Linguagens Formais e Autômatos

Lista de Exercícios Sobre Computabilidade e Gramáticas e Linguagens Livres de Contexto, Sensíveis ao Contexto, Recursivas e Recursivamente Enumeráveis

30 de junho de 2003

Prof. Mauricio Ayala Rincón

Entrega: antes do 09 de julho de 2003.

1. Construa gramáticas irrestritas para gerar as seguintes linguagens:
   (a) $\{a^i b^j a^i b^j \mid i, j \geq 0\}$;
   (b) $\{a^i b^i c^i d^i \mid i \geq 0\}$;
   (c) $\{www \mid w \in \{a, b\}^*\}$.

2. Demonstre que a seguinte gramática gera a linguagem das palavras da forma $a^i b^i c^i$, $i \geq 0$.

$$\begin{array}{l} S \to aAbc|\lambda \\ A \to aAbC|\lambda \\ Cb \to bC \\ Cc \to cc \end{array}$$

3. Demonstre que as linguagens recursivamente enumeráveis são fechadas sob união, interseção, concatenação, estrelha de Kleene e imágens homomorfas.

4. Seja G a gramática monotóna

$$\begin{array}{rl} G: & S \to SBA|a \\ & BA \to AB \\ & aA \to aaB \\ & B \to b \end{array}$$

   Determine $L(G)$ e construa uma gramática livre de contexto que gere $L(G)$.

5. Demonstre usando o lema de bombeamento para gramáticas livres de contexto que a linguagem $L = \{a^i b^{2i} a^i \mid i > 0\}$ não é livre de contexto e construa uma gramática sensível ao contexto, G, que gere L e um autômato linearmente limitado que aceite L.

6. Seja T uma árvore binária completa. Um caminho em T é uma sequência de passos descendente à esquerda (L), descendente à direita (R) ou ascendente (U). Os caminhos em T podem então ser identificados como uma linguagem sobre o alfabeto $\{L, R, U\}$. Considere a linguegem L_p de caminhos que iniciam e terminam na raiz; por exemplo, $\lambda, LU, RU, LLRUUURU$, etc. Determine a classificação de L_p na hierarquia de Chomsky.

7. Descreva uma TM que decida se dada uma TM arbitrária M com entrada w a computação pára em menos de 100 transições.

8. Demonstre que não existe algoritmo para decidir o problema de se uma TM arbitrária pára com entrada 101.

9. Demonstre que no existe algoritmo para determinar se uma TM arbitrária pára para alguma entrada.

10. Prove que é impossível decidir se uma TM arbitrária com entrada w reentra no seu estado inicial após a transição inicial.

11. Seja M uma DTM que aceita uma linguagem não-recursiva. Prove que o problema da parada para M é indecidível; i.e., que não existe TM que possa decidir se M pára com entrada w.

12. Use o teorema de Rice para provar que as seguintes propriedades de linguagens recursivamente enumeráveis são indecidíveis:

    (a) L contém uma palavra fixa w;

    (b) L é finita;

    (c) L é regular;

    (d) L é $\{0,1\}^*$.

13. Seja $L = \{R(M) \mid M$ pára executada com $R(M)\}$. Demonstre que L não é recursiva mas é recursivamente enumerável. Aqui $R(M)$ denota uma codificação binária de M, como formulado no texto de Sudkamp.

14. Demonstre que a linguagem $L_{\neq\emptyset} = \{R(M) \mid L(M) \neq \emptyset\}$ é recursivamente enumerável.

15. Encontre soluções, caso estas existam, para os seguintes sistemas de Post. Caso não exista solução, demonste-o.

    (a) $[ac, a], [abr, \lambda], [ab, ada], [ra, bra], [ad, brac]$;

    (b) $[aba, a], [aa, ab], [a, aaab]$;

    (c) $[01, 010], [100, 00], [010, 100]$;

    (d) $[st, s], [stt, tst], [tst, tts], [s, ts]$.

16. Demonstre que o problema de correspondência de Post para sistemas com um alfabeto unário é decidível.