



Instituto Politécnico Nacional.
Escuela Superior De Cómputo.



Materia:
Compiladores.

Tema:
Práctica 03.
(Reporte)

Profesor:
Roberto Tecla Parra.

Alumno:
Mario Alberto Miranda Sandoval.

Grupo:
3CM7

Tabla de Símbolos.

Descripción.

En esta práctica se guarda la posibilidad de definir variables en la calculadora de vectores, esto se logra al añadir una tabla de símbolos y modificaciones a la gramática para que si el usuario teclea "var1 = [1 2 3]" (sin las comillas) este se guarde o se busque en la tabla de símbolos.

Modificaciones al código.

```
1. %union{
2.     Vector *vector;
3.     double numero;
4.     Symbol *sym;
5. }
```

Primero a la unión le añadimos un apuntador de tipo Symbol.

```
1. %token <numero> NUMBER
2. %token <sym> VAR INDEF
3. %type <vector> exp vector component asgn
```

Ahora añadimos tokens necesarios para la tabla de símbolos.

```
1. asgn:      VAR '=' exp {$$ = $1->u.val = $3; $1->type=VAR;}
2. exp:      vector      { $$ = $1; }
3.          | VAR        { if($1-
>type == INDEF)execerror("Variable no definida, ", $1->name);
4.                                $$=$1->u.val;
5.                                }
6.          | asgn        { $$ = $1; }
```

En estas líneas comprobamos si la variable esta definida, además, de la asignación.

```
1. if(isalpha(c) && c!='X'){
2.     Symbol *s;
3.     char sbuf[200], *p=sbuf;
4.     do {
5.         *p++=c;
6.     } while ((c=getchar())!=EOF && isalnum(c) && c!='X');
7.     ungetc(c, stdin);
8.     *p='\0';
9.     if((s=lookup(sbuf))==(Symbol *)NULL)
10.        s=install(sbuf, INDEF, (Vector *)NULL);
11.     yylval.sym=s;
12.     if(s->type == INDEF){
13.         return VAR;
14.     } else {
15.         //printf("func=(%s) tipo=(%d) \n", s->name, s->type);
16.         return s->type;
17.     }
```

Ahora en yylex, añadimos la tabla de símbolos, posteriormente checamos si se encuentra la variable definida en la tabla de símbolos, sino es así, entonces la mandamos a meter en la tabla de símbolos, si esta definida entonces regresamos la variable sino regresamos el tipo.

Pruebas.

```
mario@mario-Aspire-A515-51: ~/Documentos/Compiladores/Practica 03
Archivo Editar Ver Buscar Terminal Ayuda
mario@mario-Aspire-A515-51:~/Documentos/Compiladores/Practica 03$ ./a.out
var1 = [1 2 3]
1.00 2.00 3.00
var2 = [2 4 6]
2.00 4.00 6.00
var1 + var2
3.00 6.00 9.00
var1 - var2
-1.00 -2.00 -3.00
var1 * var2
28.00
i = [1 0 0]
1.00 0.00 0.00
j = [0 1 0]
0.00 1.00 0.00
k = [0 0 1]
0.00 0.00 1.00
j X k
1.00 0.00 0.00
i X j
0.00 0.00 1.00
i X k
0.00 -1.00 0.00
j X i
0.00 0.00 -1.00
```