Assignement 3

Iván Piña Arévalo
ivan.pinaarevalo@alum.uca.es

May 10, 2019

Abstract

En esta práctica, realizaremos la implementación de una reducción polinomica. Concretamente, reduciremos el problema del coloreado de un grafo al problema 3-SAT. Una vez implementado nuestro programa, realizaremos análisis sobre su complejidad termporal, así como las relaciones entre el número de colores necesarios y el máximo clique del grafo. Se ha empleado el grafo de Petersen.

1 Introduction

En primer, lugar, comenzaremos hablando del coloreado de un grafo. Este problema consiste en colorear un grafo de tal manera que dos vértices adyacentes no pueden tener el mismo color.

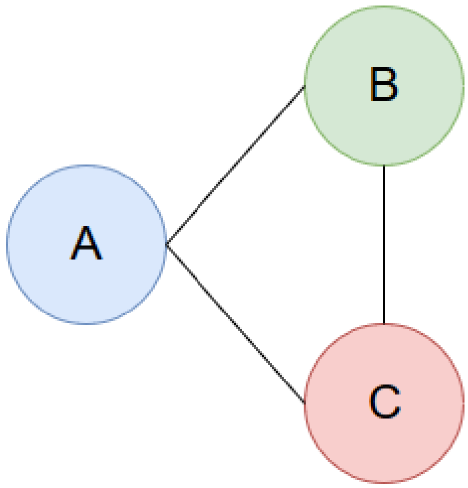

Figure 1: Grafo de ejemplo con una solución válida

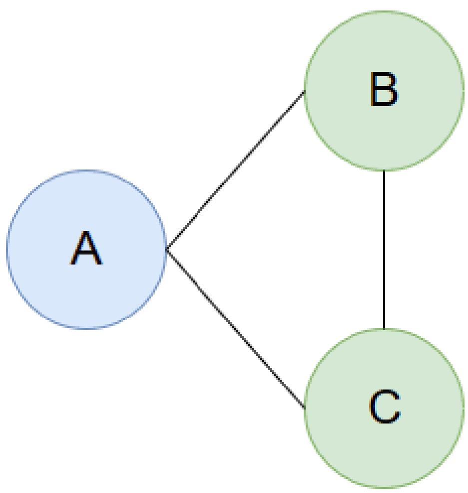

Figure 2: Grafo de ejemplo con una solución inválida

Concretamente, hemos empleado el grafo de Petersen. Hemos variado ligeramente la numeración de los nodos. Para mayor comodidad, el primer nodo lo hemos numerado como '1' en lugar de '0' y así sucesivamente (es decir, hemos incrementado en 1 el valor de cada nodo.)

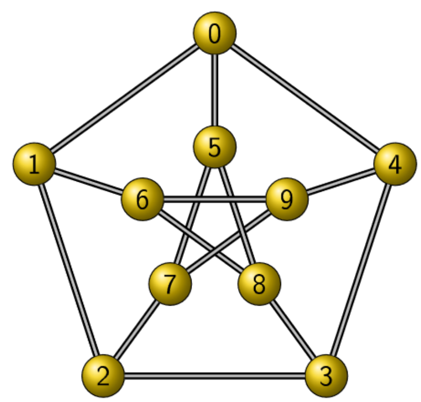

Figure 3: Grafo de Petersen

Para encontrar la solución de este problema se ha empleado el problema 3-SAT. El problema SAT fue uno de los primers en descubrirse NP-completo. Dicho problema consiste en dada una ecuación booleana, encontrar los valores para los cuales es verdadera. Además, la ecuación debe tener las siguientes características:

- La ecuación global se compone de ecuaciones enlazadas con la puerta logica and
- Cada subecuación esta compuesta por variables $(x_1, x_2, ...)$ unidas por la puerta lógica OR.

A continuación podemos ver una ecuación para el problema SAT:

$$(x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_4)$$

2 Methodology

La implementación de la práctica se ha realizado en C++, hemos dividido el programa en varias partes.

2.0.1 Lectura del grafo

Para la lectura del grafo nos hemos apoyado en un fichero. Dicho fichero debe tener la siguiente estructura:

- La definición de cada nodo se realiza en una fila.

- Los nodos a los que esta conectado el nodo que estamos definiendo estan separados por comas.

De esta manera realizamos la lectura del grafo, representado mediante listas de adyacencia. A continuación podemos ver el fichero con el grafo de ejemplo de la sección anterior.

