

# Compiladors: Examen parcial de laboratori.

21 d'abril de 2021

**ATENCIÓ:** Al Racó trobareu els jocs de proves i codi necessari per a fer l'examen. El paquet conté versions modificades dels mòduls `common/SemErrors.*` i `common/TypesMgr.*` amb els mètodes necessaris per donar els nous errors o gestionar els nous tipus que apareixen als exercicis de l'examen. **ABANS DE COMENÇAR A FER RES, llegiu les instruccions del final de l'enunciat per veure com descarregar-lo i instal·lar-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.**

**PUNTUACIÓ:** 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 específics de l'examen. La correcció és **automàtica**, a través dels jocs de proves d'aquest enunciat, més un conjunt addicional de jocs de proves privats.

**IMPORTANT:** L'examen consta de dos exercicis independents. Podeu fer-los en qualsevol ordre. Es recomana fer cada exercici incrementalment, resolent cada joc de proves abans de passar al següent.

## 1 Instruccions pack i unpack (4 punts)

Volem afegir a l'ASL les instruccions `pack` (que permet assignar una serie d'expressions als elements d'un vector), i `unpack` (que permet assignar els elements d'un vector a una serie de variables). Per exemple:

```
1 func main()
2   var a,b,c: int
3   var A: array[2] of int
4   var B: array[3] of int
5
6   unpack A into a,b;    // copiar elements de A a les variables a,b
7
8   c = A[0] + b;
9   pack a*b, a-b, c+1+A[1] into B; // omplir posicions de B amb
10                                     // el valor corresponent
11
12   unpack B into A[1], A[0]; // canviar d'ordre els elements de A
13 endfunc
```

Les instruccions `pack` i `unpack` han de complir les següents condicions, i donaran error si no se'n satisfà alguna:

- El número de variables a la dreta de l'`unpack` ha de coincidir amb la mida del vector.
- El número d'expressions a l'esquerra del `pack` ha de coincidir amb la mida del vector.

- El tipus de totes les variables a la dreta de l'**unpack** ha de ser compatible amb el tipus dels elements del vector.
- El tipus de totes les expressions a l'esquerra del **pack** ha de ser compatible amb el tipus dels elements del vector.

**Joc de proves 1 (0.5 punts).** Començarem modificant només la gramàtica per afegir les operacions **pack** i **unpack** al llenguatge, sense fer cap comprovació per ara.

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

```

1 func main()
2   var a,b,c: int
3   var A: array[2] of int
4   var B: array[3] of int
5   var C: array[5] of char
6   var x,y : float
7
8   unpack A into a,b;
9   c = A[x+1] + b;
10  pack a*b, a+2, c-1+A[1] into B;
11  unpack A into A[1], A[0];
12  unpack A into B[b-1], B[c+1];
13  B[x] = C[y-1];
14 endfunc

```

hauria de produir la sortida:

```

Line 9:8 error: Array access with non integer index.
Line 13:4 error: Array access with non integer index.
Line 13:7 error: Assignment with incompatible types.
Line 13:11 error: Array access with non integer index.

```

**Joc de proves 2 (1 punt).** A continuació comprovarem que el número d'elements coincideix amb la mida de l'array, tant al **pack** com a l'**unpack**. També cal assegurar-se que es visiten les expressions i variables implicades en un **pack** o **unpack** per tal que l'arbre tingui les decoracions necessaries:

Així arribem al tercer joc de proves:

```

1 func main()
2   var a,b,c: int
3   var A: array [2] of int
4   var B: array [3] of int
5   var C: array [5] of char
6   var x,y: float
7   var t: char
8
9   unpack A into a,b,c;
10  c = A[0] + b;
11  pack a*b, a+true, c-1+A[1] into B;
12  unpack A into A[1], A[0], A[2];
13  unpack A into B[b-1], B[x+1];
14  pack 'a', 'b', t, 'c', t into C;
15  pack 'a', 'b', t, 'c' into C;
16 endfunc

```

que ha de produir la sortida:

Line 9:2 error: The number of items mismatch wrt. array size in pack/unpack.  
Line 11:13 error: Operator '+' with incompatible types.  
Line 12:2 error: The number of items mismatch wrt. array size in pack/unpack.  
Line 13:26 error: Array access with non integer index.  
Line 15:2 error: The number of items mismatch wrt. array size in pack/unpack.

**Joc de proves 3 (0.5 punts).** El següent joc de proves comprova que la variable sobre la que actua el pack/unpack sigui realment un array.

