

3.5.1 Cuestiones resueltas de la Sesión 5

5.1 Amplía la calculadora para que admita las variables a, b, ..., z como variables distintas a A, B, ... Z.

Una solución sencilla es asignar índices distintos a las variables A, B, ...Z y a a, b, ..., z. Por ejemplo, a a:0, b:1, ..., z:25, A:26, B:27, ..., Z:52. En el analizador lexicográfico tendríamos:

```
int yylex ()
{
    . . .
    if (c >= 'A' && c <= 'Z') {
        yylval.indice = c - 'A' + 26 ;
        return VARIABLE ;
    }
    . . .
}
```

A la hora de indexar en la semántica para recuperar el nombre, hay que tener en cuenta en qué rango se mueve el índice:

```
axioma:
    . . .
        | VARIABLE '=' expresion '\n' {
            memoria [$1] = $3;
            if ($1 < 26) {
                printf ("%c", $1+'a');
            } else {
                printf ("%c", $1+'A'-26);
            }
            printf ("=%lf\n", $3);
        }
        r_expr
    ;
    . . .
```

Queda por dimensionar la memoria de las variables:

```
%{
    /* SECCION 1 */
#include <stdio.h>
double memoria [52] ;    /* Se define una zona de memoria para las variables */
%}
```

También se puede asignar los valores de índice siguiendo la distancia en la tabla ascii entre mayúsculas y minúsculas (de 32 unidades). Tendríamos entonces:

```
int yylex ()
{
    . . .
    if (c >= 'a' && c <= 'z') {
        yylval.indice = c - 'a' + 32 ;
        return VARIABLE ;
    }
    . . .
}
```

Bastaría con dimensionar el vector de variables de forma más generosa:

```
%{
    /* SECCION 1 */
#include <stdio.h>
double memoria [60] ;    /* Se define una zona de memoria para las variables */
%}
```

No sería necesario ningún cambio más respecto a la versión de calc5.y original.

5.2 Crea el analizador léxico adecuado para la calculadora obtenida en el punto anterior con *flex*.

calc5.lex

```
%{
/* Primera Seccion */
#include "calc5.tab.h"      /* Cabecera con decl. de token etc. */
extern int n_linea;
}%
%%
/* Segunda Seccion */
[ \t]          { ; } /* ignorar espacios y tabuladores */
[0-9]+\.[0-9]*| [0-9]+\.[0-9]+ { sscanf (yytext, "%lf", &yylval.valor);
return (NUMERO); }
[A-Z]          { sscanf (yytext, "%c", &yylval.indice);
yylval.indice -= 'A' ;
return (VARIABLE);
}
[a-z]          { sscanf (yytext, "%c", &yylval.indice);
yylval.indice = yylval.indice - 'a' + 32 ;
return (VARIABLE);
}
\n            { n_linea++; return ('\n'); }
.             { return (yytext [0]); } /* literales */
%%
/* Tercera Seccion */
int yywrap ()    /* se incluye para evitar un error de compilacion */
{
    return (0) ; /* para el control de fin de fichero */
}
```