**2. Coloreado de grafos con 3 colores [Graph 3-coloring]:** Dado un grafo G =

(V, E), ¿existe una función f : V 🡪 {1, 2, 3}, tal que f(u) ≠ f(v) para todos los arcos {u, v} є E?

Vamos a hacer una reducción basándonos en el problema 3SAT (comprobar si una expresión booleana, con 3 literales, tiene o no una asignación que la haga cierta). Para ello haremos una reducción intermedia con NAE-3-SAT:

Haremos la cadena de reducciones: 3SAT ≤ NAE-3-SAT ≤ G3C.

1. NAE-3-SAT ≤ G3C:

Para cada literal que aparezca en nuestro problema NAE-3-SAT vamos a considerar dos vértices, que presentan el literal y su complementario. Establecemos una unión dos a dos (el vértice del literal y el vértice del literal negado) mediante una arista. Para crear el grafo base G=(V, E), falta añadir un vértice extra, sea phi, que conectaremos con una arista a cada uno de los vértices que hemos creado.

Este grafo inicial se puede colorear sin problema alguno mediante 3 colores: primer color para phi, segundo color para los literales de nuestro problema NAE-3-SAT, tercer color para los literales negados.

Ya tenemos el grafo con los literales, tenemos que representar las cláusulas para que la reducción sea correcta. Supongamos la cláusula (x1, x2c, x3), donde x2c representa el complementario del literal x2.

Creamos un grafo aparte S donde los vértices sean los literales de la cláusula, y conectamos los tres vértices mediante aristas.

Vamos a conectar este grafo S creado a partir de la cláusula con el grafo G:

Unimos mediante una arista cada vértice de S con su correspondiente literal negado del grafo G.

Todo coloreado de un grafo con 3 colores corresponde con una cláusula del problema NAE-3-SAT.

1. 3SAT ≤ NAE-3-SAT:

Sea (x1 v x2 v x3) una instancia del problema de 3SAT. Se satisface cuando alguno de los literales es cierto, es decir, podemos tener: V F F

V V F

V V V (3V)

Sea (x1, x2, x3) una instancia del problema NAE-3-SAT. Se satisface cuando se da uno de los casos (A): V F F

V V F

El problema nos lo encontramos, por tanto, cuando se da 3V. Vamos a tratar convertido en un caso de A:

Partamos de (x1 v x2 v x3). Vamos a añadirle un literal z que asignaremos como F. Le asigno la misma variable a todas las cláusulas. Tendremos por tanto, (x1 v x2 v x3 v z) 🡪 (x1, x2, x3, z).

Lo que hemos hecho es la reducción 3SAT ≤ NAE-4-SAT. Falta descomponer las cláusulas en 3 variables. Para ello añado otra variable, y, negada y afirmada (x1, x2, x3, z, y, yc) y lo separo en cláusulas de 3 literales de modo que sean correctas.

**4. Subgrafo común maximal [Subgraph isomorphism]:** Dados los grafos G1 = (V1, E1), G2 = (V2, E2), y un entero positivo K, ¿existen subconjuntos E1’ contenido en o igual a E1 y E2’ contenido en o igual a E2 tales que |E1’| = |E2‘| ≥ K y tal que los dos subgrafos G1’= (V1, E1’) y G2’= (V2, E2’) son isomorfos?

Vamos a hacer la reducción CLIQUE ≤ ISOMORFISMO SUBGRAFOS.

Sea CLIQUE (G,K){

Return IS(G,K,) //me encuentra el isomorfismo de un grafo G a un clique (k nodos todos conectados entre sí)

}

Nos determinada si el grafo G tiene un clique de tamaño K.

**6. Empaquetado de conjuntos [Set packing]:** Dada una colección C de conjuntos finitos y un entero K ≤ |C|, ¿existen K conjuntos disjuntos en C?

Vamos a hacer una reducción basándonos en el problema de conjuntos independientes (ninguno de sus vértices es adyacente a otro).

Dado un problema de independencia de conjuntos sobre un grafo G = (V, E), vamos a crear una colección de conjuntos C donde para cada vértice Vi del grafo, tengamos un conjunto conteniendo todas las aristas adyacentes a ese vértice Vi. Es decir, C es de la forma:

C={vi / i=1,…,n} donde vi es un conjunto que contiene a todas las aristas adyacentes al vértice Vi del grafo G.

Todo empaquetamiento de conjuntos en la colección C corresponde a un conjunto independiente en G.

**8. Partición de conjuntos (bis) [Set splitting]:** Dada una familia C de subconjuntos de un conjunto finito S, ¿existe una partición de S en dos partes S1 y S2 tales ningún elemento AєC esté contenido ni en S1 o ni en S2 (todo AєC debe de tener intersección no vacía con S1 y con S2). Pista: Reducir desde 3SAT.

Uso este enlace: <http://neal.young.name/neal1/2004/cs215/wikidb/uploads/hwk4_solns.pdf>

Hacemos la reducción 3SAT ≤ Partición de conjuntos.

Sea V = {conjunto de variables del problema 3SAT}. Sea phi una fórmula bien formulada del problema.

Construyo la siguiente instancia del problema de partición de conjuntos:

S= z U V U Vc = {xc / xV}, siendo z una variable que no esté en V ni en Vc.

