Apuntes

Clase 4- GCI Sentencias Básicas
Clase 4- GCI Sentencias de Control

1. Suponga una sentencia que calcula el promedio de una lista de expresiones y se lo asigna a un identificador y cuya sintaxis está representada con las siguiente reglas gramaticales:

   A → id := P
   P → prom (L)
   L → L , E | E
   E → E + T | E – T | T
   T → T * F | T / F | F
   F → id | cte | (E)

   a) Representar la sentencia **a:= prom ([a+b, 3, c*(d-a)])** en ***polaca inversa*** de manera que toda la semántica sea resuelta en la notación intermedia
   b) Escribir las acciones semánticas en cada regla para generar código en polaca inversa para cualquier sentencia con el formato indicado.

2. Para la sentencia del ejercicio anterior ,

   a) Representar la sentencia **a:= prom ([a+b, 3, c*(d-a)])** en ***árbol sintáctico*** de manera que toda la semántica sea resuelta en la notación intermedia
   b) Escribir las acciones semánticas en cada regla para generar código en árbol sintáctico para cualquier sentencia con el formato indicado.

3. Para la sentencia del ejercicio anterior ,

   a) Representar la sentencia **a:= prom ([a+b, 3, c*(d-a)])** en ***tercetos*** de manera que toda la semántica sea resuelta en la notación intermedia
   b) Escribir las acciones semánticas en cada regla para generar código en tercetos para cualquier sentencia con el formato indicado.

4. Sea la gramática del ejercicio 9 de la práctica 1 que resuelve la asignación múltiple.

   a) Representar la sentencia ***actual:=promedio:=contador:= promedio/ 342 + (contador*contador);*** en ***polaca inversa*** de manera que toda la semántica sea resuelta en la notación intermedia
   b) Escribir las acciones semánticas en cada regla para generar código en polaca inversa para cualquier sentencia con el formato indicado.

5. Sea la gramática del ejercicio 9 de la práctica 1 que resuelve la asignación múltiple.

   a) Representar la sentencia ***actual:=promedio:=contador:= promedio/ 342 + (contador*contador);*** en ***árbol sintáctico*** de manera que toda la semántica sea resuelta en la notación intermedia
   b) Escribir las acciones semánticas en cada regla para generar código en árbol sintáctico para cualquier sentencia con el formato indicado.

6. Suponga la siguiente gramática que representa la sintaxis de un lenguaje que solo permite que sus programas tengan sentencias de selección.

   start -> programa
   programa -> programa sent
   programa -> sent
   sent -> sel | asig
   asig -> ID := exp

sel -> IF cond THEN programa ENDIF
sel -> IF cond THEN programa ELSE programa ENDIF
cond -> IF < CTE
exp -> exp + term
exp -> term
term -> term * factor
term -> factor
factor -> CTE
factor -> ID

a) Representar la siguiente sentencia en ***polaca inversa*** de manera que toda la semántica sea resuelta en la notación intermedia

```
IF a < 3 THEN
    b:= c+1
    a:= 28

ELSE
        IF   b < 245 THEN
                a:= c+ 67 * b
        ENDIF

ENDIF
```

b) Escribir las acciones semánticas en cada regla para generar código en polaca inversa para cualquier sentencia con el formato indicado.
c) Testear con las acciones escritas en el punto b), el resultado del punto a)

7. Suponga la gramática del ejercicio anterior
   a) Representar el siguiente programa en ***tercetos*** de manera que toda la semántica sea resuelta en la notación intermedia

```
IF a < 3 THEN
b:= c+1
a:= 28

ELSE
        IF   b < 245 THEN

        a:= c+ 67 * b

        ENDIF

ENDIF
```

   b) Escribir las acciones semánticas en cada regla para generar código en tercetos para cualquier sentencia con el formato indicado.
   c) Testear con las acciones escritas en el punto b), el resultado del punto a)

8. Suponga la gramática de un lenguaje que solo soporta sentencias de ciclos (del tipo FOR) y sentencias de asignación

```
start -> programa
programa  -> programa sent
programa  -> sent
sent -> ciclo | asig
asig -> ID := exp
ciclo -> FOR  inicio { programa } FOREND
inicio -> ID = CTE TO CTE
exp  -> exp + term
exp  -> term
term -> term * factor
term -> factor
factor -> CTE
factor -> ID
```

   a) Representar el siguiente programa en ***tercetos*** de manera que toda la semántica sea resuelta en la notación intermedia

```
c:=0
FOR i:=1 TO 20
{
        a:= c+ 67 * b
        b:=b+1
        c:=c+1
}
FOREND
```

   b) Escribir las acciones semánticas en cada regla para generar código en tercetos para cualquier sentencia con el formato indicado.
   c) Testear con las acciones escritas en el punto b), el resultado del punto a)

9. Suponga la siguiente gramática que representa la sintaxis de un lenguaje que solo permite que sus programas tengan sentencias de ciclos del tipo “while”.

```
start -> programa
programa -> programa sent
programa -> sent
sent -> asig | ciclo
ciclo -> WHILE  cond  programa  END
cond -> exp <= exp
asig -> id := exp
exp  -> exp + term
exp  -> exp - term
exp  -> term
term -> term * factor
term -> factor
factor -> CTE
factor -> ID
```

   a) Representar la siguiente sentencia en ***polaca inversa*** de manera que toda la semántica sea resuelta en la notación intermedia

```
WHILE a*3 <= b
        a:= a+2
        b:= 5
        WHILE b <= 5
                a:= a+1
        END
END
```

b) Escribir las acciones semánticas en cada regla para generar código en árbol sintáctico para cualquier sentencia con el formato indicado.
c) Testear con las acciones escritas en el punto b), el resultado del punto a)