Lista de exercícios para estudos
Não vale ponto – Não precisa entregar

1. Considere a seguinte especificação léxica (*"Flex-like"*):

```
(01|10)              { print "morango" }
1(01)*0              { print "banana" }
(1011*0|0100*1)      { print "laranja" }
```

   Forneça uma entrada para esse analisador léxico de forma que a string de saída seja $(\texttt{morango}^3\texttt{banana})^3(\texttt{morangolaranja})^2$, onde $\texttt{A}^\texttt{i}$ denota `A` repetido `i` vezes. Você pode usar uma notação simplificada na sua resposta, por exemplo, $(\texttt{01})^\texttt{3} = \texttt{010101}$. Lembre-se que a ordem das regras e tamanho da sequência de caracteres influenciam em qual regra será usada.

2. Considere a seguinte especificação léxica (*"Flex-like"*):

```
c*b    { print "X" }
ac     { print "Y" }
c*ac*  { print "Z" }
```

   Forneça a saída produzida pelo analisador léxico construído a partir dessa especificação para a seguinte entrada:

```
cbaccacacccbbcccbaccac
```

3. Forneça as GLCs para cada uma das seguintes linguagens. Qualquer gramática é aceitável (incluindo ambíguas), desde que estejam corretas.

   (a) o conjunto de todas as strings sobre o alfabeto $\Sigma = \{1, 2, -, *\}$ representando multiplicações entre inteiros em que a expressão gera algum valor positivo. Exemplos de strings nesta linguagem: $1 * 2$ ou $21 * -1 * -121$ ou $- - 222$. Exemplos de strings que não pertencem a esta linguagem: $2 * -2$ ou $1 * \lambda$ ou -12 ou $12 - 12$.

   (b) o conjunto de todas as strings sobre o alfabeto $\Sigma = \{0, 1\}$, em que o número de 1's é pelo menos duas vezes o número de 0's.

4. Considere as gramáticas a seguir, e faça o que se pede

   (a) Faça a fatoração à esquerda da gramática:

$$
\begin{aligned}
S &\to S + S \mid S + P \\
P &\to P * P \mid P * I \\
I &\to -I \mid (S) \mid D \\
D &\to 0 \mid 1N \\
N &\to 0 \mid 1 \mid NN \mid \lambda
\end{aligned}
$$

(b) Identifique e elimine a recursão à esquerda da gramática:

$$S \to S\ a\ S \mid U$$
$$U \to U\ u\ U \mid T$$
$$T \to t \mid f \mid T\ n \mid (S)$$

5. Considere a GLC a seguir, em que o conjunto de terminais é 0 , 1 , (,) , ;:

$$S \to (T$$
$$T \to CA\ |)$$
$$A \to ; B\ |)$$
$$B \to CA\ |)$$
$$C \to 0 \mid 1 \mid S$$

(a) Construa os conjuntos FIRST para cada um dos não terminais.

(b) Construa os conjuntos FOLLOW para cada um dos não terminais.

(c) Construa a *parsing table* LL(1) para a gramática.

(d) Mostre a sequência da pilha, entrada e ações que ocorrem durante um *parsing* LL(1) da string "(( ) ; 0)". No começo do *parse*, a pilha deve conter um único S.

6. Considere a GLC a seguir, em que o conjunto de terminais é x , y , z:

$$A \to xCBy$$
$$B \to z \mid \lambda$$
$$C \to y \mid Bx$$

(a) Construa os conjuntos FIRST para cada um dos não terminais.

(b) Construa os conjuntos FOLLOW para cada um dos não terminais. Desconsidere mostrar o símbolo \$, pois não há uma "pseudo produção" inicial.

7. Para cada uma das gramáticas a seguir, identifique e demonstre se elas são ou não LL(1).

(a)

$$X \to aY \mid Z$$
$$Y \to a \mid c$$
$$Z \to bY$$

(b)

$$P \to dR$$
$$R \to o \mid S$$
$$S \to g \mid og$$

(c)

$$J \to aKL$$
$$K \to c \mid \lambda$$
$$L \to c$$

(d)

$$J \to aKL$$
$$K \to c \mid \lambda$$
$$L \to b$$

8. A gramática a seguir não é LL(1). Use o processo necessário para transformar esta gramática, de forma que ela produza uma linguagem equivalente, e satisfaça as condições para ser uma LL(1). Será necessário remover as recursões indiretas à esquerda, para depois remover as recursões diretas à esquerda.

$$A \to B! \mid x$$
$$B \to C$$
$$C \to A? \mid y$$

9. Considere a seguinte GLC.

$$\begin{aligned} S &\to AED \mid F \\ A &\to Aa \mid a \\ B &\to Bb \mid b \\ C &\to Cc \mid c \\ D &\to Dd \mid d \\ E &\to bEc \mid bc \\ F &\to aFd \mid BC \end{aligned}$$

   (a) Mostre que esta gramática é ambígua fornecendo uma sentença que pode ser derivada de duas formas diferentes. Desenhe as duas árvores de derivação.

   (b) Forneça uma gramática não ambígua que seja capaz de gerar a mesma linguagem da gramática acima.

10. Considere a seguinte GLC, que possui o conjunto de terminais $T = \{\mathbf{id}, (,), [,], ;\}$.

$$\begin{aligned} E &\to \mathbf{id} \mid \mathbf{id}(A) \mid \mathbf{id}[E] \\ A &\to E \mid E\,;\,A \end{aligned}$$

   (a) Realize a fatoração à esquerda dessa gramática de forma que produções com o mesmo lado esquerdo não possuam lados direitos com um prefixo comum entre si.

   (b) Construir uma tabela sintática LL(1) para gramática fatorada à esquerda.

(c) Simular a operação de um *parser* LL(1) sobre a sentença de entrada **id(id[id]; id)**.

11. Considere a seguinte GLC, que possui o conjunto de terminais $T = \{\mathbf{a}, \mathbf{b}\}$.

$$\begin{aligned} S &\rightarrow X\mathbf{a} \\ X &\rightarrow \mathbf{a} \mid \mathbf{a}X\mathbf{b} \end{aligned}$$

(a) Construir o autômato de prefixos viáveis para esta gramática utilizando os itens LR(0).

(b) Identificar um conflito reduzir/empilhar nesta gramática sob as regras para construção de tabelas SLR(1).

(c) Assumindo que um *parser* SLR(1) resolve os conflitos reduzir/empilhar selecionando sempre "**empilhar**", faça a simulação do funcionamento desse parser para a sentença de entrada **aaba**.

(d) Suponha que a produção $X \rightarrow \lambda$ seja adicionada a esta gramática. Identique um conflito reduzir/reduzir na gramática resultante sob as regra de construção de tabelas SLR(1).