

Ejercicios de BNF \Leftrightarrow PAS

Nota: los ejercicios marcados con (*) al principio están sacados del libro de la cátedra
los ejercicios marcados con (°) al principio están basados en uno tomado en un final

1. (°) Dada la siguiente BNF:
 $\langle \text{sentenciaAsignacion} \rangle ::= \langle \text{identificador} \rangle \text{ OpAsignacion } \langle \text{expresion} \rangle \text{ SgPuntuacion}$
Construya una función ANSI C que implemente el PAS de esta producción

Solución

```
void SentenciaAsignacion(void) {  
    Identificador();  
    Match(OpAsignacion);  
    Expresion();  
    Match(SgPuntuacion);  
}
```

2. (°) Dada la siguiente BNF:
 $\langle \text{FACTOR} \rangle ::= \text{PAREN_IZQ } \langle \text{EXP} \rangle \text{ PAREN_DER } \mid \text{NR0}$
Construya una función ANSI C que implemente el PAS de esta producción

Solución

```
void Factor() {  
    TOKEN t = ProximoToken();  
    if (t == PAREN_IZQ) {  
        Match(PAREN_IZQ);  
        Exp();  
        Match(PAREN_DER);  
    } else if (t == NR0) {  
        Match(NR0);  
    } else {  
        errorSintactico(t);  
    }  
}
```

3. Dada la siguiente BNF:
 $\langle \text{Exp} \rangle ::= \langle \text{operando} \rangle \{ \text{SUM } \langle \text{operando} \rangle \mid \text{MULT } \langle \text{operando} \rangle \}$
Construya una función ANSI C que implemente el PAS de esta producción

Solución

```
void Exp()  
{  
    TOKEN tok;  
    Operando();  
    tok = ProximoToken();  
    while (tok == SUM || tok == MULT) {  
        Match(tok);  
        Operando();  
        tok = ProximoToken();  
    }  
}
```

4. (°) Sea el siguiente PAS

```
void PS(void) {  
    TOKEN tok = ProximoToken();  
    switch(tok) {  
        case A: Match(A); break;
```



```
        case B: Match(B); break;
        case C: Match(C); D(); Match(E); break;
        default: ErrorSintactico(tok); break;
    }
}
```

Escriba las producciones que implementa dicho PAS.

Solución

$\langle PS \rangle ::= A \mid B \mid C \langle D \rangle E$

5. (°) Sea el siguiente PAS

```
void Sent1(void) {
    Sent2();
    while(1) {
        switch(ProximoToken()) {
            case SGN0: case OP_ESP: Sent2(); break;
            default: return;
        }
    }
}
```

Escriba las producciones que implementa dicho PAS. Debe utilizar recursividad a derecha

Intermedio

$\langle Sent1 \rangle ::= \langle Sent2 \rangle \{ \langle Sent2 \rangle \}$

Solución

$\langle Sent1 \rangle ::= \langle Sent2 \rangle \mid \langle Sent2 \rangle \langle Sent1 \rangle$

(notar que Sent1 finalmente corta el loop usando la producción Sent2)

6. (°) Sea el siguiente PAS

```
void Y(void) {
    Match(H);
    while(1) {
        switch(ProximoToken()) {
            case J: Match(J); break;
            case Z: Match(Z); break;
            default: return;
        }
    }
}
```

Escriba las producciones que implementa dicho PAS. Debe utilizar recursividad a derecha

Intermedio

$\langle Y \rangle ::= H \{ J \mid Z \}$

Solución

$\langle Y \rangle ::= H \langle Aux \rangle$

$\langle Aux \rangle ::= J \langle Aux \rangle \mid Z \langle Aux \rangle \mid \epsilon$