Ejercicios NPC (trabajo individual)

Javier Montes Pérez

9 de mayo de 2024

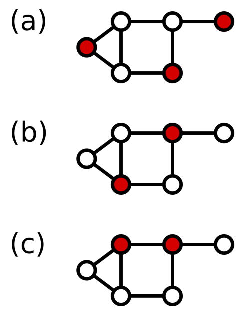
Show that DOMINATING-SET is NP-complete by giving a reduction from VERTEX-COVER

En los ejercicios propuesto se nos pide demostrar que ciertos lenguajes o problemas son NP-completos. Uno de los conceptos clave en el área de complejidad computacional es el de los problemas NP-completos. Suponiendo que demostramos que un problema es perteneciente a esta clase de complejidad, implica mostrar que es tan difícil como el problema más difícil en la clase NP, resumiéndolo de manera un poco más abstracta y sencilla al habla. En mi caso particular, nos centraremos en $DOMINGATING-SET$

$$\text{DOMINATING\_SET} = \{\langle G, k\rangle \mid G \text{ es un grafo y existe un Dominating Set en } G \text{ de tamaño } k\}$$

definiendose un Dominating Set como un subconjunto de vértices (nodos) de G tal que para todos los vértices v de G, o bien v es parte de dicho subconjunto G' (el Dominating Set) o es adyacente con al menos uno de los nodos de G'.

Estas definiciones formales que acabamos de estipular y formalizar, quizás se deben ver mejor con un ejemplo. Por ello se muestra aquí abajo un ejemplo visual y claro del tema que nos concierne.



Se puede apreciar claramente como para el grafo dado en el ejemplo, existen 3 subconjuntos de nodos que conformarían cada uno por separado Dominating Sets. Basta con fijarse en los nodos del grafo general que no están marcados como pertencientes al subconjunto de vértices, y ver que para cada uno de ellos existe una arista con al menos 1 de los nodos del subconjunto.

Una vez definido el problema sobre el cual vamos a estar trabajando, considero que podemos comenzar con la propia demostración en sí. Se nos pide demostrar que $DOMINATING - SET \in NP - completo$, es decir, que para todos los lenguajes $L \in NP$, existe una función f que es capaz de reducir dicho lenguaje a Dominating Set, o lo que es lo mismo, Dominating Set está por encima en grado de dificultad sobre todos los lenguajes/problemas de NP.

Ahora bien, para abordar la demostración existen varias estrategias a seguir, pero en mi caso seguiŕe la que se recomienda en la proposición del ejercicio en el campus virtual de la asignatura: demostrar en primer lugar que el problema/lenguaje en cuestión pertenece a NP, y tras ello proseguir con una reducción desde un lenguaje que ya sepamos que es NP-completo hacia el lenguaje que nos concierne a nosotros. En nuestro caso, se nos da como sugerencia realizar la reducción desde $VERTEX - COVER$, lenguaje el cuál explicaré y definiré más adelante.

(a) DOMINATING-SET $\in$ NP:

Como ya hemos demostrado y definido varias veces durante el trascurso de la asignatura, para abordar la demostración sobre si un determinado lenguaje/problema pertenece a NP, existen varias opciones para abordarlo. Una de ellas, no muy recomendada en la mayoría de casos por su mayor complejidad, es construir de 0 una máquina de Turing no determinista (mTnD) que sea capaz en tiempo polinómico de decidir o aceptar en lenguaje en cuestión. Por otro lado, existe la posibilidad también de abordar la demostración proporcionando un algoritmo verifiador. Los algoritimos verificadores, como su propio nombre indica, son algoritmos que reciben entradas que son pares, donde el primer elemento es una cadena con la forma de las cadenas pertenecientes al lenguaje para el cuál el verificador se está construyendo, no necesaríamente una cadena perteneciente al lenguaje. En segundo lugar, el segundo argumento se trata de un certificado que permite al verificador decidir en tiempo polinómico si la cadena de entrada (primer argumento) pertenece o no al lenguaje que concierne en dicho caso.

