

Sessions avaluació continuada CP1 (2011-2012)

Tema 3: Anàlisi sintàctica

1. Descripció de l'exercici

Donada una gramàtica en BNF d'un llenguatge imperatiu anomenat LS, i la implementació del seu analitzador sintàctic utilitzant una eina anomenada CoSel, volem ampliar les funcionalitats d'aquest llenguatge i generar el llenguatge LS+. Primer hem d'estudiar i descriure les característiques del llenguatge LS. A continuació estudiar les ampliacions demanades tenint en compte l'exemple donat i les explicacions. Per finalment dissenyar una solució. La descripció del llenguatge LS i la nova gramàtica BNF del llenguatge LS+ es presentarà als professors a la primera sessió tutoritzada de la setmana del 17 d'Octubre. Un cop presentada la solució i comentada per tots els components del grup i pel professor s'implementarà utilitzant el llenguatge CoSel i es passarà el joc de proves donat.

2. Descripció Llenguatge LS

El llenguatge LS és un llenguatge imperatiu fortament tipat.

Pot tenir declaracions de tipus de dades, variables, funcions i procediments aniuats.

No fa distinció entre majúscules i minúscules.

Feu una descripció del llenguatge LS indicant:

- **Paraules clau:** Quines
- **Definició de variables:** Com es defineixen i quines característiques tenen
- **Tipus de dades:**
- **Definició de funcions:**
- **Paràmetres:**
- **Instruccions:** Com es separen i enumerar quines hi ha i com es defineixen
- **Operadors aritmètics, lògics i de comparació:** Quines regles de precedència tenen
- Quins arbres sintàctics obtindrem si analitzem les expressions:

$A=b+c=c+d;$

If $a==b$ then if $b==c$ then $op=1$; else $op=2$;

Quines conclusions obteniu?

La descripció en forma de regles BNF la trobareu a continuació:

Sessions avaluació continuada CP1 (2011-2012)

```
<programa> ::= { <DecFun> | <DecProc> | <DecVar> | <DecTipus> }
<DecTipus> ::= Type identificador = <tipus> ";"
<tipus> ::= Integer |
           Identificador |
           Real |
           Character |
           ^ <tipus> |
           Array "[" numero "]" of <tipus> |
           Record { identificador : <tipus> ";" } "End"
<DecVar> ::= Var identificador : <tipus> ";"
<DecFun> ::= Function identificador <DecParams> : <tipus> <bloc>
<DecProc> ::= Procedure identificador <DecParams> <bloc>
<bloc> ::= "BEGIN" {
           <DecFun> |
           <DecProc> |
           <DecVar> |
           <DecTipus> |
           <instruccio>
         } "END"
<DecParams> ::= "(" [ <DecParametre> { , <DecParametre> } ] ")"
<DecParametre> ::= identificador : <tipus>
<instruccio> ::=
  <Expressio> ";" |
  if <Expressio> then <instruccio> [ else <instruccio> ] |
  while <Expressio> do <instruccio> |
  for identificador = <Expressio> to <Expressio> do <instruccio> |
  return [ <Expressio> ] ";" |
  "BEGIN" {
           <DecFun> |
           <DecProc> |
           <DecVar> |
           <instruccio> }
  "END" |
  Print <Expressio> { , <Expressio> } ";"
  ";"
<Expressio> ::= <TerBool> { || <TerBool> }
<TerBool> ::= <FacBool> { && <FacBool> }
<FacBool> ::= <ExpArit> [ ( == | != | ">" | "<" | >= | <= ) <ExpArit> ]
<ExpArit> ::= <terme> { (+|-) <terme> }
<terme> ::= <factor> { (*|/) <factor> }
<factor> ::= ! <factor> |
           - <factor> |
           "(" <Expressio> ")" |
           Numero |
           String |
           Caracter |
           Identificador (
             "(" [ <Expressio> { , <Expressio> } ] ")" |
             <acces> [= <Expressio> ]
           )
<acces> ::= { "[" <Expressio> "]" | .Identificador | ^ }
```

Sessions avaluació continuada CP1 (2011-2012)

3. Ampliació de la gramàtica

Un cop analitzat el llenguatge LS volem crear una extensió d'ell l'**LS+**.

