

Prueba puntuable del tema 6: generación de código

Procesadores de lenguaje - II26 - TE2

19 de mayo de 2008

En los siguientes ejercicios se plantean dos posibles extensiones del compilador `minicomp`. En el primero se pide escribir únicamente la correspondiente función de generación de código. En el segundo se pide describir todos los cambios necesarios, tanto para construir el AST como para generar código a partir de él. En ambos casos, la función de generación de código debe estar escrita en PYTHON y ser ejecutable.

1. **(0.125 puntos)** Supongamos que queremos modificar `minicomp` para que el lenguaje de entrada tenga un nuevo operador, el operador lógico `xor` (o exclusivo). Para ello, hemos decidido implementar una nueva clase de nodos del AST, la clase `NodoXor`. Escribe la función `codigoControl` de dicha clase. Para simplificar el problema, basta con que resuelvas el caso en que las dos etiquetas que se le pasan a la función son diferentes de `None`. Recuerda que la tabla de verdad del operador `xor` es la siguiente:

A	B	A xor B
cierto	cierto	falso
cierto	falso	cierto
falso	cierto	cierto
falso	falso	falso

2. **(0.125 puntos)** Supongamos ahora que queremos modificar `minicomp` para que el lenguaje de entrada acepte una nueva sentencia que permita invertir un vector empleando la sintaxis

```
invierte (vector);
```

donde `invierte` es una nueva palabra reservada y `vector` un identificador, posiblemente seguido de expresiones entre corchetes. El tipo de `vector` debe ser vectorial con tipo base simple. El resultado de la ejecución es que el primer elemento se intercambia con el último, el segundo con el penúltimo y así sucesivamente. Por ejemplo, dadas las declaraciones

```
v: vector[4] de cadena;  
matriz: vector [5] de vector [5] de entero;
```

las sentencias

```
invierte(v); invierte(matriz[2]);
```

invertirían todos los elementos de `v` y los de la tercera fila de `matriz`.

Ten en cuenta que los posibles efectos secundarios del acceso a `vector` se deben producir sólo una vez, y que la longitud del código generado (medida en instrucciones) debe ser independiente de la talla del vector.