En nuestro caso, el primer argumento que llegará al verificidor será una codificación de un grafo G y un número k de nodos de la forma $< G, k >$, y un certificado como segundo argumento que será un posible Dominating-Set del grafo G, si es que tiene alguno. Es por ello, por lo que nuestro verificador en tiempo pólinomico es capaz de tomar el certificado (subconjunto de vértices de G) y mostrar si el grafo original contiene o no dicho subconjunto. La naturaleza polinómica del algoritmo construido es fácil de observar, ya que como se puede deducir el verificador V solo tendrá que tomar la codificación de G ,que normalmente se trata de un paréntesis con una enumeración de nodos, seguido de un ';', seguindo de un paréntesis grande con todas las aristas codificadas como pares de vértices de la siguiente forma:

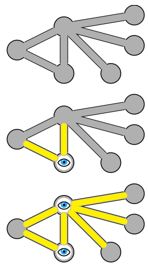
(	1	,	2	,	3	,	4	)	;	(	(	1	,	2	)	,	$\cdots$	)

y sobre ella recorrerla asegurándose, en primer lugar, de que todos los vértices del subconjunto G' pertenecen a G, y en segundo lugar, de que para todos los vertices restantes del grafo que no pertenecen al Dominating-Set existe una arista directa hacia algún vértice de G' (dependiendo de la propio codificación del grafo G quizás las aristas al ser no dirigidas solo se estipulan en la codificación una única vez, y por ello sería necesario buscar en todos los de vértices). Hablando en términos del número de vértices del grafo original n, suponiendo que la comprobación de nodos del subconjunto es polinómica $O(n)$ (ya que se debe recorrer el paréntesis con el listado de nodos totales) y de que el segundo paso puede tomar en el peor caso el siguiente tiempo: suponiendo que el dominating-set dado como certificado tenga solo 1 vértices, habría que tomar cada uno de los vértices restantes $(n-1)$, y para cada uno de ellos recorrer en el peor caso la lista entera de nodos, que para un grafo G con n aristas es como mucho de $\frac{n\cdot(n-1)}{2}$ aristas, obteniendose en este segundo paso una complejidad total $O(n^3)$, que sigue siendo polinómico.

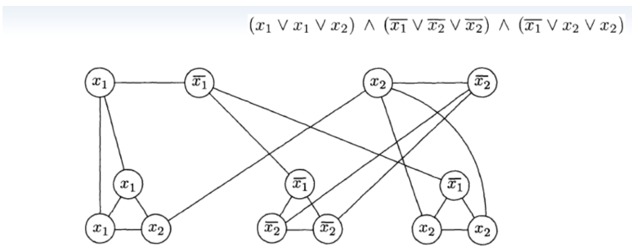
(b) Reducción desde VERTEX-COVER a DOMINATING-SET:

En el segundo paso de nuestra demostración, ya damos por supuesto que el lenguaje o problema en el cual nos estamos centrando se encuentra en la clase de complejidad NP, y eso ya está demostrado mediante verificador en el apartado anterior. Ahora bien, queremos conocer también si se trata de un lenguaje completo para dicha clase, por lo que hay que dar un paso más. Concretamente este paso extra consistirá en realizar una reducción desde un problema que ya conozcamos que se trate de $NP-completo$ hasta el nuestro, y de esta manera habríamos demostrado que Dominating-Set también se trata de un lenguaje completo.

Para ello, haremos uso de un lenguaje conocido en la asignatura denominado $VERTEX-COVER$, que realmente se trata de un problema de decisión, para el cual el *input* es un grafo G y un entero positivo k, y el output es realmente a la pregunta de: ¿tiene G una cobertura de vértices C, tal que $|C| \leq k$? Pero, ¿qué entendemos por cobertura de vértices? Se trata básicamente de lo siguiente: dado un grafo no dirigido cualquiera, una cobertura de vértices es un subconjunto de estos tales que cada arista del grafo original es incidente en al menos un vértice del subconjunto. Por ejemplo...

