Análisis de gramática para analizador Sintáctico

De primero separaremos nuestros símbolos terminales de los no terminales:

terminal T_color, circulo,rectangulo, linea, poligono, cuadrado, curva, animar, objeto, anterior, graficar,Suma, Resta, Multiplicacion, Division, Parentesis_a, Parentesis_c, coma, Numero, SLinea;

non terminal INICIO, FORMA, SIGUIENTE, ANIMAR, ANIMACION, GRAFICAR;

non terminal Integer OPERACION;

Establecemos la jerarquía de las operaciones que realizaremos:

precedence left Suma, Resta;

precedence left Multiplicación, División;

Nuestro estado inicial será: “INICIO”:

start with INICIO;

Nuestro estado INICIO produce el estado GRAFICAR o vacío:

INICIO ::=

GRAFICAR

|

En el estado GRAFICAR esperamos que venga el token graficar y luego llamamos el estado FORMA:

GRAFICAR ::=

graficar FORMA

En el estado FORMA escribimos todos los tokens que esperamos para que una figura sea aceptada llamando a otros estados como

FORMA ::=

circulo Parentesis_a OPERACION:no1 coma OPERACION:no2 coma OPERACION:no3 coma T_color:color Parentesis_c SIGUIENTE

En nuestro estado OPERACIÓN llamamos el mismo estado seguido de un símbolo matemático y luego el mismo estado de OPERACIÓN

OPERACION ::=

OPERACION:e1 Suma:s1 OPERACION:e2

En nuestro estado llamado SIGUIENTE producimos los estados GRAFICAR Y ANIMAR o vacío

SIGUIENTE ::=

GRAFICAR |

ANIMAR |

;

En ANIMAR esperamos que vengan los tokens en la forma descrita a continuación llamando de nuevo al estado de OPERACIÓN para ayudarnos:

ANIMAR ::=

animar objeto anterior Parentesis_a OPERACION coma OPERACION coma OPERACION coma ANIMACION Parentesis_c

Nuestro ultimo estado ANIMACION nos ayuda para reconocer el tipo de animación, ya sea línea o curva:

ANIMACION ::=

línea |

curva