

Examen del Primer Parcial de Compiladors 26-5-2020

Funcionament de l'examen del primer parcial

12:30 Publicació al campus virtual del enunciat de l'examen.

12:30 Reunió Teams General.

- La connexió a aquesta reunió es podrà fer sense que s'hagi d'esperar el permís del professor per connectar-se a la reunió.
- Tots els alumnes s'han de connectar a aquesta reunió.
- En aquesta reunió s'explicarà l'enunciat de l'examen i com es realitzarà.
- Una vegada acabada l'explicació, no s'ha de tancar la reunió per si el professorat ha de fer una explicació per a tota la classe.
- No silenciar la reunió per poder escoltar les explicacions del professorat.

12:30 Reunió Teams dubtes individuals:

- La connexió a aquesta reunió només es farà per consulta de dubtes sobre l'examen. Una vegada acabada la consulta, s'haurà de sortir de la reunió immediatament.
- Per entrar a la reunió l'alumne haurà d'esperar a la sala d'espera virtual fins que el professor l'atengui.
- Les consultes es faran per veu i amb la càmera activada.

Contingut de l'examen:

- L'examen serà escrit a mà. Per tant necessitareu fulls de paper i bolígraf.
- A dalt del primer full haureu de reservar un espai en blanc per posar el carnet de la UAB (NIU). D'aquesta manera, quan escanegeu o feu foto del primer full de l'examen, el vostre carnet apareixerà superposat a dalt del full.
- L'examen estarà format per algunes preguntes curtes i altres llargues, tant de teoria com de problemes.
- Podeu utilitzar tot el material de suport que considereu necessari per fer l'examen.
- Cada fulla ha de contenir nom, cognoms, NIU i número de pàgina.

14:30 o abans. Entrega de l'examen.

- L'examen escanejat o fotografiat s'entregarà com un sol fitxer a la tasca del campus virtual corresponent. El format del fitxer podrà ser un PDF o fotos JPEG de les pàgines escrites numerades de 1 en endavant. Recordeu que la primera pàgina ha d'incloure el NIU (carnet i no el número de NIU).

Exemple de la capçalera de la primera pàgina

EXAMEN PRIMER PARCIAL COMPILADORS

26-05-2020



CARNET DE LA UAB
(NIU O DNI)

NOM:

COGNOMS:

NIU:

Enunciat de l'examen

ANÀLISI LEXICOGRÀFICA

1.- (JS) Crea l'autòmat per implementar un escàner que accepta els següents símbols (2p) :

Identificador= $(l|d)^*$ Exemples: a10 casa nota10 un2tres

Numero= $d^*(\$|L|S)$ Exemples: 10 10L 10S 23L

Switch=switch Exemples: switch

Els separadors poden ser:

- Espais _
- Tabuladors \t
- Salts de línia \n
- Comentaris d'una línia de C++: // comentari

Exemples: \t \n // comentari \n

Representa el AFND, el AFD i la taula de transicions.

Estat | caràcter | nou estat

-----+-----+-----		

Posa un exemple de seqüència de símbols correcta i un altre de seqüència de símbols incorrecte. Mostra les transicions que farà el autòmat en cada crida al escàner.

Puntuación:

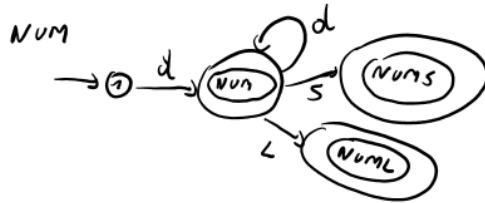
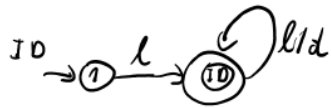
AFND 3

