

# Tarea 3

## Razonamiento automatizado

Emmanuel Peto Gutiérrez

7 de abril de 2020

Para ejecutar el programa se debe ir a la carpeta `src/Problema` (donde Problema puede ser: DPLL, Refutacion o Saturation) y con el comando `make` se compila y ejecuta cada programa. Después de haber compilado se puede ejecutar otra vez con `./<nombre_del_programa>`.

En un archivo *input* se encuentra la entrada a los programas. La entrada debe tener el siguiente formato:

- Las variables deben estar formadas por un solo caracter alfabético.
- Colocar paréntesis a cada subfórmula:  $(A < operador > B)$ . Por ejemplo, la fórmula  $p \vee q \vee r$  no es reconocida por el parser, debe ser  $p \vee (q \vee r)$  o  $(p \vee q) \vee r$ .
- Utilizar los caracteres unicode para operadores:  $\wedge$ ,  $\vee$ ,  $\rightarrow$  y  $\Leftrightarrow$ .
- Si se va a usar un conjunto de fórmulas debe empezar con `[` y terminar con `]`. Las fórmulas deben estar separadas por comas y se pueden utilizar espacios entre las comas.
- Separar cada entrada con un salto de línea.

### 1. Saturación

Se encuentra en la carpeta **Saturation** y el archivo principal es **Saturation.hs**. La resolución original de la tarea 1 se encuentra en la función *resOriginal*. La resolución unitaria se define en la función *resUnitaria* y se aplica solamente si alguna de las cláusulas es unitaria. La resolución positiva se define en la función *resPositiva* y se aplica solamente si todas las literales de alguna de las cláusulas son positivas. La función donde se aplica alguno de los tres tipos de resolución se llama *res3* y se decide con un número entero: 2 para resolución positiva, 3 para resolución unitaria y cualquier número diferente de 2 y 3 para resolución original.

## 2. Modelos con DPLL

Está en la carpeta **DPLL** y el archivo principal es **Davis.Putman.hs**.

### 2.1. Regla de cláusula unitaria

Dado un conjunto de cláusulas, primero se intenta de encontrar una cláusula unitaria con *findCU*. El resultado es **Maybe Clausula** porque un conjunto puede no tener cláusula unitaria, así que devolvería **Nothing**. Si el conjunto tiene cláusula unitaria se aplica propagación unitaria con una literal *l* en la función *reglaCU*.

### 2.2. Regla de literal pura

Se intenta encontrar una literal pura con *findLP*. Si el resultado es **Just lp** se aplica la regla de literal pura con la función *reglaLP*.

### 2.3. Regla de descomposición

Si un conjunto de cláusulas no tiene cláusula unitaria ni literal pura, se aplica la regla de descomposición. Primero se obtiene la literal más frecuente en el conjunto de cláusulas con la función *literalFrecuente*, la cual es una heurística de selección para hacer el algoritmo DPLL más eficiente. Después de obtener la literal más frecuente se aplica la función *reglaD*.

### 2.4. Modelos

Un estado (una asignación de valores a las variables) está determinado por un conjunto de variables. Si *est* es el estado y *F* la fórmula, el valor de cada  $x \in F$  está definido así:  $\mathcal{I}(x) = 1$  si  $x \in est$  y  $\mathcal{I}(x) = 0$  si  $x \notin est$ .

La búsqueda de los modelos del algoritmo DPLL se encuentra con la función *dpllCola*. En cada llamada recursiva se carga el argumento *cola*, que es una cola de literales y van a determinar el modelo para el conjunto de cláusulas que recibe la función. Cada vez que se aplica una regla se agrega una literal a la cola. Si se llega a un conjunto vacío entonces la cola será un modelo y será el resultado de la función. Si se llega a un conjunto que contiene a la cláusula vacía significa que la cola no es modelo y se devuelve una lista vacía.

Una vez que se tiene una lista de las literales que son parte del modelo se eliminan las que son negativas; esto es porque un modelo solo tiene literales positivas (variables).

## 3. Árbol DPLL

Para generar una refutación en forma de árbol se crea un tipo de dato: **ArbolDPLL**. Esta vez, en vez de devolver un conjunto de modelos, el algoritmo devuelve un árbol DPLL. En cada nodo se guarda la siguiente información: el

tipo de regla a aplicar, el conjunto de cláusulas al que se le aplica la regla y la literal usada en la regla. También se tiene un hijo si se aplica RCU o RLP, y dos hijos si se aplica la regla RD.

Las hojas del árbol pueden ser de dos tipos:  $\checkmark$  si se llega a un conjunto vacío o  $\times$  si se llega a un conjunto con una cláusula vacía. Si la fórmula es insatisfacible, el árbol DPLL debe tener  $\times$  en todas sus hojas.