

- **Gramática AVI GLC**

- $S = \text{funDec Type FunName } [' \text{ Params } ']' \text{ Body } S$
- $S = \text{Decl } S$
- $S = \varepsilon$

- **Function Declaration**

- $\text{FunName} = \text{'id'}$
- $\text{FunName} = \text{'main'}$
- $\text{Params} = \text{Params, Type 'id' ArrayOpt}$
- $\text{Params} = \text{Type 'id' ArrayOpt}$
- $\text{Params} = \varepsilon$
- $\text{FunCall} = [' \text{ Lec } ']' ';'$
- $\text{Return} = \text{'return' Ec ';'}$

- **Variable Declaration**

- $\text{Decl} = \text{Type LI}$
- $\text{LI} = \text{'id' ArrayOpt Inst}$
- $\text{LI} = \text{LI, 'id' ArrayOpt Inst}$

- **Instantiating Variables**

- $\text{Inst} = \text{'atrib' Inr}$
- $\text{Inst} = \varepsilon$
- $\text{Inr} = \text{ArrayOpt}$
- $\text{Inr} = \text{Fc}$

- **Array**

- $\text{ArrayOpt} = \text{'(' ArrayAccess}$
- $\text{ArrayOpt} = \varepsilon$
- $\text{ArrayAccess} = \text{'})'}$
- $\text{ArrayAccess} = \text{'intConst' '})'}$

- **Variable Type**

- $\text{Type} = \text{'intType'}$
- $\text{Type} = \text{'floatType'}$

- Type = 'boolType'
- Type = 'stringType'
- Type = 'reVoid'

- **Commands**

- Command = 'reFor' '[' Atr ';' Eb ';' Inc ']' Body
- Command = 'reWhile' '[' Eb ']' Body
- Command = 'reIf' '[' Eb ']' Body lfr
- lfr = 'reElseIf' '[' Eb ']' Body lfr
- lfr = 'reElse' Body
- lfr = ε
- Inc = 'constInt'
- Inc = 'id'

- **Id List:**

- IdL = 'id' ArrayAccess
- IdL = IdL ',' 'id' ArrayAccess
- IdLr = ε

- **Body**

- Body = '{' BodyScope '}'
- BodyScope = Decl BodyScope
- BodyScope = Atr ';' BodyScope
- BodyScope = Command BodyScope
- BodyScope = Return Atr ';' BodyScope
- BodyScope = ε

- **List of Expressions:**

- Lec = Fc
- Lec = Lec ',' Fc
- Lec = ε

- **Expression**

- $\text{Atr} = \text{'id' AtrR}$
- $\text{AtrR} = \text{'decreOp' ';'}$
- $\text{AtrR} = \text{'increOp' ';'}$
- $\text{AtrR} = \text{ArrayOpt 'atrib' Fc ';'}$
- $\text{AtrR} = \text{FunCall}$
- $\text{Fc} = \text{'StringConst'}$
- $\text{Fc} = \text{Eb}$
- $\text{Eb} = \text{Tb}$
- $\text{Ebr Ebr} = \text{'orOpLog' Tb Ebr // or}$
- $\text{Ebr} = \varepsilon$
- $\text{Tb} = \text{Fb Tbr}$
- $\text{Tbr} = \text{'andOpLog' Fb Tbr // and}$
- $\text{Tbr} = \varepsilon$
- $\text{Fb} = \text{'negOp' Fb // not}$
- $\text{Fb} = \text{'boolConst'}$
- $\text{Fb} = \text{Ra Fbr}$
- $\text{Fbr} = \text{Comp Ra Fbr // low/great/eq}$
- $\text{Fbr} = \varepsilon$
- $\text{Ra} = \text{Ea Rar}$
- $\text{Rar} = \text{'eqRI' Ea Rar // equal}$
- $\text{Rar} = \text{'notEqRel' 'Ea Rar // not equal}$
- $\text{Rar} = \varepsilon$
- $\text{Ea} = \text{Ta Ear}$
- $\text{Ear} = \text{'addOp' Ta Ear}$
- $\text{Ear} = \text{'subOp' Ta Ear'}$
- $\text{Ear} = \varepsilon$
- $\text{Ta} = \text{Fa Tar}$
- $\text{Tar} = \text{'divOp' Fa Tar}$
- $\text{Tar} = \text{'multOp' Fa Tar}$
- $\text{Tar} = \varepsilon$
- $\text{Fa} = \text{'(' Eb ')}$
- $\text{Fa} = \text{'subOp' Far}$
- $\text{Fa} = \text{Far}$

- **Far = 'Id'**
- **Far = 'intConst'**
- **Far = 'floatConst'**
- **Far = ε**
- **Comp = 'greRel'**
- **Comp = 'lowRel'**
- **Comp = 'greEqRel'**
- **Comp = 'lowEqRel'**