

**Parcial 02**

1) Encuentre el conjunto Primero(S) para la siguiente gramática

$S \rightarrow \text{ARM} \mid \text{iNTEL} \quad A \rightarrow \text{fT} \mid \varepsilon \quad R \rightarrow \varepsilon \mid \text{ArL} \quad M \rightarrow \text{eNx}$

Solución

$\text{Primero}(S) = \{f, r, e, i\}$

2) Dada la siguiente GIC encuentre una equivalente LL(1)

$S \rightarrow \text{fCa} \mid \text{fC} \mid \text{fCRt} \quad C \rightarrow \text{Cmd} \mid \text{zq}$

Solución

$S' \rightarrow \text{fCS}' \quad S' \rightarrow a \mid \varepsilon \mid \text{Rt}$
 $C \rightarrow \text{zqC}' \quad C' \rightarrow \text{mdC}' \mid \varepsilon$

3) Escriba la EBNF correspondiente a la siguiente PAS.

```
void Listado()  
{  
    Dato();  
    TOKEN t = ProximoToken();  
    while (t == SEP || t == PT0) {  
        Match(t);  
        Dato();  
        t = ProximoToken();  
    }  
}
```

Solución

$\langle \text{Listado} \rangle ::= \langle \text{Dato} \rangle \{ \text{SEP} \langle \text{Dato} \rangle \mid \text{PT0} \langle \text{Dato} \rangle \}$

4) Dado el siguiente fragmento del estándar de C

AND-expression:
 equality-expression
 AND-expression & equality-expression

exclusive-OR-expression:

 AND-expression
 exclusive-OR-expression ^ AND-expression

a) Indique, justificando, como asocia el operador y de bits &.

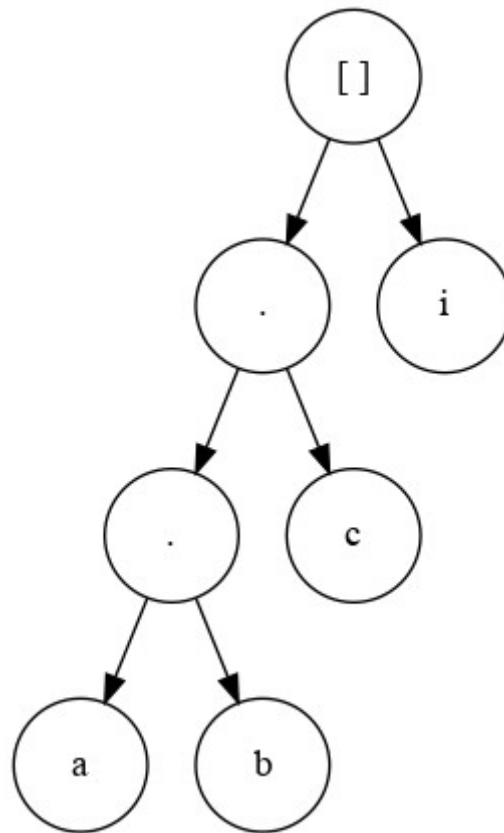
De izquierda a derecha, porque la recursión es a izquierda

b) Indique, justificando, que operador tiene mayor precedencia entre & y ^.

El operador & tiene mayor precedencia que ^ porque está más lejos del axioma.



5) Dibuje el árbol de la expresión en lenguaje C: `a.b.c[i]`



Dada la siguiente expresión en lenguaje C:

`r.v[*p]`

6) Haga las declaraciones necesarias para que sea semánticamente correcta y de tipo `double`.

```
int *p;
struct e {
    double v[5];
} r;
```

7) Haga las declaraciones necesarias para que sea semánticamente incorrecta y explique el motivo.

Si declaro `int p`, entonces hay incompatibilidad entre el operador `*` y `p`

Si declaro `int r` entonces tengo incompatibilidad entre `r` y el operador `.`



Dado el siguiente fuente

```
05 void bodoque(int n) {  
06     struct {int a, float b} e = {1};  
07     while (e.a < n) {  
08         e.a++;  
09         n--;  
10         e.a + 75l; //agregado en 2024  
11     }  
12 }
```

Conteste las siguientes preguntas.

8) A qué categoría sintáctica pertenece lo escrito en la línea 06?

Declaración/Definición (con inicialización para mayor detalle.

9) De la lista de caracteres que serán devueltos con `ungetc` por el scanner en la línea 08. Puede escribir nombres como: espacio, punto y otros similares si que más claro.

Punto , a, +

10) Indique el tipo de dato del resultado de la suma en línea 10. Justifique.

Queda `long int` dado que `75l` es constante `long int` por su sufijo y al sumar con un `int` se hacen las conversiones numéricas que llevan a ambos operando al de mayor capacidad de representación.