TEORA DA COMPUTAÇÃO

Prova 1 – 29/11/2022 – Prof. Marcus Ramos

1ª Questão (2 pontos): Por que é importante saber determinar se dois programas monolíticos quaisquer são fortemente equivalentes?

Porque, em caso de equivalência (ou seja, se ambos possuem a mesma função computada), então é possível selecionar o programa que melhor atende melhor os requisitos mais importantes do projeto: menor ocupação de memória, maior velocidade de execução, melhor estruturação etc.

2ª Questão (1,5 ponto): Qual a relação que existe entre as classes dos programas iterativos, monolíticos e recursivos no que se refere à equivalência forte? Como são feitas essas provas?

Programas interativos são menos gerias que os monolíticos, que por sua vez são menos gerais que os recursivos. A relação de inclusão é própria em ambos os casos. Para provar que todo programa iterativo possui um monolítico que é fortemente equivalente, basta notar que todo programa iterativo é também monolítico. Para provar que todo programa monolítico possui um recursivo que é fortemente equivalente, basta notar que é possível construir um recursivo equivalente a partir do monolítico. Em seguida, basta provar que, para uma determinada máquina e para um determinado programa monolítico, não existe iterativo equivalente. Além disso, que para uma determinada máquina e para um determinado programa recursivo, não existe monolítico equivalente.

3ª Questão (1,5 ponto): Conceitue:

- Linguagem recursiva;

Linguagem que é aceita por pelo menos uma Máquina de Turing que sempre pára com as cadeias da linguagem, rejeitando todas as cadeias que não pertencem à linguagem.

- Linguagem recursivamente enumerável;

Linguagem que é aceita por pelo menos uma Máquina de Turing que sempre para com as cadeias da linguagem, mas que pode rejeitar ou entrar em loop com as cadeias que não pertencem à linguagem.

- Linguagem recursivamente enumerável e não-recursiva.

Linguagem para a qual todas as Máquinas de Turing que a aceitam sempre entram em loop com pelo menos uma cadeia que não pertence à linguagem.

4ª Questão (1,5 ponto): Quais as diferenças entre um Autômato Finito e uma Máquina de Turing?

O AF tem fita de entrada com comprimento limitado, restrito à cadeia que será analisada. A MT possui uma fita com comprimento ilimitado, na qual a cadeia que será analisada é escrita. O restante das posições é preenchido inicialmente com brancos, e pode ser usado como rascunho. No AF o cursor serve apenas para leitura e se desloca apenas para a direita. Na MT o cursor serve tanto para leitura quanto para e escrita de símbolos, e pode se deslocar tanto para a esquerda quanto para a direita.

5ª Questão (2 pontos): Suponha que você projetou uma máquina M e deseja provar que ela é universal. Como você faria isso:

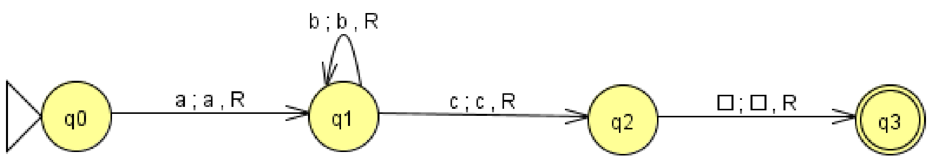
- Por meio de evidências internas?

Mostrando que, a partir das operações e testes primitivos de M, é possível criar programas que implementam operações e testes mais sofisticados, até o ponto em que fica claro que os mesmos podem ser usados para implementar qualquer algoritmo.

- Por meio de evidências externas?

Provando que M simula alguma outra máquina universal, e vice-versa.

6ª Questão (2 pontos): Obtenha uma Máquina de Turing M sobre o alfabeto $\Sigma = \{a, b, c\}$ tal que $ACEITA\ (M) = L = ab^*c$, $REJEITA\ (M) = \Sigma^* - L$ e $LOOP\ (M) = \emptyset$. Utilize o critério de aceitação "Estado final". Escolha uma sentença da linguagem e mostre a sequência de movimentos que levam à aceitação da mesma.



Considere a sentença $abbc$. Então: $(\varepsilon, q_0, abbc\beta \dots) \vdash (a, q_1, bbc\beta \dots) \vdash (ab, q_1, bc\beta \dots) \vdash (abb, q_1, c\beta \dots) \vdash (abbc, q_2, \beta \dots) \vdash (abbc\beta, q_3, \dots)$.