Profesor: Víctor Zamora Gutíerrez
Ayudantes: Edgar Quiroz Castañeda
Sara Doris Montes Incín

Instrucciones

Completen las funciones faltantes de los módulos `Logic.Tableaux` y `Logic.Normal`.

1 Tableaux

1 punto `classify :: Formula a -> Class a`

Dada una fórmula, determinar su clasificación para la creación de un tableau. Como el tipo `Formula` no puede ser restringido, se deben considerar los casos para toda fórmula. Las posibles clasificaciones son

- `Alpha`: para α-fórmulas.
- `Beta`: para β-fórmulas.
- `Lit`: para literales.
- `Na`: para constantes.

Una consideración extra es la fórmula $\neg\neg p$, que es la única expresión con una subfórmula en lugar de dos. Para homogeneizar esto, se puede definir su segunda subfórmula como T, pues esto no cambia su satisfacibilidad.

Hint: además de los casos base, basta con copiar la tabla de clasificación de fórmulas que vieron en clase.

3 puntos `makeTableau :: Formula a -> Tableau a`

Construir el tableau para una fórmula, siguiendo el algoritmo visto en clase. Pero hay algunas consideraciones extra para las clasificaciones posibles

- `Alpha, Beta`: se tratan como en el algoritmo original.
- `Lit`: sólo se añaden a la lista de literales.
- `Na`:
  - `F`: cierra automáticamente el tableau.
  - `T`: se ignora.

Hint: se puede definir la función como

```
makeTableau p = expand . Tableau $ [p] [] []
```

y definir `expand` para que expanda un tableau usando las reglas del algoritmo.

1 punto `isSatisfiable :: Eq a => [Literal a] -> Bool`

Una lista de literales es satisfacible si no contiene literales complementarias.

Hint: pueden usar la función `areComplement` para verificar si dos literales son complementarias y la función `all` para verificar que todos los elementos de una lista cumplan una propiedad.

1 punto `isOpen :: Eq a => Tableau a -> Bool`

Un tableau está abierto si alguna de sus hojas está abierta. Una hoja está abierta si su lista de literale es satisfacible. Adicinalmente, está cerrado si alguna de sus fórmulas es `F`.

¿Realmente necesitas imprimir esta hoja?

Profesor: Víctor Zamora Gutíerrez
Ayudantes: Edgar Quiroz Castañeda
Sara Doris Montes Incín

2 Formas normales

2 puntos `toNnf :: Formula a -> Formula a`: pasar una fórmula a forma normal negativa.

Hint: hay que tratar por separado las combinaciones posibles de implicaciones/equivalencias y sus negaciones.

1 punto `toCnf :: Formula a -> Formula a`: pasar fórmula a forma normal conjuntiva.

Hint: basta con pasar la fórmula a forma normal negativa y distribuir las disyunciones.

1 punto `toDnf :: Formula a -> Formula a`: pasar una fórmula a forma normal disjuntiva.

Hint: basta con pasar la fórmula a forma normal negativa y distribuir las conjunciones.

1 punto `toClausulal :: Formula a -> ClausulalForm a`: pasar fórmula a forma clausular.

Hint: basta con pasar la fórmula a forma normal conjuntiva, e implementar la función solo para $\wedge$, $\vee$ y literales.

¿Realmente necesitas imprimir esta hoja?