

# Compiladors: Examen de laboratori.

28 de maig de 2019

**ATENCIÓ:** Al Racó trobareu els jocs de proves i codi necessari per a fer l'examen. Llegiu les instruccions del final de l'enunciat per veure d'on descarregar-lo.

**ATENCIÓ:** Cal entregar l'examen en un fitxer **.tgz** pujat al Racó. Llegiu les instruccions del final de l'enunciat per veure com generar-lo.

Els tres primers punts de la nota de laboratori s'obtenen amb els jocs de proves de la pràctica base. La resta s'obtenen superant els jocs de proves següents.

## 1 Màxim d'un array (4 punts)

Volem afegir a l'ASL l'operació **.max**, que retorna el valor de l'element més gran d'un array que sempre considerarem no buit. L'operador es pot aplicar a arrays de caràcters o de tipus numèrics, però no a arrays de booleans. L'operador **.max** té la màxima prioritat.

Per exemple:

```
func main()
  var a : array [10] of int
  var b : array [10] of float
  var i : int
  var x : float

  i = 2*a.max + 1;
  while b.max < 3.0*a[i] do
    b[i] = b[i+1]*5.0;
    i = i+1;
  endwhile
endfunc
```

**Joc de proves 1 (0.5 punts).** Començarem modificant la gramàtica per afegir l'operació postfixa **.max** al llenguatge.

Un cop fets els canvis a la gramàtica, el primer joc de proves:

```
1 func main()
2   var a : array [10] of int
3   var i,j : int
4
5   i = 2*a.max + 1;
6   while a.max < 3*a[i] do
7     a[i] = a[i+1]-5.0;
8     if a[i]*7 then
9       j = j + a.max;
10      i = j>a.max and x;
11    endif
12  endwhile
13 endfunc
```

hauria de produir els errors:

Line 7:9 error: Assignment with incompatible types.  
Line 8:4 error: Instruction 'if' requires a boolean condition.  
Line 10:8 error: Assignment with incompatible types.  
Line 10:22 error: Identifier 'x' is undeclared.

**Joc de proves 2 (0.5 punts).** A continuació afegirem una primera comprovació semàntica: l'operació `.max` només es pot aplicar a arrays, sempre que no siguin de booleans.

Amb aixó, el joc de proves:

produeix la sortida:

```

1 func main()
2   var Ai: array[10] of int
3   var Af: array[10] of float
4   var Ac: array[10] of char
5   var Ab: array[10] of bool
6   var i : int
7
8   i = 0;
9   while i < 10 do
10    Af[i] = Ai.max*Af.max;
11    Ac[i] = Ac.max;
12    Ab[i] = Ab.max;
13    i = i + 1;
14  endwhile
15  if i > i.max then
16    i = x.max;
17  endif
18 endfunc

```

```

Line 12:14 error: Operator '.max' with incompatible types.
Line 15:10 error: Operator '.max' with incompatible types.
Line 16:8 error: Identifier 'x' is undeclared.

```

**Joc de proves 3 (1 punt).** Ara comprovarem que es calcula bé el tipus del resultat de l'operació `.max`.

Així arribem al  
tercer joc de proves:

que ha de produir la sortida:

```

1 func main()
2   var Ai: array[10] of int
3   var Af: array[10] of float
4   var Ac: array[10] of char
5   var Ab: array[10] of bool
6   var i,k2 : int
7
8   i = 0;
9   while i < 10 do
10    k2 = Ai.max*Af.max;
11    if k2 > 'c' then
12      write Ac.max < Ac[i]; write "\n";
13    else
14      if Ab.max then
15        i = i + k2.max;
16        i = Ab.max - i;
17      endif
18    endif
19  endwhile
20 endfunc

```

```

Line 10:7 error: Assignment with incompatible types.
Line 11:10 error: Operator '>' with incompatible types.
Line 14:11 error: Operator '.max' with incompatible types.
Line 15:18 error: Operator '.max' with incompatible types.
Line 16:7 error: Operator '.max' with incompatible types.
Line 16:12 error: Operator '-' with incompatible types.

```

**Joc de proves 4 (1 punt).** Ja podem generar el codi per a l'operació `.max`. Començarem amb els arrays d'enters i caràcters (que internament es representen com enters):

El següent joc de proves:

ha de produir el resultat:

<pre> 1 func main() 2   var before : int 3   var a : array[10] of int 4   var after : int 5   var i : int 6 7   before = 199; 8   after = 299; 9   a[0]=5; a[1]=3; a[2]=8; a[3]=9; a[4]=7; 10  a[5]=3; a[6]=4; a[7]=6; a[8]=5; a[9]=8; 11  i = 0; 12  while i &lt; 10 do 13    write "a["; write i; write "]="; 14    write a[i]; write "\n"; 15    i = i + 1; 16  endwhile 17  write "a) a.max="; write a.max; write "\n"; 18  a[0] = a[0]*5; 19  write "b) a.max="; write a.max; write "\n"; 20  a[9] = a[9] + a[0]; 21  write "c) a.max="; write a.max; write "\n"; 22 endfunc </pre>	<pre> a[0]=5 a[1]=3 a[2]=8 a[3]=9 a[4]=7 a[5]=3 a[6]=4 a[7]=6 a[8]=5 a[9]=8 a) a.max=9 b) a.max=25 c) a.max=33 </pre>
--	---

