

Indicaciones:

La duración es de 100 minutos.

La evaluación consta de 4 preguntas.

Se permite el uso de calculadora científica y tablas, copias y apuntes.

Pregunta 1 (4 puntos)

Proponga una gramática que no pertenezca a la clase LR(1). Justifique su respuesta construyendo el AFN y el AFD correspondiente, y muestre explícitamente el conflicto.

Pregunta 2 (8 puntos)

Dada la gramática:

- $S' \rightarrow S$
 - $S \rightarrow AA$
 - $A \rightarrow aA \mid b$
1. $S' \rightarrow S$
2. $S \rightarrow AA$
3. $A \rightarrow aA$
4. $A \rightarrow b$

- a) Construya los AFN y AFD correspondientes a los elementos LR(0).
- b) Construya los AFN y AFD correspondientes a los elementos LR(1).
- c) A partir del AFD de los elementos LR(1), construya el AFD de la gramática LALR(1).
- d) Derive la cadena **abab** mediante el método LALR(1).

Pregunta 3 (4 puntos)

Elabore una gramática que describa la declaración de arreglos de tamaño fijo en el lenguaje C. Luego defina un atributo semántico denominado **memoria**, que represente el tamaño total en bytes del arreglo. Por último, formule las ecuaciones semánticas correspondientes para obtener una gramática de atributos de tipo S. Algunos ejemplos de las cadenas que deben ser aceptadas en su gramática son:

- `int lista[10]` (memoria = 40 bytes)
- `char pila[50]` (memoria = 50 bytes)
- `double notas[6]` (memoria = 48 bytes)
- `int matrix[10][10]` (memoria = 400 bytes)

Pregunta 4 (4 puntos)

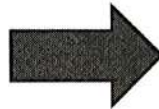
Se desea implementar un solucionador para sistemas de ecuaciones lineales ($Ax = b$) que utilice la siguiente sintaxis:

- `program ::= var_block model_block solve_stmt`
- `var_block ::= var id (, id)* ;`
- `model_block ::= model ; (equation)+ end ;`
- `equation ::= Exp = Exp ;`
- `Exp ::= Term ((+ | -) Term)*`
- `Term ::= Factor (* Factor)*`
- `Factor ::= int | id | (Exp)`
- `solve_stmt ::= solve ;`

Se solicita definir las clases que representen el Árbol de Sintaxis Abstracta (AST) y desarrollar un visitor llamado SolveVisit encargado de generar el código en Python para resolver el sistema de ecuaciones.

Ejemplo:

```
var x y z;  
model;  
  2*x + 3*y + z = 7;  
  x - y = 5;  
  x + z = 2;  
end;  
solve;
```



```
import numpy as np  
  
A = np.array([  
    [2, 3, 1],  
    [1, -1, 0],  
    [1, 0, 1]  
)  
  
b = np.array([7, 5, 2])  
x = np.linalg.solve(A, b)  
  
print("Solución:")  
  
print('x', x[0])  
print('y', x[1])  
print('z', x[2])
```