

## Compiladors (CL) GEI (2025-26)

### Pràctica 2: Compilació d'un llenguatge senzill

#### Objectiu

Fer les parts frontal i dorsal d'un compilador per un llenguatge senzill.

#### Aprendentatges

Generació de codi intermedi de tres adreces per a expressions aritmètiques i instruccions de control de flux senzilles.

#### Llenguatge font

Literals i comentaris:

- S'admeten literals enters (e.g. 123, 045) i reals (e.g. 3.1416, 0.5e-04, 1.2345e3).
- Poden haver comentaris com en Java, i.e. comentaris que comencen amb // fins a final de línia, i comentaris que es poden estendre vàries línies, iniciats amb /\* i finalitzats amb \*/.

Identificadors i expressions:

- Els identificadors i expressions tenen un tipus, que pot ser: enter (**int**) , real (**float**), Caldrà definir les variables amb el seu tipus prèviament al seu ús.  
e.g **int** a  
**float** b,c
- Les expressions aritmètiques poden estar formades per literals enters, literals reals, identificadors, parèntesis, operadors aritmètics unaris (+, -) i operadors aritmètics binaris (+, -, \*, /, %, \*\*).
- L'ordre de precedència dels operadors aritmètics binaris és: potència (\*\*) més precedència que producte, divisió i mòdul (%), producte igual precedència que divisió i mòdul, aquests major precedència que suma i resta, i suma igual precedència que resta.
- Els operadors aritmètics unaris (manteniment i canvi de signe) tenen igual precedència que la suma i la resta binàries.
- Les precedències entre operadors aritmètics s'han d'implementar a través de la definició de la gramàtica, evitant utilitzar els operadors %left i %right que proporciona el bison.

### Sentències:

- Les sentències poden ser: expressions aritmètiques, assignacions, o iteratives fixes.
- Les expressions aritmètiques i assignacions ocupen una línia.
- Les assignacions són del tipus  
 $id := expressió$   
on l'expressió és aritmètica, i  $id$  és un identificador.
- En una assignació, el tipus de l'identificador  $id$  és igual al tipus del resultat de l'expressió corresponent.
- Si s'operen expressions aritmètiques del mateix tipus, el resultat és del mateix tipus.
- Si s'operen aritmèticament enters amb reals, el tipus resultant és real.
- Els identificadors que apareguen en una expressió qualsevol han d'haver estat prèviament inicialitzats, ja que cal conèixer els seus tipus.
- La sintaxi de les sentències iteratives incondicionals és:

```
repeat expressió_aritmètica_entera do
    llista_de_sentències
done
```

El seu significat és executar la llista de sentències tantes vegades com indiqui el valor de l'expressió aritmètica entera. Podeu suposar que aquest valor és sempre més gran o igual a 1.

- Per estendre les possibilitats de les expressions aritmètiques i assignacions cal contemplar els canvis de tipus explícits.
- S'ha d'implementar tipus taules unidimensionals i el càlcul desplaçament en la generació del codi. Considereu @base de la taula en 25

La especificació de l'array:

```
int a[25]; // array de 25 elements de 0 a 24
float b[3],c[55]; //array de 3 elements de 0..2 i array de 55 elements de 0..54
```

### Programa:

- La sintaxi d'un programa és:

```
llista_de_sentències
```

### Opcions:

- Podeu afegir opcions diverses com taules multidimensionals, llistes, registres, apuntadors, etc. No s'admeten com a opcions les expressions booleanes ni noves sentències de flux de control diferents a les definides en aquest document.

## Codi de tres adreces

En el document *C3A.pdf* es descriu el llenguatge intermedi de codi de tres adreces de forma completament general. Per fer aquesta pràctica no es necessita tot, sinó només un subconjunt d'aquest llenguatge. Els detalls a tenir en compte són:

- El programa és pot considerar directament com format per un únic subprograma.
- Els tipus de dades Long i Double no s'utilitzen.
- Com no hi ha àmbits, totes les variables es poden considerar com a locals.
- La instrucció START no és necessària.
- El programa finalitza amb la instrucció HALT.

- No existeix funció potència.
- Els únics subprogrames que es necessiten són els procediments predefinits PUT.
- Per a la implementació de les taules unidimensionals cal utilitzar consultes o assignacions desplaçades
- Com no hi ha apuntadors, no cal utilitzar les proposicions de codi de tres adreces corresponents.

## Sortides

Sortida principal:

- La sortida ha de ser un programa en codi de tres adreces (C3A), respectant la descripció del C3A que es troba en el document *C3A.pdf*.
- Generació del codi de tres adreces per a les expressions aritmètiques i per a les sentències de flux de control.
- Les sentències que són només expressions (i.e. sense assignació) indiquen que es vol imprimir el seu valor per pantalla (i.e. utilitzant els procediments C3A predefinits PUT).
- Les expressions aritmètiques que només involucren literals es poden avaluar en memòria en temps de compilació, sense generar C3A pel seu càlcul.

Altres sortides:

- Convé anar guardant en un arxiu de log cada producció de la gramàtica reconeguda, per així controlar el progrés i correcció de les analisis lèxica i sintàctica. També es pot guardar al mateix arxiu qualsevol altre missatge informatiu.
- S'ha d'evitar barrejar la sortida C3A de la sortida de log.
- Quan hi ha un error al codi font s'ha d'emetre un missatge d'error indicant la posició actual a l'arxiu font, i si l'error és lèxic, sintàctic o semàntic. Mentre més informació es doni dels errors, millor.

## Exemple

Entrada:

```
// expressions
x := 5.0
i := 3
y := i + i * 3 - x

// factorial
n := 10
fact := 1
i := 2
repeat n - 1 do
    fact := fact * i
    i := i + 1
done
fact
```

Sortida (podria ser lleugerament diferent):

```

1: x := 5.0
2: i := 3
3: $t01 := i MULI 3
4: $t02 := i ADDI $t01
5: $t03 := I2F $t02
6: $t04 := $t03 SUBF x
7: y :=$t04
8: n := 10
9: fact := 1
10: i := 2
11: $t05 := n SUBI 1
12: $t06 := 0
13: $t07 := fact MULI i
14: fact := $t07
15: $t08 := i ADDI 1
16: i := $t08
17: $t06 := $t06 ADDI 1
18: IF $t06 LTI $t05 GOTO 13
19: PARAM fact
20: CALL PUTI, 1
21: HALT

```

## Lliurament

- Aquesta pràctica és individual.
- El lliurament es farà via Moodle, en les dates indicades al mateix.
- Cal lliurar un arxiu comprimit que contingui:
  - tot el codi font (sense arxius generats en el procés de compilació)
  - exemples i la seva sortida corresponent
  - scripts o Makefile per compilar-ho tot, executar el programa amb els exemples, i netejar-ho tot llevat dels arxius font i els d'exemple.
  - arxiu README (en format txt o pdf) amb instruccions per la compilació i execució, i descripció de tot allò que vulgueu destacar de la vostra pràctica (decisions de disseny, funcionalitat addicional, limitacions, etc.).
- El nom de l'arxiu lliurat ha de contenir el vostre nom i primer cognom.