

SOLUÇÃO E-FÓLIO B

a)

$G = (V, T, P, S)$

$V = \{\text{Programa, SeqInstr, Instr, Expr, Cond}\}$

$T = \{\text{begin, end, =, ., if, then, else, while, +, -, *, /, (,), VAR, CONST, <, >, !=, ==}\}$

$S = \text{Programa}$

P:

Programa \rightarrow begin SeqInstr end

SeqInstr \rightarrow Instr . SeqInstr | Instr¹

Instr \rightarrow VAR = Expr | if Cond then Instr else Instr | while Cond Instr

Expr \rightarrow Expr + Expr | Expr – Expr | Expr * Expr | Expr / Expr | (Expr) | VAR | CONST

Cond \rightarrow Expr < Expr | Expr > Expr | Expr != Expr | Expr == Expr

b)

No caso do if-then-else não existe problema porque o else é obrigatório.

Nas expressões, existe esse problema. Por exemplo, $5*3+4$ pode ser derivado de duas formas diferentes:

Expr \Rightarrow Expr + Expr \Rightarrow Expr * Expr + Expr \Rightarrow ...

Expr \Rightarrow Expr * Expr \Rightarrow Expr * Expr + Expr \Rightarrow ...

Para, por um lado, impedirmos a ambiguidade e, por outro lado, forçarmos a precedência e a associatividade à esquerda, temos de alterar as produções relativas às expressões:

Expr \rightarrow Expr + Term | Expr – Term | Term

Term \rightarrow Term * Factor | Term / Factor | Factor

Factor \rightarrow VAR | CONST | (Expr)

Neste caso, passamos a ter $V = \{\text{Programa, SeqInstr, Instr, Expr, Term, Factor, Cond}\}$.

¹ No caso de admitirmos que o programa poderia não ter instruções, poderíamos colocar ϵ em vez de Instr. No caso do ponto final no fim de cada instrução, o ponto pode ser usado nesta produção ou em cada uma das produções de Instr – nesse caso, as instruções compostas teriam sempre ponto final em cada instrução; por exemplo, em vez de if $x>5$ then $x=1$ else $x=0$. ficaria if $x>5$ then $x=1$. else $x=0$..