AFD 3

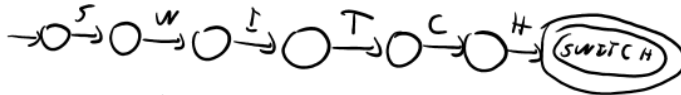
Transiciones AFND/AFD 2

Ejemplo correcto 1

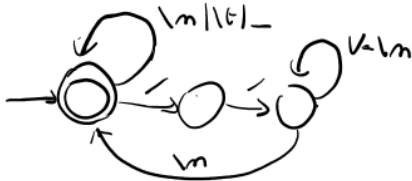
Ejemplo error 1



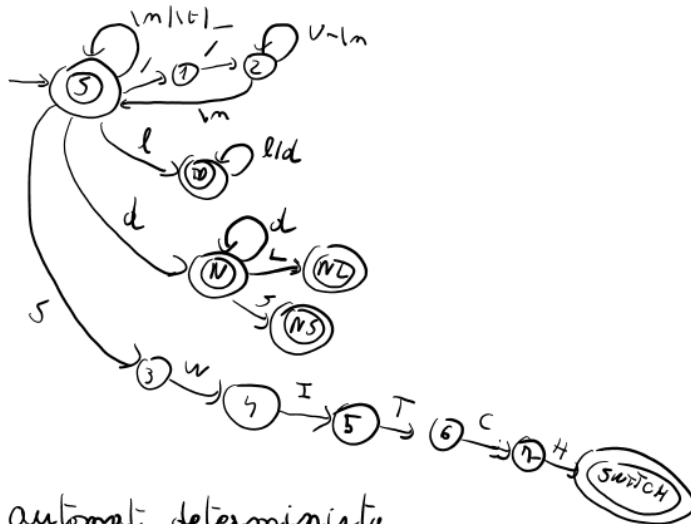
SWITCH



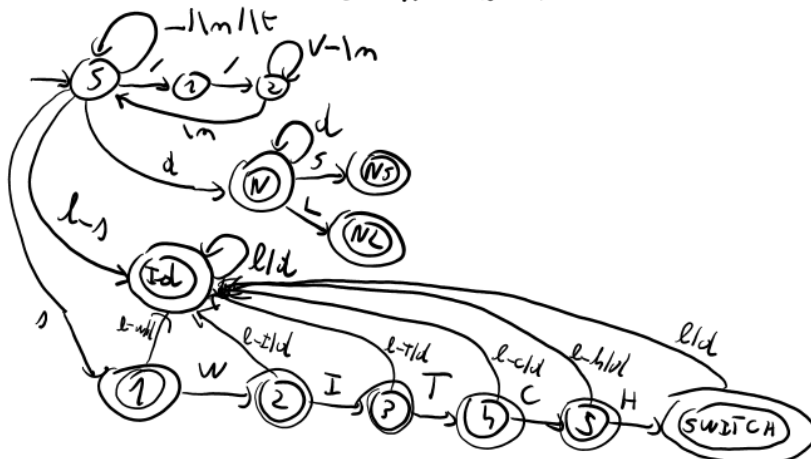
SEPARADOR*



AUTOMAT NO DETERMINISTA



automat determinista.



Transicions autòmat no determinista

$S \xrightarrow{\backslash n | \backslash t | _} S$
 $S \xrightarrow{/} 1$
 $1 \xrightarrow{/} 2$
 $2 \xrightarrow{V \backslash n} 2$
 $2 \xrightarrow{\backslash n} S$
 $S \xrightarrow{s} id, 3$
 $S \xrightarrow{l-s} id$
 $Id \xrightarrow{l|d} id$
 $S \xrightarrow{d} N$
 $N \xrightarrow{d} N$
 $N \xrightarrow{L} NL$
 $N \xrightarrow{S} NS$
 $3 \xrightarrow{w} 4$
 $4 \xrightarrow{i} 5$
 $5 \xrightarrow{t} 6$
 $6 \xrightarrow{c} 7$
 $7 \xrightarrow{h} SWITCH$

