Complejidad y Calculabilidad

Jose David Salazar Moreno
josdsalazarmor@unal.edu.co

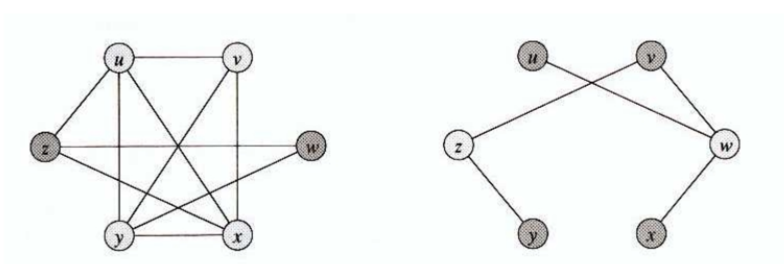
Mayo 2019

1 Clique Max NP-Completo

Antes de la reducción, necesitamos probar que clique $\in NP$. Los problemas en NP se tienen que verificar en tiempo polinomial, supongamos nos es dado un grafo G= (V, E) y un entero k, y tenemos una solución v $\in V$, revisamos si el enlace (u, v) está en E, o ahí existe un enlace entre u y v. El primero puede ser completo en V y el segundo se puede chequear en V ˆ 2 pasos, por lo tanto la verificación puede ser O(nˆ 2) entonces clique pertenece a NP. Ahora mostraremos que VERTEX-COVER puede se reducido a CLIQUE. Como VERTEXCOVER es NP completo dado un grafo no dirigido G= (V, E), definimos el complemento de G como G' = (V,E') donde E'= (u, v) :u, v V, u != v,(u, v) no pertenece a E . G' tiene los mismo vértices que G, y ninguno de los enlaces en G, Como prueba si existe un clique V' en G' entonces V - V' es un vertices cubierto en G. El algoritmo VERTEX-COVER(G,k) por reducción a Clique sigue:

- Dado un grafo G= (V, E) y un entero K
- Genere el complemento de G' = (V,E')
- Resolver el problema de Clique (G',V k)
- Si es una solución retorna Si de otra forma No

- Pandilla (clique) ={u,v,x,y}
- Cubierta de vértices = {w,z}

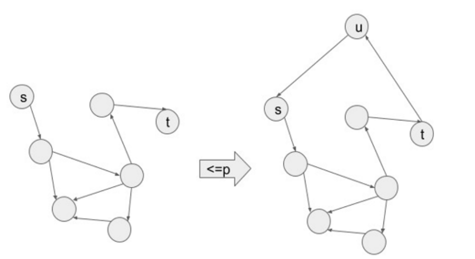


(a) (b) [1]

Para demostrar que esta reducción es correcta, debemos mostrar dos cosas: Si hay una solución para VERTEX-COVER (G, k), entonces debe haber una solución para CLIQUE (G', — V — k) Si existe es una solución para CLIQUE (G', V — k), entonces debe haber una solución para VERTEX-COVER (G, k) Primero, supongamos que G tiene una cobertura de vértices V' V, donde — V — '= k. Luego, para todo u, v V, si (u, v) E, entonces u V' o v V' o ambos, ya que la cubierta del vértice debe cubrir todos los enlaces. Lo contrario de esto es que para todo u, v V, si u no pertenece a V' y v no pertenece V', entonces (u, v) no pertenece a E y por lo tanto (u, v) E'. Esto significa que para cualquier par de vértices que no estén en la cubierta V' de G de vértice, hay un borde entre ellos en G'. La unión de todos los pares que no están todos en V' es simplemente V- V '. Por lo tanto, V-V' es clique en G', por definición, y V-V' tiene tamaño — V— - k. Supongamos que G' tiene un clique V' V, con — V '— = — V — - k Sea (u, v) cualquier enlace en E. Luego, (u, v) no pertenece E, lo que implica que al menos uno de los valores no pertenece a V ', ya que cada par de vértices en V' está conectado por un enlace de E'. En consecuencia, al menos un vértice u o v está en V - V '. Esto hace que el enlace (u, v) sea superado por V-V '. Cómo elegimos (u, v) arbitrariamente de E, cada borde en E está cubierto por V-V '. Por lo tanto, V-V' es a vértice cubre de G, con tamaño k. Hemos probado que VERTEX-COVER puede reducirse a CLIQUE.[4]

2 Circuito Hamiltoniano

Entrada: Un grafo dirigido G=(V,E). Problema: Decidir si existe un circuito en el grafo que atraviese cada nodo exactamente una vez. Camino Hamiltoniano. p Circuito Hamiltoniano. f(¡G=(V,E),s,t¿)= ¡G'=(V u ,E (u,s),(t,u))¿



[5]

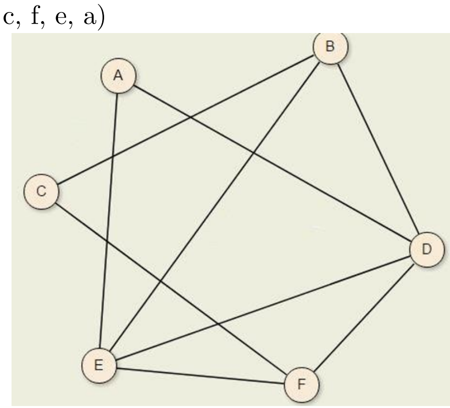
Comprobación: Si existe un camino hamiltoniano (v0=s,v1,...,vn=t) en el grafo original, entonces (u,v0=s,v1,...,vn=t,u)e un circuito hamiltoniano en el nuevo grafo. Comprobación: (u,s)y (t,u) deben estar en cualquier circuito Hamiltoniano del grafo construido. Por tanto, eliminando el nodo u obtenemos un camino desde s hasta t.

