UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA

Compiladores
CS3025
Pregrado
2025-1

Laboratorio 13

Objetivo

Elaborar el parser y visitor print de una gramática con funciones.

Programa

Se tiene implementado la siguiente gramática,

- Program ::= Body
- Body ::= VarDecList StmtList
- VarDecList ::= (VarDec)*
- VarDec ::= var Type VarList **;**
- Type ::= id
- VarList :: id (, id)*
- StmtList ::= Stmt **(** **;** Stmt **)***
- Stmt ::=
  - **id =** CExp |
  - **print (** CExp **)**
  - **if** CExp **then** Body [**else** Body] **endif**
  - **while** CExp **do** Body **endwhile**
- CExp ::= Exp [(**<**|**<=**|**==**) Exp]
- Exp ::= Term ((**+** | -) Term)*
- Term ::= Factor ((*|/) Factor)*
- Factor ::= **id** | **Num** | **Bool** | **(** Exp **)** | **ifexp (**CExp **,** CExp **,** CExp **)**
- Bool **::= true** | **false**

Problema 1

Elaborar el parser y visitor print de una gramática con funciones.

- Program ::= VarDecList FunDecList
- VarDecList ::= (VarDec)*
- FunDecList ::= (FunDec)+
- FunDec ::= **fun** Type id **(**[ParamDecList] **)** Body **endfun**
- Body ::= VarDecList StmtList
- ParamDecList ::= Type id (, Type id)*
- VarDec ::= **var** Type VarList **;**
- Type ::= id
- VarList :: id (,id)*
- StmtList ::= Stmt **(** **;** Stmt **)***
- Stmt ::= **id =** CExp |
  - **print (** CExp **)**
  - **if** CExp **then** Body [**else** Body] **endif**
  - **while** CExp **do** Body **endwhile**
  - **for** id **in range (** CExp **,** CExp **,** CExp **)** Body **endfor**
  - **return (** [CExp] **)**
- CExp ::= Exp [(**<**|**<=**|**==**) Exp]
- Exp ::= Term ((**+** | -) Term)*
- Term ::= Factor ((*|/) Factor)*
- Factor ::= id | Num **|** Bool | **(** Exp **)** | **ifexp (** CExp **,** CExp **,** CExp **)** | **id (** [ArgList] **)**
- ArgList ::= CExp (, CExp)*
- Bool **::= true** | **false**

Ejemplo

```
var int a, b;

fun void main()
  var int y;
  y = 10;
  a = 5;
  print(a+y);
  if (true) then
    var int a;
    a=100;
    print(a+y)
  endif;
  print(a+y);
  return ()
endfun
```

```
var int a, x;

fun int suma(int x)
  var int accum;
  accum = 0;
  while 0 < x do
    accum = accum + x; x = x - 1
  endwhile;
  print(accum);
  return (accum)
endfun

fun void main()
  var int y;
  x = 5;
  y = suma(3);
  print(y);
  return ()
endfun
```