Manual Técnico

PROYECTO 2

Juan Marcos Ibarra López | OLC1 | 20/04/2020
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

INTRODUCCIÓN

La aplicación fue realizada con el propósito de que pasar de un lenguaje de programación a otro no sea tan complicado, es este de caso de C# a Python.

CONTENIDO TECNICO

ANALIZADOR LEXICO

El analizador léxico se implementó en la aplicación, para que se pueda mostrar correctamente las palabras reservadas, además, si el usuario ingresa un texto con algún error el analizador léxico le ayudará a encontrar el error cometido de una manera mucho más sencilla y eficaz, mediante la generación de un archivo HTML, donde muestra los errores encontrados.

LOGICA UTILIZADA

Las palabras reservadas y símbolos que reconoce la aplicación se hacen mediante el analizador léxico, el cual se realizó mediante el método del árbol, a partir de la siguiente expresión regular:

$$[(\ “\ (L \mid D \mid S)^*\ ”\ ) \mid L+ \mid S \mid D+]$$

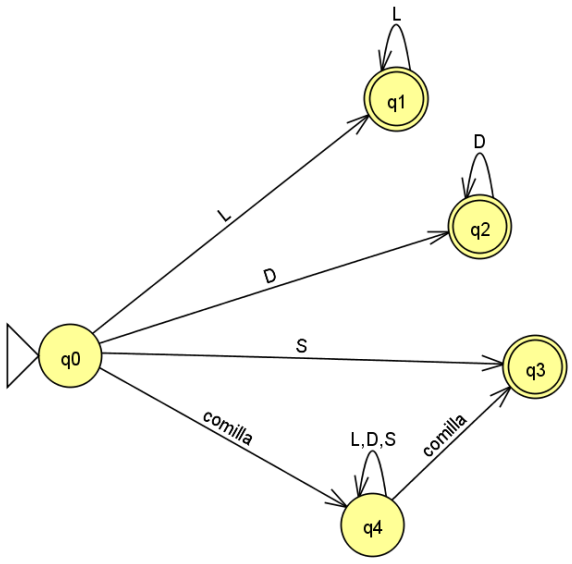
Donde:

L = Letras [A - Z]

D = Digitos [0 - 9]

