Universidade Federal de Santa Catarina / Centro Tecnológico
Departamento de Informática e de Estatística
INE5317/INE5421 - Linguagens Formais e Compiladores

Lista de Exercícios nº 1

0 – Você deve saber ... (não precisa entregar a resposta desta questão)
a) A diferença entre problemas decidíveis e problemas indecidíveis.

Se para qualquer entrada e saída pertencente ao domínio do problema o programa retorna uma resposta valida(true ou false), o problema é decidível, caso contrario ele é um problema indecidível

b) De que forma problemas computacionais em geral, podem ser vistos como problemas de Linguagem.

Todo problema computacional pode ser transformado em uma gramatica, e portanto podem ser vistos como problemas de linguagem

c) A relação entre problemas decidíveis e problemas indecidíveis com a teoria das linguagens formais.

Problemas indecidíveis correspondem a gramaticas tipo 0
Problemas decidíveis correspondem a gramaticas tipo 1, 2 ou 3

d) Diferenças entre aspectos léxicos, sintáticos e semânticos de linguagens de programação.

Aspectos léxicos são aspectos relativos a quais símbolos foram utilizados na linguagem

Aspectos sintáticos são aspectos relativos a posição dos símbolos na linguagem

Aspectos semânticos são aspectos relativos ao sentido da linguagem

e) A estrutura geral de um compilador e as funções básicas de cada fase.

O compilador eh divido em uma fase de analise e uma fase de síntese

A fase de analise eh composta por uma analise, léxica, sintática e semântica

A fase de síntese pode ter ou não uma fase geração de código intermediário e depois ter fase de optimização de código, e geração de código

f) As formas de implementação de um compilador, e os critérios usados na escolha de uma delas para uma determinada implementação.

O compilador pode ser implementado em um ou mais passos dependendo da memoria, processamento e tempo disponível para compilação do programa

g) A importância do uso de Código Intermediário na implementação de um compilador.

O código intermediário pode ser utilizado para vários fins, tais quais permitir a optimização de um código, ajudar na portabilidade permitindo transformar em um código que possa ser lido por outro

programa(inclusive um compilador) e outras

g) As técnicas de otimização de código mais comuns.

h) A definição formal de sentença, forma sentencial, gramática, linguagem, derivação e redução.

i) Mostrar que GSC são recursivas.

1 – Diferencie Linguagem finita, infinita e vazia e exemplifique cada caso usando GR.

2 - Construa uma gramática G |

a) G não seja recursiva mas possua uma GR equivalente G1 | L(G1) seja Regular, infinita e não contenha a sentença vazia.

b) G seja recursiva, L(G) não seja Livre de Contexto e L(G) seja finita e contenha a sentença vazia..

3 - Toda linguagem finita é uma L.L.C. ? é uma L. Regular ? Justifique sua resposta.

4 – a) A União e a Concatenação de duas Linguagens (L1 e L2) de um mesmo tipo, resultam sempre em Linguagens do mesmo tipo que L1 e L2? Justifique e exemplifique.

b) Toda linguagem LLC pode ser gerada por uma GLC formada apenas por produções da forma $A \rightarrow B\ C \mid a$ onde A, B e C $\in V_n$ e a $\in V_t$?

c) Dada uma GLC ou uma GR qualquer, existe algoritmo para determinar se G é vazia? Se sim, forneça esse algoritmo (em passos gerais); senão, justifique.

d) Dada uma GLC ou uma GR qualquer G, existe algoritmo para verificar se L(G) é ou não finita? Se sim, forneça esse algoritmo (em passos gerais); senão, justifique.

5 - Construa uma Gramática Regular G |

a) $L(G) = \{ x \mid x \in (a, b, c)^+ \wedge$ todos os b's de x sejam consecutivos$\}$

b) $L(G) = \{ x \mid x \in a^n\ (b, c)^* \wedge n + \#c\text{'s é par} \wedge x$ não possui b's consecutivos$\}$

c) $L(G) = \{ 0^n1^m2^k \mid n,m,k \geq 0,$ m seja par $\wedge\ n + m + k$ seja ímpar$\}$

d) $L(G) = \{ x \mid x \in (a, b)^* \wedge \#a\text{'s seja par ou } \#b\text{'s seja divisível por 3}\}$