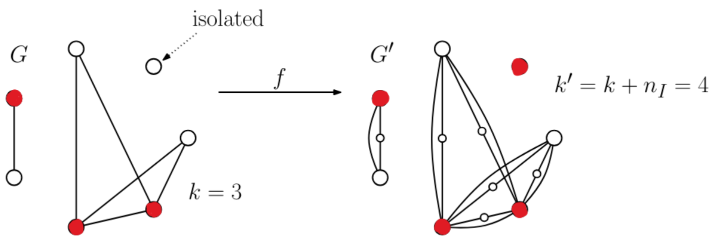


Como se puede ver en la imagen de arriba, para el grafo inicial donde todos sus vértices estaban coloreados de gris, hemos encontrado un subconjunto de nodos/vértices tales que todas las aristas del grafo inciden en alguno de ellos. La demostración de que $VERTEX-COVER$ es contenido de la asignatura, aunque su explicación de manera abreviada es: (*1) se trata de un problema NP ya que con un simple certificado que fuera el propio subconjunto de vértices que forman la cobertura podríamos probar en tiempo polinómico si un par $< G, k >$ tiene una cobertura de tamaño menor o igual que k. Y por otro lado, (*2) para demostrar que es completo en dicha clase de complejidad se construye una reducción entre $3SAT$ (fórmulas boolean en forma normal conjuntiva con hasta 3 literales por cláusula) y VCP (Vertex Cover Problem), de manera que $3SAT \leq_p VCP$. En la demostración de esta parte, observamos que una asignación de variables en f corresponde a dar valor verdadero o falso a cada una de sus variables. Podemos representar esto en G con una arista entre un nodo que represente a una variable y el nodo que represente a su negada. Lo hacemos para todas las variables. Un nodo de cada par conectado por una arista deberá estar obligatoriamente en la cobertura del grafo. Se añade al grafo tres nodos por cada cláusula que representen a cada uno de los literales de dcha cláusula. Todos ellos conectados entre sí. Finalmente, cada nodo de las ternas lo conectamos por medio de una arista con su respectivo «nodo variable». De esta forma, un ejemplo posible sería...



Una vez mostrado de manera genérica como se consigue obtener la conclusión de que $3SAT \leq_p VCP$, y por consiguiente de que VCP es NPC, podemos proseguir con nuestro ejercicio.

Para demostrar que $VCP \leq_p DominatingSet$, queremos concluir que dada una instancia del problema VC con la forma (G, k), podemos producir una instancia equivalente del problema DS en tiempo polinómico. Creamos un grafo G' de la siguiente manera. Inicialmente, $G' = G$. Para cada arista $\{u, v\}$ en G, vamos a crear un nuevo vértice adicional en G' de manera que además de añadirse, se incluyan también 2 aristas nuevas que unan este nuevo vértice x_{uv} con los vértices u y v. De esta manera, podríamos ver el proceso como "transformar todas las aristas de G en triángulos, donde la arista previa se mantenga y aparezca 2 nuevas. Si asumimos que el grafo G del cual partiamos contenía un vertex cover de tamaño k, para cada arista de dicho grafo que se componga de un par de vértices u, v tenemos por seguro que como mínimo uno de ellos forma parte del vertex cover, ya que por la propia definición de cobertura de vértices (propocionada anteriormente), se trata de un subconjunto de vértices tales que para cada arista del grafo del que se parte, incide en alguno de los vértices de este subconjunto. Por ello, si encontramos una arista donde ninguno de los extremos pertenece al VC entonces habríamos roto la restricción. Ahora con lo añadido como parte de la demostración, para cada nuevo vértice x_{uv} este será adyacente tanta a u como a v, uno de los cuales pertenecerá al VC por la explicación que acabamos de dar. Por ello, el conjunto de vértices que forma el VC en el grafo G, son los mismos que formarán en DominatingSet en G'. Sin embargo, no siempre será así. El caso en el que esto cambiará será si el grafo G del que partimos contiene vértices aislados. Denotaremos al conjunto de vértices aislados de G como I, por tanto en ese caso por la propia definición de DominatingSet, para cada vértice del grafo o bien pertenece al DS, o bien está conectado mediante una arista a alguno de ellos. Como son aislados, nos vemos en la obligación de tener que incluirlos en el conjunto DS. Por ello, sabemos con firmeza que: $k' = k + |I|$. Un ejemplo de aplicación de la función de reducción que acabamos de construir sería:



Como se puede apreciar, concretamente hemos elegido para ilustrar un ejemplo donde partimos de un grafo que contiene un vértice aislado. En el caso de VC, el vértices no suponía ningún problema ya que de él no salían aristas, pero para Dominating Set es necesario tenerlo en cuenta, como hemos explicado previamente. En rojo se pueden apreciar los conjuntos que forman VC y DS en ambos grafos ilustrados.

Ahora bien, aunque hayamos encontrado una transformación que nos convierte un grafo que contiene una cobertura de vértices VC en uno que presenta la propiedad de Dominating-Set DS, esto no nos garantiza aún que dicha transformación sea polinómica, por la definición de $\leq_p$. Para ello hacemos lo siguiente.

