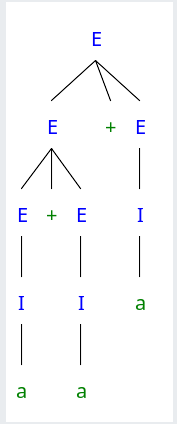
Preparation Activity PA07 – Context-Free Grammars (CFGs)

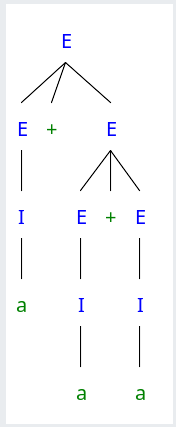
1. Tendo em consideração a CFG G1:  
   E → I | E+E | E×E | (E)   
   I → a | b | Ia | Ib | I0 | I1  
     
   a) – A CFG apresenta ambiguidade pois existem casos em que se pode obter duas arvores de derivação diferentes para a mesma string, havendo mais que uma forma alternativa de derivar uma dada string com as produções fornecidas. Isto deve-se ao facto de existirem produções como E → E+E, que apresentam recursividade à esquerda e à direita.

Um exemplo seria a string “a+a+a” que apresenta duas arvores de derivação:

Primeira Derivação Possível



Segunda Derivação Possível



b) – Uma derivação *leftmost* possível seria:   
  
E => (E) => (E+E) => ((E) + E) => ((I) + E) => ((a) + E) => ((a) + (E)) => ((a) + (E×E)) => ((a) + (I×E)) => ((a) + (a×E)) => ((a) + (a×I)) => ((a) + (a×b))

1. a) – Considerando a seguinte gramática G2:

E → I | E+I | E×I | (E)   
I → a | b | Ia | Ib | I0 | I1  
  
G2 não é ambígua, mas, por outro lado, não representa a mesma linguagem que G1, pois em G2 não é possível ter expressões do tipo E + (E) ou E × (E), isto porque o lado direito das somas e multiplicações em G2 não tem recursividade para E (devido a uma tentativa de remover ambiguidade, certamente) e, sendo assim, do lado direito apenas se poderão ter identificadores sem parêntesis à volta (que são as produções de I), estando apenas disponível derivar E+I ou E×I em G2.

b) – Considerando a gramática G3:

E → F | E+F | E×F   
F → I | (E)   
I → a | b | Ia | Ib | I0 | I1

G3 não é ambígua e representa a mesma linguagem que G1, sendo que, resolve o problema da gramática anterior criando uma variável intermédia F que contempla a possibilidade do operando direito as expressões ser tanto um identificador como uma expressão entre parêntesis, resolve também o problema da ambiguidade pois remove a recursividade de E no lado direito, criando apenas um a derivação possível para as somas e multiplicações.

c) – Considerando a gramática G4:  
E → J | E×J   
J → I | J+I   
I → a | b | Ia | Ib | I0 | I1 | (E)

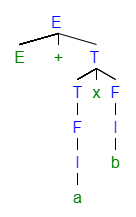
O problema de G4 é que admite que um identificador (I) pode ser representado por uma expressão entre parêntesis, daí ser uma extensão da linguagem G1, pois admite mais casos possíveis para identificadores a gramática original, sendo que, G1 apenas admite identificadores constituídos pelos símbolos “a”, “b”, “0” ou “1” e sem parêntesis.

d) – Considerando a gramática G5:

E → T | E+T   
T → F | T×F   
F → I | (E)   
I → a | b | Ia | Ib | I0 | I1

A seguinte gramática respeita a precedência aritmética da multiplicação sobre a soma, pois, encontrando uma expressão que contenha ambas, tal como se poderia ver pela arvore de derivação resultante, calcular-se-ia primeiro uma multiplicação e apenas a seguir a soma.

Logo, tendo um exemplo do gênero “a + a × b”, a arvore de derivação seria:



Considerando que a análise da árvore seria feita de baixo para cima, das folhas até à raiz, a multiplicação teria sempre precedência.