Transicions autòmat determinista

$S \xrightarrow{\backslash n | \backslash t | _} S$
 $S \xrightarrow{/} 1$
 $1 \xrightarrow{/} 2$
 $2 \xrightarrow{V \backslash n} 2$
 $2 \xrightarrow{\backslash n} S$
 $S \xrightarrow{s} 3$
 $S \xrightarrow{l-s} id$
 $Id \xrightarrow{l|d} id$
 $S \xrightarrow{d} N$
 $N \xrightarrow{d} N$
 $N \xrightarrow{L} NL$
 $N \xrightarrow{S} NS$
 $3 \xrightarrow{w} 4$
 $3 \xrightarrow{l-i|d} id$
 $4 \xrightarrow{i} 5$
 $4 \xrightarrow{l-i|d} id$
 $5 \xrightarrow{t} 6$
 $5 \xrightarrow{l-c|d} id$
 $6 \xrightarrow{c} 7$
 $6 \xrightarrow{t-h|d} id$
 $7 \xrightarrow{h} SWITCH$
 $7 \xrightarrow{l|d} id$

Seqüència correcta

$_10S$
 $S \xrightarrow{_} S \xrightarrow{1} N \xrightarrow{0} N \xrightarrow{S} NS$

Seqüència incorrecta

$/a$
 $S \xrightarrow{/} 1 \xrightarrow{a} Error$

ANÀLISI SINTÀCTICA

2.- (JB) Respecte a la següent gramàtica (2.5p) :

$\langle \text{Instrucció} \rangle ::= \langle \text{exp} \rangle \text{ " ; " } \mid \text{ if } \langle \text{exp} \rangle \text{ then } \langle \text{instrucció} \rangle \text{ [else } \langle \text{instrucció} \rangle \text{]}$
 $\langle \text{Exp} \rangle ::= \langle \text{terme} \rangle \{ \text{ " + " } \mid \text{ " - " } \} \langle \text{terme} \rangle$
 $\langle \text{terme} \rangle ::= \text{ num } \{ + \text{ num } \} \mid \text{ identificador}$

- a) Quin error hi ha a la gramàtica?. Indica on està, justifica la resposta i posa un exemple de seqüència de símbols mal analitzada.

Hay dos maneras diferentes para poder hacer la suma de dos números: desde la regla Exp y desde la regla terme, permiténdose tener dos árboles distintos para una misma secuencia de símbolos (por ejemplo: 3 + 4;)

También el Else lleva a gramàtica ambigua.

- b) Calcula les produccions de la gramàtica i els primers de tots els símbols no terminals de les produccions. Explica pas a pas el desenvolupament fins a trobar la solució.

Producciones:

$\langle \text{exp} \rangle \rightarrow \langle \text{Terme} \rangle \langle \text{exp_1} \rangle$
 $\langle \text{exp_1} \rangle \rightarrow \langle \text{exp_2} \rangle \langle \text{Terme} \rangle \langle \text{exp_1} \rangle$
 $\langle \text{exp_1} \rangle \rightarrow \lambda$
 $\langle \text{exp_2} \rangle \rightarrow -$
 $\langle \text{exp_2} \rangle \rightarrow +$
 $\langle \text{Instruccio} \rangle \rightarrow \langle \text{Instruccio_1} \rangle$
 $\langle \text{Instruccio_1} \rangle \rightarrow \text{ if } \langle \text{exp} \rangle \text{ then } \langle \text{Instruccio} \rangle \langle \text{Instruccio_2} \rangle$
 $\langle \text{Instruccio_1} \rangle \rightarrow \langle \text{exp} \rangle ;$
 $\langle \text{Instruccio_2} \rangle \rightarrow \text{ else } \langle \text{Instruccio} \rangle$
 $\langle \text{Instruccio_2} \rangle \rightarrow \lambda$
 $\langle \text{Terme} \rangle \rightarrow \langle \text{Terme_1} \rangle$
 $\langle \text{Terme_1} \rangle \rightarrow \text{ Identificador}$
 $\langle \text{Terme_1} \rangle \rightarrow \text{ Numero } \langle \text{Terme_2} \rangle$
 $\langle \text{Terme_2} \rangle \rightarrow + \text{ Numero } \langle \text{Terme_2} \rangle$
 $\langle \text{Terme_2} \rangle \rightarrow \lambda$

Primeros:

