Programación para la Inteligencia Artificial
Haskell - Práctica 2
Uso de módulos y directorios en GHCi

Presentación de imágenes en formato ASCII

Para realizar esta tarea, usaremos el módulo `Pictures.hs` que contiene la implementación de una serie de funciones para manejar imágenes en formato ASCII. La imágenes, renombradas con el nombre `Picture`, son en este caso listas de listas. Algunas de las funciones son:

- `printPicture` muestra una imagen en el terminal.
- `above` Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una encima de la otra.
- `beside` Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una al lado de la otra.
- `superimpose` Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una sobre otra.
- `flipH` da la vuelta a la imagen sobre el eje horizontal.
- `flipV` da la vuelta a la imagen sobre el eje vertical.
- `invertColour` invierte el blanco y el negro en la imagen.

1. **Uso del módulo** Podemos ejecutar `GHCi` directamente con:

   ```
   > GHCi Pictures.hs
   ```

   o bien, dentro del entorno hacer un `load` del módulo:

   ```
   Prelude> :l Pictures
   ```

   o

   ```
   Prelude> :m +Pictures
   ```

   Para evitar los mensajes de advertencia acerca de los tabuladores en la versión 8.0.1, activar la opción correspondiente:

   ```
   Prelude> :set -Wno-tabs
   ```

   En caso de que el módulo no se encuentre en el directorio correspondiente, podemos ver el directorio por defecto con la opción:

   ```
   Prelude> :show paths
   ```

Si nos interesa modificar esta ruta, lo hacemos con:

`Prelude> :cd otroDirectorio`[1]

En las siguiente sesión, este será el directorio por defecto.

2. **Uso de las funciones** Podemos usar cualquiera de las funciones sobre la imagen `horse` ya predefinida. Por ejemplo:

```
*Pictures> printPicture (horse `beside` (flipV horse))
```

Presentación de imágenes en un navegador web

Para realizar esta tarea, usaremos el módulo `PicturesSVG.hs` que contiene la implementación de una serie de funciones para manejar imágenes en formato SVG (Scalable Vector Graphics), que es una especificación para describir gráficos vectoriales bidimensionales en formato XML. En este caso las imágenes se definen como un nuevo tipo de dato, denominado `Picture`. Algunas de las funciones son:

- `render` muestra una imagen en el navegador.
- `above` Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una encima de la otra.
- `beside` Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una al lado de la otra.
- `over` Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una sobre otra.
- `flipH` da la vuelta a la imagen sobre el eje horizontal.
- `flipV` da la vuelta a la imagen sobre el eje vertical.
- `invertColour` niega cada pixel de la imagen.

1. **Uso del módulo** Podemos ejecutar `GHCi` directamente con:

   `> GHCi PicturesSVG.hs`

   o bien cargarlo tal y como hemos comentado antes.

   En el mismo directorio en el que estemos trabajando, incluimos las imágenes que queremos visualizar, en formato `jpg` y los ficheros `showPic.html` o `refresh.html`.

   Abrimos con el navegador `Firefox` o `Google Chrome` el fichero `showPic.html` o bien `refresh.html` si queremos que la imagen en el navegador se actualice automáticamente.

2. **Uso de las funciones** Podemos usar cualquiera de las funciones sobre la imagen `horse` ya predefinida usando el fichero `blk_horse_head.jpg`. Por ejemplo:

```
*PicturesSVG> render (horse `beside` (flipV horse))
```

[1] El separador de directorios es "/" en sistemas Linux y "\" en sistemas Windows.

3. Definición de nuevas funciones

Ej.1 Definir una función que tome como entrada una imagen y devuelva otra con la imagen cuadruplicada con el siguiente formato:

IMAGEN	IMAGEN EN ESPEJO Y CON COLORES INVERTIDOS
IMAGEN CON COLORES INVERTIDOS	IMAGEN EN ESPEJO

Para este ejercicio se recomienda almacenar las funciones definidas en un archivo, en el que debemos importar la librería correspondiente, añadiéndola en la cabecera:

```
import PicturesSVG
```

Puede probarse con la imagen predefinida `horse`.

Ej.2 Usando las funciones de la librería y las imágenes `black` y `white`, definir una función recursiva que, tomando como entrada dos números `n` y `m`, devuelva una imagen con un tablero tipo ajedrez que contenga las filas y columnas indicadas, respectivamente, por `n` y `m`.

Ej.3 Finalmente, usar las imágenes `black`, `white` y `smallHorse` para conseguir un tablero de ajedrez de 8×8, de manera que aparezca, sobre cada casilla de la diagonal blanca, un caballo negro mirando a la izquierda y, sobre cada casilla de la diagonal negra, un caballo blanco mirando hacia la derecha.