

Compiladors: Examen parcial de laboratori.

6 d'abril de 2022

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 Instrucció de mapeig d'arrays (**3.5** punts)

Volem afegir a l'ASL una instrucció de mapeig d'arrays, que permet construir un array a partir dels elements d'un altre.

La sintaxi és: `a = [cond ? exp1 : exp2 for x in b]`

La semàntica de l'operació és que per a cada element de `b` s'avalua la condició `cond`. Si el resultat és cert, s'assigna a la posició corresponent de `a` el valor de `exp1`, i si és fals, s'hi assigna el valor de `exp2`.

Per a que la instrucció sigui correcta, s'han de complir les següents condicions:

- `a` (array destí) i `b` (array origen) són arrays de la mateixa mida.
- `cond` és una expressió booleana.
- `exp1` i `exp2` són expressions de tipus assignables als elements de `a`.
- `x` és una variable existent i els elements de `b` són d'un tipus assignable a `x`.

Observeu que es tracta d'una **instrucció** i per tant no es separable, és a dir, la llista a la dreta de l'assignació **no** és una expressió i no es pot usar en cap altre context.

Un exemple de codi:

```
1 func main()
2   var a,b,c: int
3   var q : bool
4   var A: array[10] of int
5   var B: array[10] of int
6   var C: array[10] of bool
7
8   // assignar a cada element de A l'element de la
9   // mateixa posicio en B, incrementat en 1.
10  A = [ true ? b+1 : 0 for b in B ];
11
12  // modificar els elements de B de forma diferent
13  // depenent del seu signe.
14  B = [ c>=0 ? a*c : b*(-c+1) for c in B ];
15
16  // omplir un array de booleans segons els valors
17  // dels elements de a.
18  C = [ a>c ? a%2==0 : a*b>5 for a in A ];
19
20  // assignar diferents valors als elements de B
21  // segons el contingut boolea de C.
22  B = [ q or b!=3 ? b*2 : b/5 for q in C];
23 endfunc
```

Joc de proves 1 (0.5 punts). El primer pas és estendre la gramàtica amb la nova instrucció. Creeu una regla **nova** per aquesta instrucció, i **no modifiqueu** la regla de l'assignació, així evitarem introduir errors en el codi existent.

De moment, no farem encara cap comprovació de tipus.

El primer joc de proves:

```
1 func main()
2   var a,b,c: int
3   var q : bool
4   var A: array[10] of int
5   var B: array[10] of int
6   var C: array[10] of bool
7
8   A[c] = B[q] - 1;
9   A = [ true ? b+1 : 0 for b in B ];
10  if B[q] == C[b*a] then
11    B = [ c>=0 ? a*c : b*(-c+1) for c in B ];
12  endif
13  C[3] = q or A[3.2+b]==0;
14  C = [ a>c ? a%2==0 : a*b>5 for a in A ];
15  B = [ q or b!=3 ? b*2 : b/5 for q in C];
16  b[5] = A[B[2]] + B[A];
17 endfunc
```

genera els errors:

```
Line 8:11 error: Array access with non integer index.
Line 10:7 error: Array access with non integer index.
Line 10:10 error: Operator '==' with incompatible types.
Line 13:16 error: Array access with non integer index.
Line 16:2 error: Array access to a non array operand.
Line 16:21 error: Array access with non integer index.
```

Joc de proves 2 (1 punt). A continuació començarem a comprovar el tipus dels elements de la nova instrucció. En aquest joc de proves, cal comprovar que els arrays tenen la mateixa mida i que la variable del `for` és del tipus dels elements de l'array origen.

Així, el segon joc de proves:

```
1 func main()
2   var a,b,c: int
3   var s : char
4   var q : bool
5   var A: array[10] of int
6   var B: array[10] of int
7   var C: array[5] of bool
8
9   A[c] = B[q] - a(x);
10  A = [ true ? b+1 : 0 for q in B ];
11  if B[q] == C[b*a] then
12    B = [ c>=0 ? a*c : b*(-c+1) for b in B ];
13  endif
14  C[a] = q or A[3.2+b]=='x';
15  C = [ a>c ? a%2==0 : a*b>5 for s in A ];
16  B = [ q or b!=3 ? b*2 : b/5 for q in C];
17  b[5] = A[B[2]] + B[A];
18 endfunc
```