6 - Construa uma Gramática Livre de Contexto (GLC) G |

a) $L(G) = \{ a^n b^m (c,d)^* \mid 0 \leq n \leq m \wedge \#c\text{'s} = 2 * \#d\text{'s}\}$

b) $L(G) = \{ a^n\ b^m\ c^k\ d^p \mid n,m,k,p \geq 0,\ n \geq p \wedge k \geq m \}$

c) $L(G) = \{ x \mid x \in (a, b)^* (c, d)^* \wedge \#a\text{'s} + \#b\text{'s} \neq \#c\text{'s} + \#d\text{'s}\}$

d) $L(G) = \{ a^n\ b^m\ c^j \mid n,m,j \geq 0 \wedge n = m \vee m = j \}$

7 – Construa uma GLC que especifique:

a) a sintaxe da **declaração** de procedimentos (métodos) de uma linguagem de programação qualquer. OBS. Para simplificar, considere apenas parâmetros de tipos simples.

b) a sintaxe da **chamada** de procedimentos (métodos) de uma linguagem de programação qualquer.

c) a sintaxe de uma seqüência de estruturas de controle if-then-else e while-do (possivelmente aninhadas).

8 – Construa uma gramática G, do maior tipo possível |

a) $L(G) = \{ x \mid x \in (a, b)^* \wedge \#\underline{a}\text{'s} \neq \#\underline{b}\text{'s} \}$

b) $L(G) = \{ a^n b^m c^k \mid n,m,k \geq 0 \wedge m > n \wedge n > k \}$

c) $L(G) = \{ x \mid x \in (a, b)^* \wedge x \text{ seja palíndrome} \}$

d) $L(G) = \{ x \mid x \in a^n (b,c)^* \wedge n \text{ é par e } \#b\text{'s} + \#c\text{'s é divisível por 3} \}$

9 - Construa uma GSC (Gramática Sensível ao Contexto) G |

α) $L(G) = \{ a^n b^n c^m \mid n \geq 0, m \geq 0 \wedge n \neq m \}$

β) $L(G) = \{ a^n b^m c^p d^q \mid n,m,p,q \geq 0 \wedge n > p \wedge q < m \}$

χ) $L(G) = \{ x \mid x \in (a, b, c)^* \wedge \#\underline{a}\text{'s} \neq \#\underline{b}\text{'s} \neq \#c\text{'s} \}$

δ) $L(G) = \{ x \mid x \in (a, b, c)^* d^+ \wedge \#\underline{a}\text{'s} > \#c\text{´s} \wedge \#b\text{'s} = \#d\text{'s} \}$

10 - Seja G a seguinte gramática:

$S \rightarrow a\ S \mid S\ C \mid c\ S\ A \mid b$

$A\ C \rightarrow C\ A \qquad C\ A \rightarrow A\ C$

$b\ C \rightarrow b\ c \qquad b\ A \rightarrow b\ a$

$a\ A \rightarrow a\ a \qquad a\ C \rightarrow a\ c$

$c\ A \rightarrow c\ a \qquad c\ C \rightarrow c\ c$

Pede-se:

a) Verifique se x = acba e y = abca pertencem a L(G), usando o algoritmo apropriado;

b) Determine L(G);

c) Construa, se possível, uma GLC $G_1 \mid L(G1) = L(G)$

d) Construa $G_2 \mid$ G2 seja do mesmo tipo de G $\wedge L(G_2) = L(G) \cup \{ \varepsilon \}$

11 - Determine L(G) onde G é dada por:

a) $S \rightarrow 0\ S\ 1 \mid 1\ S\ 0 \mid 0\ 1 \mid 1\ 0$

b) $S \rightarrow a\ S \mid b\ A \mid b \mid a\ C \mid b\ D$
$A \rightarrow b\ S \mid a\ A \mid a \mid b\ C \mid a\ D$
$C \rightarrow d\ C \mid c\ D \mid c$
$D \rightarrow d\ D \mid c\ C \mid d$

c) $S \rightarrow a\ S\ B\ C \mid X \mid Y$
$X \rightarrow a\ X\ C \mid a\ C$
$Y \rightarrow Y\ Y \mid B$
$C\ B \rightarrow B\ C$
$B\ C \rightarrow C\ B$
$B \rightarrow b$
$C \rightarrow c$

d) $S \rightarrow a\ A \mid b\ A \mid a \mid b$ (* $L(G) \cup \{ \varepsilon \}$ é também uma LR? *)
$A \rightarrow a\ B \mid b\ B$
$B \rightarrow a\ C \mid b\ C$
$C \rightarrow a\ D \mid b\ D$

D → a E | b E | a | b
E → a S | b S

e) S → C D
C → a C A | b C B
A D → a D
B D → b D
A a → a A
A b → b A
B a → a B
B b → b B
C → ε
D → ε

f) S → AD (* L(G) é também uma LLC? *)
A → a A C | a A | a
D → B D d | D d | d
C B → B C
B C → C B
B → b
C → c

g) S → a S C | b S D | a S D | b S C | ε (* L(G) é também uma LLC? *)
D C → C D
C D → D C
C → c
D → d