



4º Ingeniería Informática

II26 Procesadores de lenguaje

Control de la práctica de miniint (19 de abril de 2008)

INSTRUCCIONES:

- La duración del examen es de dos horas.
- Antes de empezar, asegúrate de que el usuario con el que estás trabajando coincide con el del recuadro del final de este enunciado.
- Rellena el recuadro con tus datos.
- Crea en el *home* de tu usuario un fichero llamado `datos` que contenga tres líneas: la primera con tu DNI, la segunda con tu nombre y la tercera con tus apellidos.
- Al introducir el USB, debería montarse automáticamente en el directorio `/media/disk`, si no tiene ninguna partición, o en `/media/disk1`, si los datos están en la partición 1. Si fuera necesario, puedes montarlo y desmontarlo mediante `mount` y `umount` sobre el directorio apropiado.
- Cuando termines el examen, entrégnos esta hoja.

PREGUNTA 1

(0,5 PUNTOS)

El objetivo de este ejercicio es que tu intérprete MINIINT acepte, además de lo que se pedía en la práctica 3, sentencias iterativas (bucles) con los requisitos que se especifican a continuación. Para ello, añade al esquema de traducción lo siguiente:

```
<Sentencia> ->
  para <AccesoVariable> desde <Expresion> hasta <Expresion> hacer <Sentencia> fin
  @Sentencia.arb= AST.NodoPara(AccesoVariable.arb, Expresion1.arb, Expresion2.arb, Sentencial1.arb, para.nlinea)@
  ;
```

y añade "para", "desde", "hasta" y "hacer" a la lista de palabras reservadas que se encuentra en el mismo fichero¹. Sin modificar nada más en el esquema de traducción, debes implementar lo necesario para poder ejecutar programas en los que se utilicen ese tipo de sentencias, teniendo en cuenta lo siguiente:

1. Se debe comprobar que tanto la variable de control del bucle como las expresiones inicial y final sean de tipo entero. Si no es así, el mensaje que debes pasar a `errores.semantico` es exactamente "Error de tipos en para."
2. El comportamiento del bucle debe ser el que ilustra la figura 1 (y, por tanto, las expresiones inicial y final se deben evaluar antes de la primera iteración).
3. La variable de control del bucle es una variable del programa como cualquier otra, que debe ser declarada igual que las demás y puede ser utilizada en cualquier otra sentencia antes, durante y después de la ejecución del bucle. En particular, se debe poder modificar dentro del bucle, afectando con ello al valor que tendrá en la siguiente iteración, y será responsabilidad del usuario que ello no se convierta en un bucle sin fin. Su valor al terminar el bucle será, tal como se deduce del diagrama anterior, el resultado del último incremento si se ha realizado alguna iteración, y el resultado de evaluar la expresión inicial si no se ha realizado ninguna iteración.
4. La variable de control del bucle podría ser una componente de un vector unidimensional o multidimensional, siempre que sea una componente de tipo entero. En ese caso, será también responsabilidad del usuario que el comportamiento del programa no dependa de cuántas veces se evalúen las expresiones de los índices del vector.
5. Se deben poder utilizar bucles dentro de bucles, de forma anidada.

¹Tras modificar ese fichero, debes compilarlo de nuevo utilizando, por ejemplo, `../metacomp/metacomp miniint.mc -s miniint`

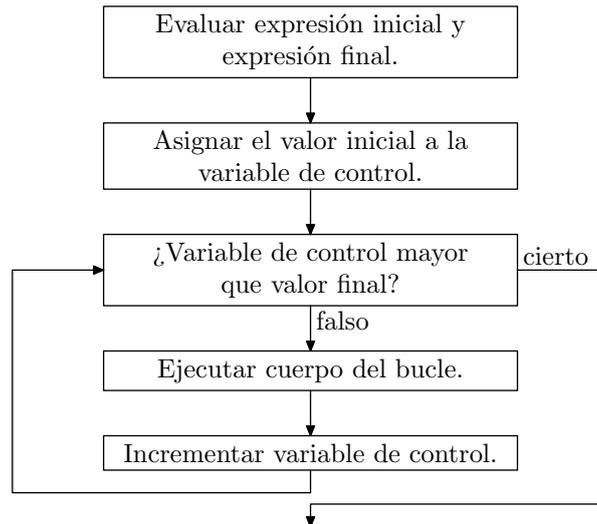


Figura 1: Diagrama de flujo del bucle para.

6. Se deben poder utilizar bucles dentro de funciones. En ese caso, la ejecución de una sentencia `devuelve` mientras se ejecuta un bucle debe tener el comportamiento habitual, finalizando la ejecución de la función y por tanto también del bucle.

El siguiente ejemplo ilustra algunos de estos aspectos (todos los bucles se comportan de acuerdo con el diagrama anterior, pero presta especial atención al valor de la variable de control al terminar cada uno y al comportamiento del tercer bucle, en el que se modifica la variable de control dentro del bucle):

```

globales
  v : vector[10] de vector[20] de vector[30] de entero;
  i : entero;
fin
secuencia
  para i desde 2 + 3 hasta 9 - 2 hacer
    secuencia
      escribe i; escribe " ";
    fin
  fin
  escribe i; nl;

  para i desde 333 hasta 222 hacer
    secuencia
      escribe i; escribe " ";
    fin
  fin
  escribe i; nl;

  para i desde 100 hasta 500 hacer
    secuencia
      escribe i; escribe " ";
      i := i * 2;
      escribe i; escribe " ";
    fin
  fin
  escribe i; nl;

  para v[3][4][5] desde 6 hasta 9 hacer
    secuencia
      escribe v[3][4][5] * 10; escribe " ";
    fin
  fin
  escribe v[3][4][5]*10; nl;
fin

```

Al ejecutar este ejemplo con tu solución, la salida debe ser la siguiente (diseña tú mismo pruebas de los demás requisitos si lo consideras conveniente):

```
5 6 7 8
333
100 200 201 402 403 806 807
60 70 80 90 100
```

Cuando termines de resolver el ejercicio, guarda los ficheros del intérprete en el directorio `miniint` del home de tu usuario. El fichero principal debe llamarse `miniint` y ser ejecutable.