- Construir f.
- Comprobar que f es computable en tiempo polinómico

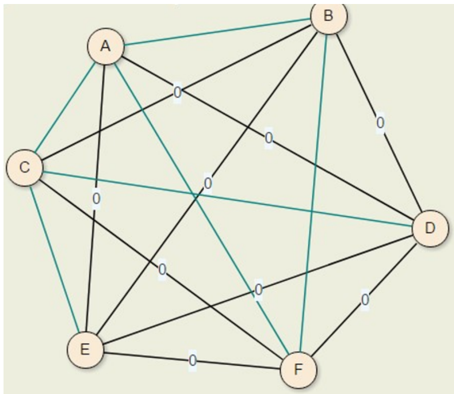
- Probar que f es una reducción, esto es:
- Si w $\in CaminoHamiltoniano$. Entonces f(w) $\in CircuitoHamiltoniano$.
- Sif(w)$\in CircuitoHamiltoniano$. Entonces w $\in CaminoHamiltoniano$

3 Agente Viajero es NP completo

El problema de agente viajero es un problema NP, debido a que su complejidad es O(— V— !), esto es porque en el peor de los casos debo probar todas las posibles permutaciones del conjunto de vértices del grafo. Reducción Dada una instancia de circuito hamiltoniano G = (V, E) Cree una copia de G de la siguiente forma: G' = (V', E') con K = 0 V' = V E' = (u, v) u, v $\in V$, Es decir, con los mismos vértices que G, pero haciendo que G' sea completo. para todo e = (u, v) $\in E$, W(e) = 0 si (u, v) $\in E$, W(e) = 1 si (u, v) $\in E$, Para arista en G' presentes también en G, se asigna peso de 0 Para arista en G' que no están en G0 se asigna un peso de 1 Verificación de instancias positivas y negativas Para instancias positivas en G, de tal manera que exista un circuito hamiltoniano, siempre existirá un ciclo en G' con longitud = 0, esto debido a que las aristas de G que se encuentran en G' tienen un peso de cero. Para instancias negativas en G, de tal manera que no exista un circuito hamiltoniano, no existirá un ciclo en G' con longitud = 0, debido a que las nuevas aristas en G' tienen un peso de 1. Dado un grafo G una instancia positiva de circuito hamiltoniano S = (a, d, b, c, f, e, a)



Se construye un grafo G', de tal manera que las arista en azul tiene peso 1 y las negras tiene peso 0, con esto el mismo conjunto S forma ciclo longitud = 0



4 Parada'

Dando un contraejemplo: Teóricamente Sea f(P,Q,n) = Parada', y r(P,Q) = Parada (instancia de parada de turing) f es una instancia de parada porque f = h(r(P,Q),n) donde h es una funcion que calcula el numero de operaciones hasta parar o hasta antes de entrar en un estado de no parada, h compuesto r por reducción dice que h no es computable, y r tampoco es computable asi que f no es computable.[3] Algorítmicamente Para demostrar que Parada no es computable, usando de ejemplo un lenguaje de programación como python, supongamos que Parada' existe y es computable def paradaPrima(P, Q ,n) : if operaciones(P(Q)) ¡ n : return True return False Sea Contra la función que recibe como entrada el programa P, y un entero n no negativo, y en la salida no termina def Contra (P, n) : while True : pass def Contra(Contra, n) : if paradaPrima(Contra, Contra, n) : while True :pass Si parada Prima retornara True significaria que Contra termina, y luego entraria a un bucle infinito donde no terminaria, y si paradaPrima retornara False significaria que Contra no termina, y luego no siguen mas instrucciones asi que termina Esta demostracion se vuele en esencia el Argumento de la diagonal de Cantor[2]

References

[1] *http://profesores.elo.utfsm.cl/ agv/elo320/01and02/NP*$_c$*ompleteness/NP*$_C$*ompletitud.pdf.*

[2] *https://es.wikipedia.org/wiki/Argumento*$_d$*e*$_l$*a*$_d$*iagonal*$_d$*e*$_C$*antor.*

[3] *https://es.wikipedia.org/wiki/Problema*$_d$*e*$_l$*a*$_p$*arada.*

[4] *http://www.sc.ehu.es/jiwhehum2/TC/temas/[4]reducciones.pdf.*

[5] *http://www.cs.toronto.edu/ekzhu/teaching/csc373summer2015/tut9.pdf.* 2015.