Método de Tableaux en LP

Laboratorio 3
Lógica para Computación
2024

El objetivo de este laboratorio es implementar en Haskell una variedad de funciones para fórmulas y razonamientos de Lógica Proposicional (LP) que tendrán como base principal la decisión de satisfacibilidad basada en el método de Tableaux.

Una consecuencia lógica en el lenguaje de LP, cuyas fórmulas son representadas por el tipo L, será representada aquí mediante el tipo

$$\mathsf{data\ Consecuencia} \ = \ [\mathsf{L}] \ :|= \ \mathsf{L}$$

donde el primer parámetro es una lista de premisas y el segundo es la conclusión.

Ejercicios

Implementar las siguientes funciones:

1. **esConsistente :: [(Var, Bool)] → Bool**
   Determina si una lista de asignaciones de valores de verdad sobre variables es *consistente*. Diremos que es *consistente* si no tiene pares complementarios de asignaciones, o sea no le asigna distintos valores a la misma variable [1].

   Ejemplos: $\mathsf{esConsistente}\ [(\mathsf{p}, \mathsf{False}), (\mathsf{q}, \mathsf{True}), (\mathsf{p}, \mathsf{True})] \ = \ \mathsf{False}$
   $\mathsf{esConsistente}\ [(\mathsf{p}, \mathsf{False}), (\mathsf{q}, \mathsf{True}), (\mathsf{r}, \mathsf{True})] \ = \ \mathsf{True}$

2. **int2f :: I → L**
   Convierte una interpretación dada como lista (no vacía) de asignaciones a una fórmula, específicamente una conjunción de literales.

   Ejemplos: $\mathsf{int2f}\ [(\mathsf{p}, \mathsf{False}), (\mathsf{q}, \mathsf{True}), (\mathsf{r}, \mathsf{True})] \ = \ \neg \mathsf{p} \wedge \mathsf{q} \wedge \mathsf{r}$
   $\mathsf{int2f}\ [(\mathsf{p}, \mathsf{True})] \ = \ \mathsf{p}$

3. **tableau :: L → Tableau**
   Aplica el método de Tableaux sobre la fórmula dada, devolviendo el árbol correspondiente.

   Ejemplo: $\mathsf{tableau}\ ((\mathsf{p} \wedge \neg \mathsf{q}) \vee \mathsf{q})$ se imprime en consola con el formato

```
[((p ∧ ¬q) ∨ q)]
+-  [(p ∧ ¬q]
|    '-  [p, ¬q]
 '-  [q]
```

[1] Cuando esto se cumple, dicha lista representa una interpretación. Recordar que una interpretación es una función (aquí dada por extensión), y como tal no debe asignarle a una misma variable más de un valor de verdad.

4. **sat :: L → Bool**
   Decide si una fórmula de LP es satisfacible, aplicando el método de Tableaux.

   Ejemplos: $\mathsf{sat}\ ((\mathsf{p} \wedge \neg \mathsf{q}) \vee \mathsf{q}) \ = \ \mathsf{True}$
   $\mathsf{sat}\ ((\mathsf{p} \wedge \neg \mathsf{p}) \wedge \mathsf{q}) \ = \ \mathsf{False}$

5. **modelos :: L → [I]**
   Devuelve todos los modelos de una fórmula, utilizando el método de Tableaux. Notar que a las hojas del árbol de Tableaux le pueden faltar letras de la fórmula original, por lo que habrá que utilizar funciones auxiliares con las que *completar* una interpretación con respecto a una fórmula.

   Ejemplos: $\mathsf{modelos}\ ((\mathsf{p} \wedge \neg \mathsf{q}) \vee \mathsf{q})) \ = \ [[(\mathsf{p}, \mathsf{False}), (\mathsf{q}, \mathsf{True})], [(\mathsf{p}, \mathsf{True}), (\mathsf{q}, \mathsf{False})], [(\mathsf{p}, \mathsf{True}), (\mathsf{q}, \mathsf{True})]]$
   $\mathsf{modelos}\ ((\mathsf{p} \wedge \neg \mathsf{p}) \wedge \mathsf{q}) \ = \ []$

6. **clasificar :: L → Clase**
   Determina a qué clase pertence una fórmula dada.

   Ejemplos: $\mathsf{clasificar}\ (((\mathsf{p} \supset \mathsf{q}) \supset \mathsf{p}) \supset \mathsf{p}) \ = \ \mathsf{Tau}$
   $\mathsf{clasificar}\ ((\mathsf{p} \wedge \neg \mathsf{q}) \vee \mathsf{q}) \ = \ \mathsf{Conti}$

7. **cons2f :: Consecuencia → L**
   Convierte una consecuencia, de forma $\mathsf{ps} : | = \mathsf{c}$, donde ps es la lista de premisas y c es la conclusión, a una fórmula de LP siguiendo la ley de consecuencia lógica: $\Gamma \vDash \alpha$ ssi $\models \bigwedge \Gamma \supset \alpha$. [2]

   Ejemplos: $\mathsf{cons2f}\ ([\mathsf{p} \supset \mathsf{q},\ \mathsf{p}]\ :| =\ \mathsf{q}) \ = \ ((\mathsf{p} \supset \mathsf{q}) \wedge \mathsf{p}) \supset \mathsf{q}$
   $\mathsf{cons2f}\ ([]\ :| =\ \mathsf{p} \wedge \neg \mathsf{q}) \ = \ \mathsf{p} \wedge \neg \mathsf{q}$

8. **valida :: Consecuencia → Bool**
   Decide si una consecuencia lógica es válida.

   Ejemplos: $\mathsf{valida}\ ([\mathsf{p} \supset \mathsf{q},\ \mathsf{p}]\ :| =\ \mathsf{q}) \ = \ \mathsf{True}$
   $\mathsf{valida}\ ([\mathsf{p} \supset \mathsf{q},\ \mathsf{q}]\ :| =\ \mathsf{p}) \ = \ \mathsf{False}$

9. **fnd :: L → L**
   Convierte una fórmula a FND. Para tal fin, fnd puede usar funciones previas que usted considere convenientes. Cómo la FND no es única, el resultado puede ser cualquiera de ellas.

   Ejemplos: $\mathsf{fnd}\ (\mathsf{p} \supset \mathsf{q}) \ = \ (\mathsf{p} \wedge \mathsf{q}) \vee (\neg \mathsf{p} \wedge \mathsf{q}) \vee (\neg \mathsf{p} \wedge \neg \mathsf{q})$
   $\mathsf{fnd}\ \neg(\mathsf{p} \supset \mathsf{q}) \ = \ \mathsf{p} \wedge \neg \mathsf{q}$

10. **fnc :: L → L**
    Convierte una fórmula a FNC. Análoga a la anterior.

    Ejemplos: $\mathsf{fnc}\ (\mathsf{p} \supset \mathsf{q}) \ = \ \mathsf{p} \vee \neg \mathsf{q}$
    $\mathsf{fnc}\ \neg(\mathsf{p} \supset \mathsf{q}) \ = \ (\mathsf{p} \vee \mathsf{q}) \wedge (\neg \mathsf{p} \vee \mathsf{q}) \wedge (\neg \mathsf{p} \vee \neg \mathsf{q})$

[2]Donde $\bigwedge \Gamma$ abrevia la conjunción de las premisas $\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n$.