Volem afegir les funcionalitats següents:

- a) Declaracions de múltiples variables de l'estil: **var a,b:integer;**
- b) Declaracions de múltiples camps a un record i també declaracions de múltiples paràmetres amb un únic tipus de l'estil: **c1,c2:integer**
- c) Declaracions amb inicialitzacions de variables, de l'estil:
var a:integer=10; var b:array[5] of integer={1,2,3,4,5};
Penseu que quan hi ha arrays n-dimensionals podem tenir inicialitzacions de l'estil:
={{{1,2},{3,4}},{{1,2},{3,4}}}
I el que pot anar a cada una de les posicions pot ser tota una expressió.
- d) Declaracions amb inicialitzacions de variables, de l'estil:
Type r1=record n1:integer; n2: integer; End;
Var a,b:r1=r1(10,15);
a=r1(15,30);
La primera línia inicialitza els dos records i la segona canvia els valors del record a.
Penseu que a cada posició pot anar tota una expressió, i que r1(15,30) podria ser el paràmetre d'una funció.
- e) Declaracions de funcions de l'estil dels llenguatges orientats a objecte. A on el nom i el tipus de l'objecte va davant del nom de la funció. De l'estil

Function a:integer.m(b:real):real

...

- f) Crides a funció de l'estil dels llenguatges orientats a objecte. A on primer posem el nom de la variable sobre la que volem aplicar el mètode i a continuació un punt i el nom del mètode amb els seus paràmetres. De l'estil:

a.m(4.7*f(36))

Considerarem que el paràmetre a és el primer paràmetre de la funció i el passem per referència. Per això només acceptarem variables o els camps o dimensions d'alguna variable.

- g) La instrucció with de l'estil:

with a do

/* A on a és un record i dins del with es poden utilitzar els camps del record directament com si fossin variables, i no podem definir variables dins del with amb el mateix nom que els camps del record.*/

/* dins del with poden haver definicions de funcio, procediments, variables, tipus i instruccions de tota mena*/

...

End

- h) La instrucció for la volem canviar de la següent manera:

For (inicialitzacio;condicio;increment)

Begin

...

end

On la inicialització és una assignació d'una variable o una declaració de variable amb la seva inicialització, la condició és una condició és una expressió booleana, l'increment és una expressió, normalment una assignació. De l'estil

(Var a:integer=10;a<20;a=a+1)

(a=10;a<20;a=a+1)

...

- i) L'operador potencia denotat per ****** . Hem de tenir present que aquest operador té associativitat per la dreta.
- j) Arrays amb múltiples dimensions, de l'estil: **a:array [5,5] of integer**

Sessions avaluació continuada CP1 (2011-2012)

On l'accés a aquests arrays es farà de la següent manera: $a[i,j]$, on i i j poden ser expressions matemàtiques enteres.

4. Lliuraments

Hi ha dues sessions associades a aquest enunciat una presencial i l'altre via web:

a) Sessió de la setmana del 17 d'Octubre:

Per aquesta setmana s'ha de portar a la sessió el següent:

- Descripció del llenguatge LS (Veure apartat 2)
- Resposta a la pregunta plantejada a l'apartat 2 sobre els arbres sintàctics generats i les característiques d'aquest llenguatge.
- Disseny de la gramàtica del llenguatge LS+
- Enumeració dels nous tokens requerits.

b) Sessió de la setmana del 31 d'Octubre:

Per aquesta setmana s'ha de lliurar via web la implementació de l'analitzador sintàctic del llenguatge LS+. Aquesta definició ha de complir els següents requisits:

- Adaptar-se a les característiques descrites
- Passar el joc de proves associat.

5. Implementació en CoSel

Què es necessita per implementar l'analitzador sintàctic en CoSel:

- Instal·lar CoSel (CrossVisions)
- Descarregar-se el mòdul del generador (Com.csm)
- Descarregar-se la plantilla de la implementació del parser del llenguatge LS (cp1sint.csl) i completar-la.
 - Substituir els valors del NIA i el Nom i cognoms per les dades dels components del grup.
("nom1","Cognoms1","0000001"),
("nom2","Cognoms2","0000002"),
("nom3","Cognoms3","0000003"),
("nom4","Cognoms4","0000004"),
("nom5","Cognoms5","0000005")
 - Completar la llista dels terminals utilitzats en el nostre llenguatge.
TERMINALS = + - ...
 - Completar les regles.
 - Passar l'autotest:
 - Modificar el fitxer sintactic.csl com hem indicat abans
 - Executar el cosel: cosel.cip
 - Carregar el fitxer: load sintactic.csl
 - Executar autotest: autotest()

Us donem els següents fitxers:

- Enunciat: enunciatsint1112.pdf
- Parser del llenguatge LS: sintactic.csl
- Joc de proves: dins del sintactic.csl es posarà un joc de proves que us podreu descarregar a partir del dia 18 d'octubre (ATENCIÓ AQUEST FITXER S'ACTUALITZARÀ A LA WEB A PARTIR DEL DIA 18-10-2010)