TEORIA DA COMPUTAÇÃO

Prova 1 – 15/02/2017

Prof. Marcus Ramos

1. (1,0 ponto) Sem entrar em detalhes, apresente um esboço (ou seja, a ideia e as linhas gerais) das provas dos seguintes teoremas:

   a. Todo programa iterativo possui um programa monolítico que é fortemente equivalente;

Basta notar que todo programa iterativo é também um programa monolítico.

   b. Todo programa monolítico possui um programa recursivo que é fortemente equivalente;

Basta codificar o programa monolítico na forma de um programa recursivo, de tal forma que cada instrução do primeiro corresponde à um subprograma do segundo.

   c. Nem todo programa monolítico possui um programa iterativo que é fortemente equivalente;

Através de contra-exemplo: o programa monolítico que duplica a entrada na Máquina de Um Registrador não possui um iterativo que lhe seja equivalente nesta mesma máquina.

   d. Nem todo programa recursivo possui um programa monolítico que é fortemente equivalente.

Através de contra-exemplo: o programa recursivo que determina se um número é par ou ímpar na Máquina de Um Registrador não possui um monolítico que lhe seja equivalente nesta mesma máquina.

2. (1,0 ponto) Quais são as condições que precisam ser satisfeitas para se poder afirmar que uma máquina N simula uma outra máquina M?

É necessário que, para todo programa P de M,

- Exista um programa Q de N;
- Exista uma função de codificação da entrada;
- Exista uma função de decodificação da saída;

De tal modo que as funções computadas coincidam.

3. (1,0 ponto) Os programas M_1 e M_2 a seguir são fortemente equivalentes (exercício 3.3 do livro do Menezes)? Justifique a sua resposta aplicando o método de verificação da equivalência forte de programas.

M_1

```
1: faça F vá_para 2
2: se T então vá_para 3 senão vá_para 5
3: faça G vá_para 4
4: se T então vá_para 1 senão vá_para 0
```

```
5: faça F vá_para 6
6: se T então vá_para 7 senão vá_para 2
7: faça G vá_para 8
8: se T então vá_para 6 senão vá_para 0
```

M_2

```
1: faça F vá_para 2
2: se T então vá_para 3 senão vá_para 1
3: faça G vá_para 4
4: se T então vá_para 1 senão vá_para 0
```

M_1

```
1: (F,2)(F,2)
2: (G,3)(F,4)
3: (F,2)(Término,ε)
4: (G,5)(F,4)
5: (G,5)(Término,ε)
```

M_2

```
1: (F,2)(F,2)
2: (G,3)(F,2)
3: (F,2)(Término,ε)
```

M

```
1: (F,2)(F,2)
2: (G,3)(F,4)
3: (F,2)(Término,ε)
4: (G,5)(F,4)
5: (G,5)(Término,ε)
6: (F,7)(F,7)
7: (G,8)(F,7)
8: (F,7)(Término,ε)
```

$B_0=\{(1,6)\}$
$B_1=\{(2,7)\}$
$B_2=\{(3,8),(4,7)\}$
$B_3=\{(5,8)\}$

Os programas não são fortemente equivalentes.

4. (1,0 ponto) Em que consiste a Hipótese de Church e quais as principais conseqüências da sua adoção?

A Hipótese de Church afirma que tudo que pode ser computado por ser computado por uma Máquina de Turing. A principal conseqüência é que ela permitiu que a Máquina de Turing fosse adotada como representação formal da noção de algoritmo, possibilitando a obtenção de uma série de resultados teóricos sobre os limites e as possibilidades da computação.

5. (1,0 ponto) O que é uma Máquina Universal e como se pode determinar se uma máquina qualquer é ou não universal?

Uma Máquina Universal é uma máquina para a qual se pode implementar qualquer algoritmo na forma de um programa para a mesma. A determinação pode ser feita através de evidências internas ou externas. No primeiro caso, mostrando que virtualmente qualquer nova operação e teste pode ser implementado através de operações e testes primitivos, ou então operações e testes implementados previamente. No segundo caso, mostrando a equivalência dela com outra máquina já aceita como sendo universal.

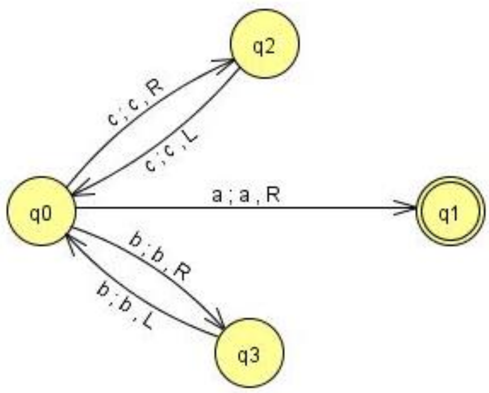
6. (1,0 ponto) O que afirma o Teorema Fundamental da Aritmética e qual a importância do mesmo?

