Parseo de código
Recuperatorio Primer Parcial
Licenciatura en Informática
Universidad Nacional de Quilmes
26 de Noviembre de 2025

Instrucciones:

1. No se permite el uso de material.
2. Todos los ejercicios deben estar justificados rigurosamente.
3. Se valorará la prolijidad, formalidad y profundidad de la solución propuesta.

Ejercicio 1

S → id T | (S) S'
T → = Expr | ε
S' → S S' | ε
Expr → Term Expr'
Expr' → + Term Expr' | ε
Term → id | INT | (Expr)

a) Calcula los conjuntos **FIRST** y **FOLLOW** para todos los no terminales.
b) Construye la **tabla LL(1)** completa (representarla como matriz o listado de entradas). Marca explícitamente si aparecen entradas vacías (ε).
c) Probar formalmente si G es LL(1) o no:
    i) si lo es, justificar por qué no hay conflictos
    ii) si no lo es, describir exactamente dónde se genera el conflicto (FIRST/FIRST o FIRST/FOLLOW) y qué transformación permitiría volverla LL(1)
d) Indica la pila, la entrada restante y la producción aplicada en cada paso. Muestra el **árbol sintáctico** final.

Ejercicio 2

Dada la siguiente gramática clásica para expresiones:

S → StmtList
StmtList → StmtList Stmt | Stmt
Stmt → IDENT := Expr ;
Expr → Expr + Term | Expr - Term | Term
Term → Term * Factor | Term / Factor | Factor
Factor → (Expr) | IDENT | INT

```
if x > y then
   x := x + 1;
else
   y := y / 2;
end
```

Secuencia: a := id + id * id ;

1. Indica la **secuencia de tokens** producida por un lexer que usa las reglas de la gramática propuesta.
2. Indica si la **sintaxis** con la gramática acepta la secuencia; si no, explica por qué y qué transformación/producción se necesitaría.
3. Si la cadena es aceptada, muestra **el árbol de parseo** .
4. Construir la tabla LL(1) con todo lo que conlleva desde cero. Presenta conflictos? En caso de que así sea, indíquelos.

Ejercicio 3:

Sea la siguiente gramática:

```
S → if E then S else S
  | if E then S
  | repeat S until E
  | begin StmtList end
  | stmt
StmtList → S ; StmtList | S
E → cond | E and E | E or E | not E | ( E )
```

1. La gramática propuesta ¿Es ambigua?. En caso afirmativo demuestrelo.
2. Construir un árbol de derivación completo de la siguiente cadena:

   if cond then repeat stmt ; stmt until cond else stmt

   Explica paso a paso las producciones aplicadas.

3. De ser posible, convertir la gramática a Forma Normal de Chomsky (CNF).Justifique paso a paso.