

3.3. ANALISIS LEXICOGRAFICO.

El código fuente escrito en código ASCII es analizado para la obtención de tokens que serán posteriormente utilizados por el Analizador Sintáctico.

Se analiza una cadena de trabajo carácter por carácter, siendo el espacio en blanco el carácter nulo de separación entre palabras.

Si la palabra forma parte de la lista de palabras reservadas, se devuelve el número de la palabra reservada que se halla en la tabla de símbolos; si es otra palabra (que puede significar un valor numérico literal), se aumenta un lugar con este lexema en la tabla de símbolos, se carga su valor real y se retorna ésta ubicación de Tabla de Símbolos.



3.4. ANALIZADOR SINTACTICO.

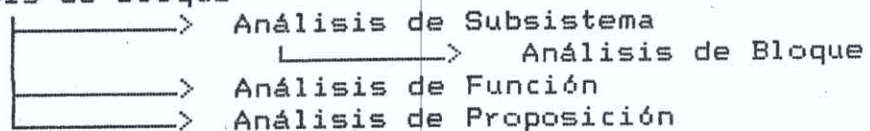
El Analizador Sintáctico, siguiendo las reglas definidas por el lenguaje se utiliza para generar el Código Intermedio.

Los tokens ingresados sirven para alimentar los procesos que son recursivos.

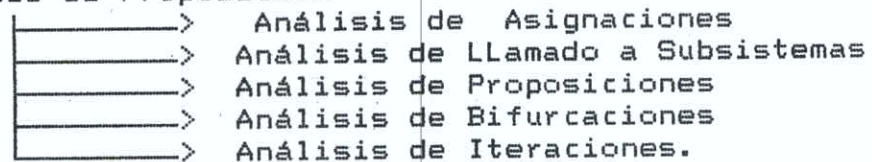


Enseguida se muestran algunos procedimientos recursivos para ser utilizados en el Analizador Sintáctico (en la Descripción del Código se hallan todos los procedimientos).

Análisis de Bloque



Análisis de Proposición



3.5. CODIGO INTERMEDIO.

El proceso de interpretación implica la traducción de un lenguaje fuente complejo hacia otro lenguaje más simple a nivel de máquina. En este caso se llega hasta un nivel conocido como intermedio lineal.

El Código Intermedio es una cuarteto que se almacena en un registro de la siguiente manera:

Type

```

tipocode = record
    op, arg1, arg2, arg3: integer;
end;
  
```

```

Var tcode = array[1..MaxCode] of tipocode;
  
```

donde :

```

op : Operador del Código
arg1, arg2, arg3 : Argumentos del código, apuntan a un
lugar de la tabla de símbolos.
  
```

MaxCode : Tamaño del arreglo de código intermedio.

Algunas operaciones consideradas en el Código Intermedio se muestran enseguida:

Operación	Simb.	Operador 1	Operador 2	Operador 3
Adición	+	X	X	X
Sustracción	-	X	X	X
Multiplicación	*	X	X	X
División	/	X	X	X
Llamar Rutina	CALL	X	-	-
Retorno Rutina	RET	-	-	-
Asignación	ASIG	X	X	X
Or Lógico	tOR	X	X	X
And Lógico	tAND	X	X	X

3.6. TRATAMIENTO DE ERRORES.

Los errores se manejan con el método del pánico. Es decir, cada vez que se encuentra un error éste se muestra al usuario y se retorna al ambiente del sistema.

Tenemos tres tipos de errores : Los lexicográficos, los sintácticos y los de tiempo de ejecución.

3.6.1. ERRORES LEXICOGRAFICOS.

Son los errores que se hallan al momento de realizarse el Análisis lexicográfico. Entre los errores de este tipo tenemos:

- Error al toma una constante real o constante muy grande:
19999100000231231321123.123131312209999922200200
- Error de carácter extraño en una palabra :

AB1₇SW1~1A

3.6.2. ERRORES SINTACTICOS.

Basándose en la gramática previamente definida, se detectan los errores en el código fuente que no cumplen con alguna regla. Como ejemplo tenemos:

- Esperando una paréntesis:

A = (C*D/3 ;

3.6.3. ERRORES EN TIEMPO DE EJECUCION.

En los casos que se producen errores al ejecutarse el código intermedio como :

- División por cero

3.7. EJECUCION DEL CODIGO INTERMEDIO.

El Código Intermedio generado por el Analizador Sintáctico es efectuado linealmente. Las operaciones se realizan cambiando el estado o valor de los lugares de la tabla de símbolos asignados por su valor de los operandos.