genera els errors:

```
Line 9:11 error: Array access with non integer index.
Line 9:16 error: Identifier 'a' is not a callable function.
Line 9:18 error: Identifier 'x' is undeclared.
Line 10:2 error: Instruction 'map' with incompatible control variable.
Line 11:7 error: Array access with non integer index.
Line 11:10 error: Operator '==' with incompatible types.
Line 14:16 error: Array access with non integer index.
Line 14:22 error: Operator '==' with incompatible types.
Line 15:2 error: Instruction 'map' with incompatible control variable.
Line 15:2 error: Instruction 'map' with non arrays operands or with arrays of different size.
Line 16:2 error: Instruction 'map' with non arrays operands or with arrays of different size.
Line 17:2 error: Array access to a non array operand.
Line 17:21 error: Array access with non integer index.
```

Joc de proves 3 (1 punt). Seguint amb les comprovacions, de tipus, cal verificar també que la condició sigui booleana i que les expressions assignades siguin del mateix tipus que els elements de l'array destí.

El tercer joc de proves:

```

1 func main()
2   var a,b,c: int
3   var s : char
4   var q : bool
5   var A: array[10] of int
6   var B: array[10] of int
7   var C: array[5] of bool
8
9   A[c] = B[q] - a(x);
10  A = [ true ? b+1 : 0 for q in B ];
11  if B[q] == C[b*a] then
12    B = [ c*4.2+1 ? true : b*(-c+1) for b in B ];
13  endif
14  C[a] = q or A[3.2+b]=='x';
15  C = [ s=='z' ? a%2==0 : a*b for s in A ];
16  B = [ 5 + (q or b!=3) ? b*2 : b>0 for q in C];
17  b[5] = A[B[2]] + B[A];
18 endfunc

```

ha de produir els errors:

```

Line 9:11 error: Array access with non integer index.
Line 9:16 error: Identifier 'a' is not a callable function.
Line 9:18 error: Identifier 'x' is undeclared.
Line 10:2 error: Instruction 'map' with incompatible control variable.
Line 11:7 error: Array access with non integer index.
Line 11:10 error: Operator '==' with incompatible types.
Line 12:5 error: Instruction 'map' with incompatible assigned expression(s).
Line 12:5 error: Instruction 'map' with non-boolean condition.
Line 14:16 error: Array access with non integer index.
Line 14:22 error: Operator '==' with incompatible types.
Line 15:2 error: Instruction 'map' with incompatible assigned expression(s).
Line 15:2 error: Instruction 'map' with incompatible control variable.
Line 15:2 error: Instruction 'map' with non arrays operands or with arrays of different size.
Line 16:2 error: Instruction 'map' with incompatible assigned expression(s).
Line 16:2 error: Instruction 'map' with non arrays operands or with arrays of different size.
Line 16:2 error: Instruction 'map' with non-boolean condition.
Line 16:10 error: Operator '+' with incompatible types.
Line 17:2 error: Array access to a non array operand.
Line 17:21 error: Array access with non integer index.

```

Joc de proves 4 (0.5 punts). Ara cal tenir en compte les coercions enter-real. Les següents combinacions són vàlides i, per tant, **no** han de donar error:

- Alguna de les expressions `exp1` o `exp2` és entera, i els elements de l'array destí són reals.
- La variable del `for` és real, i els elements de l'array origen són enters.

El quart joc de proves:

```

1 func f(x : int) : float
2   var M : array[4] of int
3   var R : array[4] of float
4   var y : float
5   var i : int
6
7   R = [ y>0 ? y+1 : -i-1 for y in M ];
8   M = [ y>0 ? i+1 : -y-1 for y in M ];
9   R = [ y>0 ? y+1 : -i-1 for i in R ];
10  R = [ -y*2+1 ? i+1 : -i-1 for i in R ];
11  return R[0] + M[i];
12 endfunc
13
14 func main()
15   var a,b,c: int
16   var s : char
17   var q : bool
18   var A: array[10] of int
19   var B: array[10] of int
20   var C: array[5] of bool
21
22   A[c] = B[q] - f(x);
23   A = [ true ? b+1 : 0 for q in B ];
24   if B[q] == C[b*a] then
25     B = [ c*4.2+1 ? true : b*(-c+1) for b in B ];
26   endif
27   C[a] = q or A[3.2+b]=='x';
28   C = [ s=='z' ? a%2==0 : a*b for s in A ];
29   B = [ 5 + (q or b!=3) ? b*2 : b>0 for q in C];
30   b[5] = A[B[2]] + B[A];
31 endfunc

