TEORIA DA COMPUTAÇÃO

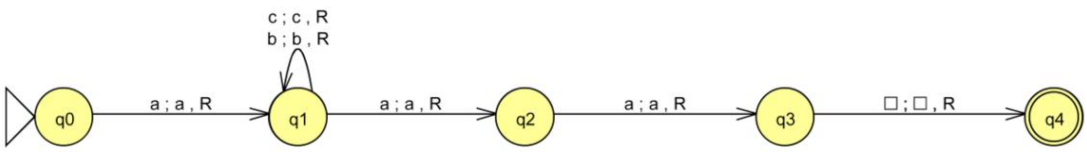
Prova 1 - 08 de maio de 2012 - Prof. Marcus Ramos

Questão 1 (1 ponto) - Conceitue:

a. Programa para uma máquina;
   Programa tal que todos os seus identificadores de operação e de teste são implementados através de funções na referida máquina;
b. Computação de um programa monolítico;
   Cadeia de elementos do tipo (r_i, v_i) onde r_i indica o rótulo da próxima instrução a ser executada e v_i o conteúdo completo da memória nessa situação; pode ser finita ou infinita;
c. Função computada por um programa.
   Computação onde o valor inicial da memória é obtido pela aplicação da função entrada, e, ao término da computação, o valor da função saída é aplicada ao conteúdo da memória.

Questão 2 (1 ponto) - Qual a relação que existe entre equivalência forte de programas e poder computacional? Justifique a sua resposta.
Nenhuma relação. A equivalência forte de programas estabelece uma relação entre programas que possuem a mesma função computada em qualquer máquina. O poder computacional, por outro lado, estabelece apenas que, para um dado programa de um certo paradigma e uma máquina, existe um outro programa em outro paradigma e uma outra máquina tal que as funções computadas coincidem. Na equivalência forte de programas, os programas recursivos são mais gerais que os monolíticos, e esse por sua vez mais gerais que os iterativos. No poder computacional os três paradigmas são equivalentes.

Questão 3 (1 ponto) - Obtenha uma Máquina de Turing que aceite a linguagem $a(b|c)^*aa$ e que sempre pare.



Questão 4 (1 ponto) - Conceitue:

a. Máquina Universal
   Máquina capaz de executar qualquer programa (ou algoritmo);
b. Máquina de Turing Universal;
   Máquina de Turing que aceita como entrada a descrição codificada de uma outra Máquina de Turing e a sua entrada, e produz como saída o mesmo resultado que a máquina simulada produziria diretamente;
c. Linguagem aceita pela Máquina de Turing Universal;
   $L_u = \{< M, w > | w \in L(M)\}$

Questão 5 (1 ponto) - Uma Máquina de Turing M aceita uma linguagem L e é tal que $REJEITA(M) = \emptyset$ e $LOOP(M) \neq \emptyset$. A linguagem L é não-recursiva? Justifique sua resposta.
Não necessariamente. O fato de existir uma Máquina de Turing que aceita L e entra em loop infinito com cadeias que não pertencem à L não implica necessariamente que L seja não-

recursiva. Para isso, seria necessário demonstrar que TODAS as máquinas que aceitam L sempre entram em loop para pelo menos uma cadeia não pertencente à mesma.

Questão 6 (1 ponto) - Prove que a classe das linguagens recursivamente enumeráveis contém a classe das linguagens recursivas.
Basta lembrar que toda linguagem recursiva é, por definição, também uma linguagem recursivamente enumerável. Ou seja, se L é uma linguagem recursiva, então L é aceita por uma Máquina de Turing que sempre pára. Portanto, L é também uma linguagem aceita por uma Máquina de Turing que sempre pára quando a cadeia pertence à linguagem e pára ou entra em loop quando a cadeia não pertence à linguagem.

Questão 7 (1 ponto) - Apresente, com suas próprias palavras, um esboço da prova de que a linguagem $L_d = \{w_i | w_i \notin L(M_i)\}$ não é recursivamente enumerável.
Trata-se de usar o argumento da diagonalização de Cantor. Considerando-se a tabela onde linhas e colunas correspondem à cadeias que representam codificações de Máquinas de Turing ordenadas lexicograficamente, a diagonal indica se uma certa máquina M_i aceita a sua própria codificação w_i. Ao construir a linguagem L_d como o complemento dessa diagonal, define-se uma linguagem que difere da linguagem aceita por cada uma das Máquinas de Turing da tabela em pelo menos uma cadeia. Logo, não existe Máquina de Turing que aceite L_d e L_d não é recursivamente enumerável.

Questão 8 (1 ponto) - Prove que o complemento de uma linguagem recursivamente enumerável porém não-recursiva não pode ser recursivamente enumerável.
Se o complemento de uma linguagem recursivamente enumerável for também recursivamente enumerável então, conforme o teorema visto em sala de aula, ambas são recursivas. Como se sabe que a linguagem original não é recursiva, segue não ser possível que o complemento da mesma seja recursivamente enumerável.

Questão 9 (1 ponto) - De que maneira a técnica da redutibilidade pode ser usada para provar que um problema de natureza desconhecida é decidível (ou indecidível)?
Para provar que ele é decidível, basta apresentar uma redução dele para um problema reconhecidamente decidível. Para provar que ele é indecidível, basta apresentar uma redução de um problema reconhecidamente indecidível para ele.

Questão 10 (1 ponto) - Prove que o problema de determinar se uma linguagem recursivamente enumerável é decidível é indecidível.
Ser decidível é uma propriedade não-trivial das linguagens recursivamente enumeráveis. Por exemplo, L_u é uma linguagem recursivamente enumerável porém não-recursiva, e $a(b|c)^*$ é uma linguagem recursivamente enumerável porém recursiva. Portanto, pelo Teorema de Rice, pode-se concluir que o problema é indecidível.