Analicemos paso a paso de manera detallada que ocurre en la transformación. Como se ha explicado anteriormente, lo que realiza nuestra función es en primer lugar iterar sobre la codificación del grafo de partida G y para cada arista que este contenida en el mismo, generar 1 vértice nuevo y sus 2 aristas correspondientes adicionales. Esta operación, si consideramos n como el número de vértices del grafo G, podemos pensarla como: recorrer la cinta de la mT en la cual se encuentre la codificación para encontrar el paréntesis de fin de lista de vértices y en ese momento mover todo a la derecha para poder incluir el nuevo vértice. Como la cinta centra tamaño, a lo sumo $1+n+n+1+1+1+(n*(n-1)/2)*6+1$ (teniendo en cuenta todos los detalles de parentesis, comas, etc... y $n+(n*(n-1)/2)$ generalizando, que pertenece a $O(n^2)$. Además, se deben añadir también las 2 aristas asociadas, pero generalizando de nuevo el orden de complejidad seguirá siendo similar, de manera que será $O(n^2+n^2+n^2) = O(3n^2) = O(n^2)$. Como este proceso se debe repetir para cada arista presenta en el grafo, que a lo sumo serán $\frac{n\cdot(n-1)}{2}$, podemos concluir que el proceso de añadir todos los vértices y aristas adicionales consume $O(n^2 * \frac{n\cdot(n-1)}{2}) \in O(n^4)$. Tras esto, solo queda recorrer de nuevo la codificación del nuevo grafo, comprobando si existe algún vértice aislado y guardarlo en algún contador (que con una mT multicinta no es una tarea tediosa), y escribir al final de la codificación k', que será igual a $k+|I|$, siendo I el conjunto de vértices aislados que la máquina de Turing multicinta haya almacenado en su último recorrido. Por ello la complejidad final queda la forma $O(n^4+n+1) \in O(n^4)$. Queda demostrado por tanto que la reducción se opera en tiempo polinómico.

Para establecer la corrección de la reducción, necesitamos mostrar que G tiene un vertex cover de tamaño k si y solo si G_0 tiene un conjunto dominante de tamaño k'.

($\Rightarrow$) Primero debatimos que si V' es un vertex cover para G, entonces $V'' = V' \cup V_I$ es un Dominating Set para G'. Se observa que $|V''| = |V' \cup V_I| \leq k + nI = k'$. Se tiene en cuenta que $|V' \cup V_I|$ podría tener un tamaño menor que $k + nI$, si no hay vértices aislados en V'. Si es así, podemos agregar cualquier vértice que deseemos para hacer que el tamaño sea igual a k'. Para ver que V'' es un conjunto dominante, primero observa que todos los vértices aislados están en V'' y por lo tanto están dominados. Segundo, cada uno de los vértices especiales x_{uv} en G' corresponde a una arista $\{u, v\}$ en G, lo que implica que u o v está en el vertex cover V'. Por lo tanto, x_{uv} está dominado por el mismo vértice en V''. Finalmente, cada uno de los vértices originales no aislados v es incidente a al menos una arista en G, y por lo tanto o está en V' o todos sus vecinos están en V'. En cualquier caso, v está en V'' o adyacente a un vértice en V''.

($\Leftarrow$) De manera inversa, podemos probar que si G' contiene un Dominating Set V'' de tamaño $k' = k + nI$ entonces G tiene una cobertura de vértices V' de tamaño k. Primero tengamos en cuenta que todos los vértices aislados de G' para a pertenecer a su Dominating Set. Construimos un conjunto V''' que se obtiene de tomar V'' y quitarle los vértices aislados I. Podríamos pensar que directamente V''' es un vertex cover para G, pero no necesariamente debe ocurrir eso, ya que V''' puede que tenga vértices que no pertenecen al grafo original (vértices adicionales) del que partiamos al comienzo de la demostración. Sin embargo, por sencillez vamos a suponer que V''' no puede contener este tipo de vértices, y que por tanto si un vértice adicional x_{uv} pertenece a V''' se sutituye por u o v. (tomamos por ejemplo u). Además, salimos ganando, ya que si algún vertice adicional pertenecía a V''' solo dominaba a u y v, sin embargo al haberlo sustituido por uno de estos 2 últimos, quizás u domine ademas de x_{uv} y v, a más vértices. De esta manera, el conjunto de vértices resultante constituye un vertex cover para G.

Referencias

https://es.wikipedia.org/wiki/Cobertura$_{d}e_{v}C3A9rtices$
https://en.wikipedia.org/wiki/Dominating$_{s}et$
https://es.wikipedia.org/wiki/NP-completo