Un cop fets els canvis, el segon joc de proves:

```
1 func main()
2   var a,b,c: int
3   var A: array [2] of int
4   var B: array [3] of int
5   var C: array [5] of char
6   var x,y : float
7
8   unpack x into a,b;
9   c = A[0] + b;
10  pack a*b, a+true, c-1+A[1] into B;
11  pack A,C,y,a into b;
12  pack a[1], b-1, 3 into B;
13  unpack A into A[1], A[0];
14  unpack A into B[b-1], B[x+1];
15  x = C[b+c-2];
16  C[y-1] = y+1;
17 endfunc
```

hauria de produir la sortida:

Line 8:2 error: Pack/unpack with non array operand.  
Line 10:13 error: Operator '+' with incompatible types.  
Line 11:2 error: Pack/unpack with non array operand.  
Line 12:7 error: Array access to a non array operand.  
Line 14:26 error: Array access with non integer index.  
Line 15:4 error: Assignment with incompatible types.  
Line 16:4 error: Array access with non integer index.  
Line 16:9 error: Assignment with incompatible types.

**Joc de proves 4 (1 punt).** A continuació comprovarem que el tipus de les expressions del pack siguin assignables als elements del vector (tenint en compte possibles coercions).

Amb això, el següent joc de proves:

```
1 func main()
2   var a,b,c: int
3   var A: array[2] of int
4   var B: array[3] of float
5   var C: array[5] of char
6   var x,y : float
7   var t : char
8
9   unpack A into a,b,c;
10  c = A[0] + b;
11  pack a*b, x+true, c-1.0+y into B;
12  unpack A into A[1], A[x+1], c;
13  pack a+1, 'b', x, 'c', t into C;
14  pack a, true, x+1, 'c' into A;
15 endfunc
```

ha d'escruiure:

```
Line 9:2 error: The number of items mismatch wrt. array size in pack/unpack.
Line 11:13 error: Operator '+' with incompatible types.
Line 12:2 error: The number of items mismatch wrt. array size in pack/unpack.
Line 12:24 error: Array access with non integer index.
Line 13:2 error: Item #1 with incompatible types in pack/unpack operation.
Line 13:2 error: Item #3 with incompatible types in pack/unpack operation.
Line 14:2 error: The number of items mismatch wrt. array size in pack/unpack.
Line 14:2 error: Item #2 with incompatible types in pack/unpack operation.
Line 14:2 error: Item #3 with incompatible types in pack/unpack operation.
Line 14:2 error: Item #4 with incompatible types in pack/unpack operation.
```

**Joc de proves 5 (1 punt).** Finalment, comprovarem que el tipus de les variables destí de l'unpack també coincideix amb el tipus del vector (també admetent possibles coercions).

Així passarem el darrer joc de proves:

```
1 func main()
2   var a,b,c: int
3   var A: array[2] of int
4   var B: array[3] of float
5   var C: array[5] of char
6   var x,y : float
7   var t : char
8
9   unpack A into a,b,c;
10  c = A[0] + b;
11  pack a*b, a+true, x-1+A[1] into B;
12  unpack A into A[1], A[0], x;
13  unpack A into B[b-1], B[x+1];
14  pack a+1, 'b', x, 'c', t into C;
15  pack a, true, x+1, 'c' into A;
16  unpack B into x, y, B[0];
17  unpack B into a, y, t, x;
18 endfunc
```

que produeix la sortida:

```
Line 9:2 error: The number of items mismatch wrt. array size in pack/unpack.
Line 11:13 error: Operator '+' with incompatible types.
Line 12:2 error: The number of items mismatch wrt. array size in pack/unpack.
Line 13:26 error: Array access with non integer index.
Line 14:2 error: Item #1 with incompatible types in pack/unpack operation.
Line 14:2 error: Item #3 with incompatible types in pack/unpack operation.
Line 15:2 error: The number of items mismatch wrt. array size in pack/unpack.
Line 15:2 error: Item #2 with incompatible types in pack/unpack operation.
Line 15:2 error: Item #3 with incompatible types in pack/unpack operation.
Line 15:2 error: Item #4 with incompatible types in pack/unpack operation.
Line 17:2 error: The number of items mismatch wrt. array size in pack/unpack.
Line 17:2 error: Item #1 with incompatible types in pack/unpack operation.
Line 17:2 error: Item #3 with incompatible types in pack/unpack operation.
```

