A Grammar Example

Stefan D. Bruda

CS 310, Winter 2025

A REALISTIC MINIMAL EXAMPLE

⟨block⟩	::=	{ ⟨decls⟩ ⟨stmts⟩ }
⟨decls⟩	::=	⟨decl⟩ ⟨decls⟩ | ε
⟨decl⟩	::=	⟨type⟩ `ID` ;
⟨type⟩	::=	⟨type⟩ `[` `NUM` `]` | `BASIC`
⟨stmts⟩	::=	⟨stmt⟩ ⟨stmts⟩ | ε
⟨stmt⟩	::=	⟨loc⟩ = ⟨bool⟩ ; |
	|	`IF` (⟨bool⟩) ⟨stmt⟩ `ELSE` ⟨stmt⟩
	|	`WHILE` (⟨bool⟩) ⟨stmt⟩
	|	⟨block⟩
⟨loc⟩	::=	⟨loc⟩ [⟨bool⟩] | `ID`
⟨bool⟩	::=	⟨join⟩ `||` ⟨bool⟩ | ⟨join⟩
⟨join⟩	::=	⟨equality⟩ `&&` ⟨join⟩ | ⟨equality⟩
⟨equality⟩	::=	⟨equality⟩ == ⟨rel⟩
	|	⟨equality⟩ `!=` ⟨rel⟩
	|	⟨rel⟩
⟨rel⟩	::=	⟨expr⟩ < ⟨expr⟩
	|	⟨expr⟩ > ⟨expr⟩
	|	⟨expr⟩
⟨expr⟩	::=	⟨term⟩ + ⟨expr⟩
	|	⟨term⟩ - ⟨expr⟩
	|	⟨term⟩
⟨term⟩	::=	⟨unary⟩ * ⟨term⟩
	|	⟨unary⟩ / ⟨term⟩
	|	⟨unary⟩
⟨unary⟩	::=	! ⟨unary⟩ | - ⟨unary⟩ | ⟨factor⟩
⟨factor⟩	::=	(⟨bool⟩) | ⟨loc⟩ | `NUM` | `REAL` | `TRUE` | `FALSE`

A REALISTIC MINIMAL EXAMPLE

⟨block⟩ ::= { ⟨decls⟩ ⟨stmts⟩ }
⟨decls⟩ ::= ⟨decl⟩ ⟨decls⟩ | ε
⟨decl⟩ ::= ⟨type⟩ ID ;
⟨type⟩ ::= ⟨type⟩ [NUM] | BASIC
⟨stmts⟩ ::= ⟨stmt⟩ ⟨stmts⟩ | ε
⟨stmt⟩ ::= ⟨loc⟩ = ⟨bool⟩ ;
| IF (⟨bool⟩) ⟨stmt⟩ ELSE ⟨stmt⟩
| WHILE (⟨bool⟩) ⟨stmt⟩
| ⟨block⟩
⟨loc⟩ ::= ⟨loc⟩ [⟨bool⟩] | ID
⟨bool⟩ ::= ⟨join⟩ || ⟨bool⟩ | ⟨join⟩
⟨join⟩ ::= ⟨equality⟩ && ⟨join⟩ | ⟨equality⟩
⟨equality⟩ ::= ⟨equality⟩ == ⟨rel⟩
| ⟨equality⟩ != ⟨rel⟩
| ⟨rel⟩
⟨rel⟩ ::= ⟨expr⟩ < ⟨expr⟩
| ⟨expr⟩ > ⟨expr⟩
| ⟨expr⟩
⟨expr⟩ ::= ⟨term⟩ + ⟨expr⟩
| ⟨term⟩ - ⟨expr⟩
| ⟨term⟩
⟨term⟩ ::= ⟨unary⟩ * ⟨term⟩
| ⟨unary⟩ / ⟨term⟩
| ⟨unary⟩
⟨unary⟩ ::= ! ⟨unary⟩ | - ⟨unary⟩ | ⟨factor⟩
⟨factor⟩ ::= (⟨bool⟩) | ⟨loc⟩ | NUM | REAL | TRUE | FALSE

Recursive descent parsing grammar

⟨block⟩ ::= { ⟨decls⟩ ⟨stmts⟩ }
⟨decls⟩ ::= ⟨decl⟩ ⟨decls⟩ | ε
⟨decl⟩ ::= ⟨type⟩ ID ;
⟨type⟩ ::= BASIC ⟨typeCl⟩
⟨typeCl⟩ ::= [NUM] ⟨typeCl⟩ | ε
⟨stmts⟩ ::= ⟨stmt⟩ ⟨stmts⟩ | ε
⟨stmt⟩ ::= ⟨loc⟩ = ⟨bool⟩ ;
| IF (⟨bool⟩) ⟨stmt⟩ ELSE ⟨stmt⟩
| WHILE (⟨bool⟩) ⟨stmt⟩
| ⟨block⟩
⟨loc⟩ ::= ID ⟨locCl⟩
⟨locCl⟩ ::= [⟨bool⟩] ⟨locCl⟩ | ε
⟨bool⟩ ::= ⟨join⟩ || ⟨bool⟩ | ⟨join⟩
⟨join⟩ ::= ⟨equality⟩ && ⟨join⟩ | ⟨equality⟩
⟨equality⟩ ::= ⟨rel⟩ ⟨equalityCl⟩
⟨equalityCl⟩ ::= == ⟨equalityCl⟩ | != ⟨equalityCl⟩ | ε
⟨rel⟩ ::= ⟨expr⟩ ⟨relTail⟩
⟨relTail⟩ ::= < ⟨expr⟩ | > ⟨expr⟩ | ε
⟨expr⟩ ::= ⟨term⟩ ⟨exprTail⟩
⟨exprTail⟩ ::= + ⟨expr⟩ | - ⟨expr⟩ | ε
⟨term⟩ ::= ⟨unary⟩ ⟨termTail⟩
⟨termTail⟩ ::= * ⟨term⟩ | / ⟨term⟩ | ε
⟨unary⟩ ::= ! ⟨unary⟩ | - ⟨unary⟩ | ⟨factor⟩
⟨factor⟩ ::= (⟨bool⟩) | ⟨loc⟩ | NUM | REAL | TRUE | FALSE