Publicamos a continuación tres trabajos realizados por alumnos de la especialidad de Cálculo Automático de la Facultad de Ciencias para la asignatura de Compilación y Sistemas Operativos. Dado su volumen no publicamos los listados de los programas que están a disposición de quien quiera consultarlos en el CCUM.

ANALIZADOR MORFOLOGICO PARA UN MINI-LENGUAJE

Por Pedro Roa Medina

Se ha realizado un procesador basado en las ideas expuestas por David Gries en su libro Compiler Generator for Digital Computers, pags. 66-71. Trabaja en la forma bottom-up.

Este precompilador no está orientado al tratamiento de ningún lenguaje usual en particular, si bien su metodología tiene la suficiente generalidad como para adaptarse a algunos de ellos si fuese necesario.

La gramática de las U.S. analizadas puede darse en la FNB como sigue:

< U.S.> ::= <posible ident.> |<entero> |<signo simple> |
<signo doble> |<comentario><U.S.> | ϕ <U.S.>

<pos. ident.> ::= <letra> |<pos. ident.> <letra> |
<pos. ident.> <digito> |<pos. ident.> ϕ

<entero> ::= <digito> |<entero><digito> |<entero> ϕ

<signo simple> ::= / | + | * | = | (|) | ,

<signo doble> ::= //

<comentario> ::= /* cadena */

<cadena> ::= <caracter> |<cadena><caracter> | Λ

<caracter> ::= <cualquier carácter permitido en la máquina.

<letra> ::= A | B .... | Z

<dígito> ::= 0 | 1 ..... | 9

El programa se ha realizado en el lenguaje SNOBOL-3 que representa verdaderas ventajas de manejo de cadenas de caracteres, si bien es lento el proceso.

Las funciones que realiza son:

a) Listado del programa objeto a reconocer.

b) Reconocimiento de las U.S., dando mensajes de error si este existiera.

c) Asociación a cada U.S. reconocida con un código propio de ella.

d) Evaluación de número y almacenamiento en tabla de constantes

e) Reconocimiento, dentro del tipo de U.S. posible ident. , si se trata de palabras reservadas, en cuyo caso asociaria el código de dicha palabra; en otro caso asociaria el de identificador.

LLos códigos asociados con las distintas U.S. son:

identificador 1

/ 6

\+ 7

\- 8

* 9

(.....10

) 11

= 12

// 13

ABS 20

BEGIN 21

DO 22

DIMENSION 23

END 24

EQUIVALENCE 25

GØ TØ 26

IF 27

INTERGER 28

NAMELIST. 29

PRINT 30

READ 31

STØP 32

WRITE 33

A la U.S. <numero> se le asociará un código tal que el primer caracter será una C, el segundo serâ igual al primer digito del número y el tercero y sucesivos digitos indican

la colocación en la tabla de las constantes que empiezan por ese digito. Así C94 indicaría que la U.S. analizada es una - constante que comienza por 9 y es la cuarta de éstas encontrada hasta ahora. Dada la facilidad del lenguaje SNOBOL para - concatenar cadenas para formar nombre y la facilidad del direccionamiento indirecto. el acceso a la tabla de constantes es fácilmente realizable; por ejemplo, para conseguir el valor - de una de ellas, cuyo código fuese C47 y estuviese almacenado en CØDIGØ podría hacerse de la forma

```
     I = "1"
     CØDIGØ  * A/"1"*  *B/"1"*  =
UP   $(B  "TC")  D  *NUM*  ","
     D = D NUM ","
     I = .LT(I,CØDIGØ) I + "1"   /S(UP)
```

El tratamiento de palabras reservadas se efectua de la forma siguiente:

Las palabras irán almacenadas en distintas cadenas llamadas APR, BPR, ... , según empiecen por A, B, etc.; estos nombres pueden resumirse, si tenemos la primera letra del posible - identificador en FIRL, en el nombre $(FIRL "PR") y así todas las cadenas APR...ZPR pueden tratarse dentro de un mismo nombre ; se almacenarán dentro de cada cadena por orden creciente de longitudes llevando a la izquierda de la palabra su longitud sin blancos intermedios (aunque en el progra

ma fuente puede llevarlos), y a su derecha el código que le queremos asociar; la consulta a la tabla se efectuará siempre que encontremos un posible identificador y solo tendrá que consultarse la cadena que nos indique la 1ª - letra del posible identificador, y dentro de ella, solo - mientras que las palabras consultadas de la tabla sean de longitud menor o igual a la del posible identificador. El tratamiento en el programa de la consulta a la tabla, se hace con la función J que tomará valor 0 si no está en - la tabla y si estuviera en ella tomará el código asociado.

