TEORIA DA COMPUTAÇÃO

04 de julho de 2016

Prova 2

Prof. Marcus V. M. Ramos

1) (1,0 ponto) O que é uma Máquina de Turing Universal e qual a relação da mesma com os computadores modernos?

É uma Máquina de Turing M que aceita como entrada a codificação de uma outra Máquina de Turing N e a entrada w da mesma. A partir disso, M simula N com a entrada w. Se N pára e aceita, então M pára e aceita. Se N pára e rejeita, M pára e rejeita. Se N entra em loop, M também entra em loop. A linguagem da Máquina de Turing Universal é:

$$L_U = \{<N,w> \mid N \text{ aceita } w\}$$

Máquinas de Turing Universais são o modelo matemático dos computadores modernos, que lêem seus programas de meios externos e executam as instruções contidas neles. Dessa forma, ambos são genéricos e não precisam sofrer modificação quando executam programas diferentes.

2) (1,0 ponto) O que significa dizer que um problema é decidível?

É um problema para o qual existe um algoritmo (Máquina de Turing) que sempre pare com todas as instâncias do problema, não importando se afirmativas ou negativas. Corresponde à classe das linguagens recursivas.

3) (1,0 ponto) O que significa dizer que um problema é indecidível?

É um problema para o qual (i) não existe um algoritmo (Máquina de Turing) que sempre pare com todas as instâncias afirmativas do problema, ou (ii) toda e qualquer Máquina de Turing que aceite as instâncias afirmativas entra necessariamente em loop com alguma instância negativa do problema.

4) (1,0 ponto) Como se pode provar que um problema é decidível?

Basta apresentar um algoritmo que sempre pára com qualquer instância do problema, afirmativa ou negativa. Alternativamente, pode-se provar que existe uma redução do mesmo para algum problema decidível conhecido.

5) (1,0 ponto) Como se pode provar que um problema é indecidível?

Pode-se fazer uma prova por contradição, mostrando que não existe algoritmo (Máquina de Turing) capaz de decidir o problema. Pode-se ainda provar que existe uma redução de algum problema indecidível conhecido para o mesmo. Se se tratar de alguma propriedade não-trivial das linguagens recursivamente enumeráveis, pode-se usar o Teorema de Rice.

6) (1,0 ponto) Prove que a classe das linguagens não-recursivamente enumeráveis não é fechada em relação à operação de complementação.

Se uma linguagem é não-recursivamente enumerável, então todas as Máquinas de Turing que podem ser construídas para ela entram em loop com pelo menos uma instância afirmativa do problema. Quanto às instâncias negativas, pode ser que (i) todas sejam rejeitadas por alguma Máquinas de Turing, ou então que (ii) alguma instância faça todas as Máquinas de Turing entrar em loop. No primeiro caso, a complementação produz uma linguagem recursivamente enumerável. No segundo caso, a complementação produz uma linguagem não-recursivamente enumerável. Logo, a classe não é fechada em relação à operação de complementação.

7) (1,0 ponto) Prove que a relação "A reduz para B" é transitiva.

Se A reduz para B, então existe uma função f que mapeia instâncias afirmativas de A em instâncias afirmativas de B, e instâncias negativas de A em instâncias negativas de B. Se B reduz para C, então existe uma função g que mapeia instâncias afirmativas de B em instâncias afirmativas de C, e instâncias negativas de B em instâncias negativas de C. Logo, a composição $g \circ f$ (ou simplesmente $g(f(n))$) é uma redução de A para C.

8) (1,0 ponto) Prove que o problema MTLE4 é decidível:

MTLE4 = {<M> | M é uma Máquina de Turing com quatro ou menos estados}

Basta contar a quantidade de estados distintos de M. Se for menor ou igual a 4, aceitar. Senão, rejeitar.

9) (1,0 ponto) Prove que o problema NOT111 é indecidível:

NOT111 = { <M> | M é uma Máquina de Turing sobre ∑={0,1} e
L(M) não contém nenhuma sentença que contenha a subcadeia 111}

Como M é uma Máquina de Turing, segue que L é uma linguagem recursivamente enumerável. Como a propriedade "conter uma cadeia que contenha a subcadeia 111" é não-trivial, segue que o Teorema de Rice pode ser aplicado e NOT111 é indecidível. Para provar que a propriedade é não-trival, basta apresentar uma linguagem recursivamente enumerável que não contenha nenhuma cadeia que contenha a subcadeia 111 (por exemplo L_1={101}) e outra que contenha alguma cadeia que contenha a subcadeia 111 (por exemplo L_2={111}).

10) (1,0 ponto) Prove que PCP é decidível para alfabetos com um único símbolo e listas A e B com no máximo duas cadeias cada (dica: considere os comprimentos relativos das cadeias w_i e $x_{i)}$.

Basta gerar cadeias que tenham o mesmo comprimento, já que o alfabeto possui um único símbolo. Para saber se isto é possível, basta considerar:

Caso 1: Cada lista contém uma única cadeia, respectivamente w_1 e x_1. Então:

Caso 1.1: Se $w_1=x_1$, existe solução (1).

Caso 1.2: Se $w_1 \neq x_1$, não existe solução.

Caso 2: Cada lista contém duas cadeias, w_1, w_2 e x_1, x_2. Então:

Caso 2.1: Se existir i tal que $w_i = x_i$, então i (1 ou 2) é solução.

Caso 2.2: Se não existir i tal que $w_i = x_i$, então:

Caso 2.2.1: Se $|w_1| > |x_1|$ e $|w_2| > |x_2|$, então não há solução.

Caso 2.2.2: Se $|w_1| < |x_1|$ e $|w_2| < |x_2|$, então não há solução.

Caso 2.2.3: Se $|w_1| > |x_1|$ e $|w_2| < |x_2|$, então 1^n2^m, com $m=|w_1|-|x_1|$ e $n=|x_2|-|w_2|$ é solução, pois ambas as cadeias possuirão o mesmo comprimento.

Caso 2.2.4: Se $|w_1| < |x_1|$ e $|w_2| > |x_2|$, então 1^n2^m, com $m=|x_1|-|w_1|$ e $n=|w_2|-|x_2|$ é solução, pois ambas as cadeias possuirão o mesmo comprimento.

Em todos os casos, o problema tem solução. Por intermédio de um raciocínio semelhante, é possível provar que o mesmo resultado vale para instâncias PCP com listas A e B com mais de duas cadeias.