

COMPLEJIDAD COMPUTACIONAL

EL PROBLEMA 3-SATISFIABILITY

3SAT

Alberto Delgado Soler
Germán Alfonso Teixidó
Tomás González Martín
Pier Paolo Tarasco

6 de noviembre de 2019

1. Introduction

1.1. Descripción del problema

La 3-satisfactibilidad es un caso especial de satisfactibilidad (SAT), en la que cada cláusula contiene exactamente 3 literales. El problema SAT es el problema de saber si, dada una expresión booleana con variables y sin cuantificadores, hay alguna asignación de valores para sus variables que hace que la expresión sea verdadera.

1.2. Paso de SAT a 3SAT

Siendo $U = \{u_1, u_2, \dots, u_n\}$ un conjunto de variables, y $C = \{c_1, c_2, \dots, c_m\}$ un conjunto de cláusulas, queremos construir un conjunto C' de cláusulas de tres literales en un nuevo conjunto U' , de tal forma que C' sea satisfactoria si y solo si C es satisfactoria. Para la construcción de C' vamos a contemplar distintos casos, los cuales se dividen en función del número de variables disponibles en U y el número de variables necesarias para completar las tripletas. Supóngase que C contiene k literales:

- Caso 1. Si $k = 1$ significa que $C_i = z_1$. Por lo tanto, se necesitan dos variables adicionales v_1 y v_2 de tal forma que se obtienen cuatro nuevas cláusulas de 3 literales.

$$C' = \{v_1, v_2, z_1\}, \{v_1, \overline{v_2}, z_1\}, \{\overline{v_1}, v_2, z_1\}, \{\overline{v_1}, \overline{v_2}, z_1\}$$

El motivo de esto es que tanto v_1 como v_2 pueden ser verdaderos, o no. Por ello, se deben presentar todos los casos (incluyendo los literales negados).

- Caso 2. Si $k = 2$ significa que $C_i = \{z_1, z_2\}$. Ya que contamos como dos literales, solo se necesita un literal adicional, v_1 . Por ello, los dos posibles valores que tome el literal adicional añadido.

$$C' = \{v_1, z_1, z_2\}, \{\overline{v_1}, z_1, z_2\}$$

- Caso 3. Si tenemos $k = 3$ el SAT ya cumple el formato 3SAT
- Caso 4. Si $k > 3$ se deben crear " $k-3$ " variables adicionales, incrementando el número de cláusulas en $k-2$. El conjunto C' estará compuesto por 3 partes. La primera parte es la primera tripleta del conjunto, formada por los dos primeros k literales, añadiendo la primera variable adicional.

$$\{z_1, z_2, v_1\}$$

Las siguientes tripletas (excepto la última) serán de la forma

$$\{\overline{v_i}, z_{i+2}, v_{i+1}\}$$

Por último, el conjunto se completa con

$$\{\overline{v_{k-3}}, z_{k-1}, z_k\}$$

Por lo tanto, para el Caso 4, donde $k > 3$, tenemos el problema 3SAT con la forma:

$$C' = \{z_1, z_2, v_1\} \cup \{\overline{v_i}, z_{i+2}, v_{i+1}\} \cup \{\overline{v_{k-3}}, z_{k-1}, z_k\}$$

1.3. Código desarrollado

El código, desarrollado en el lenguaje *C++*.

1.3.1. Estructura del programa

- */include*: ficheros **.h* que contienen las cabeceras de las clases utilizadas.
- */obj*: contiene el ejecutable del programa, además de las entradas y salidas, junto a los ficheros objeto (**.o**).
- */src*: ficheros con extensión **.cpp* que contienen la implementación de las clases, y el programa principal.

1.3.2. Descripción de ficheros principales

- *SAT*: Contiene la definición formal del problema SAT.
- *SAT3*: Siendo un caso particular del SAT, hereda de éste, implementando la concatenación de clausulas de los casos visto anteriormente.
- *SAT2SAT3*: *SAT to 3SAT* tiene como propósito convertir el problema del SAT en el problema del 3SAT utilizando la capacidad de concatenar *SAT3*
- *input*: fichero que presenta el problema de entrada en la forma normal conjuntiva (cnf) siguiendo el estandar, donde:
 - Las primeras líneas pueden tener una *c* seguida de un comentario en cada línea en la que quieras poner comentarios.
 - Después la letra *p* seguido de *cnf* y luego el número de variables y el número de cláusulas.
 - A continuación debe haber una línea por cada cláusula y, en cada cláusula, se ponen los números de las variable que se encuentran en la cláusula (sin nada o con un “-” delante para indicar que está negado) y al final de la cláusula el un “0” para indicar que se terminó la cláusula