**Joc de proves 5 (0.5 punts).** Ara generarem el codi també pels arrays de reals (ja que cal usar una comparació diferent a la `tvm`):

Amb aixó passarem el darrer joc de proves:

que escriu la sortida:

<pre> 1 func init(aC : array[6] of float) 2   aC[0]=-5.4; aC[1]=-3.7; aC[2]=-8.2; 3   aC[3]=-5.1; aC[4]=-7.9; aC[5]=-10; 4 endfunc 5 6 func main() 7   var a : array[6] of float 8   var b : array[3] of char 9   var i : int 10 11  init(a); 12  b[0] = 'h'; b[1] = 'b'; b[2] = 'o'; 13  i = 0; 14  while i &lt; 6 do 15    write "a["; write i; write "]="; 16    write a[i]; write "\n"; 17    i = i + 1; 18  endwhile 19  i = 1000; 20  while i &lt; 1003 do 21    write "b["; write i-1000; write "]="; 22    write b[i-1000]; write "\n"; 23    i = i + 1; 24  endwhile 25  write "a) a.max="; write a.max; write "\n"; 26  write "b) b.max="; write b.max; write "\n"; 27 endfunc </pre>	<pre> a[0]=-5.4 a[1]=-3.7 a[2]=-8.2 a[3]=-5.1 a[4]=-7.9 a[5]=-10 b[0]=h b[1]=b b[2]=o a) a.max=-3.7 b) b.max=o </pre>
---	---

**Joc de proves 6 (0.5 punts).** A continuació, comprovem que l'operació `.max` funciona també amb arrays rebuts com a paràmetres:

Amb aixó passarem el darrer joc de proves:

que escriu la sortida:

```
1 func f(a : array[3] of int, b : int) : int      a) f(a, 23)=37
2   return b + a.max;                            b) g('z', b)=o
3 endfunc                                          c) g('e', b)=a
4
5 func g(a : char, b: array[3] of char) : char
6   if a > b.max then
7     return b.max;
8   else
9     return b[2];
10  endif
11 endfunc
12
13 func main()
14   var a : array[3] of int
15   var b : array[3] of char
16
17   a[0] = 7;   a[1] = 3;   a[2] = 14;
18   b[0] = 'o'; b[1] = 'c'; b[2] = 'a';
19
20   write "a) f(a, 23)="; write f(a, 23); write "\n";
21   write "b) g('z', b)="; write g('z', b); write "\n";
22   write "c) g('e', b)="; write g('e', b); write "\n";
23 endfunc
```

## 2 Accés segur a arrays (3 punts)

El segon exercici consisteix en dotar a l'ASL de validació en temps d'execució del rang de l'accés a arrays.

Per exemple, el següent codi:

```
func main()
  var a : array [10] of int
  var i,s : int
  s = 0; i = 0;
  while (i<20) do
    s = s + a[i];
    i = i+1;
  endwhile
  write s;
endfunc
```

hauria de donar la sortida:

Runtime Error: Line 6:14. Access out of range: a[i]

Llegiu la *informació important* al final de l'enunciat per saber on trobareu codi que us permetrà generar fàcilment aquest error.

En aquest cas, no cal fer cap modificació a la gramàtica ni a la comprovació de tipus. Només cal que cada accés a un array generi un t-codi addicional que comprovi si el valor de l'índex està dins dels límits de la mida de l'array, i en cas contrari, doni un missatge d'error i aturi l'execució del programa.

**Joc de proves 7 (0.5 punts).** El primer pas serà provar un programa amb errors semàntics:

El primer joc de proves:	genera els següents errors:
1 func f(a: array[10] of float, 2 c : float) : float 3 var i : int 4 5 return (a[-23]+a[23]) / (2.0*i); 6 endfunc 7 8 func main() 9 var A: array[10] of int 10 var k, z: int 11 12 z = k*f(A, k) % 2; 13 write z + A; write "\n"; 14 write (v / 4.5) > 'c'; 15 endfunc	Line 12:10 error: Parameter #1 with incompatible types in call to 'f'. Line 12:16 error: Operator '%' with incompatible types. Line 13:10 error: Operator '+' with incompatible types. Line 14:9 error: Identifier 'v' is undeclared. Line 14:18 error: Operator '>' with incompatible types.

