

## LL(l) Grammar

$\langle \text{program} \rangle \rightarrow \langle \text{decl\_list} \rangle$

$\langle \text{decl\_list} \rangle \rightarrow \langle \text{decl} \rangle ';' \langle \text{decl\_list} \rangle \mid \lambda$

$\langle \text{decl} \rangle \rightarrow \text{var id ':'} \langle \text{type} \rangle$   
|  $\text{constant id '='} \langle \text{expr} \rangle$   
|  $\text{function id} \langle \text{formal\_list} \rangle \text{' ':'} \langle \text{type} \rangle \langle \text{block} \rangle$   
|  $\text{procedure id} \langle \text{formal\_list} \rangle \langle \text{block} \rangle$

$\langle \text{type} \rangle \rightarrow \text{integer} \mid \text{boolean} \mid \text{string}$

$\langle \text{formal\_list} \rangle \rightarrow '(' \langle \text{formal\_list\_tail} \rangle$

$\langle \text{formal\_list\_tail} \rangle \rightarrow ')' \mid \langle \text{formals} \rangle ')'$

$\langle \text{formals} \rangle \rightarrow \text{id ':'} \langle \text{type} \rangle \langle \text{formals\_tail} \rangle$

$\langle \text{formals\_tail} \rangle \rightarrow ',' \text{id ':'} \langle \text{type} \rangle \langle \text{formals\_tail} \rangle \mid \lambda$

$\langle \text{stmt} \rangle \rightarrow \text{id} \langle \text{id\_tail} \rangle$   
|  $\text{if} \langle \text{expr} \rangle \text{ then} \langle \text{stmt} \rangle \langle \text{stmt\_tail} \rangle$   
|  $\text{while} \langle \text{expr} \rangle \text{ do} \langle \text{stmt} \rangle \text{ od}$   
|  $\text{for id ':'} \langle \text{expr} \rangle \text{ to} \langle \text{expr} \rangle \text{ do} \langle \text{stmt} \rangle \text{ od}$   
|  $\text{read}(' \text{id}')$   
|  $\text{write}(' \text{id}')$   
|  $\text{return} '(' \langle \text{expr} \rangle ')'$   
|  $\langle \text{block} \rangle$

$\langle id\_tail \rangle \rightarrow ' := ' \langle expr \rangle | \langle arg\_list \rangle$

$\langle stmt\_tail \rangle \rightarrow fi | else \langle stmt \rangle fi$

$\langle block \rangle \rightarrow begin \langle var\_decl\_list \rangle \langle stmt\_list \rangle end$

$\langle var\_decl\_list \rangle \rightarrow \langle var\_decl \rangle ';' \langle var\_decl\_list \rangle | \lambda$

$\langle var\_decl \rangle \rightarrow var\ id\ ':' \langle type \rangle$

$\langle stmt\_list \rangle \rightarrow \langle stmt \rangle ';' \langle stmt\_list \rangle | \lambda$

$\langle arg\_list \rangle \rightarrow '(' \langle arg\_list\_tail \rangle$

$\langle arg\_list\_tail \rangle \rightarrow ')' | \langle args \rangle ')'$

$\langle args \rangle \rightarrow \langle expr \rangle \langle args\_tail \rangle$

$\langle args\_tail \rangle \rightarrow ';' \langle args \rangle | \lambda$

$\langle expr \rangle \rightarrow \langle expr1 \rangle \langle expr\_tail \rangle$

$\langle \text{expr\_tail} \rangle \rightarrow \langle \text{rel\_conj} \rangle \langle \text{expr1} \rangle \langle \text{expr\_tail} \rangle \mid \lambda$

$\langle \text{expr1} \rangle \rightarrow \langle \text{expr2} \rangle \langle \text{expr1\_tail} \rangle$

$\langle \text{expr1\_tail} \rangle \rightarrow \langle \text{rel\_op} \rangle \langle \text{expr2} \rangle \langle \text{expr1\_tail} \rangle \mid \lambda$

$\langle \text{expr2} \rangle \rightarrow \langle \text{expr3} \rangle \langle \text{expr2\_tail} \rangle$

$\langle \text{expr2\_tail} \rangle \rightarrow \langle \text{arith\_op\_tail1} \rangle \langle \text{expr3} \rangle \langle \text{expr2\_tail} \rangle \mid \lambda$

$\langle \text{expr3} \rangle \rightarrow \langle \text{expr4} \rangle \langle \text{expr3\_tail} \rangle$

$\langle \text{expr3\_tail} \rangle \rightarrow \langle \text{arith\_op\_tail2} \rangle \langle \text{expr4} \rangle \langle \text{expr3\_tail} \rangle \mid \lambda$

$\langle \text{expr4} \rangle \rightarrow \langle \text{unary\_op} \rangle \langle \text{expr4} \rangle \mid \langle \text{expr5} \rangle$

$\langle \text{expr5} \rangle \rightarrow \text{id} \langle \text{expr\_id\_tail} \rangle \mid \text{integer} \mid \text{string} \mid \text{true} \mid \text{false} \mid$   
 $\text{'('} \langle \text{expr} \rangle \text{'}$

$\langle \text{expr\_id\_tail} \rangle \rightarrow \langle \text{arg\_list} \rangle \mid \lambda$

$\langle \text{arith\_op\_tail1} \rangle \rightarrow \text{'+'} \mid \text{'-'}$

$\langle arith\_op\_tail2 \rangle \rightarrow '*' \mid '/'$

$\langle rel\_op \rangle \rightarrow '=' \mid '!=' \mid '<' \mid '<=' \mid '>' \mid '>='$

$\langle rel\_conj \rangle \rightarrow and \mid or$

$\langle unary\_op \rangle \rightarrow - \mid not$