$\text{Primers}(\langle \text{exp} \rangle) = \text{ Identificador Numero}$
 $\text{Primers}(\langle \text{Instruccio} \rangle) = \text{ if Identificador Numero}$
 $\text{Primers}(\langle \text{Instruccio_1} \rangle) = \text{ if Identificador Numero}$
 $\text{Primers}(\langle \text{Instruccio_2} \rangle) = \lambda \text{ else}$
 $\text{Primers}(\langle \text{exp_1} \rangle) = \lambda - +$
 $\text{Primers}(\langle \text{Terme} \rangle) = \text{ Identificador Numero}$
 $\text{Primers}(\langle \text{exp_2} \rangle) = - +$
 $\text{Primers}(\langle \text{Terme_1} \rangle) = \text{ Identificador Numero}$
 $\text{Primers}(\langle \text{Terme_2} \rangle) = \lambda +$

3.- (JS) Com modificaries la següent gramàtica per que accepti les instruccions break i continue només dintre de bucles.
(1p)

```
<instruccio> ::=
    <Expressio> ";" |
    if <Expressio> then <instruccio> [ else <instruccio> ] |
    while <Expressio> do <instruccio> |
    <Bloc> |
    ";"
```

```
<bloc> ::=
    "{" {
        <DecFun> |
        <DecProc> |
        <DecVar> |
        <instruccio>
    }
    "}"
```

Posa un exemple d'una seqüència de símbols on es vegi com la gramàtica accepta l'ús de les instruccions continue i break dintre de bucles i un altre on no accepta break i continue fora de bucle. Construeix el corresponent arbre sintàctic per les dos seqüències (ne la erronia fins on detecta l'error). No desenvolupis el arbre per els símbols no terminals de la gramàtica que no tenen la corresponent regla BNF (exemple: <expressio> <DecFun>, etc.).

```
<instruccio> ::=
    <Expressio> ";" |
    if <Expressio> then <instruccio> [ else <instruccio> ] |
    while <Expressio> do <instruccioBucle> |
    <Bloc> |
    ";"

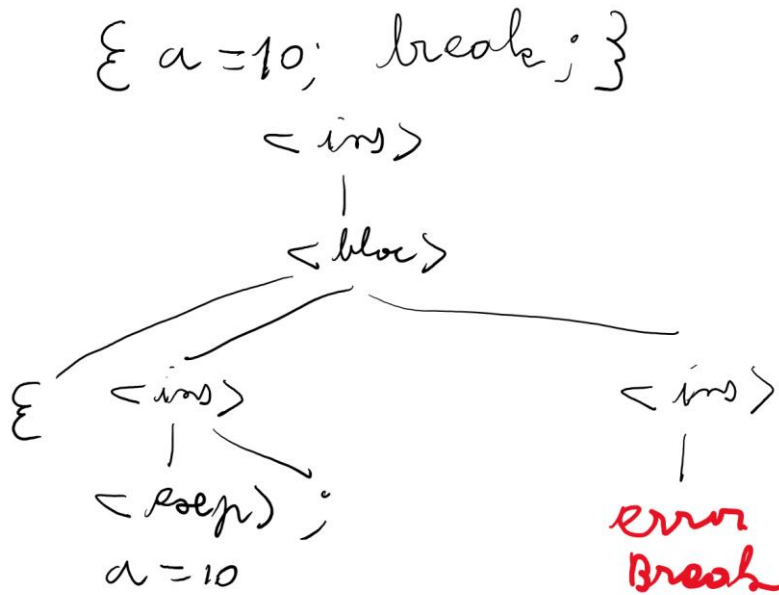
<instruccioBucle> ::=
    <Expressio> ";" |
    if <Expressio> then <instruccioBucle> [ else <instruccioBucle> ] |
    while <Expressio> do <instruccioBucle> |
    <BlocBucle> |
    ";" |
    break ";" | continue ";"

<bloc> ::=
    "{" {
        <DecFun> |
        <DecProc> |
        <DecVar> |
        <instruccio>
    } "}"

<blocBucle> ::=
    "{" {
        <DecFun> |
        <DecProc> |
        <DecVar> |
        <instruccioBucle>
    } "}"
```

Seqüència incorrecte:

