

Programación para la Inteligencia Artificial

Haskell - Práctica 5 (Parte I)

Listas por comprensión o intensionales

El ejercicio consiste en modelar los datos de préstamos almacenados en una **biblioteca**. Se supone que la biblioteca usa una base de datos para almacenar los registros de los libros prestados a diferentes lectores. Por tanto, debemos:

- Definir el tipo de dato adecuado para la base de datos de la biblioteca, por ejemplo `[(Persona, Libro)]`, aunque no tiene que ser necesariamente éste.
- Definir funciones para extraer información de la BD. En particular, debemos definir funciones para las siguientes tareas:
 - Dada una persona, obtener los libros que tiene en préstamo (función `libros`). Podría tener el tipo:


```
libros :: BD -> Persona -> [Libro]
```
 - Dado un libro, obtener todos los lectores que lo tienen en préstamo (función `lectores`). Podría tener el tipo:


```
lectores :: BD -> Libro -> [Persona]
```
 - Dado un libro, obtener la información acerca de si está o no prestado (función `prestado`). Podría tener el tipo:


```
prestado :: BD -> Libro -> Bool
```
 - Dada una persona, encontrar el número de libros que tiene prestados (función `numPrestados`). Podría tener el tipo:


```
numPrestados :: BD -> Persona -> Int
```
- Definir funciones para modificar la BD. En particular, son necesarias dos funciones:
 - Una para actualizar la BD con un nuevo préstamo (función `realizarPrestamo`). Podría tener el tipo:


```
realizarPrestamo :: BD -> Persona -> Libro -> BD
```
 - Otra para actualizar la BD con una devolución de libro (función `devolverPrestamo`). Podría tener el tipo:


```
devolverPrestamo :: BD -> Persona -> Libro -> BD
```

La BD puede ser modelada de diferentes formas aunque, en cualquier caso, debe consistir en una lista con la información de los préstamos. Esta información puede almacenarse como pareja de persona y libro, o pareja de persona con su lista de libros prestados o bien pareja de libro con lista de personas que lo tienen en préstamo. Por otro lado, usaremos siempre la base de datos como parámetro de entrada y de salida en las funciones de actualización.

Posible ampliación

Podríamos añadir una lista con el número de ejemplares de cada libro (por ejemplo, como una lista de parejas (`Libro`,`Int`), a la que podríamos llamar `NumEjem`), y gestionarla en los accesos a la BD de préstamos. Para ello debemos modificar las funciones de préstamo y devolución, para poder consultar y actualizar la lista `NumEjem`. Podrían tener el nuevo tipo:

```
realizarPrestamo :: BD -> NumEjem -> Persona -> Libro -> (BD,NumEjem)
y
devolverPrestamo :: BD -> NumEjem -> Persona -> Libro -> (BD,NumEjem)
```

Necesitaríamos, además, funciones auxiliares, por ejemplo:

- Para comprobar si hay al menos un ejemplar disponible de un libro:
`disponibleLibro :: NumEjem -> Libro -> Bool`
- Para añadir un ejemplar nuevo a la biblioteca:
`nuevoEjemplar :: NumEjem -> Libro -> NumEjem`
- Y dos funciones para devolver y extraer, respectivamente, un ejemplar:
`devuelveEjemplar,extraeEjemplar :: NumEjem -> Libro -> NumEjem`