**Joc de proves 8 (1 punt).** A continuació generarem codi amb accés segur quan l'accés a l'array apareix en una *right expression*. En primer lloc comprovant que l'índex accedit no és negatiu.

Usarem el següent codi:

que ha de donar la sortida:

<pre> 1 func main() 2   var a : array[10] of int 3   var i,x : int 4 5   i = 5; a[2] = 2222; 6   while i &gt; -5 do 7     x = a[i]; 8     write "a["; write i; 9     write "]" ok"; 10    if i == 2 then write a[i]; endif 11    write "\n"; 12    i = i - 1; 13  endwhile 14  write "bad message!"; write "\n"; 15 endfunc </pre>	<pre> a[5] ok a[4] ok a[3] ok a[2] ok2222 a[1] ok a[0] ok Runtime Error: Line 7:8. Access out of range: a[i] </pre>
--	---

**Joc de proves 9 (0.5 punts).** I en segon lloc comprovant que l'índex tampoc sobrepassa el límit superior.

Usarem el següent codi:

que ha de donar la sortida:

<pre> 1 func main() 2   var a : array[10] of int 3   var i,x : int 4 5   i = 5; 6   while i &lt; 10 do 7     x = a[i]; 8     write "a["; write i; 9     write "]" ok"; write "\n"; 10    i = i + 1; 11  endwhile 12  write a[10]; 13  write "bad message!"; write "\n"; 14 endfunc </pre>	<pre> a[5] ok a[6] ok a[7] ok a[8] ok a[9] ok Runtime Error: Line 12:8. Access out of range: a[10] </pre>
---	---

**Joc de proves 10 (0.5 punts).** A continuació comprovarem que els índexs hauran de ser vàlids també quan volem assignar a una posició d'un array (*left expression*).

Usarem el següent codi:

que ha de donar la sortida:

<pre> 1 func main() 2   var a : array[10] of int 3   var i,x : int 4 5   i = 5; 6   while i &lt; 10 do 7     a[i] = i; 8     write "a["; write i; 9     write "]="; write a[i]; write "\n"; 10    i = i + 1; 11  endwhile 12  a[10] = 9999; 13  write "bad message!"; write "\n"; 14 endfunc </pre>	<pre> a[5]=5 a[6]=6 a[7]=7 a[8]=8 a[9]=9 Runtime Error: Line 12:2. Access out of range: a[10] </pre>
---	--

**Joc de proves 11 (0.5 punts).** Finalment, tractarem l'accés incorrecte quan es troba en una funció que rep l'array com a paràmetre.

El codi d'aquest joc de proves:

ha de donar la sortida:

```
1 func init(aI : array[10] of int,
2         k : int)
3     var i : int
4
5     i = 0;
6     while i < 10 do
7         aI[i] = i*i + k; i = i + 1;
8     endwhile
9 endfunc
10
11 func f(a : array[10] of int)
12     var i,x : int
13
14     i = 0;
15     while i < 15 do
16         x = a[i+1];
17         write "a["; write i+1;
18         write "]="; write x; write "\n";
19         i = i + 1;
20     endwhile
21     write "bad message!"; write "\n";
22 endfunc
23
24 func main()
25     var b,c: array[10] of int
26
27     init(b, 5);
28     c = b;
29     f(c);
30 endfunc
```

a[1]=6  
a[2]=9  
a[3]=14  
a[4]=21  
a[5]=30  
a[6]=41  
a[7]=54  
a[8]=69  
a[9]=86  
Runtime Error: Line 16:8. Access out of range: a[i+1]

## Informació important

**CODI PER A L'EXAMEN:** Al Racó (`examens.fib.upc.edu`) trobareu un fitxer `examen.tgz` amb el següent contingut:

- Codi necessari per a l'examen
  - `RuntimeError.h`: Funció que genera el codi d'error a emetre a l'exercici 2, quan es detecta un índex fora de rang.  
Feu `#include "RuntimeError.h"` des de `CodegenListener.cpp` (o copieu-hi la funció).
  - Directori `common` amb lleugeres modificacions necessàries per a l'examen. Useu el `common` proporcionat, sense modificar res.
  - `main.cpp` original de la pràctica, que s'usarà per a la correcció automàtica. No useu mains modificats.
- `jps`: Subdirectori amb jocs de proves (`jp_chkt_XX.asl` i `jp_genc_YY.asl`), i la seva corresponent sortida esperada (`jp_chkt_XX.err` per als jocs de proves de validació sintàctica i semàntica, `jp_genc_YY.in/.out` per als jocs de proves de generació de codi). En els JPs de generació, no es compara el codi generat, sinó la sortida que produeix la tVM en executar-lo.
- `avalua.sh`: Script que executa tots els jocs de proves i compara els resultats amb la sortida esperada.
- `empaqueta.sh`: Script que crea un fitxer `examen-nom.cognom.tgz` amb la vostra solució. Aquest és el fitxer que cal pujar al Racó.