

Lab08: Mínimo vertex cover usando SAT

1. Reducción de vertex cover decisional a SAT

El problema de decidir si es posible encontrar un recubrimiento $R = \{v_1, \dots, v_k\}$ de todas las aristas de un grafo $G(V, E)$, no dirigido, de tamaño $k \geq 1$, se puede reducir al problema SAT.

Dado un grafo $G(V, E)$, la fórmula booleana $\varphi_{G(V, E), k}$, en forma normal conjuntiva, constará de n variables donde n es el número de nodos del grafo. Esto es, $\{x_1, x_2, \dots, x_n\}$. Cada variable x_i , con $(1 \leq i \leq n)$, evaluará a *true* si, y sólo si, el nodo $(i - 1)$ forma parte del recubrimiento. Esto es, $(i - 1) \in R$.

La fórmula $\varphi_{G(V, E), k}$ debe contener las siguientes cláusulas:

- Un conjunto de cláusulas que expresan que cada arista $(a, b) \in E$ del grafo debe tener al menos uno de sus dos nodos en R ($a \in R \vee b \in R$).
- Un conjunto de cláusulas para indicar que las variables que evalúan a *true* tienen que ser como mucho k variables. Esto es, $x_1 + x_2 + \dots + x_n \leq k$. Este tipo de restricciones están implementadas en PySAT y vamos a usar la función `CardEnc.atmost(list_variables, bound)` (<https://pysathq.github.io/docs/html/api/card.html>)

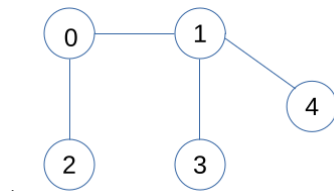


Figure 1: Grafo A

El conjunto de variables para el grafo A de la figura es $\{x_1, x_2, x_3, x_4, x_5\}$. La codificación del problema para un $k = 2$ será la fórmula $\varphi_{A, 2}$ que contiene las cláusulas $(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_4) \wedge (x_2 \vee x_5)$, además de las que devuelve la llamada a `CardEnc.atmost`.

2. Tareas a realizar

Debes implementar las siguientes funciones para conseguir devolver el mínimo recubrimiento de un grafo no dirigido.

- Función `reduce_VC_to_SAT` que, dado un grafo $G(V, E)$ y un $k \geq 1$, devuelva la lista de cláusulas que codifica la fórmula $\varphi_{G(V, E), k}$.
- Función `encuentra_k` que, dado un grafo, una cota inferior *low* y otra superior *up* para el tamaño del recubrimiento, devuelva el tamaño k del mínimo recubrimiento con $low \leq k \leq up$. La búsqueda de k deberá ser dicotómica y su implementación requerirá llamar a `reduce_VC_to_SAT` y preguntar al SAT-solver.
- Funciones `cover_2` y `greedy` vistas en clase.

3. Entrega

Junto al enunciado se adjunta un juego de pruebas con los resultados obtenidos por nosotros. Además de tu código, debes entregar un fichero pdf similar con el tiempo que tarda tu solución (no hace falta incluir los grafos).