Tabla Índice	Líneas de Código Intermedio No.	Operandos	
-----	111	SUMA 120 121 122	Ejecución \ /
-----	112	RESTA	
-----	113		
-----	114		

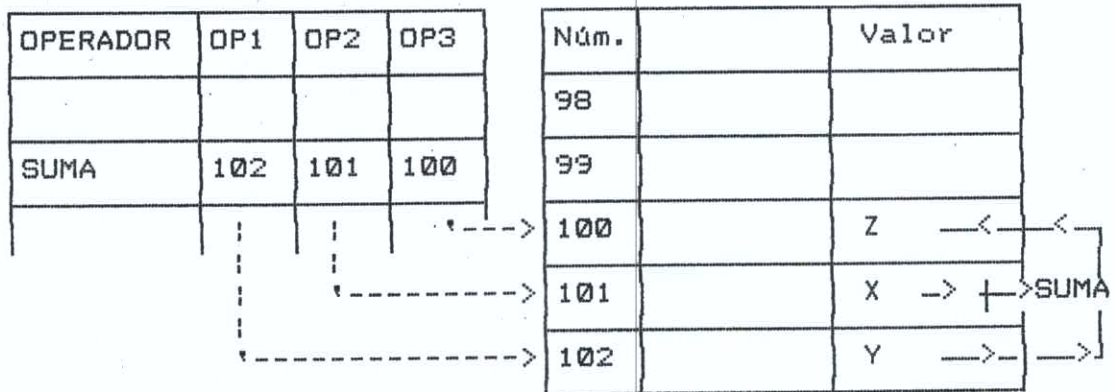
Enseguida se describen las operaciones.

3.7.1. OPERACIONES BASICAS.

Son las operaciones como Suma, Resta, Multiplicación y División. Se trabaja de la siguiente manera:

Las operaciones se realizan con los valores a los que apuntan los valores de los Operadores de la Tabla del Código Intermedio.

TABLA DE CODIGO INTERMEDIO TABLA DE SIMBOLOS



$$Z = X + Y$$

Se asigna al valor de la Tabla de Símbolos en el lugar apuntado por OP3 (tercer operador), el resultado de la

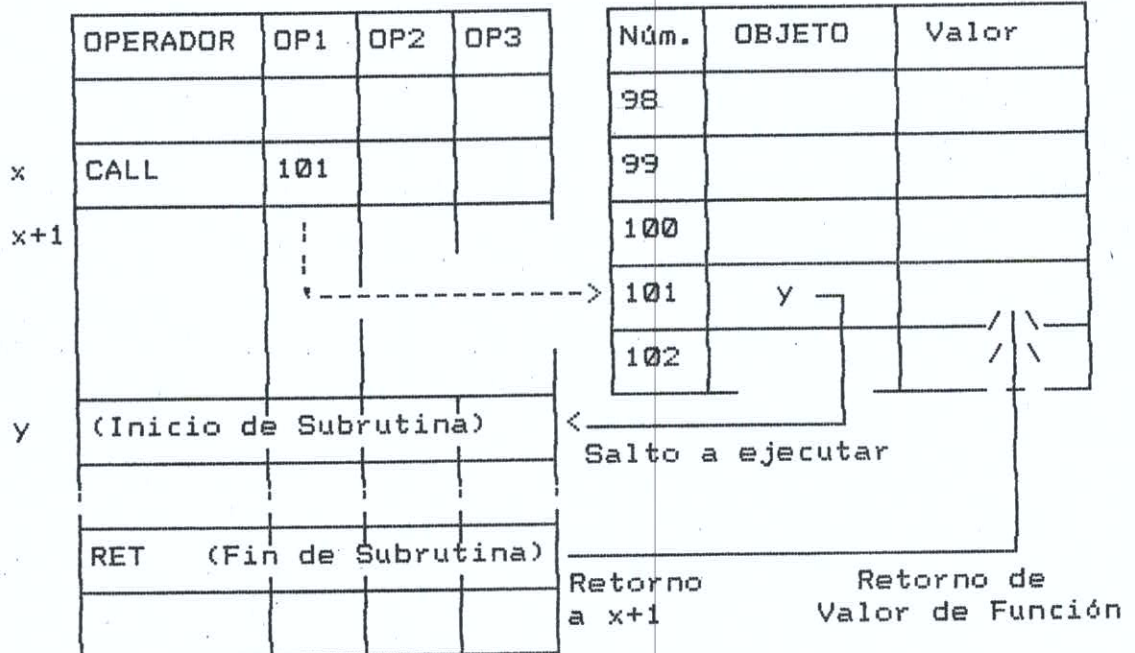
operación que se realiza con los valores de la TS apuntados por los Operadores Primero y Segundo.