```

produirà els errors:

```

Line 8:3 error: Instruction 'map' with incompatible assigned expression(s).
Line 9:3 error: Instruction 'map' with incompatible control variable.
Line 10:3 error: Instruction 'map' with incompatible control variable.
Line 10:3 error: Instruction 'map' with non-boolean condition.
Line 22:7 error: Assignment with incompatible types.
Line 22:11 error: Array access with non integer index.
Line 22:18 error: Identifier 'x' is undeclared.
Line 23:2 error: Instruction 'map' with incompatible control variable.
Line 24:7 error: Array access with non integer index.
Line 24:10 error: Operator '==' with incompatible types.
Line 25:5 error: Instruction 'map' with incompatible assigned expression(s).
Line 25:5 error: Instruction 'map' with non-boolean condition.
Line 27:16 error: Array access with non integer index.
Line 27:22 error: Operator '==' with incompatible types.
Line 28:2 error: Instruction 'map' with incompatible assigned expression(s).
Line 28:2 error: Instruction 'map' with incompatible control variable.
Line 28:2 error: Instruction 'map' with non arrays operands or with arrays of different size.
Line 29:2 error: Instruction 'map' with incompatible assigned expression(s).
Line 29:2 error: Instruction 'map' with non arrays operands or with arrays of different size.
Line 29:2 error: Instruction 'map' with non-boolean condition.
Line 29:10 error: Operator '+' with incompatible types.
Line 30:2 error: Array access to a non array operand.
Line 30:21 error: Array access with non integer index.

```

Joc de proves 5 (0.5 punts). Finalment, s'incorporen casos en els que alguns dels elements de la nova operació és de tipus **error**. Per simplificar, podeu suposar que els arrays origen i destí sempre estaran declarats, i seràn arrays.

El darrer joc de proves:

```

1 func f(x : int) : float
2   var M : array[4] of int
3   var R : array[4] of float
4   var y : float
5   var i : int
6
7   R = [ y>0 ? y+1 : -i-1 for z in M ];
8   M = [ y>0 ? i+1 : -y-1 for y in M ];
9   R = [ q[4-R] ? y+1 : -i-1 for i in R ];
10  R = [ -y*2+1 ? i+1 : -i-1 for i in R ];
11  return R[0] + M[i];
12 endfunc
13
14 func main()
15   var a,b,c: int
16   var s : char
17   var q : bool
18   var A: array[10] of int
19   var B: array[10] of int
20   var C: array[5] of bool
21
22   A[c] = B[q] - f(x);
23   A = [ true ? b+1 : 0 for q in B ];
24   if B[q] == C[b*a] then
25     B = [ a[3] ? true : b*(-c+1) for b in B ];
26   endif
27   C[a] = q or A[3.2+b]=='x';
28   C = [ s=='z' ? a%2==0 : a*b for s in A ];
29   B = [ 5 + (q or b!=3) ? b*2 : b>0 for y in C ];
30   b[5] = A[B[2]] + B[A];
31 endfunc

```

produirà els errors:

```
Line 7:30 error: Identifier 'z' is undeclared.
Line 8:3 error: Instruction 'map' with incompatible assigned expression(s).
Line 9:3 error: Instruction 'map' with incompatible control variable.
Line 9:9 error: Identifier 'q' is undeclared.
Line 9:12 error: Operator '-' with incompatible types.
Line 10:3 error: Instruction 'map' with incompatible control variable.
Line 10:3 error: Instruction 'map' with non-boolean condition.
Line 22:7 error: Assignment with incompatible types.
Line 22:11 error: Array access with non integer index.
Line 22:18 error: Identifier 'x' is undeclared.
Line 23:2 error: Instruction 'map' with incompatible control variable.
Line 24:7 error: Array access with non integer index.
Line 24:10 error: Operator '==' with incompatible types.
Line 25:5 error: Instruction 'map' with incompatible assigned expression(s).
Line 25:11 error: Array access to a non array operand.
Line 27:16 error: Array access with non integer index.
Line 27:22 error: Operator '==' with incompatible types.
Line 28:2 error: Instruction 'map' with incompatible assigned expression(s).
Line 28:2 error: Instruction 'map' with incompatible control variable.
Line 28:2 error: Instruction 'map' with non arrays operands or with arrays of different size.
Line 29:2 error: Instruction 'map' with incompatible assigned expression(s).
Line 29:2 error: Instruction 'map' with non arrays operands or with arrays of different size.
Line 29:2 error: Instruction 'map' with non-boolean condition.
Line 29:10 error: Operator '+' with incompatible types.
Line 29:40 error: Identifier 'y' is undeclared.
Line 30:2 error: Array access to a non array operand.
Line 30:21 error: Array access with non integer index.
```


