

PROGRAMACIÓN DECLARATIVA G. INFORMÁTICA CURSO 2013-14
PRÁCTICA OBLIGATORIA

Objetivo de la práctica: Programar en Haskell un intérprete de un lenguaje imperativo sencillo.

Descripción: Las características del lenguaje considerado son las siguientes:

- Solo se consideran variables de tipo entero, que no hace falta declarar.
- Hay tres tipos de instrucciones:
 - Instrucción de asignación: $x := e$, donde x es una variable y e es una expresión aritmética. Una expresión aritmética está formada por enteros, variables, sumas, restas y multiplicaciones de otras expresiones.
 - Instrucción condicional: $\text{Cond } b \ p \ p'$, donde b es una expresión booleana y p, p' son programas. Las expresiones booleanas son: comparaciones ($=, <, >, <=, >=$) de expresiones aritméticas, negaciones, conjunciones y disyunciones de otras expresiones booleanas.
 - Instrucción de bucle: $\text{While } b \ p$, donde b es una expresión booleana y p es un programa.
- El efecto de ejecutar una instrucción es modificar el estado (valor actual) de las variables, de acuerdo con la forma estándar de entender el significado de las instrucciones.
- Un programa no es más que una secuencia de instrucciones que se ejecutan, claro, en secuencia.
- El estado inicial de la ejecución de un programa no se determina en el propio programa, sino que se supone dado de manera adicional al comenzar la ejecución.
- El resultado final de la ejecución de un programa es el valor que tenga al final una variable distinguida de nombre R .

Ejemplo: El siguiente programa calcula el factorial de X (más exactamente, del valor que tenga la variable X en el estado inicial):

```
Y := X
R := 1
While Y>0  R := R*Y
           Y := Y-1
```

Si ejecutamos ese programa a partir de un estado inicial en el que X tenga almacenado el valor 3, al terminar la ejecución nos encontramos el valor 6 en la variable R .

¿Qué se debe implementar?

- Como mínimo (por un valor de 1 punto):
 - Tipos (o alias de tipos) para representar adecuadamente instrucciones, programas, expresiones aritméticas, expresiones booleanas, estados, etc.
 - Una función `ejecuta :: Programa -> Estado -> Integer` que proporcione el resultado de la ejecución de un programa a partir de un estado inicial. Por supuesto, pueden ser necesarias algunas funciones auxiliares.

- Sendas funciones de aridad cero **factorial** y **s0** (para ejemplos) cuyos valores sean, respectivamente:
 - el programa para el cálculo del factorial escrito arriba, de acuerdo con la representación determinada por los tipos que se hayan definido,
 - el estado inicial en el que la variable **X** vale 3 y las demás no tienen valor definido.

Por ejemplo, de acuerdo con una representación que tengo yo en mente (y que podéis si queréis utilizar vosotros también) tendríamos:

```
factorial = ["Y" := V "X",
            "R" := I 1,
            While (I 0 < V "Y")
              ["R" := V "R" :* V "Y",
              "Y" := V "Y" :- I 1
            ]
]

s0 = [("X",3)]
```

Y la ejecución del programa en el intérprete de GHC resultaría en lo siguiente:

```
> ejecuta factorial s0
6
```

- **No se pide** que se haga ningún tipo de análisis sintáctico de los programas, sino que se supone que están escritos en la representación abstracta dada por los tipos (como en el ejemplo de **factorial** anterior).

■ Otro contenido posible (por valor de 0,5 puntos):

Realizar alguna interacción simple con el usuario, como por ejemplo:

- Suponiendo que el programa a ejecutar está escrito en un fichero (pero, igual que antes, en su representación abstracta), se le pide al usuario que dé por pantalla el nombre del fichero y el estado inicial.
- Permitir realizar la ejecución paso a paso (controlada, por ejemplo, por pulsaciones de una tecla) en la que se vaya viendo cómo se va modificando el estado paso a paso.

¿Qué, cómo y cuándo se debe entregar?

- La entrega se realizará a través del Campus Virtual y consistirá en un solo fichero de extensión **.hs** en el que las explicaciones irán como comentarios Haskell.
- **Fecha límite** para la entrega: **5 de junio**

¿Cuánto influye la calificación de la práctica en la calificación final?

- La nota de la práctica supone el **15 % de la nota final (1,5 puntos)**.

- El trabajo es individual. La copia de otros compañeros o de cualquier otra fuente, así como facilitar la copia a otros, será severamente castigado en la calificación **global** de la asignatura. Ante las dudas, consultad con el profesor.