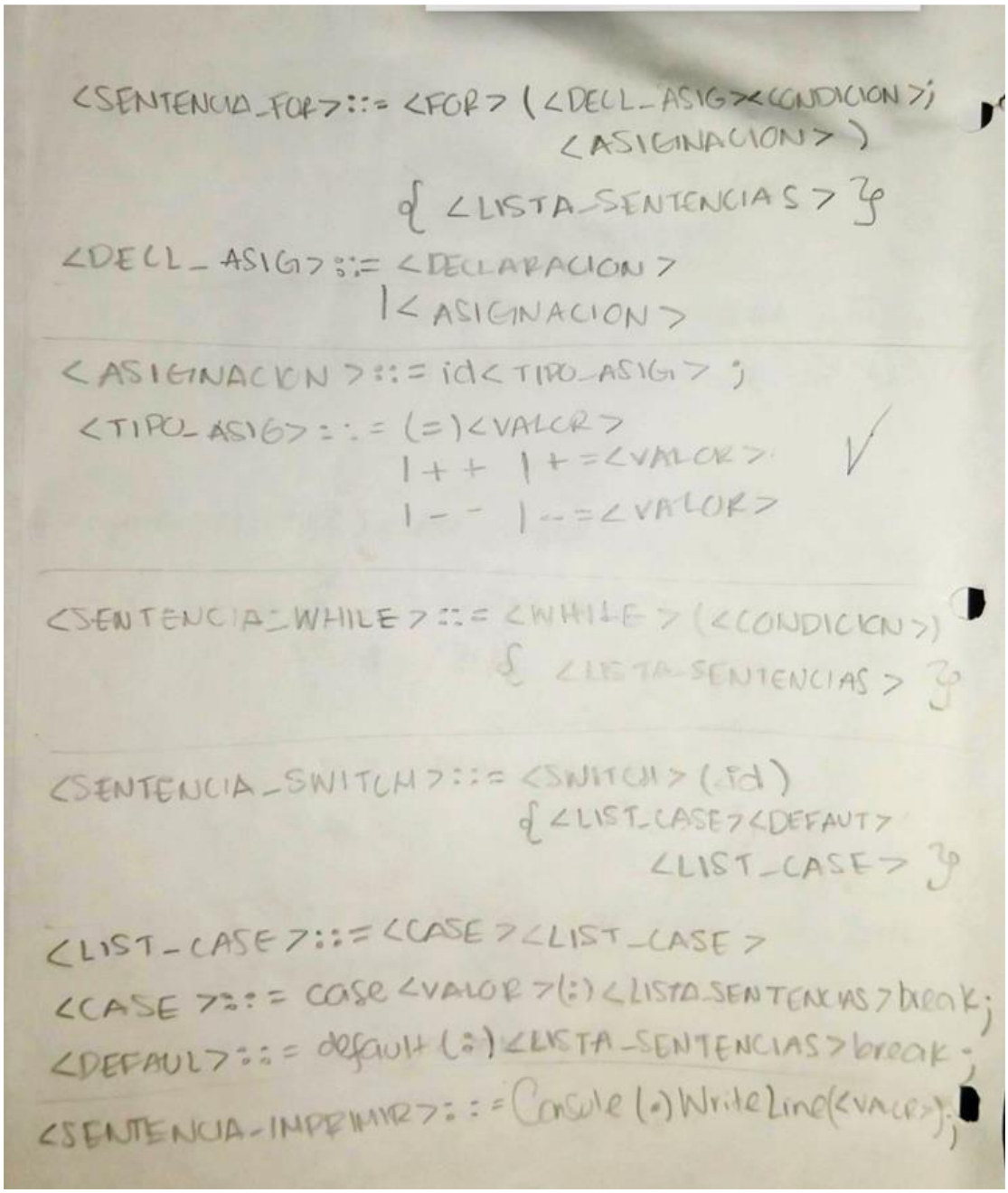
S = Simbolos definidos [dos puntos, punto y coma, porcentaje, corchetes]

