

Simulacro de recuperatorio del primer parcial - AYEDI Taller

Soluciones.hs

Mails de contacto:

benjamin.alvarez@mi.unc.edu.ar

rocio.perez.sbarato@mi.unc.edu.ar

Ejercicio 1

Se va a representar el juego **pedra-papel-tijera** usando tipos en Haskell.



Para ello:

a) Definir el tipo `Forma` que consta de los constructores sin parámetros `Piedra`, `Papel` y `Tijera`. El tipo `Forma` **no debe** estar en la clase `Eq`. Luego programar la función

```
leGana :: Forma -> Forma -> Bool
```

que dadas dos valores `f1` y `f2` de tipo `Forma` devuelve `True` si y sólo si la forma `f1` gana sobre la forma `f2`. La regla es que:

- La piedra le gana a la tijera,
- La tijera le gana al papel
- El papel le gana a la piedra

Se puede seguir la siguiente tabla:

f1	f2	leGana f1 f2
Piedra	Tijera	True
Piedra	Piedra	False (empate)

Piedra	Papel	False (la forma f1 perdió)
Papel	Piedra	True
Papel	Papel	False (empate)
Papel	Tijera	False (la forma f1 perdió)
Tijera	Papel	True
Tijera	Tijera	False (empate)
Tijera	Piedra	False (la forma f1 perdió)

b) Definir el tipo `Nombre` como un sinónimo de `String`. Luego definir el tipo `Jugador` que consta de un único constructor `Mano` que toma dos parámetros, el primero de tipo `Nombre` y el segundo de tipo `Forma`. Por último programar la función:

```
ganador :: Jugador -> Jugador -> Maybe Nombre
```

que dados dos valores `j1` y `j2` del tipo `Jugador` debe devolver el nombre del jugador ganador (el de `j1` o `j2` según corresponda) usando el constructor `Just`, o `Nothing` en caso que no haya ganador.

NOTA: dejar como comentario un ejemplo con el que hayas probado la función.

Ejercicio 2

Programar la función

```
quienJugo :: Forma -> [Jugador] -> [Nombre]
```

que dada una forma `f` y una lista de jugadores `js` devuelve los nombres de los jugadores de `js` que usaron la forma `f`

NOTA: dejar como comentario un ejemplo con el que hayas probado la función que tenga una lista de al menos 3 jugadores.