2 Structs simples (3.5 punts)

El segon exercici consisteix en dotar el llenguatge ASL del tipus `struct`. Així podriem escriure per exemple:

```
1 func main()
2   var i : int
3   var x: float
4   var b: bool
5   var t : struct {k3:int, z:bool, i:float}
6   var q : struct {q2:float, b:int}
7
8   t.k3 = 3*i;
9   t.z = q.q2 > 5.2;
10  q.q2 = q.b*x-1;
11  x = t.i + q.q2;
12 endfunc
```

Les condicions correctes d'utilització dels structs són les següents:

- Els camps del struct poden ser només de tipus bàsics, altrament es produeix un error sintàctic.
- Declarar dos camps amb el mateix nom dins del mateix struct produeix un error semàntic.
- Accedir a un camp inexistent d'un struct produirà un error semàntic.
- L'utilització dels camps del struct 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 struct, i l'accés als seus camps, per ara sense fer cap comprovació de tipus. L'operador d'accés al camp d'un struct, com el d'accés a un element d'un array, té màxima prioritat sobre els altres.

El primer joc de proves:

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

genera els errors:

```
Line 8:12 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 structs, afegint al `SymbolsVisitor` el codi necessari per crear el tipus struct quan es declara una variable. També caldrà modificar el codi del `TypeCheckVisitor` per afegir-hi la comprovació que el nom del camp en un accés al struct és un nom de camp vàlid.

El segon joc de proves:

```
1 func main()
2   var i : int
3   var x: float
4   var b: bool
5   var t : struct {u:int, h4:bool, f:float}
6   var q : struct {f:float, i:int}
7
8   t.u = 3*(i+b);
9   t.h4 = q.f > 5.2;
10  if x+i then
11    q.f = q.h * x-1;
12  endif
13  x = t.y + q.f;
14  q.p = 3.2;
15  i = x*2;
16 endfunc
```

genera els errors:

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

Nota: Al mòdul `TypesMgr` modificat trobareu els mètodes necessaris (`createEmptyStructTy`, `addStructField`, `isStructTy`, `getStructFieldTy`, etc). Mireu el fitxer `common/TypesMgr.h` per veure la llista completa i els paràmetres de cada mètode. Els mètodes `equalTypes` i `copyableTypes` ja han estat degudament adaptats per tractar també structs.

Joc de proves 8 (1 punt). En aquest pas comprovarem que quan s'accedeix a un camp d'una variable, aquesta sigui realment un struct, i obtindrem el tipus i l'atribut `IsLValue` del camp accedit.

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 : struct {m:int, j:bool, k:float}
7   var q : struct {w2:float, q:int}
8
9   t.m = 3*(i+b);
10  x.h = q.w2 > 5.2;
11  t.j = b.k > 5.2;
12  if x+i then
13    q.w2 = q.q*x - 1;
14    y = not t.j;
15  endif
16  x = 2*t.k + i.y;
17  y.fds = 3.2;
18 endfunc
```

ha de donar la sortida:

```
Line 9:12 error: Operator '+' with incompatible types.
Line 10:2 error: Struct access to a non struct operand.
Line 11:8 error: Struct access to a non struct operand.
Line 12:2 error: Instruction 'if' requires a boolean condition.
Line 14:7 error: Assignment with incompatible types.
Line 16:14 error: Struct access to a non struct operand.
Line 17:2 error: Struct access to a non struct operand.
```

Joc de proves 9 (0.5 punts). Ara comprovarem que el càlcul del tipus en l'accés a un camp d'un struct s'ha fet correctament, tant si apareix en una expressió com en una *left-expression*.

