P1 - Teoria da Computação

1. Seja EQ_{AFD} = {<A,B>|| A e B sÃo AFDs e L(A) = L(B)}. Prove que EQ_{AFD} é decidível.

Vacuidade

V_{AFD} = {<A> | A é um AFD e L(A) = Ø

Teorema: V_{AFD} é uma linguagem decidível

Prova:

T = "Sobre uma entrada <A>, onde A é um AFD "

1. Marque o estado inicial de A;
2. Até que nenhum estado novo seja marcado repita:

   2.1 Marque qualquer estado que receba uma transição de um estado já marcado
3. Se nenhum estado de aceitação estiver marcado, aceite, do contrário rejeite.

Teorema: EQ_{AFD} é uma linguagem decidível

I = "Sobre a cadeia de entrada <A,B>, onde A e B são AFDs"

1. Construa um AFD C, tal que C aceite somente cadeia aceitas por A ou por B mas não por ambas.
2. Aplique o teorema da vacuidade;
3. Se a MT aceitar, então aceite. Do contrário, rejeite.

2. O que diz a Tese de Church-Turing? Existe prova formal para ela?

Segundo a Tese de Church-Turing, toda 'função que seria naturalmente considerada computável' pode ser computada por uma Máquina de Turing, ou seja, todo problema possível de ser computável pode ser traduzido em uma Máquina de Turing e, consequentemente, toda a Máquina de Turing pode ser traduzida em uma linguagem de propósito geral. Portanto, a Tese de Church-Turing equivale a dizer que todo algoritmo possa ser expresso por uma linguagem de propósito geral. No entanto, tal tese não pode ser totalmente comprovada, mas pode ser refutada caso haja uma máquina com um poder de expressão maior do que a conseguida através da Máquina de Turing.

3. Descreva uma máquina de Turing que receba como entrada uma string w ∈ {1}*, que representa um número inteiro não negativo, representado em notação unária, e produza como sada na ta, ao nal da computação, o resto da divisão desse número por 2, também escrito em representação unária. Descreva o programa dessa MT usando pseudocódigo.

M = "Sobre cada entrada w ∈ {1}*

1. Se inicialmente a entrada da fita contém branco então aceita e encerra a execução.
2. Marque o primeiro símbolo;
3. Enquanto a fita ler '1' avança a cabeça da fita para a direita;
4. Ao ler um caractere branco(fim da fita), retorna (esquerda) sobrescrevendo cada símbolo '1' por branco;
5. Ao chegar no caractere marcado (1º), desmarca e aceita a entrada.

4. Se L1 e L2 são linguagens recursivamente enumeráveis, então L1 U L2 pode ser recursivamente enumerável ou não. Justifique.

Se duas linguagens (L1 e L2) são recursivamente enumeráveis, então elas estão fechadas nas seguintes propriedades:

L1 U L2 é recursivamente enumerável

L1 ∩ L2 é recursivamente enumerável

L1 concatenado L2 também é recursivamente enumerável

5. Toda linguagem livre de contexto é decidível?

Toda linguagem livre de contexto é turing-reconhecível, mas não decidível pois pode entrar em loops como: S → aSb

6. Linguagem regular, linguagem livre de contexto e linguagem recursiva são todas recursivamente enumeráveis?

Sim, pois é possível construir uma MT que reconheça tais linguagens.