

Problema 3-Satisfiability

3SAT

...

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

Descripción del problema

Se trata de un caso específico del problema SAT, donde se pretende tener cláusulas de 3 literales cada una. El problema del SAT trata de saber si, dada una expresión booleana con variables sin cuantificadores, hay alguna asignación de valores para sus variables que hace que la expresión sea verdadera.

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, se quiere construir un conjunto C' de cláusulas de 3 literales

La construcción de C' depende del número de variables necesarias para completar las tripletas.

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 esta forma, se obtienen cuatro nuevas cláusulas de 3 literales

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

Caso 2

Si $k = 2$, significa que $C_i = \{z_1, z_2\}$. Para tener 3 literales solo nos falta uno. Se necesita una nueva variable v_1 y nos quedaremos con dos cláusulas de 3 literales.

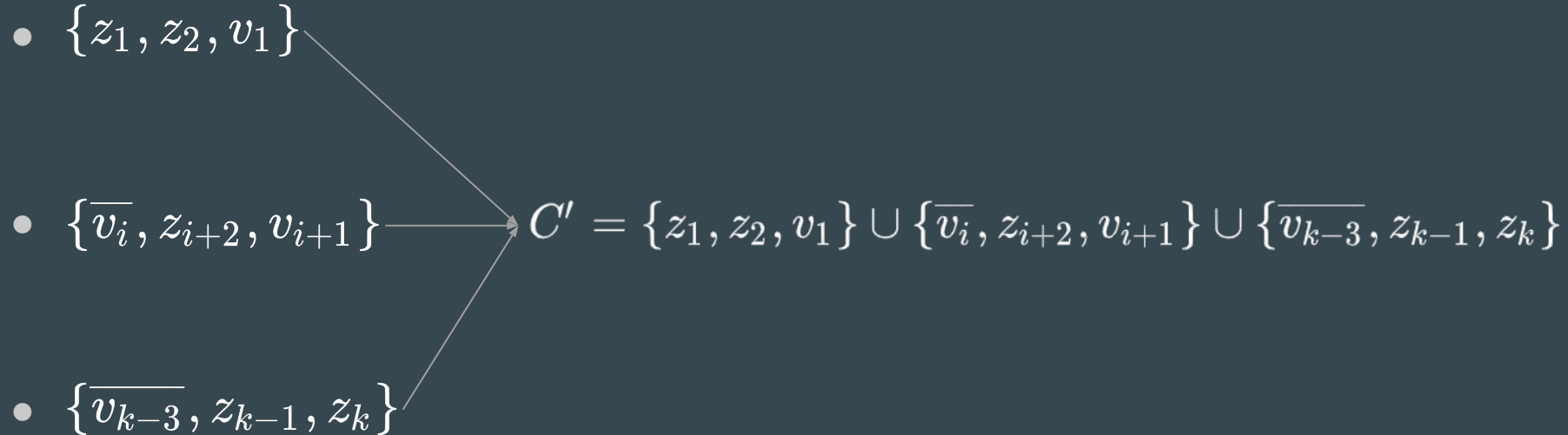
$$C' = \{v_1, z_1, z_2\}, \{\overline{v_1}, z_1, z_2\}$$

Caso 3

Si tenemos $k = 3$, la cláusula ya cumple el formato 3SAT, por lo que no es necesario realizar ningún procedimiento.

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 unidas entre sí.



Código desarrollado

El código está desarrollado en C++

Las cabeceras de las clases se encuentran en archivos **.h* dentro de la carpeta */include*.

En la carpeta */src* los archivos **.cpp* donde desarrollamos el código.

La carpeta */obj* se almacenan los archivos objeto con el código compilado y los archivos de entrada y salida del programa..

Código desarrollado

Descripción de los principales ficheros:

- “SAT” define un objeto con las cláusulas y sus literales en formato SAT.
- “SAT3” hereda de SAT pero solo permite que se agreguen cláusulas de 3 literales.
- “SAT2SAT3” realiza los pasos explicados anteriormente para que a partir de un objeto del tipo SAT se obtenga un objeto del tipo SAT3.
- “convertir” recibe el fichero con el problema SAT y lo almacena en un objeto del tipo SAT, luego el objeto se pasa por un objeto SAT2SAT3 para obtener un objeto del tipo SAT3 y se almacena con el formato cnf en un fichero diferente al recibido.

Formato del fichero de entrada

El fichero de entrada está en forma normal conjuntiva (cnf).

```
1  c frrrr
2  p cnf 4 4
3  -1 -4 0
4  1 -2 -3 0
5  -2 -3 4 1 0
6  1 0
```

Está guiado bajo el estándar *cnf*, 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 variables 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.

Bibliografía

- Computers and intractability : a guide to the theory of NP-completeness
- Estandar cnf: <https://people.sc.fsu.edu/~jburkardt/data/cnf/cnf.html>
- https://es.wikipedia.org/wiki/Problema_de_satisfacibilidad_booleana#3-Satisfactibilidad