

## Final Grammar: Left Factoring and Eliminating Left Recursion

$A \rightarrow \underline{\text{let}} \ B \ \underline{\text{in}} \ I \ \underline{\text{end}}$

$B \rightarrow C \ B \mid \epsilon$

$C \rightarrow D \mid F$

$D \rightarrow \underline{\text{var}} \ ID \ D'$

$D' \rightarrow := \ K \mid : \ E$

$E \rightarrow \underline{\text{int}} \mid \underline{\text{string}} \mid \underline{\text{void}}$

$F \rightarrow \underline{\text{function}} \ ID \ ( \ F'$

$F' \rightarrow G \ ) : E = I \ \underline{\text{end}} \mid \ ) : E = I \ \underline{\text{end}}$

$G \rightarrow H \ G'$

$G' \rightarrow , \ H \ G' \mid \epsilon$

$H \rightarrow ID : E$

$I \rightarrow J \ I'$

$I' \rightarrow ; \ J \ I' \mid \epsilon$

$J \rightarrow ID := J' \mid ID \ ( \ J'' \mid \text{return } J''' \mid \underline{\text{printint}} \ ( \ K \ ) \mid \underline{\text{printstring}} \ ( \ K \ )$

$J' \rightarrow K \mid \text{getint} \ ( \ )$

$J'' \rightarrow \ ) \mid N \ )$

$J''' \rightarrow K \mid \epsilon$

$K \rightarrow L \ K'$

$K' \rightarrow + \ L \ K' \mid - \ L \ K' \mid \epsilon$

$L \rightarrow M \ L'$

$L' \rightarrow * \ M \ L' \mid / \ M \ L' \mid \epsilon$

$M \rightarrow ( \ K \ ) \mid \text{NUMBER} \mid \text{STRING\_LITERAL} \mid ID \ M'$

$M' \rightarrow ( \ ) \mid ( \ N \ ) \mid \epsilon$

$N \rightarrow K \ N'$

$N' \rightarrow , \ K \ N' \mid \epsilon$

## Original Language

program = A  $\rightarrow$  let B in I end

decs = B  $\rightarrow$  C B |  $\epsilon$

dec = C  $\rightarrow$  D | F

var\_dec = D  $\rightarrow$  var ID := K | var ID : E

type = E  $\rightarrow$  int | string | void

function\_dec = F  $\rightarrow$  function ID (G) : E = I end | function ID ( ) : E = I end

parameters = G  $\rightarrow$  G , H | H

parameter = H  $\rightarrow$  ID : E

statements = I  $\rightarrow$  I ; J | J

statement = J  $\rightarrow$  ID := K | printint ( K ) | printstring ( K ) | ID := getint( ) | ID ( ) | ID ( N ) | return K | return

expr = K  $\rightarrow$  K + L | K - L | L

term = L  $\rightarrow$  L \* M | L / M | M

factor = M  $\rightarrow$  ( K ) | NUMBER | STRING\_LITERAL | ID | ID ( ) | ID ( N )

expr\_list = N  $\rightarrow$  N , K | K