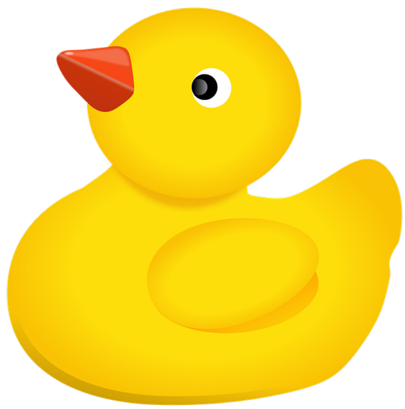

Figure 4: Fichero de entrada para el grafo de ejemplo

De igual manera, el grafo de Petersen representado mediante listas de adyacencia.

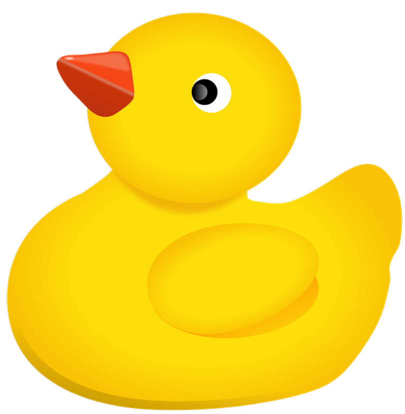

Figure 5: Fichero de entrada para el grafo de Petersen

2.0.2 Transformación del 3-Col to SAT

Lo primero al realizar esta parte ha sido analizar profundamente la estructura de las ecuaciones booleanas que representan al grafo que deseamos colorear. Estas ecuaciones se pueden dividir en dos grupos.

- Ecuaciones correspondientes al color del propio nodo.

- Ecuaciones correspondientes al color de los nodos adyacentes al nodo actual.

Si nos fijamos, en los grafos que hemos empleado las aristas son bidireccionales. Bastaría con definirlas una sola vez, sin embargo, para mantener la legibilidad del código he decicido permitir que se escriban dos veces (una en cada sentido). Este aumento en número de ecuaciones no afecta a PicoSAT. A continuación mostramos la salida para el grafo de Petersen

2.0.3 Escritura de las ecuaciones booleanas

Finalmente es necesario escribir en un fichero las ecuaciones correspondientes al grafo. Estas ecuaciones deben ir en formato DIMACS. El fin de este formato es el correcto procesamiento por parte del programa PicoSAT.

Simplemente en lugar de ir mostrando las ecuaciones por la salida estándar las hemos dirigido hacia una cadena. Igualmente hemos definido una variable que se encarga de contar cuantas ecuaciones tiene y otra más para el número de variables. Volcando las cadenas y los contadores en un fichero conseguimos tener la información lista para PicoSAT.

El programa ha sido paramétrizado de tal forma que tanto el grafo como el número de colores pueda modificarse.

2.0.4 PicoSAT

Finalmente, para ejecutar el programa hemos empleado la orden:

$$./picosatsalida$$

Si deseamos ver todas las soluciones

$$./picosat--allsalida$$

3 Results and Discussion

Para averiguar el tipo de reducción obtenida hemos realizado un estudio de la complejidad temporal del programa: Por lo tanto, nuestro programa posee una complejidad polinómica. Consecuentemente hemos obtenido una reducción polinómica. Empleando tres colores, obtenemos las ecuaciones: Una solución para el grafo de Petersen es esta:

Así mismo, todas las posibles soluciones son:

Si ahora reducimos el numero de colores a 2, obtenemos estas soluciones: Y como podemos observar, no existe ninguna solución.

Finalmente, analizando la estructura del grafo tenemos que el Clique Máximo del grafo de Petersen es de tamaño 2. Por ejemplo:

4 Conclussion

El Clique máximo del grafo es 2. Así mismo no podemos colorear el grafo con menos de tres colores. Esto, intuitivamente nos sugiere que

$$clique < numero_colores$$

Siendo clique el numero de nodos que forman el máximo clique y numero_colores el número de colores mínimo necesario para colorear el grafo. Ahora a fin de contrastar esta suposición, decidimos probar con el grafo de ejemplo que hemos mostrado al principio del report.

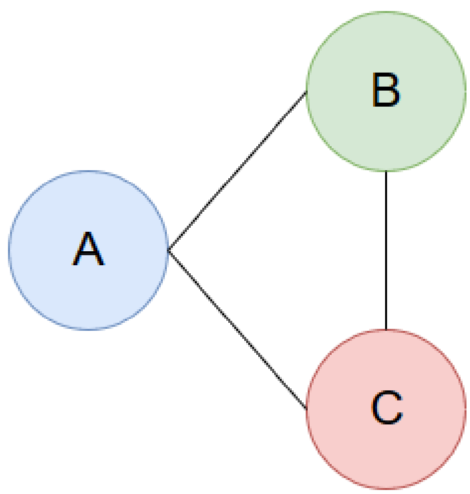

Figure 6: Grafo de ejemplo

Aquí, sin embargo, el clique máximo del grafo es 3 y se puede colorear correctamente con tres colores. Por tanto

$$clique \leq numero_colores$$