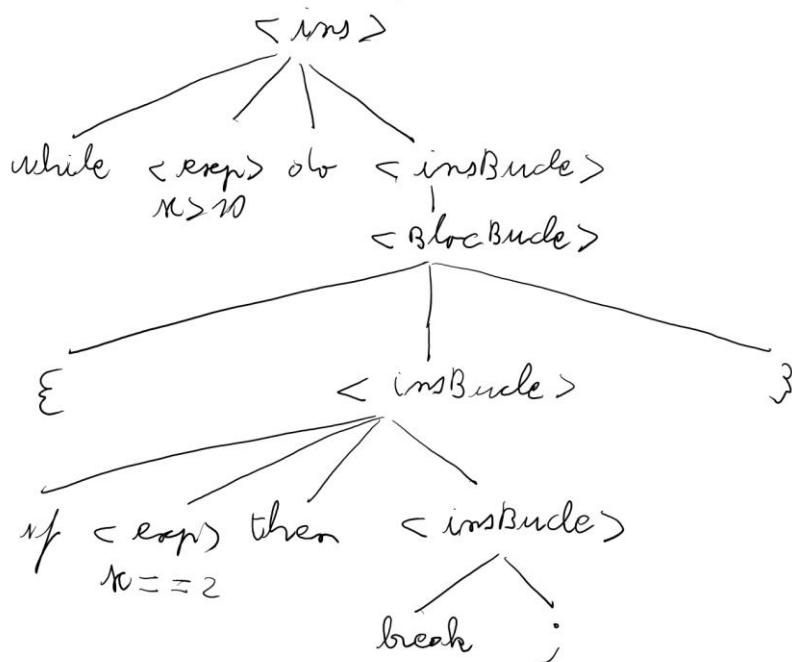
{ a=10 ; break; }



Seqüència correcta:

while x>10 do { if x==2 then break; }

while x>10 do { if x==2 then break; }



Gramàtica 6 punts

Exemple correcta 2 punts

Exemple incorrecte 2 punts.

ANÀLISI SEMÀNTICA

4.- (JS) Donada la regla BNF que a analitza sintàctica i semànticament les declaracions de variables a LOOS amb l'ampliació de declaració i inicialització de variables per desconstrucció d'objectes. Indica si funciona correctament o hi ha errors. Si hi ha errors, crea una prova que mostra cada error i fes la correcció del codi de la regla (3p).

La declaració per desconstrucció d'objectes correspon a declarar tantes variables com camps que tingui l'objecte. Aquestes variables quedaran inicialitzades amb els valors dels camps del objecte.

Exemple:

```
Class Point {
    m_X:Real;
    m_Y:Real;
    m_Etiqueta:Array [10] of character;
}
Procedure main()
{
    Var obj:Point;
    Var {x,y,etiqueta}=obj; // Declaració per desconstrucció del objecte obj.
}
```

El efecte de Var {x,y,etiqueta}=obj; es el següent:

- Declara la variable x de tipo real que inicialitza amb obj.m_X
- Declara la variable y de tipo real que inicialitza amb obj.m_Y
- Declara la variable etiqueta de tipo array [10] of character que inicialitza amb obj.m_Etiqueta

```

Rule <DecVar>::= @var nom,t;
  Var ( //començar declaració per desconstrucció sense var 1p.
    identificador#(nom)
    @TS.ComprovarDuplicat(nom);
    : <tipus(t,Unbound)>
    @var DescriptorsParametres=[];
    [<parametres(DescriptorsParametres)>]
    @t.ComprovarConstruible(DescriptorsParametres);
    @TS.Insertar(ETSVariable(nom,t));
    |
    @Var lNoms,nom;
    "{"
      identificador#(nom)
      @TS.ComprovarDuplicat(nom);
      @lNoms=[nom];
      { , identificador#(nom)
        @TS.ComprovarDuplicat(nom);
        // No detectar duplicitats dintre de la mateixa declaració 1p.
        @if (-)(n<-lNoms, n==nom)) throw SemanticError("Declaracio duplicada");

        @lNoms=lNoms++[nom];
      }
    "}"
    @Var dvExp=DescriptorValor();
    = <expressio(dvExp,true)>
    @{
      // no detecta que el resultat de l'expressió no es un objecte 1p.
      if (!TypeP(TClasse,dvExp.Tipus)) throw SemanticError("Error de tipus");
      // no verifica que es pugui llegir el resultat de l'expressió (opcional)
      if (!dvExp.PermisLectura) throw SemanticError("No es pot llegir");
      var membres=[ m | m<-dvExp.Tipus.Membres, TypeP(ETSVariable,m)];
      // El número de variables declarades ha de correspondre al de camps 1p.
      if (membres.Length!=lNoms.Length) throw SemanticError("El número de variables
no correspond al numero de camps");
      for (nom<-lNoms,camp<-membres) {
        TS.Insertar(ETSVariable(nom,camp.Tipus,true,true));
        // Comprovar que es pot copiar el camp (construcció per copia) 1p.
        camp.Tipus.ComprovarConstruiblePerCopia();
      }
    }
  )
  ";"

