TEORIA DA COMPUTAÇÃO

25 de maio de 2016

Prova 1

Prof. Marcus V. M. Ramos

1) (2,4 ponto) Defina:

   a. "Programa".

   Conjunto de instruções que definem as operações e testes que devem ser executados, e em qual ordem.

   b. "Estrutura de Controle".

   Instruções que definem a ordem em que as operações e testes de um programa devem ser executados.

   c. "Máquina".

   Dispositivo no qual é executado um programa. Define o significado associado aos identificadores de operação e teste usados no progrma. Possui armazenamento e capacidade de ler e enviar informações para o meio externo.

   d. "Programa para uma Máquina".

   P é um programa para uma máquina M se todos os identificadores de operação e teste usados em P estiverem definidos em M através das correspondentes funções.

   e. "Computação".

   Seqüência de estados assumidos por uma máquina durante a execução de um programa. Cada elemento dessa seqüência é uma dupla, cujo segundo componente representa o conteúdo completo de memória da máquina. O primeiro componente depende do tipo de programa: se monolítico, é o rótulo da próxima instrução a ser executada; se iterativo é o restante do programa a ser executado; se recursivo, é o restante da expressão a ser avaliada.

   f. "Função Computada".

   Função que associa valores de entrada com valores de saída produzidos pela execução de um programa em uma máquina.

   g. "Equivalência Forte de Programas".

   Dois programas que possuem a mesma função computada em qualquer máquina.

   h. "Máquinas Equivalentes".

   Quando uma máquina pode simular a outra e vice-versa.

2) (0,6 ponto) Como se relacionam as classes dos programas iterativos, recursivos e monolíticos no que se refere à equivalência forte de programas?

A classe dos programas iterativos está contida na classe dos programas monolíticos, que por sua vez está contida na classe dos programas recursivos. As relações de inclusão são próprias. Em outras palavras, todo programa iterativo possui um monolítico que é fortemente equivalente, todo programa monolítico possui um recursivo que é fortemente equivalente, nem todo programa monolítico possui um iterativo que seja fortemente equivalente e nem todo programa recursivo possui um monolítico que seja fortemente equivalente.

3) (1,0 ponto) Descreva, com suas próprias palavras, a estratégia estudada para verificar a equivalência forte de programas monolíticos (com um único identificador de teste).

Deve-se:

- Converter os programas monolíticos para programas monolíticos com instruções rotuladas compostas;
- Simplificar os ciclos infinitos dos programas;
- Fazer a união disjunta dos conjuntos de instruções;
- Verificar se os rótulos iniciais dos programas são fortemente equivalentes. Para isso, deve-se:
    - Verificar se eles são consistentes;
    - Obter os pares sucessores;
    - Repetir os dois passos acima até que não existam novos pares ou surjam inconsistências;

      Os rótulos serão fortemente equivalentes se não houverem novos pares e todos os anteriores forem consistentes; os programas serão fortemente equivalentes se os rótulos forem fortemente equivalentes.

4) (0,6 ponto) Defina “Máquina Universal”.

Máquina que permite a representação de qualquer algoritmo na forma de um programa para a mesma.

5) (1,0 ponto) Descreva, com exemplos, duas estratégias através das quais é possível caracterizar uma máquina como sendo universal.

Evidências internas: extensões ou variações dos recursos primitivos da máquina não aumentam o seu poder computacional e podem ser usados para representar qualquer algoritmo. Exemplo: definição de macros para a Máquina Norma.

Evidências externas: provas bidirecionais de equivalência com outros dispositivos considerados universais. Exemplos: equivalência da Máquina Norma com Dois Registradores com a Máquina de Turing, e da Máquina de Post com a Máquina de Turing.

6) (1,0 ponto) Em que consiste e qual a importância:

   a. Da Hipótese de Church?

   Ela estabelece a equivalência entre a noção (informal) de algoritmo e a Máquina de Turing, declarando que esta última é capaz de representar qualquer algoritmo. A Hipótese permite que algoritmos sejam tratados matematicamente, revelando propriedades que são fundamentais para a Teoria da Computação.

   b. Do Teorema Fundamental da Aritmética?

Permite representar qualquer número como o produto de um conjunto finito de números primos. Sua importância reside no fato de que ele possibilita representar objetos variados (ênuplas, vetores, programas etc) de forma unívoca na forma de um número composto, reduzindo o espaço dos problemas ao universo dos números naturais.

7) (1,2 ponto) Defina:
   a. Cada uma das três linguagens associadas a uma mesma Máquina de Turing qualquer.

ACEITA(M): conjunto das cadeias que levam a Máquina de Turing até a parada numa configuração final;

REJEITA(M): conjunto das cadeias que levam a Máquina de Turing até a parada numa configuração não-final;

LOOP(M): conjunto das cadeias que levam a Máquina de Turing a um loop infinito;

   b. "Linguagem Recursivamente Enumerável".

Linguagem que é aceita por uma Máquina de Turing. Cadeias não pertencentes à linguagem podem levá-la à rejeição ou loop.

   c. "Linguagem Recursiva".

Linguagem que é aceita por uma Máquina de Turing que sempre pára, não importando se a cadeia pertence ou não à linguagem por ela definida.

   d. "Linguagem Recursivamente Enumerável não-Recursiva".

Linguagem que é aceita apenas por Máquinas de Turing que entram em loop com pelo menos uma cadeia que não pertence à linguagem por ela aceita.

8) (0,6 ponto) Defina:
   a. "Função Computável".

Função que, ao ser computada por uma Máquina de Turing, pára quando ela é definida para o(s) argumento(s) e pode entrar em loop quando não é definida (função parcial).

   b. "Função Computável Total".

Função que, ao ser computada por uma Máquina de Turing, sempre pára para qualquer entrada (função total).

9) (1,0 ponto) Descreva o poder computacional da Máquina com Pilhas conforme o número de pilhas que ela possui.

Conforme abaixo:

- 0 pilhas: reconhece a classe das linguagens regulares (autômato finito);
- 1 pilha: reconhece a classe das linguagens livres de contexto (autômato de pilha);
- 2 pilhas: reconhece a classe das linguagens recursivamente enumeráveis (Máquina de Turing);
- 3 ou mais pilhas: idem à máquina com duas pilhas.

10) (0,6 ponto) Além de possuirem o mesmo poder computacional, a Máquina de Turing e o Autômato com Duas Pilhas exibem uma característica em comum que as distingue de outros dispositivos, como é o caso da Máquina Norma, da Máquina de Post e da Máquina com Pilhas. Que característica é esta?

Nos dois primeiros casos, o programa a ser executado faz parte da definição da máquina, através da especificação das suas transições. Nos demais, o programa é representado separadamente através de um fluxograma ou então de um texto com sintaxe própria.