



# Lógica Computacional 2017-2

## Práctica 2: LP en Haskell

Lourdes del Carmen González Huesca

Roberto Monroy Argumedo

Fernando A. Galicia Mendoza

Facultad de ciencias, UNAM

Fecha de entrega: Martes, 28 de febrero del 2017

Esta práctica puede ser entregada en equipo de a lo mas dos personas.

Utilizando los módulos `LProp`, `LPS` y `LConj` hechos en el laboratorio y realiza lo indicado en las siguientes secciones.

## 1. Clases para `LProp`

Modifica el archivo `LProp` según indican las siguientes instrucciones:

- Crea una instancia de la clase `Show` para que la impresión sea la siguiente:
  - `TTrue`: Se muestre como `T`.
  - `FFalse`: Se muestre como `F`.
  - `V x`: Se muestre como `x'`, donde `x'` es la cadena representante del número entero `x`.
  - `Neg p`: Se muestre como `~ p`, donde `p` es la cadena representante de la fórmula `p`.
  - `Conj p q`: Se muestre como `p ^ q`, donde `p,q` son las cadenas representantes de las fórmula `p` y `q`.
  - `Disy p q`: Se muestre como `p v q`, donde `p,q` son las cadenas representantes de las fórmula `p` y `q`.
  - `Imp p q`: Se muestre como `p -> q`, donde `p,q` son las cadenas representantes de las fórmula `p` y `q`.

- **Equiv**  $p \ q$ : Se muestre como  $p \leftrightarrow q$ , donde  $p, q$  son las cadenas representantes de las fórmula  $p$  y  $q$ .
- Crea una instancia de la clase **Eq** donde dadas  $\varphi, \psi$  dos elementos de tipo **Prop**,  $\varphi = \psi$  syss son sintácticamente iguales.
- Crea una instancia de la clase **Ord** donde las relaciones de orden van a estar definidas de acuerdo al peso de cada fórmula.

## 2. Semántica sobre conjuntos de fórmulas

Amplia el módulo **LPS** realizando los siguientes ejercicios:

1. Define una función que dado un conjunto de fórmulas, devuelva el conjunto de todos los posibles estados del conjunto de fórmulas. Nómbrala **estadosConj**.
2. Define una función que dado un conjunto de fórmulas, devuelva el conjunto de todos los posibles modelos del conjunto de fórmulas. Nómbrala **modelosConj**.
3. Define una función que dada una interpretación y un conjunto de fórmulas, indique si el conjunto es satisfacible en el estado dado. Nómbrala **satisfenConj**.
4. Define una función que dado un conjunto de fórmulas, indique si el conjunto es satisfacible. Nómbrala **satisfConj**.
5. Define una función que dada una interpretación y un conjunto de fórmulas, indique si el conjunto es insatisfacible en el estado dado. Nómbrala **insatisfenConj**.
6. Define una función que dado un conjunto de fórmulas, indique si el conjunto es insatisfacible. Nómbrala **insatisfConj**.

## 3. Relaciones binarias

Crea un archivo llamado **Rel**, ingresa las siguientes expresiones a tu archivo:

```
-- | U. Lista que representa un conjunto.
type U a = [a]

-- | R. Lista que representa una relacion binaria.
type R a = [(a,a)]
```

Y realiza los siguientes ejercicios:

- Considera un conjunto  $A$  y una relación  $R$  tal que  $R \subseteq A \times A$ , define las siguientes funciones:
  - Que devuelva la inversa de  $R$ , nómbrala **invR**.
  - Que construya la relación binaria total dado  $A$ , nómbrala **total**.
  - Que determine si  $R$  es reflexiva, nómbrala **refl**.
  - Que determine si  $R$  es simétrica, nómbrala **symm**.

## 4. La forma normal conjuntiva

Crea un archivo llamado **FNC** y realiza los siguientes ejercicios:

1. Define una función que reciba una fórmula de la lógica proposicional y devuelva el número de negaciones aplicadas a fórmulas distintas de variables proposicionales y constantes booleanas, nómbrala **nN**.
2. Define una función que reciba una fórmula de la lógica proposicional y devuelva una fórmula equivalente tal que este en forma normal negativa, nómbrala **fnn**.
3. Define una función que reciba una fórmula de la lógica proposicional y devuelva una fórmula equivalente tal que este en forma normal conjuntiva, nómbrala **cnf**.

## 5. Teoría

Los siguientes ejercicios pueden ser entregadas en el archivo **README** o bien impreso al inicio de la ayudantía el día de entrega de esta práctica.<sup>1</sup>

1. Sean  $\varphi$  un elemento de tipo **Prop**, demuestra que si  $\varphi$  cumple con la definición de ser una fórmula en forma normal negativa, entonces  $\text{nN } \varphi = 0$ .
2. Sean  $\varphi$  un elemento de tipo **Prop**, demuestra que  $\text{fnn } \varphi$  cumple con la definición de ser una fórmula en forma normal negativa.
3. Sea  $\varphi$  un elemento de tipo **Prop**, demuestra que  $\text{nN } (\text{fnn } \varphi) = 0$ .

### Reglas:

- Todo archivo debe seguir los estándares establecidos en el laboratorio.
- Todas las funciones deben ser recursivas.
- Para esta práctica no es necesario importar bibliotecas brindadas por el lenguaje.

*Si así fue, así pudo ser; si así fuera, así podría ser; pero como no es, no es. Es cuestión de lógica.*

-Charles Lutwidge Dodgson (Lewis Carroll)

---

<sup>1</sup>En caso de ser impresa, se sugiere utilizar hojas de rehuso.