```

Exemples de errors:

començar declaració per desconstrucció sense var 1p.

```
{a,b}=obj;
```

No detectar duplicitats dintre de la mateixa declaració 1p.

```
Var {a,a}=obj;
```

No detecta que el resultat de l'expressió no es un objecte 1p.

```
Var {a,b}=10;
```

El número de variables declarades ha de correspondre al de camps 1p.

```
Class a { c1:integer; c2:real; }
```

```
Var obj:a;
```

```
Var {x,y,z}=obj;
```

Comprovar que es pot copiar el camp (construcció per copia) 1p.

```
Class a {
```

```
    a1:integer;
```

```
    Constructor a();
```

```
}
```

```
Class b {
```

```
    b1:a; // No es pot construir per copia
```

```
}
```

```
Var obj:b;
```

```
Var {c,d}=obj;
```

5. (JB) Taula de Simbols (1.5p) :

a) Mostra el contingut de la Taula de Simbols al punt especificat del codi:

```
=====
TAULA DE SIMBOLS =====
=====
ETSTipus: Nom=point, Descriptor=
  TClasse (point): Base=TVoid, Membres=
    ETSVariable: Nom=m_X, Tipus=TReal
    ETSVariable: Nom=m_Y, Tipus=TReal
    ETSFuncio: Nom=Imprimir, Classe=point, TipusFuncio=TFProcediment, TeCos=True, TipusRetorn=TVoid
ETSTFuncio: Nom=f, Classe=Void, TipusFuncio=TFFuncio, TeCos=True, TipusRetorn=TInteger
  Parametre 1: Nom=x, Referencia=False ,Tipus=TInteger
ETSTFuncio: Nom=main, Classe=Void, TipusFuncio=TFProcediment, TeCos=True, TipusRetorn=TVoid
===== TSSeparadorFuncio
===== TSSeparador =====
ETSTVariable: Nom=a, Tipus=point
ETSTVariable: Nom=b, Tipus=
  TArray: Mida=5, TipusElements=
    TArray: Mida=3, TipusElements=point
=====
```

b) Explica el recorregut que fa el analitzador semàntic per fer la cerca de a.m_X

1. **Búsqueda en la Tabla de Símbolos. Encuentra la a.**
2. **Dentro de PossibleLValue va a regla <Acces> y encuentra el campo m_X (llama a BuscarCamp)**
3. **No ha de salir del ámbito local**

c) Explica el recorregut que fa el analitzador semàntic per fer la cerca de m_Y dins de la funció imprimir

1. **Búsqueda en la Tabla de Símbolos. No encuentra en el ámbito en que está el m_Y**
2. **Encuentra separador función**
3. **Busca dentro de la clase BuscaClase si es miembro (lo es)**

```
class point
{
    m_X:Real;
    m_Y:Real;
    procedure imprimir();
}
procedure point::imprimir()
{
    PrintLn m_X," ",m_Y;
}
function f(x:integer):integer
{
    var c:integer;
    return x+c;
}
procedure main()
{
    var a:point;
    a.m_X=10.0;
    a.m_Y=20.0;
    var b:array[5] of array[3] of point;
    IMPRIMIR_TAUULA_DE_SIMBOLS;
    a.imprimir();
    print f(40);
}
```