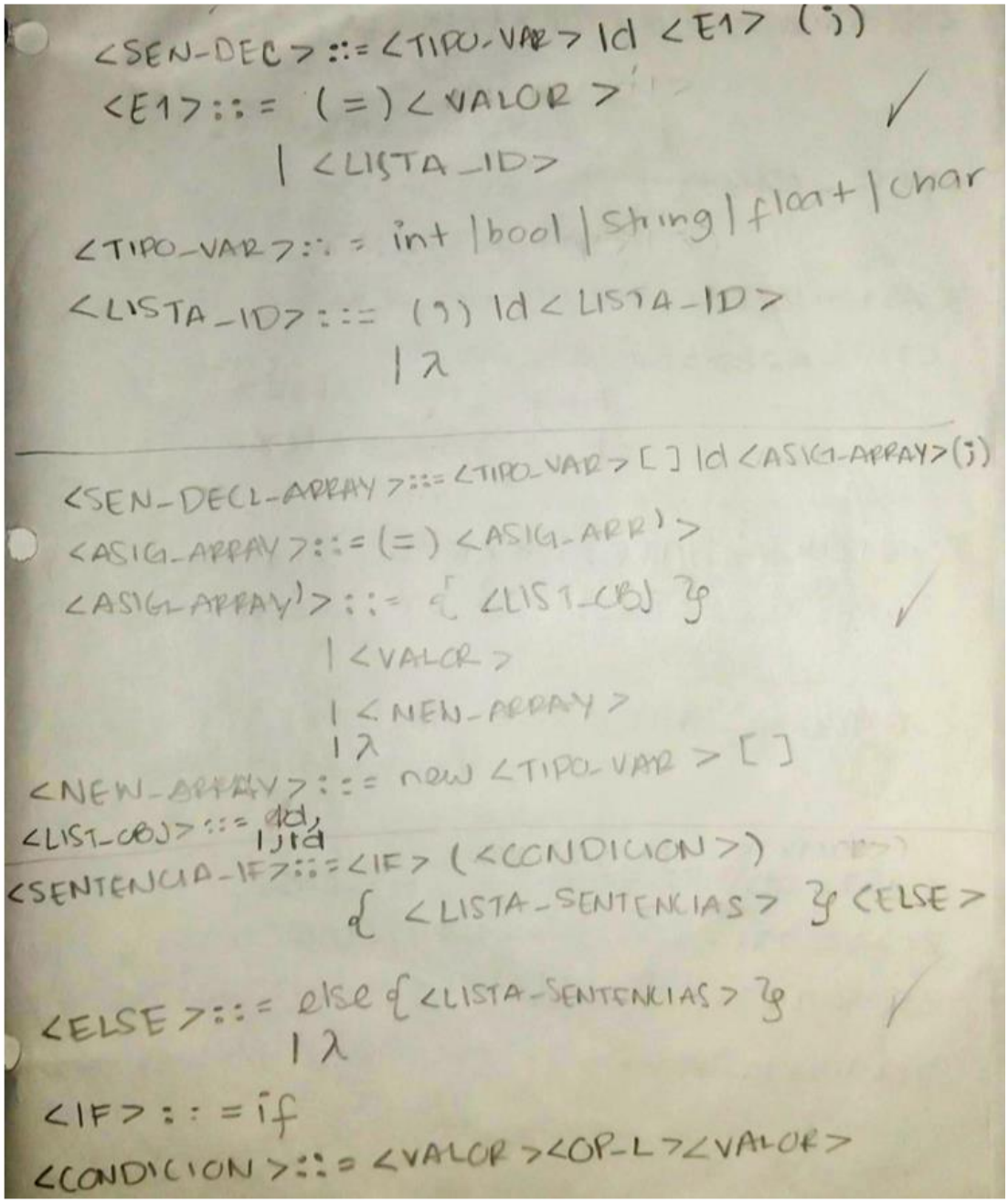
AFD Obtenido:



ANALIZADOR SINTACTICO

Para realizar el analizador sintáctico primero se procedió a crear las gramáticas expresadas en el sistema BNF.




<SEN-DEC> ::= <TIPO-VAR> id <E1> (;)
<E1> ::= (=) <VALOR>
| <LISTA_ID>
<TIPO_VAR> ::= int | bool | string | float | char
<LISTA_ID> ::= (,) id <LISTA-ID>
| λ
<SEN-DECL-ARRAY> ::= <TIPO_VAR> [ ] id <ASIG-ARRAY> (;)
<ASIG-ARRAY> ::= (=) <ASIG-ARR'>
<ASIG-ARRAY'> ::= { <LIST_OBJ } ;
| <VALOR>
| <NEW-ARRAY>
| λ
<NEW-ARRAY> ::= new <TIPO_VAR> [ ]
<LIST_OBJ> ::= id,
| id
<SENTENCIA-IF> ::= <IF> (<CONDICION>)
{ <LISTA-SENTENCIAS> } <ELSE>
<ELSE> ::= else { <LISTA-SENTENCIAS> }
| λ
<IF> ::= if
<CONDICION> ::= <VALOR> <OP-L> <VALOR>