3.7.2. LLAMADAS A SUBRUTINAS.

Los subsistemas y las funciones se inician en algún lugar de la Tabla de Código Intermedio y terminan cuando se halla la Orden de Retorno, que regresa el control a la línea siguiente del código que invocó a la subrutina.

TABLA DE CODIGO INTERMEDIO

TABLA DE SIMBOLOS



El retorno del Valor se realiza si la subrutina es una función.

3.7.3. CREACION DE ARCHIVOS INTERMEDIOS.

El Código Intermedio se ejecuta desde la primera proposición, tantas veces como se hayan definido las variables de control. Es decir el horizonte de integración es desde un momento inicial (DESDE_T) hasta el final (HASTA_T) incrementándose cada vez en un DT definido por el usuario.

En cada iteración se salvan los valores necesarios para

graficar los resultados. Se define un archivo tipo ASCII donde se guardan los datos de las variables que se graficarán luego.

Ejemplo, si $DT = 0.5$:

Tiempo	Variable 1	Variable 2
1.0	199.000	2.000
1.5	200.000	4.000
2.0	201.000	6.000

...

3.8. GENERACION DE RESULTADOS.

Se utiliza el modulo graficador y el archivo de resultados intermedios para generar un gráfico en la pantalla, mostrando los cambios de las variables contra el tiempo u otra variable (diagrama de fase).

Además se muestran los resultados de los valores máximo y mínimo de las variables mostradas (hasta un máximo de ocho).

3.9. ORDENES PREDEFINIDAS.

Existen dos ordenes predefinidas para el ambiente gráfico.

Tenemos las siguiente orden que se escribe inmediatamente después de haber escrito las variables de CONTROL.

PON_GRAFICO(<Variable en el eje de Abscisas>, <Límites>)

La orden siguiente puede colocarse como cualquier proposición:

GRAFICA (<Variable>, <Tipo Graf.> [, <Límite I>, <Límite S>])

<Variable> : Es el nombre de la variable a graficar.

<Tipo Graf.> : Valores enteros entre 1 y 4 para seleccionar el tipo de trazado que muestre el gráfico.

<Límite I> : Límite Inferior de la variable en el eje de las Ordenadas.

<Límite S> : Límite Superior de la variable en el eje de las Ordenadas.

3.10. INTEGRACION DEL SISTEMA.

El ambiente del sistema, que es la unidad de mayor jerarquía y controla el resto del sistema, tiene los módulos siguientes :

- a) Administrador de Archivos.
- b) Editor.
- c) Compilador.
- d) Ejecutor.
- e) Cambios de Condiciones iniciales
- f) Unidad de Análisis de Sensibilidad.

3.10.1. Administrador de Archivos.

En el módulo para seleccionar archivos de código fuente y de trabajo.

3.10.2. Editor.

Unidad externa para la edición del código fuente.

3.10.3. Compilador.

Se invoca al modulo de Traductor del Código fuente al Código Intermedio.

3.10.4. Ejecutor.

Ejecuta el código intermedio creando los archivos de resultados.

3.10.5. Unidad Gráfica.

Este modulo muestra los resultados del modelo, que son básicamente los cambios de valor de las variables, en el tiempo.

3.10.6. Unidad de Análisis de Sensibilidad.

Estudian la sensibilidad del modelo frente a cambios en las valores iniciales de una de las variables, partiendo de un valor inicial hasta un valor final, con un cierto incremento.

3.10.7. CARTA ESTRUCTURAL.

En la siguiente página se muestra una carta estructural del sistema.

CARTA ESTRUCTURAL DEL INTERPRETE.