Ele afirma que todo número natural correspondente ao produto de um conjunto de finito de numeros primos. A importância reside no fato de permitir representar virtualmente qualquer informação de forma unívoca através de um único número natural.

7. (1,0 ponto) Descreva uma estratégia que pode ser usada para provar que a Máquina Norma é uma Máquina Universal.

Fora vistas duas estratégias em sala de aula. Na primeira, usando evidências internas, foram construídas macros que implementavam testes e operações sucessivamente mais complexas, na forma de programas monolíticos para a mesma. Na segunda, foi provada que Máquina Norma e Máquina de Turing podem se simular mutuamente.

8. (1,0 ponto) Considere a Máquina de Turing M apresentada a seguir (critério de aceitação "Entrada") e ∑={a,b,c}:



   a. Determine ACEITA (M);

a(a|b|c)*

   b. Determine REJEITA (M);

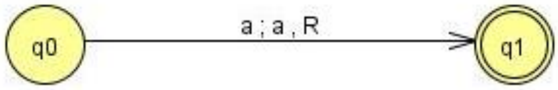
b|ba|bc|c|ca|cb

   c. Determine LOOP (M);

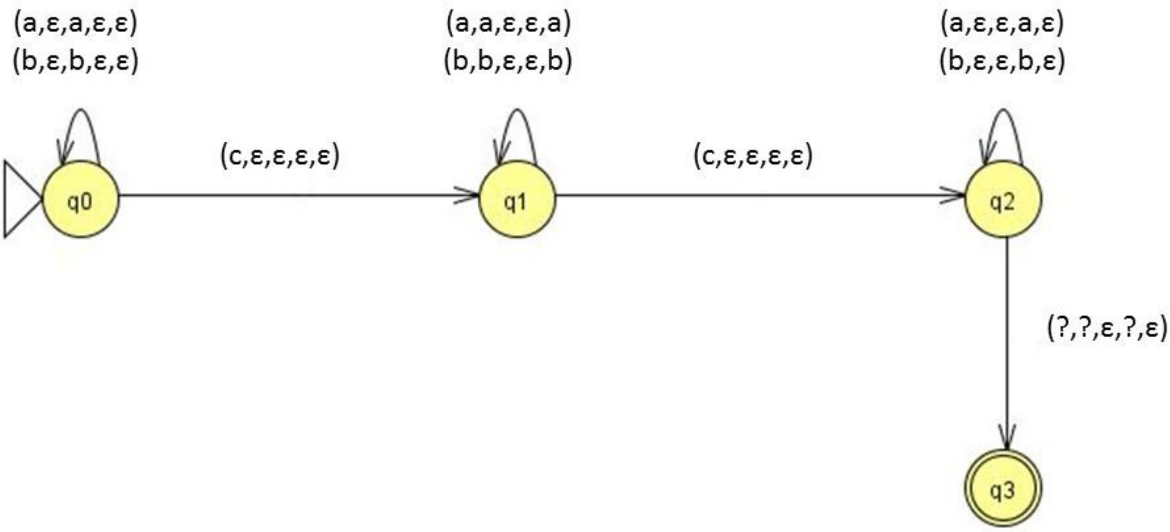
bb(a|b|c)*|cc(a|b|c)*

   d. ACEITA (M) é recursiva, recursivamente enumerável não-recursiva ou não-recursivamente enumerável? Justifique a sua resposta.

A linguagem aceita por esta máquina é regular (a(a|b|c)*) e portanto recursiva. A Máquina de Turing abaixo aceita esta mesma linguagem e sempre pára:



9. (1,0 ponto) Obtenha um Autômato com Duas Pilhas que aceite a linguagem $\{wcw^Rcw \mid w \in \{a,b\}^+\}$.



10. (1,0 ponto) Descreva, de forma concisa e com as suas próprias palavras, a estratégia usada para obter uma Máquina de Turing determinística a partir de outra não-determinística porém equivalente.

Essencialmente, deve-se construir uma Máquina de Turing determinística que simula os movimentos da versão não-determinística, através da geração e análise das suas configurações a partir de uma certa entrada. A análise destas configurações, no entanto, deve ser feita em largura (e não em profundidade) para evitar entrar em loop desnecessariamente. Sempre que a máquina original produzir uma configuração final, a nova máquna pára e aceita. Sempre que a máquina original não produzir nenhuma nova configuração, a nova máquina pára e rejeita. E sempre que a máquina original entrar em loop, a nova máquina, por construção, também fará o mesmo.