ALGUNAS YA IMPLEMENTADAS

```
/*-----------------------------------------------------------------
SENTENCIA IF
-----------------------------------------------------------------   */
private Sentencia_IF():void
{
    if (this.tokenActual.tipoToken === TipoToken.IF)
    {

        this.emparejar(TipoToken.IF);
        this.Traduccion += this.tabulacion+"if ";
        this.emparejar(TipoToken.PARENTESIS_APERTURA);
        this.Condicion();
        this.Traduccion += " :\n";
        this.emparejar(TipoToken.PARENTESIS_CIERRE);
        this.emparejar(TipoToken.LLAVE_APERTURA);
        this.AumentarTab();
        this.LISTA_TODAS_SENTENCIAS();
        this.emparejar(TipoToken.LLAVE_CIERRE);
        this.disminuirTab();
        this.ELSE();
    }
    this.LISTA_TODAS_SENTENCIAS();  //CUANDO TERMINE UNA PASE A LA SIGUIENTE
}
```

```
/*-----------------------------------------------------------------
    SENTENCIA WHILE  -----------
-----------------------------------------------------------------   */
private Sentencia_While():void {
    if (this.tokenActual.tipoToken === TipoToken.WHILE)
    {
        this.contadorBucles++;
        this.emparejar(TipoToken.WHILE);
        this.Traduccion +=  this.tabulacion + "while";
        this.Traduccion += " " + this.tokenActual.lexemaToken;
        this.emparejar(TipoToken.PARENTESIS_APERTURA);
        this.Condicion();
        this.Traduccion += "" + this.tokenActual.lexemaToken + "\n"; //salto de linea
        this.emparejar(TipoToken.PARENTESIS_CIERRE);

        this.emparejar(TipoToken.LLAVE_APERTURA);
        this.AumentarTab();
        this.LISTA_TODAS_SENTENCIAS();
        this.disminuirTab();
        this.emparejar(TipoToken.LLAVE_CIERRE);
    }
    this.contadorBucles--;
    this.LISTA_TODAS_SENTENCIAS();  //CUANDO TERMINE UNA PASE A LA SIGUIENTE
}
/*-----------------------------------------------------------------
```

```
/*-----------------------------------------------------------------
    SENTENCIA SWITCH
-----------------------------------------------------------------  */
public Sentencia_Switch():void {
    if (this.tokenActual.tipoToken == TipoToken.SWITCH)
    {
        this.emparejar(TipoToken.SWITCH);
        this.emparejar(TipoToken.PARENTESIS_APERTURA);
      //  this.Traduccion += "" + this.tokenActual.lexemaToken + " == ";
        this.variableCondicionCase = this.tokenActual; // GUARDAMOS EL NOMBRE DE LA VARIABLE QUE SE VA A USAR EL LOS CASES
        this.emparejar(TipoToken.IDENTIFICADOR);
        this.emparejar(TipoToken.PARENTESIS_CIERRE);

        this.emparejar(TipoToken.LLAVE_APERTURA);

        this.Traduccion += this.tabulacion + "def switch(case, " + this.variableCondicionCase.lexemaToken + "):\n";
        this.AumentarTab();
        this.Traduccion += this.tabulacion + "switcher = {\n";
        this.AumentarTab();
        this.List_Case();
        this.Case_Default();
        this.List_Case();

        this.emparejar(TipoToken.LLAVE_CIERRE);

        this.disminuirTab();
        this.Traduccion += this.tabulacion + "}\n";
        this.disminuirTab();

        this.contadorCase = 0; // REINICIAR EL CONTADOR DE CASE
    }
    this.LISTA_TODAS_SENTENCIAS();  //CUANDO TERMINE UNA PASE A LA SIGUIENTE
}
```

```
/*-----------------------------------------------------------------
    SENTENCIA IMPRIMIR
-----------------------------------------------------------------  */
public Sentencia_Imprimir():void {
    this.emparejar(TipoToken.CONSOLE);
    this.emparejar(TipoToken.PUNTO);
    if(this.tokenActual.tipoToken === TipoToken.WRITELINE){
        this.emparejar(TipoToken.WRITELINE);
    }
    else{
        this.emparejar(TipoToken.WRITE);
    }
    this.Traduccion += this.tabulacion + "print";
    this.emparejar(TipoToken.PARENTESIS_APERTURA);
    this.Traduccion += "(";
    this.ValorDato();
    this.emparejar(TipoToken.PARENTESIS_CIERRE);
    this.Traduccion += ")";
    this.emparejar(TipoToken.PUNTO_COMA);
    this.Traduccion += "\n";
    this.LISTA_TODAS_SENTENCIAS();  //CUANDO TERMINE UNA PASE A LA SIGUIENTE
}
```