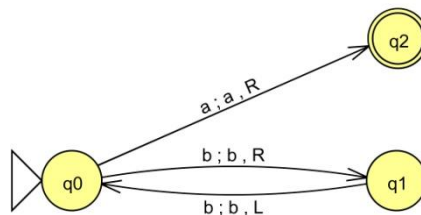


## Teoria da Computação

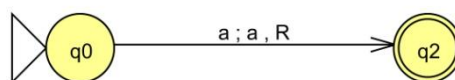
Prova 1 - 29/04/2010 - Prof. Marcus Ramos - UNIVASF

1. (1 ponto) Conceitue:
  - a. Programa  
*Conjunto de instruções que determinam a ordem em que operações e testes são executados.*
  - b. Máquina  
*Interpretador para as operações e testes referenciados num programa.*
  - c. Máquina universal  
*Máquina que permite representar qualquer algoritmo na forma de um programa para a mesma.*
  - d. Computação  
*Seqüência de estados de representam a execução de um programa numa certa máquina. A definição de "estado" depende do tipo de programa considerado.*
  - e. Função computada  
*Função parcial que representa o mapeamento de valores do conjunto de entrada no conjunto de valores de saída realizado por um certo programa.*
  - f. Programas equivalentes em uma máquina  
*Programas que possuem a mesma função computada nessa máquina.*
  - g. Equivalência forte de programas  
*Programas que possuem a mesma função computada em qualquer máquina.*
  - h. Máquinas equivalentes  
*Máquinas que podem simular uma à outra.*
  
2. (1 ponto) Responda:
  - a. Qual a importância prática de se avaliar a equivalência forte de programas?  
*Ela permite analisar programas do ponto de vista da sua complexidade estrutural, ou outros critérios considerados relevantes (legibilidade etc), uma vez que do ponto de vista funcional eles são equivalentes.*
  - b. Dado um programa P qualquer (iterativo, monolítico ou recursivo), é sempre possível obter um outro programa P' (iterativo, monolítico ou recursivo) tal que  $P \equiv P'$ ? Justifique a sua resposta.  
*No caso geral, não. É possível demonstrar que existem programas monolíticos que não possuem iterativos que sejam fortemente equivalentes, da mesma forma que existem programas recursivos que não possuem monolíticos que sejam fortemente equivalentes. A menos desses casos, a resposta é afirmativa.*
  
3. (1 ponto) Em que consiste a Hipótese de Church? Por quê ela não pode ser demonstrada formalmente?  
*Ela estabelece a equivalência entre a noção intuitiva de algoritmo e a Máquina de Turing. Não pode ser demonstrada formalmente pois o conceito de algoritmo é informal.*
  
4. (1 ponto) Em que consiste o Teorema Fundamental da Aritmética? Qual a sua importância no estudo das máquinas universais?  
*Ele permite mapear ênuplas de números inteiros positivos em um único número inteiro positivo, de forma unívoca. Ele serve, por exemplo, para permitir a representação de vetores em máquinas de registradores ou ainda para codificar programas.*

5. (1 ponto) Descreva as principais características e modo de funcionamento das seguintes máquinas:
- Máquina Norma  
*Máquina com uma quantidade ilimitada de registradores. Cada registrador armazena um número natural. As operações em cada registrador são incremento, decremento e teste se zero.*
  - Máquina de Post  
*Máquina que manipula uma fila. Símbolos são lidos e removidos do início da fila. Inserções são feitas sempre no final da fila.*
  - Máquina com Pilhas  
*Máquina que manipula um conjunto de pilhas. Símbolos são lidos e removidos do topo de cada pilha. Inserções são feitas sempre no topo de cada pilha.*
  - Autômato com Duas Pilhas  
*Como a Máquina com Pilhas, porém com duas pilhas. Ao invés de um programa, são usados estados para estabelecer o fluxo de execução.*
6. (1 ponto) Suponha que M seja um novo modelo de máquina que está sendo proposto para o estudo da teoria da computação. Que tipo de verificação você poderia fazer para demonstrar que M possui status de máquina universal?  
*A melhor opção seria tentar demonstrar que (i) M pode simular qualquer Máquina de Turing e (ii) que qualquer Máquina de Turing pode simular M. Outra alternativa seria fazer uma verificação exaustiva das possibilidades de simulação das principais características de linguagens de programação em M.*
7. (1 ponto) Considere a Máquina de Turing abaixo:



- Qual a linguagem aceita por ela (considere  $\Sigma=\{a,b,c\}$ )?  
 $a(a|b|c)^*$
- Essa linguagem é recursivamente enumerável e/ou recursiva? Prove a sua resposta.  
*A máquina acima aceita essa linguagem, apesar de ela entrar em loop com a entrada "bb...". Logo, ela é recursivamente enumerável. A máquina abaixo aceita a mesma linguagem e pára com qualquer entrada, inclusive "bb...". Logo, ela é recursiva também.*



8. (1 ponto) Quais são as classes de linguagens que são aceitas por Máquinas com Pilhas, com respectivamente 0, 1, 2 ou mais pilhas?
- 0 pilhas: linguagens regulares*  
*1 pilha: linguagens livres de contexto*  
*2 ou mais pilhas: linguagens recursivamente enumeráveis*

9. (1 ponto) Descreva o comportamento geral de uma Máquina de Turing determinística que é construída para simular uma outra Máquina de Turing não-determinística. *O método geral de conversão produz uma máquina determinística que reproduz e analisa as configurações da máquina não-determinística original. A árvore de caminhos que mostra todas as possibilidades de movimentação da máquina não-determinística é reproduzida e analisada nível por nível, a partir da raiz (configuração inicial), e sempre em largura primeiro.*
10. (1 ponto) Obtenha uma Máquina de Turing que reconheça a linguagem  $\{a^i b^j \mid (i > 1) \wedge (j \geq i) \wedge (j \text{ é par})\}$ .

