



Normas del examen

- Se dispone de **3 horas** para su realización.
 - La entrega debe consistir en una serie de hojas de examen, debidamente **numeradas**, con las respuestas del alumno. Todas las hojas deben llevar los **apellidos**, el **nombre** y el número de **D.N.I.** del alumno.
 - El examen consta de **dos partes** que deben entregarse en **hojas de examen independientes** para facilitar su corrección. La parte I corresponde a las preguntas 1, 2 y 3 relativas al 1^{er} Cuatrimestre, y la parte II a las preguntas 4, 5, 6 y 7 relativas al 2º.
 - Para superarlo con éxito es necesario obtener una calificación **igual o superior a 5 puntos**.
-

PARTE I

1. **[1 punto]** Formaliza mediante definiciones regulares la siguiente descripción de una categoría léxica que se necesita en un cierto lenguaje para reconocer URLs.

Esta sería una *URL* típica: *esquema://autoridad/ruta?consulta#fragmento*.

El *esquema* siempre aparece y puede ser **http** o **https**. La *autoridad* también es obligatoria y admite dos formas: la primera consiste en uno o más *nombres* separados por puntos (donde los *nombres* comienzan con una *letra* mayúscula o minúscula entre la **A** y la **Z**, y pueden seguir con más *letras* o *dígitos* decimales). La segunda consiste en cuatro *números* (entre **0** y **255**) separados también por puntos. Opcionalmente puede venir tras la *autoridad* el símbolo */* y una *ruta*. La *ruta* consiste en uno o más *nombres* separados por */*. Si hay *ruta*, es posible que le siga el símbolo **?** seguido de una *consulta* (un simple *nombre*). Finalmente, haya o no *ruta*, la *URL* siempre puede acabar con el símbolo **#** seguido de un *fragmento* (otro *nombre*).

2. **[1 punto]** Acondiciona la siguiente gramática de atributos, eliminando la recursión a izquierdas y factorizando las producciones que puedan resultar problemáticas para la implementación del correspondiente analizador descendente predictivo recursivo.

Exp ::= Exp : Term	{Exp ₀ .cont = Exp ₁ .cont : Term.cont}
Exp ::= Term	{Exp.cont = Term.cont}
Term ::= Fich ^ Term	{Term ₀ .cont = Fich.cont [Term ₁ .cont]}
Term ::= Fich Pos	{Term.cont = pasa(Fich.cont, Pos.cont)}
Fich ::= Id	{Fich.cont = lee(Id.lex)}

3. **[1 punto]** Traduce el siguiente programa a pseudocódigo P para una máquina a pila con contador de instrucciones, según la arquitectura y los esquemas de traducción clásicos que hemos estudiado.

```
padre: integer;
hijo: integer
&
while padre <> hijo do
  if padre > hijo then
    padre := padre - hijo
  else
    hijo := hijo + 1
```

PARTE II

4. [3 puntos] Considera el siguiente programa:

```
tipo tcadena = array [10] of character;
tipo puncadena = ^tcadena;
tipo tcelda = reg item: int; cad: puncadena; freg
tipo puncelda = ^tcelda;
tipo tlista = array [20] of puncelda;

lista: tlista;
msg: tcadena;

procedure crealista ( var l: tlista) {
    i : int;
    & (*1*)
    for i:=0 to 19 do{
        new(l[i]);
        l[i]^item = i;
        new(l[i]^cad);
    }
};

procedure inserta(var l: tlista, pos: int, cadena: tcadena) {
    procedure copiacadena ( var dest: tcadena, orig: tcadena) {
        j: int;
        &
        for j:=0; to 9 do
            dest[j]:=orig[j];
    }
    (*2*)
    copiacadena(l[pos]^cad^, cadena); (*3*)
};

&
crealista(lista);
msg[0]='h'; msg[1]='o'; msg[2]='l';msg[3]='a';
inserta(lista, 2, msg);
```

- [1 punto] Escribe el contenido de la tabla de símbolos en el punto (*1*). Considera que los tipos básicos (enteros, punteros y carácter) ocupan una posición de memoria.
- [1 punto] Haz un esquema detallado de la memoria de la máquina P en el punto (*2*). Incluye todo el detalle posible sobre los marcos de activación, memoria dinámica y contenido de las celdas. Considera que la memoria tiene únicamente 1000 posiciones ([0,999]).
- [1 punto] Escribe el código P necesario para realizar la llamada al procedimiento y el paso de parámetros en la invocación al método copiacadena() (*3*).

5. [2 puntos] Considera la siguiente gramática incontextual:

$A \rightarrow BCD$

$B \rightarrow CDE$

$B \rightarrow \lambda$

$C \rightarrow Da$

$C \rightarrow \lambda$

$D \rightarrow b$

$D \rightarrow \lambda$

$E \rightarrow CDc$

a) Encuentra los conjuntos primeros y siguientes de sus no terminales

b) Calcula el conjunto de directores. ¿Es ésta una gramática LL(1)? Justifica tu respuesta.

6. [1,5 puntos] Considera la siguiente gramática incontextual:

$A \rightarrow BBC$

$B \rightarrow bB$

$B \rightarrow b$

$C \rightarrow Cc$

$C \rightarrow c$

Encuentra el automata LR(0) de esta gramática. ¿Es ésta una gramática LR(0)? ¿Y una gramática SLR(1)? Justifica tu respuesta.

7. [0,5 puntos] Describe cual sería el estado E resultante de ejecutar el siguiente programa en nuestro interprete Prolog:

```
run([apila(4),apila(2),suma,apila(3),desapila_dir(100)],E).
```