TEORIA DA COMPUTAÇÃO

Prova 2 – 29/08/2017 – Prof. Marcus Ramos

1ª Questão (1,5 ponto): Descreva, em linhas gerais, como é feita a representação e a manipulação da fita de entrada de uma Máquina de Turing em um Autômato com Duas Pilhas que simula o mesmo.

- A primeira pilha representa o conteúdo da fita de entrada situado do lado esquerdo da cabeça de leitura/escrita (símbolo mais à direita no topo);
- A segunda pilha representa o conteúdo da fita de entrada situado do lado direito e sob a cabeça de leitura/escrita (símbolo mais à esquerda no topo);
- Movimentações da MT para a direita removem o símbolo do topo da segunda pilha e inserem o novo símbolo no topo da primeira pilha;
- Movimentações para a esquerda substituem o símbolo no topo da segunda pilha pelo novo símbolo, e além disso também transferem o símbolo situado no topo da primeira pilha para o topo da segunda pilha.

2ª Questão (1,0 ponto): Considere o modelo da Máquina de Turing determinística e com fita ilimitada em ambos os sentidos. Cite 4 extensões/restrições deste modelo que não afetam o seu poder computacional.

- Não-determinismo;
- Múltiplas trilhas;
- Múltiplas fitas de entrada;
- Fita limitada à esquerda (ou direita).

3ª Questão (1,5 ponto): Considere a árvore de movimentações exibida por uma Máquina de Turing não-determinística M_1 com uma certa cadeia de entrada. Explique por que a varredura desta árvore, por uma Máquina de Turing determinística M_2 que pretende simular M_1, deve ser feita em largura e não em profundidade.

Porque se a pesquisa for feita em profundidade e M_1 entrar em loop com a cadeia entrada em algum caminho, então M_2 também poderá entrar em loop e talvez não seja possível concluir a simulação, mesmo que a cadeia pertença à ACEITA (M_1). Ao contrário, se a pesquisa for feita em profundidade, então M_2 só entrará em loop se a cadeia pertencer à LOOP (M_1). Em todos os outros casos, M_2 pára e aceita ou pára e rejeita, conforme a cadeia de entrada seja aceita ou rejeitada por M_1.

4ª Questão (1,5 ponto): Prove que a linguagem L_u é recursivamente enumerável.

$L_u = \{< M, w >|\ M \text{ é } uma\ MT\ e\ M\ aceita\ a\ entrada\ w\}$

L_u é aceita pela Máquina de Turing Universal U. Ou seja, U aceita todas as cadeias $< M, w >$ tais que $< M, w > \in L_u$. Para isto, basta U simular M com a entrada w: Se M aceita w, então U

aceita $< M, w >$. Se M rejeita w, então U rejeita $< M, w >$. Se M entra em loop w, então U entra em loop com $< M, w >$. Logo, U aceita L_u.

5ª Questão (1,5 ponto): Descreva, de forma concisa e objetiva, como é feita a redução de L_u para $PARA_{MT}$ usada na prova da indecidibilidade desta última.

$L_u = \{< M, w > | M \text{ é } uma\ MT\ e\ M\ aceita\ a\ entrada\ w\}$
$PARA_{MT} = \{< M', w > | M' \text{ é } uma\ MT\ e\ M'\ \text{pára}\ com\ a\ entrada\ w\}$

Construímos uma MT M' que, inicialmente, a partir da entrada w, simula M com w:

- Se M aceita w, então M' pára;
- Se M rejeita w, então M' entra em loop.

Além disso, é fácil perceber que, se M entra em loop com w, então M' também o fará. Logo, M' pára com a entrada w se e somente se M aceita w e temos uma redução de L_u para $PARA_{MT}$.

6ª Questão (1,5 ponto): Considere o problema de determinar se uma linguagem recursivamente enumerável qualquer não é regular. Este problema é decidível ou indecidível? Prove a sua resposta.

O problema é indecidível e pode ser prova com auxílio do Teorema de Rice. Segundo ele, qualquer propriedade não-trivial das linguagens recursivamente enumeráveis é indecidível. Para provar que "não ser regular" é uma propriedade não-trivial, basta considerar as linguagens:

- $a^n b^n, n \geq 0$
- $a^* b^*$

A primeira é recursivamente enumerável e não-regular. A segunda é recursivamente enumerável e regular. Portanto, a propriedade "não ser regular" é não-trivial e o problema é indecidível.

7ª Questão (1,5 ponto): O problema de aceitação de uma cadeia qualquer é indecidível para Máquinas de Turing e decidível para Autômatos Linearmente Limitados. Você concorda com essa afirmação? Justifique a sua resposta.

Sim. O fato de o ALL possuir uma quantidade finita de configurações para uma certa cadeia de entrada permite determinar se ele está ou não em loop durante o processamento da mesma. Desta forma, é possível saber se a cadeia pertence ou não à linguagem aceita pelo dispositivo, em qualquer situação (aceitação, rejeição ou loop). Como a MT possui uma quantidade infinita de configurações para uma mesma cadeia de entrada, esta verificação não pode ser feita.