Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales
Departamento de Ciencia de la Computación
Análisis de Lenguajes de Programación

Nombre y Apellido:

Examen Parcial 3

1. La siguiente mónada es utilizada para cómputos que llevan información adicional, aparte de su valor de cómputo.

```
newtype Writer w a = Writer {runWriter :: (a, [w])}
instance Monad (Writer w) where
  return a = Writer (a, [])
  (Writer (a, w)) >>= f = let (a', w') = runWriter (f a)
                          in Writer (a', w ++ w')
```

a) Probar que Writer w es una mónada. Puede asumir que $+\!\!+$ es asociativa sin demostrarlo.

b) Para combinar los efectos de la mónada Writer con los efectos de Maybe se definió el siguiente tipo de datos:

```
newtype WriterMaybe w a = WM {runWM :: (Maybe a, [w])}
```

Dar la instancia de Monad para WriterMaybe w.

c) Definir las siguientes operaciones sobre la mónada WriterMaybe:

i. tell :: $[w] \to$ WriterMaybe w (), agrega la información dada.

ii. fail :: WriterMaybe e a, que representa una falla.

d) Se desea utilizar la mónada WriterMaybe para definir un firewall simple que filtre paquetes según reglas. Además de filtrar paquetes el firewall producirá mensajes de logs sobre su actividad.

Dadas las siguientes definiciones que representan los tipos para reglas de filtrado, paquetes y resultados de la aplicación de una regla:

```
type Rule = String
type Packet = String
data Result = Accepted | Rejected
  deriving (Show, Eq)
```

Suponinedo definida una función match :: [Rule] → Packet → [(Rule, Result)], que dada una lista de reglas y un paquete, devuelva una lista con las reglas que matchearon el paquete y el resultado de la aplicación de éstas sobre el paquete; definir una función filterPacket :: [Rule] → Packet → WriterMaybe Char Packet, que determine si un paquete es aceptado por las reglas y agregue mensajes del resultado como logs.

Si un paquete P no machea ninguna regla debe mandar un mensaje de log "UNMATCHED PACKET P" y fallar, sino debe agregar un mensaje "MATCHED P WITH RULE R" por cada regla macheada junto con un mensaje "RULE Accepted" o "RULE Rejected" (dependiendo si la regla acepta o no el paquete). Si todas las reglas aceptan el paquete lo retorna y sino falla.

Ayuda: Puede usar la función mapM :: $(a \to m\ b) \to [a] \to m\ [b]$ definida en práctica.

2. Probar que T es un functor.

```
data T a = Uno a | Dos (a, a) | More (a, a) (T a)
```