

Práctica 01
Lógica Computacional
Universidad Nacional Autónoma de México

Dr. Miguel Carrillo Barajas
Estefanía Prieto Larios
Mauricio Esquivel Reyes

1. Estructuras

1.1. Naturales

Consideremos la siguiente representación de los números naturales

```
data Natural = Cero | Suc Natural deriving (Eq, Show)
```

1.1.1. `mayorQue :: Natural -> Natural -> Bool`

Dados dos naturales nos dice si el primero es mayor que el segundo.
Ejemplos:

- `Main>mayorQue Cero (Suc Cero)`
`False`
- `Main>mayorQue (Suc Cero) Cero`
`True`

1.1.2. `menorQue :: Natural -> Natural -> Bool`

Dados dos naturales nos dice si el primero es menor que el segundo.
Ejemplos:

- `Main>menorQue Cero (Suc Cero)`
`True`
- `Main>menorQue (Suc Cero) Cero`
`False`

1.1.3. `igual :: Natural -> Natural -> Bool`

Dados dos naturales nos dice si son iguales.

Ejemplos:

- `Main>igual Cero (Suc Cero)`
`False`
- `Main>igual (Suc Cero) (Suc Cero)`
`True`

1.2. Lista de naturales

Consideremos la siguiente definición de las listas de naturales.

```
data ListaDeNaturales = Nil | Cons Natural ListaDeNaturales
```

1.2.1. `concat :: ListaDeNaturales -> ListaDeNaturales -> ListaDeNaturales`

Dadas dos listas de naturales regresar la concatenación de ambas.

Ejemplos:

- `Main>concat (Cons (Suc Cero) Nil) (Cons Cero (Cons (Suc (Suc Cero)) Nil))`
`Cons (Suc Cero) (Cons Cero (Cons (Suc (Suc Cero)) Nil))`
- `Main>concat (Cons Cero (Cons (Suc (Suc Cero)) Nil)) (Cons (Suc Cero) Nil)`
`Cons Cero (Cons (Suc (Suc Cero)) (Cons (Suc Cero) Nil))`

1.2.2. `reversa :: ListaDeNaturales -> ListaDeNaturales`

Dada una lista regresar la reversa de dicha lista.

Ejemplos:

- `Main>reversa (Cons Cero (Cons (Suc (Suc Cero)) (Cons (Suc Cero) Nil)))`
`Cons (Suc Cero) (Cons (Suc (Suc Cero)) (Cons Cero Nil))`

2. Lógica Proposicional

Consideremos la siguiente representación de la lógica proposicional.

```
-- Tipo de dato indice
type Indice = Int

-- Tipo de dato fórmula
data PL = Top | Bot | Var Indice
        | Oneg PL
        | Oand PL PL | Oor PL PL
        | Oimp PL PL deriving (Eq, Show)
```

2.1. Conjunciones

2.1.1. conj :: PL ->[PL]

Dada una fórmula regresar una lista con las conjunciones de dicha fórmula.

Ejemplo:

```
■ Main>conj Oor (Oand (Var 1) Oneg $ Var 2) (Oand Bot (Var 3))
Oand (Var 1) Oneg $ Var 2, Oand Bot (Var 3)
```

2.1.2. numConj :: PL ->Int

Dada una fórmula regresar el número de conjunciones que tiene dicha fórmula.

Ejemplo:

```
■ Main>conj Oor (Oand (Var 1) Oneg $ Var 2) (Oand Bot (Var 3))
2
```

2.2. Semántica

Consideremos los tipos de datos Valuación y Modelo de la forma

```
type Valuacion = Indice -> Bool
type Modelo = [Indice]
```

2.2.1. satMod :: Modelo ->PL ->Bool

Dada una fórmula proposicional, evaluar la fórmula de acuerdo a un modelo.

Ejemplos:

- Main>satMod [1] (Oneg (Var 1))
False
- Main>satMod [] (Oimp (Var 1) (Var 2))
True

2.2.2. **satPL :: Valuacion -> PL -> Bool**

Dada una fórmula proposicional, evaluar la fórmula de acuerdo a una valuación.

(Nota: en clase realizamos la función modeloToValuacion)

2.3. Formas normales

2.3.1. **esClausula :: PL -> Bool**

Dada una fórmula nos indica si es una clausula.

2.3.2. **esCNF :: PL -> Bool**

Dada una fórmula nos indica si esta en forma normal de conjunción.

2.3.3. **esTermino :: PL -> Bool**

Dada una fórmula nos indica si es un término.

2.3.4. **esDNF :: PL -> Bool**

Dada una fórmula nos indica si esta en forma normal de disyunción.

2.3.5. **toCNF :: PL -> PL**

Dada una fórmula en NNF y sin implicaciones, dar su CNF, tal que sean logicamente equivalentes.