Compiladores — Folha laboratorial 3

Pedro Vasconcelos, DCC/FCUP

Outubro 2020

Gramáticas independentes de contexto

Exercício 1

(a) Considere a gramática apresentada na aula teórica com símbolos terminais $\{a, b\}$, não-terminais $\{S, B\}$, símbolo inicial S e as produções seguintes.

$$\begin{array}{ll} S \to aSB & B \to Bb \\ S \to \varepsilon & B \to b \\ S \to B & \end{array}$$

Descreva a linguagem reconhecida por esta gramática numa frase. (Sugestões: Onde podem ocorrer as e bs? Qual relação entre as suas contagens?)

(b) Escreva uma gramática para a linguagem de palavras com o mesmo número de as e bs (em qualquer ordem)

(c) Escreva uma gramática para a linguagem de parêntesis casados. Alguns exemplos:

Palavras aceites	Palavras não aceites
ε (palavra vazia)	(
()	)
(())	)(
()()	(()
(()(()))	()())

(d) Escreva uma gramática para a linguagem da expressão regular `((ab*a)|(ba*b))`

(e) Escreva uma gramática para a linguagem da expressão regular `((0|1)+"."(0|1)*)|((0|1)*"."(0|1)+)`

Exercício 2

Considere uma gramática de programas sequências apresentada na aula teórica:

$$\begin{array}{ll} S \to S \texttt{ ; } S & E \to \text{ident} \\ S \to \text{ident} \texttt{ = } E & E \to \text{num} \\ S \to \text{ident} \texttt{ ++} & E \to E \texttt{ + } E \end{array}$$

(a) Mostre que a gramática acima é ambígua, i.e., encontre uma palavra com duas derivações correspondentes a árvores distintas. Consegue encontrar mais do que um exemplo?
(b) Re-escreva a gramática de forma a eliminar a ambiguidade.

Exercício 3

Considere a gramática para expressões aritméticas fatorizada de forma a eliminar ambiguidade e recursão à esquerda:

$$
\begin{array}{lll}
E \to T\ E' & T \to F\ T' & \\
E' \to +\ T\ E' & T' \to *\ F\ T' & F \to \text{num} \\
E' \to -\ T\ E' & T' \to /\ F\ T' & F \to (\ E\ ) \\
E' \to \varepsilon & T' \to \varepsilon &
\end{array}
$$

Partindo do esqueleto de código no repositório *Git*, implemente um *parser* para esta gramática usando descida recursiva. O *parser* deve aceitar expressões bem formadas como `123*(45+6)` ou `(1+2*3)/(4+5)` e rejeitar expressões incorretas como `)12+53(` ou `(1++2*3)`.

Exercício 4

Considerando de novo a gramática do Exercício 3:

(a) Determine o valor de NULLABLE e dos conjuntos FIRST e FOLLOW para os não-terminais desta gramática. (Sugestão: comece por determinar as repostas por inspeção das produções e só depois use as equações apresentadas na aulas para verificar se a sua intuição está correta.)

(b) Mostre que a gramática acima é $LL(1)$ calculado a tabela de *parsing* preditivo.