El codi d'aquest joc de proves:

```
1 func f(a: bool) : bool
2   return a > 1;
3 endfunc
4
5 func main()
6   var i : int
7   var x,y: float
8   var b: bool
9   var t : struct {x1:int, x2:bool, x3:float}
10  var q : struct {a1:float, a2:int}
11
12  t.x1 = 3*(i+b);
13  t.x2 = 1 - q.a1 * 5;
14  t.x2 = q.a2 > 5.2;
15  if x+i then
16    q.a1 = q.a2 <= x-1;
17    y = t.x2;
18  endif
19  x = 2*t.x3 + q.a1;
20  y = q.a2 - f(q.a1);
21  if f(t.x2) then
22    b = t.x2 or x>0;
23  endif
24  z = i + f(not b);
25  q.z6 = 3.2;
26 endfunc
```

ha de generar els errors:

```
Line 2:11 error: Operator '>' with incompatible types.
Line 12:13 error: Operator '+' with incompatible types.
Line 13:7 error: Assignment with incompatible types.
Line 15:2 error: Instruction 'if' requires a boolean condition.
Line 16:10 error: Assignment with incompatible types.
Line 17:7 error: Assignment with incompatible types.
Line 20:11 error: Operator '-' with incompatible types.
Line 20:15 error: Parameter #1 with incompatible types in call to 'f'.
Line 24:2 error: Identifier 'z' is undeclared.
Line 24:8 error: Operator '+' with incompatible types.
Line 25:2 error: Struct access to a non existent field name.
```

Joc de proves 10 (0.5 punts). Finalment, en la declaració d'un struct, també cal comprovar que no hi hagi noms de camps repetits. En cas que hi siguin, els camps amb noms repetits s'ignoren i només es declara el primer que apareix amb aquell nom.

Amb això, el programa següent:

```
1 func g(p: struct {x1:int, x2:bool, x3:float}, a: bool) : int
2   var s : struct {b1:bool, b2:bool, a:char}
3   if a then
4     s.b1 = p.x2 and not p.x1<3;
5   else
6     s.b2 = p.x3 != 0.3;
7   endif
8   s.a = p+3;
9   s.a = 'c';
10 endfunc
11
12 func main()
13   var i : int
14   var x,y: float
15   var b: bool
16   var t : struct {u:int, h4:bool, u:bool, t:float}
17   var p1 : struct {x1:int, x2:bool, x3:float}
18   var p2 : struct {i:int, b:bool, f:float}
19   var q,q2 : struct {f:float, i:int}
20
21   t.u = 3*(i+b);
22   t.h4 = q.f > 5.2;
23   if t.u then
24     q.f = q.h * x-1;
25   endif
26   t.u = true;
27   x = t.y + t.u;
28   q.f = q.i + 3.2;
29   i = x*2 + q;
30   y = g(b, q.a2);
31   if g(p2, p2.b) then
32     b = p1.x2 or x>0;
33   endif
34   z = i + g(p1, b) - g(q, p2.b);
35   z1.k = 33 > t;
36   t.t = z2.n1 * 5;
37   q = p1;
38   q2 = q;
39 endfunc
```

ha de donar la sortida:

Line 4:21 error: Operator 'not' with incompatible types.
Line 4:29 error: Operator '<' with incompatible types.
Line 8:6 error: Assignment with incompatible types.
Line 8:9 error: Operator '+' with incompatible types.
Line 16:10 error: Struct with some field already declared.
Line 21:12 error: Operator '+' with incompatible types.
Line 23:2 error: Instruction 'if' requires a boolean condition.
Line 24:11 error: Struct access to a non existent field name.
Line 26:6 error: Assignment with incompatible types.
Line 27:6 error: Struct access to a non existent field name.
Line 29:4 error: Assignment with incompatible types.
Line 29:10 error: Operator '+' with incompatible types.
Line 30:8 error: Parameter #1 with incompatible types in call to 'g'.
Line 30:11 error: Struct access to a non existent field name.
Line 31:2 error: Instruction 'if' requires a boolean condition.
Line 31:7 error: Parameter #1 with incompatible types in call to 'g'.
Line 34:2 error: Identifier 'z' is undeclared.
Line 34:23 error: Parameter #1 with incompatible types in call to 'g'.
Line 35:2 error: Identifier 'z1' is undeclared.
Line 35:12 error: Operator '>' with incompatible types.
Line 36:8 error: Identifier 'z2' is undeclared.
Line 37:4 error: Assignment with incompatible types.

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-2022.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
...
```

- L'accés al servidor de disc pot ser lent en alguns casos. Si és necessari podeu compilar amb l'script `fast-make.sh` que compila tot l'examen en un directori temporal local, evitant problemes de col·lapse de la xarxa.

```
cd asl
../fast-make.sh
```

- 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**.