Construimos C con los siguientes conjuntos:

Para cada variable x en phi, construimos el conjunto Sx = {x, xc}

Para cada cláusula cli єCL construimos el conjunto Scli={variables de la cláusula U z}

Queda C={Sx U Scli}.

Para cada cláusula phi verdadera del problema 3-SAT tomamos sus variables como S1 y las demás de S como S2.

**10. Matching 3D [Perfect 3-Matching]:** Dados tres conjuntos U, V, W de tamaño n y un conjunto S contenido en o igual a U x V x W, ¿existe un subconjunto T contenido en o igual a S de tamaño n que incluya cada elemento de U υ V υ W exactamente una vez?

NOTA: El Matching bidimensional está en P (p.ej. se puede resolver como un problema de flujo en redes).

Vamos a reformularlo:

Sean (u1, v1, w1), (u2, v2, w2) є T con (u1, v1, w1) ≠ (u2, v2, w2) 🡪 u1≠u2, v1≠v2, w1≠w2. Hagamos la reducción 3SAT ≤ Matching 3D.

Sea un problema 3SAT con variables li que englobamos en un conjunto L={li tal que i=1,….,n} y un conjunto de cláusulas C={ci tal que i=1,…,m}.

Construimos los conjuntos U, V y W:

Para cada variable li vamos a introducir en U los elementos li[j], lic[j], en V cierto elementos ai[j] y en W ciertos elementos bi[j], j=1,…,m y i=1,...,n.

En el conjunto S introduzco las tripletas Tit={( lic[j], ai[j], bi[j]): j=1,...,m} y

Tif = {( li[j], ai[j+1], bi[j]): j=1,…,m-1} U {( li[m], ai[1], bi[m])}

Con esta asignación de conjuntos conseguimos que cualquier subconjunto T tenga que contener los conjuntos L o Lc.

Ahora tenemos que tener en cuenta las cláusulas del conjunto C:

Añadimos los elementos s1[j] a V

Añadimos los elementos s2[j] a W.

Añadimos a S las tripletas:

{(li[j], s1[j], s2[j]) tal que li está en cj} U {(lic[j], s1[j], s2[j]) tal que ¬li está en cj}

Los subconjuntos T tendrán que contener alguna de estas tripletas.

Finalmente añadimos los siguientes elementos:

Añadimos g1[k] con 1≤ k ≤ m(n-1) a V.

Añadimos g2[k] con 1≤ k ≤ m(n-1) a W.

Añadimos a S las tripletas:

(li[j], g1[k], g2[k]), (lic[j], g1[k], g2[k]), 1≤ k ≤ m(n-1), 1≤ i ≤ n, 1≤ j ≤ m.

**12. Partición en caminos de longitud 2 [Path partition]:** Dado un grafo G =

(V, E), con |V| = 3q donde q es un entero positivo, ¿existe una partición de V en q conjuntos disjuntos V1, V2, …, Vq de tal forma que para cada Vi = {vi[1], vi[2], vi[3]}, al menos dos de los tres posibles arcos, {vi[1], vi[2]}, {vi[1], vi[3]}, {vi[2], vi[3]} está en E.

Pista: Reducir PERFECT 3-MATCHING...

**14. Numeración en grafos [Graph Grundy Numbering]:** Dado un grafo dirigido G = (V,A), ¿existe una numeración L : V 🡪 N, donde el mismo número puede asignarse a más de un vértice y tal que cada L(u) es igual al mínimo de todos los valores enteros que no están en {L(v) : (u, v) є A}.

**16. Circuito hamiltoniano alternativo [Another Hamiltonian cycle]:** Dado un grafo y un circuito hamiltoniano, determinar si el grafo tiene otro circuito hamiltoniano.

Vamos a hacer la reducción basándonos en el recubrimiento de vértices:

CV ≤ CH

Sea G=(V,E) un grafo y K≤|V| un problema de cubrimiento por vértices.

Construiremos un grafo G’=(V’,E’) e modo que la existencia de un circuito hamiltoniano para G’ sea equivalente a la existencia de un recubrimiento de tamñano ≤ K para G. Veámoslo:

Ponemos tres nodos en G’ por cada arista de G. Supongamos que la arista (i,j) une los vértices i, j. Llamaremos a estos tres nodos de G’ como (i,j)m con m=1,2,3. Añadiremos también K nuevos nodos h=1,2,…K.

Relacionamos los nodos de G’, es decir, consideramos el conjunto E’ de la siguiente forma:

E’=A U B U C donde:

Los nodos (i,j)m están relacionados por las siguientes aristas:

A={ ((i,j)1, (i,j)2) , ((i,j)2, (i,j)1) , ((i,j)2, (i,j)3) , ((i,j)3, (i,j)2) }

Se añaden dos aristas entre cada uno de los nodos h=1,2,…,K y los nodos de tipo (i,j)

B={ ((i,j) 1,h) , (h,(i,j) 1) , ((i,j) 3,h) , (h,(i,j)3)}

Añadimos finalmente C={ ((h,i),(i,j)) para (h,i),(i,j)єE, h≠j}.

Si G tiene un recubrimiento de vértices formado por k nodos, entonces G’ tendrá un circuito hamiltoniano.

**18. Equivalencia de expresiones regulares sin estrella [Star-Free Regular**

**Expression Equivalence]:** Dadas dos expresiones regulares E1 y E2 sobre el alfabeto A que no contienen el operador de clausura \*, ¿representan estas expresiones regulares lenguajes distintos sobre A?

12 14 18 20