$$

b) Uma expressão lambda que representa a função f.

$$x := \ Y(\lambda xy.\overline{if}\ \left(\overline{zero}\ (\overline{sub}\ y\ \overline{1})\right)\overline{1}\ \left(\overline{add}\ y\ \left(x\ (\overline{sub}\ y\ \overline{1})\right)\right)$$

onde Y é um operador de ponto fixo (Turing ou Curry).

É permitido o uso identificado de termos que representam operações aritméticas e outros que se fizerem necessários.