## 2 Tuples a l'estil de Python (3 punts)

El segon exercici consisteix en dotar el llenguatge ASL de tuples a l'estil de python. És a dir, tipus similars a *structs* de C, però on els camps no s'identifiquen per nom sinó simplement per la seva posició. Així podriem escriure per exemple:

```
1 func main()
2   var i : int
3   var x: float
4   var b: bool
5   var t : {int, bool, float}
6   var q : {float, int}
7
8   t{0} = 3*i;
9   t{1} = q{0} > 5.2;
10  q{0} = q{1}*x-1;
11  x = t{2} + q{0};
12 endfunc
```

Les condicions correctes d'utilització de les tuples són les següents:

- Els camps de la tupla poden ser només de tipus bàsics, altrament es produeix un error sintàctic.
- Els accessos a un camp de la tupla, tant si és en una expressió o en una *left-expression* només es poden fer amb una constant entera, altrament es produeix un error sintàctic.
- Accedir a una posició inexistente a la tupla produirà un error semàntic.
- L'utilització dels camps de la tupla seguirà les mateixes comprovacions de tipus que qualsevol altra variable o expressió.

**Joc de proves 6 (0.5 punts).** El primer pas és afegir a la gramàtica el tipus tupla, i l'accés als seus camps, per ara sense fer cap comprovació de tipus.

El primer joc de proves:

```
1 func main()
2   var i : int
3   var x: float
4   var b: bool
5   var t : {int, bool, float}
6   var q : {float, int}
7
8   t{0} = 3*(i+b);
9   t{1} = q{0} > 5.2;
10  if x+i then
11    q{0} = q{1}*x-1;
12  endif
13  x = t{2} + q{0};
14  i = x*2;
15 endfunc
```

genera els errors:

```
Line 8:13 error: Operator '+' with incompatible types.
Line 10:2 error: Instruction 'if' requires a boolean condition.
Line 14:4 error: Assignment with incompatible types.
```

**Joc de proves 7 (1 punt).** A continuació cal tractar la declaració de tuples, afegint al `SymbolsVisitor` el codi necessari per crear el tipus tupla. També caldrà modificar el codi del `TypeCheckVisitor` per afegir-hi la comprovació que l'índex d'accés a un camp de la tupla està en el rang adequat.

El segon joc de proves:

```
1 func main()
2   var i : int
3   var x: float
4   var b: bool
5   var t : {int, bool, float}
6   var q : {float, int}
7
8   t{0} = 3*(i+b);
9   t{1} = q{0} > 5.2;
10  if x+i then
11    q{0} = q{4}*x-1;
12  endif
13  x = t{3} + q{0};
14  q{5} = 3.2;
15  i = x*2;
16 endfunc
```

genera els errors:

```
Line 8:13 error: Operator '+' with incompatible types.
Line 10:2 error: Instruction 'if' requires a boolean condition.
Line 11:12 error: Tuple access to a non existent field number.
Line 13:6 error: Tuple access to a non existent field number.
Line 14:2 error: Tuple access to a non existent field number.
Line 15:4 error: Assignment with incompatible types.
```

*Nota:* Al mòdul `TypesMgr` modificat trobareu els mètodes necessaris (`createTupleTy`, `isTupleTy`, `getTupleSize`, etc). Mireu el fitxer `common/TypesMgr.h` per veure la llista completa i els parametres de cada mètode.

**Joc de proves 8 (0.5 punts).** En aquest pas comprovarem que quan s'accedeix a un camp d'una variable, aquesta sigui realment una tupla

Amb això, el programa següent:

```
1
2 func main()
3   var i : int
4   var x,y: float
5   var b: bool
6   var t : {int, bool, float}
7   var q : {float, int}
8
9   t{0} = 3*(i+b);
10  x{1} = q{0} > 5.2;
11  t{1} = b{1} > 5.2;
12  if x+i then
13    q{0} = q{1}*x-1;
14    y = not t{1};
15  endif
16  x = 2*t{2} + i{0};
17  y{5} = 3.2;
18 endfunc
```

ha de donar la sortida:

```
Line 9:13 error: Operator '+' with incompatible types.
Line 10:2 error: Tuple access to a non tuple operand.
Line 11:9 error: Tuple access to a non tuple operand.
Line 12:2 error: Instruction 'if' requires a boolean condition.
Line 14:7 error: Assignment with incompatible types.
Line 16:15 error: Tuple access to a non tuple operand.
Line 17:2 error: Tuple access to a non tuple operand.
```

**Joc de proves 9 (1 punt).** Finalment, afegirem el càlcul del tipus d'un camp quan hi ha un accés a la tupla, tant si apareix en una expressió com en una *left-expression*.