EEl analizador morfológico está confeccionado en forma de una ffuncion llamada SCANNER que a su vez llama a la función J antes ddescrita, si bien en el listado adjunto existe un programa pprincipal que solo nos sirve para llamar sucesivas veces a la ssubrutina SCANNER con fines de prueba.

EEl trabajo se ha realizado según el diagrama de estados adjuntto, donde las acciones semanticas indicadas significan:

GCH .- poner en CHAR el siguiente caracter del input

GNB .- " " " " " " " no blanco

CLAS .- calcular CLAS de acuerdo con:

CLAS=1 si CHAR es digito.

" 2 " " " letra

" 3 " " " /

" 4 " " " + ó -ó*ó(ó)ó=

ADD .- concatenar el trozo de U.S. almacenado hasta ahora

con CHAR.

La programación de estas acciones se ha efectuado con un conjunto de instrucciones con etiquetas de salida que admiten ser definidas antes de que se efectue la transferencia, y por tanto nos permiten usarlas de forma parecida a una subrutina (tengase en cuenta que en el lenguaje de programación usado no existe la posibilidad de usar subrutinas).

El programa fué realizado en el C.C.U.M.; su puesta en marcha fué laboriosa dado que no existia práctica en el manejo del lenguaje usado ni en el de su compilador .

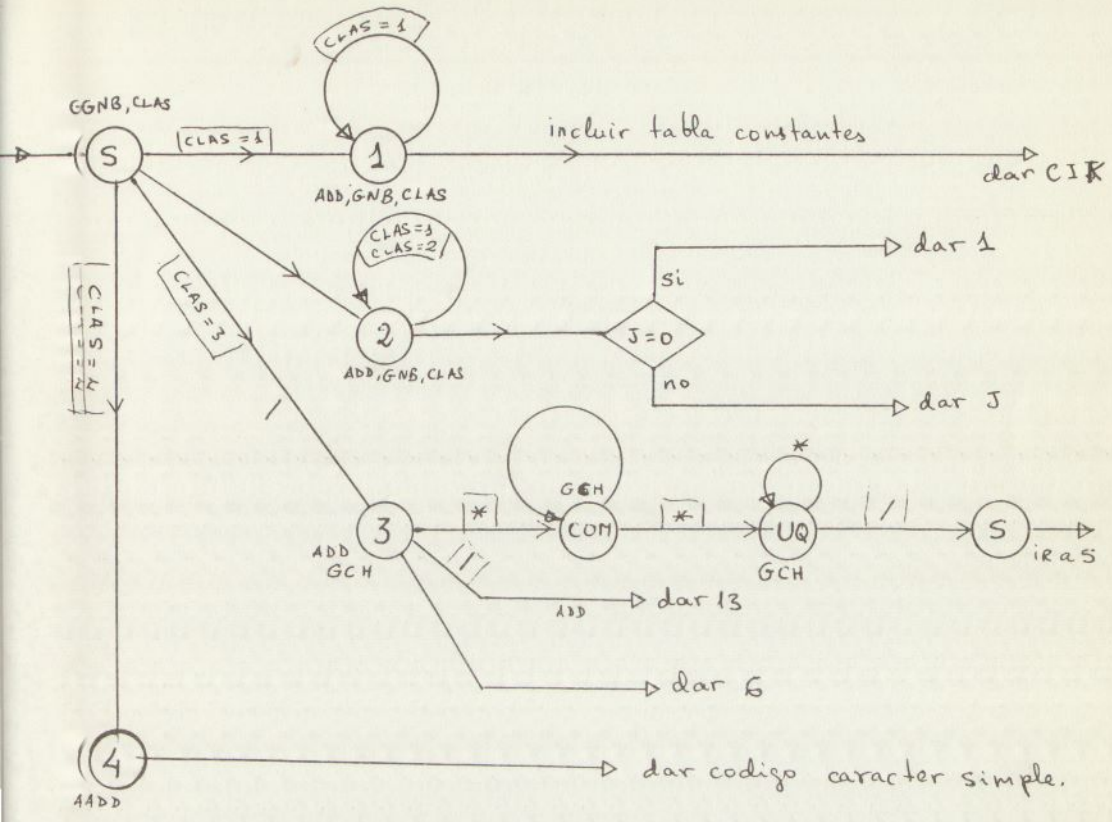

DIAGRAMA DE ESTADOS CON ACCIONES SEMANTICAS
INDICADAS DE LA SUBROUTINA SCANNER