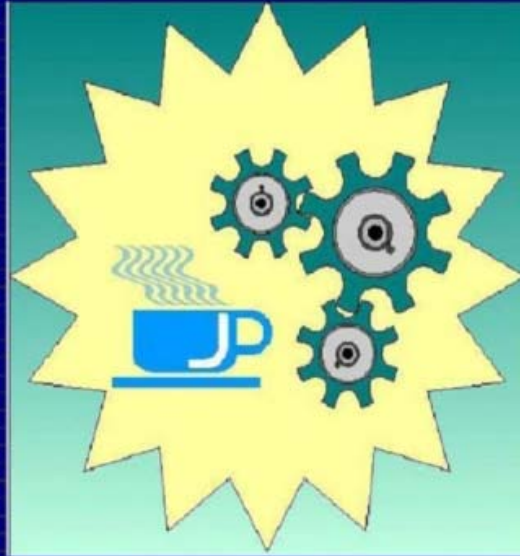


# Compiladores

## Teoría y práctica

*con Java, JLex, Cup y Ens2001*



**Compiladores: Teoría Y Práctica Con Java, Jlex, Cup Y Ens2001**

**Jacinto Ruiz Catalán**

**Editor:** Lulu.com (6 de noviembre de 2008)

**ISBN-10:** 1409230384, **ISBN-13:** 978-1409230380

#

#

Ahora, pondremos un ejemplo de utilización para ver como funciona. Supongamos un subconjunto del lenguaje Pascal. Supondremos que se permite la definición de tipos de registro y de vector, aparte de los dos tipos básicos **integer** y **boolean**. Es decir, se permiten estas definiciones, por ejemplo:

```
type vector=array[0..10] of integer;
type registro=record
  x:integer;
  y:boolean;
end;
```

Supongamos que tenemos el siguiente programa y veremos cómo va cambiando el contenido de la tabla de tipos conforme se avanza en el análisis del programa.

```
1: program p;
2:   type vector=array[0..9] of integer;
3:   type registro=record
4:     x:integer;
5:     y:boolean;
6:   end;
7:   procedure p1;
8:     type vector=array[0..8] of boolean;
9:     type r1=record
10:       y:integer;
11:       z:integer;
12:     end;
13:   begin
14:   end;
15: begin
16: end.
```

#### Programa 1

Vamos a suponer que este subconjunto de Pascal es sensible a las mayúsculas ( aunque realmente Pascal no lo es ).

Comencemos procesando el programa. Inicialmente, se añaden los tipos básicos a la tabla de tipos.

#

#

#

Cod	Nombre	TipoBase	Padre	Dimensión	Mínimo	Máximo	Ambito
0	integer	-1	-1	1	-1	-1	0
1	boolean	-1	-1	1	-1	-1	0

Ahora, procesamos la línea 1, luego la 2.

Cod	Nombre	TipoBase	Padre	Dimensión	Mínimo	Máximo	Ambito
0	integer	-1	-1	1	-1	-1	0
1	boolean	-1	-1	1	-1	-1	0
2	vector	0	-1	10	0	9	0

Procesamos la línea 3.

Cod	Nombre	TipoBase	Padre	Dimensión	Mínimo	Máximo	Ambito
0	integer	-1	-1	1	-1	-1	0
1	boolean	-1	-1	1	-1	-1	0
2	vector	0	-1	10	0	9	0
3	registro	-1	-1	-1	-1	-1	0

Procesamos la línea 4 y 5.

Cod	Nombre	TipoBase	Padre	Dimensión	Mínimo	Máximo	Ambito
0	integer	-1	-1	1	-1	-1	0
1	boolean	-1	-1	1	-1	-1	0
2	vector	0	-1	10	0	9	0
3	registro	-1	-1	2	-1	-1	0
4	x	0	3	1	-1	-1	0
5	y	1	3	1	-1	-1	0

#

#

#

#

Una aclaración. Como hemos visto, hemos incluido los dos campos del registro en la tabla de tipos. Esto es necesario ya que cuando declaremos una variable de este tipo, deberemos acceder a cada campo de la variable por el nombre que tiene el campo en la tabla de tipos. Además, hemos calculado la dimensión del tipo 3 ( registro ) sumando las dimensiones de sus hijos ( sus campos ). Esto se ha hecho a posteriori ya que en el momento de procesar la línea 3 no se ha procesado aún la 4 ni la 5. Más adelante se comentará cómo calcular la dimensión de los tipos complejos.

Procesemos las líneas 6, 7 y 8. Tenemos la siguiente tabla.

Cod	Nombre	TipoBase	Padre	Dimensión	Mínimo	Máximo	Ámbito
0	integer	-1	-1	1	-1	-1	0
1	boolean	-1	-1	1	-1	-1	0
2	vector	0	-1	10	0	9	0

#

3	registro	-1	-1	2	-1	-1	0
4	x	0	3	1	-1	-1	0
5	y	1	3	1	-1	-1	0
6	vector	1	-1	9	0	8	1

Vemos que como estamos en un ámbito más interno que el anterior, se ha incrementado el número del ámbito. También vemos que hemos definido un tipo con el mismo nombre de otro tipo anterior. Esto es posible ya que está en otro ámbito. Si queremos buscar el tipo de una variable, comenzaremos por el final, por lo que el que utilizaremos será el del ámbito 1.

Procesamos la línea 9.

#

Cod	Nombre	TipoBase	Padre	Dimensión	Mínimo	Máximo	Ámbito
0	integer	-1	-1	1	-1	-1	0
1	boolean	-1	-1	1	-1	-1	0
2	vector	0	-1	10	0	9	0
3	registro	-1	-1	2	-1	-1	0
4	x	0	3	1	-1	-1	0
5	y	1	3	1	-1	-1	0
6	vector	1	-1	9	0	8	1
7	rl	-1	-1	-1	-1	-1	1

#

#

#

Ahora, procesamos las líneas 10 y 11.

Cod	Nombre	TipoBase	Padre	Dimensión	Mínimo	Máximo	Ambito
0	integer	-1	-1	1	-1	-1	0
1	boolean	-1	-1	1	-1	-1	0
2	vector	0	-1	10	0	9	0
3	registro	-1	-1	2	-1	-1	0
4	x	0	3	1	-1	-1	0
5	y	1	3	1	-1	-1	0
6	vector	1	-1	9	0	8	1
7	rl	-1	-1	2	-1	-1	1
8	y	0	7	1	-1	-1	1
9	z	0	7	1	-1	-1	1

Vemos que se ha modificado la dimensión del registro al procesar los dos campos. Procesemos ahora las líneas 12 y 13. La tabla no cambia. Pero al procesar la 14, terminamos la zona del procedimiento y por tanto, eliminamos el ámbito y por tanto los tipos que están en él. La tabla quedaría así.

Cod	Nombre	TipoBase	Padre	Dimensión	Mínimo	Máximo	Ambito
0	integer	-1	-1	1	-1	-1	0
1	boolean	-1	-1	1	-1	-1	0
2	vector	0	-1	10	0	9	0
3	registro	-1	-1	2	-1	-1	0

#

4	x	0	3	1	-1	-1	0
5	y	1	3	1	-1	-1	0

Finalmente se procesan las líneas 15 y 16. Al procesar la línea 16 ( la última línea del programa ), ya no necesitamos la tabla de tipos y por lo tanto la podemos eliminar.

#