

## Compiladores – Lista #1

1. Considere a gramática livre de contexto

$$S \rightarrow S S + \mid S S \times \mid a$$

- (a) Mostre que a cadeia  $aa+a\times$  pode ser gerada por esta gramática.
  - (b) Construa a árvore gramatical para esta cadeia.
  - (c) Qual é a linguagem gerada por esta gramática? Justifique sua resposta.
2. Construa uma gramática livre de contexto para cuja linguagem gerada são as listas, associativas à direita, de identificadores separados por vírgulas.
  3. Construa um esquema de tradução dirigida pela sintaxe que traduz expressões aritméticas da notação infixa para a prefixa, na qual um operador figura antes de seus operandos: por exemplo,  $-xy$  é a notação prefixa para  $x-y$ . Forneça as árvores gramaticais anotadas para as entradas  $9-5+2$  e  $9-5\times 2$
  4. Construa um parser recursivo descendente para as seguintes gramáticas:
    - (a)  $S \rightarrow 0 S 1 \mid 0 1$
    - (b)  $S \rightarrow + S S \mid - S S \mid a$
    - (c)  $S \rightarrow S ( S ) S \mid \epsilon$
  5. Na linguagem de programação C o enunciado **for** possui a forma

**for** ( **expr1**; **expr2**; **expr3** ) **cmd**

A primeira expressão é executada antes do laço; é tipicamente usada para inicializar o índice do laço. A segunda expressão é um teste feito antes de cada iteração do laço; o laço é abandonado se a expressão se torna 0. O laço em si consiste do enunciado { **cmd** **expr3** ; }. A terceira expressão é executada ao fim de cada iteração; é tipicamente usada para incrementar o índice do laço. O significado do enunciado **for** é similar a

**expr1** ; **while** ( **expr2** ) { **cmd** **expr3** }

Construa um esquema de tradução dirigida pela sintaxe para traduzir enunciados **for** em C em código de máquina de pilha.