<html><head></head><body><pre style="word-wrap: break-word; white-space: pre-wrap;">
```
-- LMF
(3Âº del Grado en MatemÃ¡ticas)
-- 2Âº examen de evaluaciÃ³n continua (30 de abril de 2014)
-- ----------------------------------------------------------------------

-- Nombre:
-- Apellidos:

import SintaxisSemantica
import FormasNormales
import Data.List

-- ----------------------------------------------------------------------
-- Ejercicio 1: Definir la funciÃ³n
--    modelosFormulaFN :: Prop -&gt; [InterpretaciÃ³n]
-- tal que (modelosFormulaFN f) es la lista con los modelos de la
-- fÃ³rmula f, calculados usando formas normales. Por ejemplo,
--    modelosFormulaFN ((p --&gt; q) /\ (no(q --&gt; p)))
--    ==&gt; [[no p,q],[q,no p]]
--    modelosFormulaFN ((p --&gt; q) /\ (no q --&gt; no p))
--    ==&gt; [[no p,q],[no p],[q],[q,no p]]
-- ----------------------------------------------------------------------

modelosFormulaFN :: Prop -&gt; [InterpretaciÃ³n]
modelosFormulaFN f =
    filter (not . tieneParComplementario) (listaM (formaNormalDisyuntiva f))
      where tieneParComplementario xs = or [(no p) `elem` xs | p &lt;- xs]

-- O bien

modelosFormulaFN' :: Prop -&gt; [InterpretaciÃ³n]
modelosFormulaFN' f =
  [xs | xs &lt;- listaM (formaNormalDisyuntiva f), not (tieneParComplementario xs)]
    where tieneParComplementario xs = or [(no p) `elem` xs | p &lt;- xs]


listaM :: Prop -&gt; [[Prop]]
listaM (Disj f g) = (listaM f) `union`  (listaM g)
listaM f = [conjALista f]

conjAListа :: Prop -&gt; [Prop]
conjALista f | literal f     = [f]
conjAlista (Conj f g) = (conjAlista f) `union` (conjAlista g)
```

</pre></body></html>