El codi d'aquest joc de proves:

```
1 func f(p: {int, bool, float}, a: bool) : bool
2   var s : {bool, bool, char}
3   if a then
4     s{0} = p{1} and not p{0}<3;
5   else
6     s{1} = p{2} != 0.3;
7   endif
8   s{2} = p+3;
9   s{2} = 'c';
10 endfunc
11
12 func main()
13   var i : int
14   var x,y: float
15   var b: bool
16   var t : {int, bool, float}
17   var q : {float, int}
18
19   t{0} = 3*(i+b);
20   t{1} = 1 - q{0} * 5;
21   t{1} = q{1} > 5.2;
22   if x+i then
23     q{0} = q{1} <= x-1;
24     y = t{1};
25   endif
26   x = 2*t{2} + q{0};
27   y = q{1} - f(t, q{0}) + f(b, q{1});
28   z = i + f(t, b);
29   z = 4 - f(q, not b);
30   q{5} = 3.2;
31 endfunc
```

ha de generar els errors:

```
Line 4:21 error: Operator 'not' with incompatible types.
Line 4:29 error: Operator '<' with incompatible types.
Line 8:7 error: Assignment with incompatible types.
Line 8:10 error: Operator '+' with incompatible types.
Line 19:13 error: Operator '+' with incompatible types.
Line 20:7 error: Assignment with incompatible types.
Line 22:2 error: Instruction 'if' requires a boolean condition.
Line 23:10 error: Assignment with incompatible types.
Line 24:7 error: Assignment with incompatible types.
Line 27:11 error: Operator '-' with incompatible types.
Line 27:15 error: Parameter #1 with incompatible types in call to 'f'.
Line 27:18 error: Parameter #2 with incompatible types in call to 'f'.
Line 27:24 error: Operator '+' with incompatible types.
Line 27:28 error: Parameter #1 with incompatible types in call to 'f'.
Line 27:31 error: Parameter #2 with incompatible types in call to 'f'.
Line 28:2 error: Identifier 'z' is undeclared.
Line 28:8 error: Operator '+' with incompatible types.
Line 28:12 error: Parameter #1 with incompatible types in call to 'f'.
Line 29:2 error: Identifier 'z' is undeclared.
Line 29:8 error: Operator '-' with incompatible types.
Line 29:12 error: Parameter #1 with incompatible types in call to 'f'.
Line 30:2 error: Tuple access to a non existent field number.
```



## Informació important

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

- `parcial-lab-CL-2021.pdf`: Aquest document, amb l'enunciat i les instruccions.
- `jps`: Subdirectori amb jocs de proves (`jp_chkt_XX.asl`), i la seva corresponent sortida esperada (`jp_chkt_XX.err`).
- `common`: Subdirectori amb els mòduls auxiliars `SemErrors` i `TypesMgr` ampliat amb el codi necessari per a l'examen.
- `avalua.sh`: Script que executa tots els jocs de proves i diu si se superen o no.
- `empaqueta.sh`: Script que crea un fitxer `examen-nom.cognom.tgz` amb la vostra solució. Aquest és el fitxer que cal pujar al Racó.

### PASSOS A SEGUIR:

- Feu una còpia de les carpetes `asl` i `common` de la vostra pràctica a un nou directori `examen`.

```
mkdir examen
cp -r practica/asl practica/common examen/
```
- Canvieu al nou directori `examen`, i descomprimiu-hi el fitxer `examen.tgz` del Racó:

```
cd examen
tar -xzf examen.tgz
```

Això extreurà el contingut del paquet, **afegint** al vostre directori `examen` els fitxers llistats anteriorment.

**IMPORTANT:** Feu-ho en l'ordre especificat (primer una còpia de la vostra pràctica i després descomprimir el `.tgz`). Fer-ho en l'ordre invers causarà que us falti codi necessari a `common` i que els JPs no siguin els adequats.
- Trebal·leu normalment a la carpeta `examen/asl`.

```
cd asl
make antlr
make
...
```
- Per veure les diferències entre la sortida del vostre `asl` i la sortida esperada en un joc de proves concret de type check, podeu fer:

```
./asl ../jps/jp_chkt_XX.asl | diff -y - ../jps/jp_chkt_XX.err
```

(Podeu ignorar la línia “There are semantic errors: no code generated” que genera el main)
- Per executar tots els jocs de proves i veure si els passeu, executeu `../avalua.sh`.
- Executeu `../empaqueta.sh` per crear el fitxer d'entrega `../examen-USERNAME.tgz` que cal pujar al Racó. Els paquets creats sense usar aquest script seran qualificats com **NO PRESENTAT**.