

Tarea 1

Emmanuel Peto Gutiérrez

25 de febrero de 2020

El programa está hecho en *Haskell*, así que para compilar y ejecutar se teclea lo siguiente en la carpeta src:

- `ghc Saturacion.hs`
- `./Saturacion`

1. Forma normal conjuntiva

Para pasar cualquier fórmula a forma normal conjuntiva primero se eliminaron equivalencias (\leftrightarrow) e implicaciones (\rightarrow). Luego se colocaron las negaciones solamente en atómicas para pasar a forma normal negativa. Finalmente de FNN se transformó a FNC usando reglas de distribución.

2. Conjunto de cláusulas

Para transformar una fórmula en FNC a su forma de conjuntos, primero se hizo una función que dada una cláusula devuelve una lista de sus literales. Después se usó esta función para hacer una lista de cláusulas.

3. Saturación

Se realizó una función que encuentra literales complementarias entre dos cláusulas; si no tienen literales complementarias devuelve **Nothing**. La función *resBin* realiza resolución binaria de dos cláusulas dada una literal **l**. Después se hizo una función que hace resolución binaria de todas las cláusulas en un conjunto y la une con el conjunto original. Finalmente se aplica el algoritmo de saturación de forma recursiva y este se detiene si el conjunto de cláusulas no cambia o aparece el conjunto vacío en algún punto.

4. Saturación con simplificación

Para simplificar un conjunto de cláusulas se realizan los siguientes pasos:

- Se eliminan las literales repetidas de cada cláusula por la propiedad de idempotencia.
- Se eliminan las cláusulas que son tautología. Es decir, las que tienen literales complementarias o que tienen **T**.
- Se eliminan las cláusulas que son subsumidas por otras. Para esto se creó una función que determina si una cláusula C es superconjunto de algún elemento en un conjunto de cláusulas Cl ; si lo es se elimina C .

Una vez que se tiene la función de simplificación, se concatena la simplificación con la resolución y se aplica el algoritmo de saturación.