UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE CIENCIAS
LÓGICA COMPUTACIONAL

Profesora: ESTEFANÍA PRIETOS LARIOS
Ayudante: HERNÁNDEZ OLVERA MAURICIO E.
Ayudante Lab: SALAZAR GONZÁLEZ EDWIN MAX

Fecha de Entrega: Jueves 28 de Febrero, 2019. 23:59

PRÁCTICA 1. Parte 3.

RECURSIÓN.
LÓGICA PROPOSICIONAL.

OBJETIVOS

- Aplicar el concepto de recursión en un lenguaje funcional como lo es Haskell.
- Reforzamiento y aplicación de temas sobre la lógica proposicional.

INSTRUCCIONES

Su misión, si deciden aceptarla es:

Descargar el archivo *Practica1p3.hs* y resolver los ejercicios definidos sobre éste.

EJERCICIOS

Función **neighbors**
Recibe un nodo, una gráfica y regresa la lista de nodos que son vecinos del nodo dado como parámetro.

```
*Practica1p3> let G = Graph {gId = 1, nodes = ["A", "B", "C", 'D'"], edges = [("A", "B"), ("C", "D"), ("D", "A")]}

*Practica1p3> neighbors "A" G

["B", "D"]
```

Función **mindegree**
Recibe una gráfica y nos devuelve el grado menor de la gráfica.

```
*Practica1p3> let G = Graph {gId = 1, nodes = ["A", "B", "C", 'D'"], edges =
[("A", "B"), ("C", "D"), ("D", "A"), ("A", "C")]}

*Practica1p3> mindegree G

1
```

Función **maxdegree**
Recibe una gráfica y devuelve el grado mayor de la gráfica

```
*Practica1p3> let G = Graph {gId = 1, nodes = ["A", "B", "C", 'D'"], edges =
[("A", "B"), ("C", "D"), ("D", "A"), ("A", "C")]}

*Practica1p3> maxdegree G

3
```

Función **path**
Recibe dos nodos y una gráfica. Simplemente regresa True en caso de que exista un camino entre ambos nodos.

```
*Practica1p3> let G = Graph {gId = 1, nodes = ["A", "B", "C", 'D'"], edges =
[("A", "B"), ("C", "D")]}

*Practica1p3> path "A" "C" G

False
```

Función **delete**
Recibe un nodo y una gráfica. Elimina el nodo de la gráfica.

```
*Practica1p3> let G = Graph {gId = 1, nodes = ["A", "B", "C", 'D'"], edges =
[("A", "B"), ("C", "D"), ("D", "A"), ("A", "C")]}

*Practica1p3> delete "A"  G

Graph {gId = 1, nodes = ["B", "C", 'D'"], edges = [("C", "D")]}
```

Función **simplify**
Elimina dobles negaciones de una expresión.

```
*Practica1p3> let p = (Neg (Neg "Q"))

*Practica1p3> simplify p

Q
```

Función **deleteImpl**
Mediante equivalencia de operadores, elimina implicaciones y equivalencias de una Preposición.

```
*Practica1p3> let p = (Impl (Var "P") (Var "Q"))

*Practica1p3> deleteImpl p

Disy ((Neg (Var "P")) (Var "Q"))
```

Función **demorgan**
Mediante equivalencia de las reglas de De Morgan, regresa una Proposición equivalente.

```
*Practica1p3> let p = (Neg (Conj (Var "P") (Var "Q")))

*Practica1p3> demorgan p

Disy ((Neg (Var "P")) (Neg (Var "Q")))
```

Función **inter**
Haciendo uso del estado dado, interpreta la preposición.

```
*Practica1p3> interp (Conj (Var "P") (Neg (Var "P"))) [("P", Falso)]

False
```

Función **fnc**
Devuelve la forma normal conjuntiva de una Preposición.

```
*Practica1p3> let p = (Conj  (Var "P") (Impl (Var "Q") (Var "R")))

*Practica1p3> fnc  p

Conj  (Var "P")  (Disy (Neg (Var "Q")) (Var "R"))
```

Función **esFNC**
Nos dice si una proposición esta en forma normal conjuntiva.

```
*Practica1p3> let p = Conj  (Var "P")  (Impl (Neg (Var "Q")) (Neg (Var "R")))

*Practica1p3> esFNC  p

False
```

Función **correcto**
Recibe un conjunto de proposiciones y una conclusión. La función *correcto* nos dice si el argumento es lógicamente correcto o no.

```
*Practica1p3> let p = correcto [(Syss (Var "P") (Var "Q")), (Var "P")] (Var "Q")

*Practica1p3> correcto p

True
```

ESPECIFICACIONES

- ✓ Respetar las firmas de las funciones.
- ✓ Todas las funciones deberán estar documentadas.
- ✓ La legibilidad y documentación tendrá un impacto sobre la calificación de la práctica.
- ✓ La práctica se podrá realizar en equipos de a lo más 3 personas.
- × Cualquier copia entre equipos será evaluado a 0.
- × Cualquier copia de internet, sin entender el código será 0 sobre la función.
- × Cualquier práctica entregada posterior a la fecha límite no será tomada en cuenta.

Se deberá contar con un directorio cuyo nombre sea Practica1. Dentro del directorio se debe tener:

- README.txt, donde se incluya número de cuenta y comentarios sobre la práctica.
- Practica1p3.hs, script requerido para ésta práctica.

Comprimir el directorio con el formato ApellidoNombreP1p3. Comprimir con extensión .tar.gz o .zip

Si la práctica se realiza en equipo, agregar el nombre y número de cuenta de los integrantes en el README.txt

Enviar la práctica al correo ciclomax9@ciencias.unam.mx con el asunto [LC-Apellido-Nombre-P1p3].

Suerte

"Hardware: las partes de un ordenador que pueden ser pateadas"

-- Jeff Pesis