1 Gramàtica

Els símbols terminals o tokens estan marcats en **negreta**.

$$
\begin{aligned}
\mathit{program} &\rightarrow \mathit{declList} \\
\mathit{declList} &\rightarrow \mathit{declList}\ \mathit{decl} \\
&\mid \mathit{decl} \\
\mathit{declList} &\rightarrow \mathit{varDecl} \\
&\mid \mathit{funcDecl} \\
&\mid \mathit{procDecl} \\
\mathit{varDecl} &\rightarrow \mathit{const}\ \mathit{type}\ \mathit{varDeclList}\ \mathit{varInit}\ \mathbf{;} \\
\mathit{varDeclList} &\rightarrow \mathbf{ID} \\
&\mid \mathit{varDeclList}\ \mathbf{,}\ \mathbf{ID} \\
\mathit{varInit} &\rightarrow \mathbf{=}\ \mathit{exprSimple} \\
&\mid \lambda \\
\mathit{const} &\rightarrow \mathbf{const} \\
&\mid \lambda \\
\mathit{funcDecl} &\rightarrow \mathbf{func}\ \mathbf{ID}\ \mathbf{(}\ \mathit{args}\ \mathbf{)}\ \mathbf{:}\ \mathit{type}\ \mathbf{begin}\ \mathit{bloc}\ \mathbf{end}\ \mathbf{ID}\ \mathbf{;} \\
\mathit{procDecl} &\rightarrow \mathbf{proc}\ \mathbf{ID}\ \mathbf{(}\ \mathit{args}\ \mathbf{)}\ \mathbf{begin}\ \mathit{bloc}\ \mathbf{end}\ \mathbf{ID}\ \mathbf{;} \\
\mathit{args} &\rightarrow \mathit{argList} \\
&\mid \lambda \\
\mathit{argList} &\rightarrow \mathit{type}\ \mathbf{id}\ \mathit{argListCont} \\
\mathit{argListCont} &\rightarrow \mathbf{,}\ \mathit{argList} \\
&\mid \lambda \\
\mathit{bloc} &\rightarrow \mathit{statement}\ \mathbf{;}\ \mathit{bloc} \\
&\mid \lambda \\
\mathit{statement} &\rightarrow \mathit{varDecl} \\
&\mid \mathit{iterationStatement} \\
&\mid \mathit{conditionalStatement} \\
&\mid \mathit{returnStatement} \\
&\mid \mathbf{break} \\
\mathit{iterationStatement} &\rightarrow \mathit{forIteration} \\
&\mid \mathit{whileIteration} \\
\mathit{forIteration} &\rightarrow \mathbf{for}\ \mathbf{(}\ \mathit{forInit}\ \mathbf{;}\ \mathit{exprSimple}\ \mathbf{;}\ \mathit{forPostExpression}\ \mathbf{)}\ \mathbf{do}\ \mathit{bloc}\ \mathbf{end}\ \mathbf{;} \\
\mathit{forInit} &\rightarrow \mathit{expression} \\
&\mid \lambda \\
\mathit{forPostExrpession} &\rightarrow \mathit{expression} \\
&\mid \lambda
\end{aligned}
$$

$$
\begin{aligned}
\mathit{whileIteration} &\rightarrow \textbf{while } (\ \mathit{exprSimple}\ ) \textbf{ do } \mathit{bloc} \textbf{ end} \\
\mathit{condStatement} &\rightarrow \mathit{ifStatement} \\
&\mid \mathit{switchStatement} \\
\mathit{ifStatement} &\rightarrow \textbf{if } \mathit{exprSimple} \textbf{ do } \mathit{bloc}\ \mathit{elseIfStatement} \textbf{ end} \\
\mathit{elseIfStatement} &\rightarrow \textbf{else if } \mathit{exprSimple} \textbf{ do } \mathit{bloc}\ \mathit{elseIfStatement} \\
&\mid \mathit{elseStatement} \\
\mathit{elseStatement} &\rightarrow \textbf{else do } \mathit{bloc} \\
&\rightarrow \lambda \\
\mathit{switchStatement} &\rightarrow \textbf{switch } (\ \mathit{exprSimple}\ ) \textbf{ begin } \mathit{caseBloc} \textbf{ end} \\
\mathit{caseBloc} &\rightarrow \textbf{case } \mathit{literal}? \textbf{ do } \mathit{bloc} \textbf{ end } \mathit{caseBloc} \\
&\mid \textbf{default do } \mathit{bloc} \textbf{ end} \\
&\mid \lambda \\
\mathit{exprSimple} &\rightarrow \mathit{exprSimple} \textbf{ and } \mathit{exprSimple} \\
&\mid \mathit{exprSimple} \textbf{ or } \mathit{exprSimple} \\
&\mid \mathit{exprSimple} \textbf{ not } \mathit{exprSimple} \\
&\mid (\ \mathit{exprSimple}\ ) \\
&\mid \mathit{relExpr} \\
&\mid \mathit{aritExptr} \\
\mathit{aritExpr} &\rightarrow \mathit{aritExpr} + \mathit{aritExpr} \\
&\mid \mathit{aritExpr} \ / \ \mathit{aritExpr} \\
&\mid \mathit{aritExpr} \ \textbf{*}\ \mathit{aritExpr} \\
&\mid \mathit{aritExpr} + \mathit{aritExpr} \\
&\mid \mathit{operand} \\
\mathit{relExpr} &\rightarrow \mathit{exprSimple} == \mathit{exprSimple} \\
&\mid \mathit{exprSimple} \ \textbf{!}= \mathit{exprSimple} \\
&\mid \mathit{exprSimple} <= \mathit{exprSimple} \\
&\mid \mathit{exprSimple} < \mathit{exprSimple} \\
&\mid \mathit{exprSimple} >= \mathit{exprSimple} \\
&\mid \mathit{exprSimple} > \mathit{exprSimple} \\
\mathit{operand} &\rightarrow \mathit{mutable} \\
&\mid \mathit{immutable} \\
\mathit{mutable} &\rightarrow \textbf{id} \\
\mathit{immutable} &\rightarrow \mathit{literal} \\
&\mid \mathit{funcCall} \\
\mathit{funcCall} &\rightarrow \textbf{id } (\ \mathit{largs}\ ) \\
\mathit{largs} &\rightarrow \mathit{operand}\ \mathit{contlargs} \\
&\mid \lambda
\end{aligned}
$$

$$
\begin{aligned}
contlargs \rightarrow & \ \mathbf{,}\ operand\ largs \\
& \mid \lambda
\end{aligned}
$$