UNIVERSITAT JAUME·I

4º Ingeniería Informática

II26 Procesadores de lenguaje

Control de la práctica de minicomp (29 de mayo de 2010)

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.
- Ejecuta el programa `./prepara.py` pasándole como parámetro tu DNI. Este creará un fichero con tu DNI y nombre y un directorio llamado `minicomp`, donde debes guardar tu compilador.
- Al introducir el USB, debería montarse automáticamente. Si fuera necesario, puedes montarlo y desmontarlo mediante `mount` y `umount` sobre el directorio apropiado.
- Cuando termines el examen, entréganos esta hoja.

EJERCICIO (1,5 PUNTOS)

En este ejercicio debes cambiar tu compilador minicomp, que debe incluir obligatoriamente las modificaciones que se pedían en las versiones 0 a 9 de la práctica 4 y, opcionalmente, las 10 y 11. Tras el cambio, minicomp deberá aceptar asignaciones múltiples con los requisitos siguientes:

1. Las asignaciones seguirán teniendo una expresión en la parte derecha, pero en la parte izquierda pasarán a tener una lista de uno o más accesos a variable separados por comas.
2. Los posibles efectos secundarios de evaluar la expresión de la parte derecha ocurren sólo una vez y antes de acceder a cualquiera de las variables de la parte izquierda. Por ejemplo, en la sentencia

   ```
   v[llama f(1)]= llama g(2);
   ```

   la llamada a `g` sucede antes que la llamada a `f`.
3. Los posibles efectos secundarios de cada acceso a variable en la parte izquierda de la asignación también suceden una sola vez y de izquierda a derecha, aunque no se define si ocurrirán antes o después de la asignación al resto de accesos a variable. Por ejemplo, en la sentencia

   ```
   v[llama f(1)], v[llama g(2)] = 4;
   ```

   no se garantiza que en la llamada a `g` el valor de `v[llama f(1)]` sea `4`, pero sí que se llamará a `f` antes que a `g`.
4. Las reglas de compatibilidad de tipos son las mismas que en la versión 9 de minicomp, lo que implica, por ejemplo, que se pueden asignar enteros a variables reales o vectores entre sí. Sin embargo, se exige que todos los accesos a variable de la parte izquierda tengan el mismo tipo.
5. Del mismo modo, se sigue aplicando el requisito de la versión 9: "Al realizar asignaciones entre vectores, la longitud del código generado, medida en instrucciones, debe ser independiente del tamaño de los vectores".
6. Debes cambiar todos los mensajes de error dentro de la función `compsemanticas` del `NodoAsignacion` de modo que el mensaje que se pasa a `errores.semantico` sea exactamente `"Asignacion incorrecta."` (sin acento y con un punto).

Por ejemplo, suponiendo las declaraciones

```
r: real;
n: entero;
v: vector [30] de real;
m: vector [10] de vector [20] de vector [30] de real;
```

algunas sentencias válidas serían:

```
n = 7;  // La asignación habitual sigue siendo válida
v[11], r, v[n] = 2*n+1; // Hay promoción de entero a real
m[4+4][5-5], m[n][8] = v; // Podemos asignar vectores
```

Para que tengas una prueba inicial de tu compilador, hemos dejado en el *home* de tu usuario el fichero `ejemplo.i` junto con los correspondientes `.o` y `.e`. Obviamente, debes preparar tus propias pruebas para verificar que tu compilador se comporta correctamente también en aquellos aspectos no abarcados por las nuestras.

Recuerda: los ficheros de tu compilador deben estar en el directorio `minicomp` del *home* de tu usuario. El fichero principal debe llamarse `minicomp` y ser ejecutable. Puedes utilizar el *script* `verifica.sh` de tu directorio *home* para comprobar que la estructura es la que pedimos. **Se penalizarán los exámenes para los que este script dé errores.**