Problema NP-Completo: EXACT-COVER

Hecho por Marcos Hidalgo Baños a día 19 de mayo del 2022

Definición formal en forma de lenguaje.

EXACT-COVER es un problema de **decisión** que pretende dar respuesta a la pregunta: 'Dado un par de conjuntos <S, V>, ¿existen uno o más subconjuntos de S que cubran exactamente los valores del conjunto V?' donde el concepto cubrir exactamente hace referencia al número de veces que un elemento puede ser escogido por un conjunto, siendo en este caso particular uno.

Una instancia de EXACT-COVER es definida por una pareja de conjuntos (S, V) tal que S es un conjunto de elementos y $S^* \subseteq P(S)$ es un conjunto de subconjuntos de S. Su objetivo es decidir si existe un subconjunto $E \subseteq P$ tal que sea una partición de S.

Definición formal del problema: **{ $\langle S, V \rangle$ | $S^* \subseteq \wp(V)$ ¿S* contiene una cobertura exacta de V? }**

<u>Ejemplo sencillo válido</u>

S = {A, B, C, D}

V = {1, 2, 3, 4, 5}

A = {2, 4}

B = {3, 4, 5}

C = {1, 3}

D = {5}

S* = { {A, C, D} }

<u>Ejemplo sencillo inválido</u>

S = {A, B, C, D}

V = {1, 2, 3, 4, 5}

A = {2, 4}

B = {3, 4, 5}

C = {1, 3}

D = {4}

S* = { }

Notar que una pequeña modificación en la creación de los conjuntos que componen el macro-conjunto S* hace que la cobertura no sea exacta. Si no se ha encontrado ninguna posible combinación de elementos de S que cubran exactamente a V, decimos que para dichos conjuntos no existe ninguna cobertura exacta.

Aplicaciones de EXACT-COVER.

Normalmente, el problema EXACT-COVER suele ser utilizado por otros para establecer una **reducibilidad** en tiempo polinómico y así determinar que también son problemas NP-Completos.

Algunos ejemplos son:

Sudoku - Hundir la Flota - Problema de las N-reinas - Tetris - Pentomino

Destacar que estos problemas requieren de una **solución muy concreta y específica** para cada instancia, como si de un puzzle se tratase, en el que hay que encajar las piezas de una determinada manera. Sin embargo, en este tipo de problemas (al igual que en EXACT-COVER) la solución no tiene por qué ser única, por lo que se deberá proporcionar un conjunto S* de soluciones que para cada instancia cumplan las restricciones.

EXACT-COVER es NP-Completo.

Para demostrar que EXACT-COVER es un problema NPC tal y como Karp determinó en 1972, deberemos realizar las siguientes comprobaciones:

1) **EXACT-COVER $\in$ NP**
   - ➔ Mediante verificador
   - ➔ Mediante una mTnD

2) **$\forall$ A $\in$ NP: A $\leq_p$ EXACT-COVER**

   Para ello, nos valdremos de otro conocido problema NPC.

   Una vez realizada la reducción, sabremos que EXACT-COVER $\in$ NPC.

Breve Introducción a 3-COLOR.

Si un grafo es **3-coloreable** (RGB) entonces:

- Cada nodo es instanciado a uno de los tres colores.
- No hay dos nodos conectados que tengan asignado el mismo color.

Aplicaciones: Problemas de optimización, problemas de resolución de conflictos, coloreado de mapas, identificación de grafos bipartitos... (Típicos problemas de coloreado de un grafo, al ser este un caso particular con 3 colores).

Es fácil ver el parecido entre ambos problemas, siendo la solución de una instancia de 3-COLOR una de las tantas que seguramente cumplan las condiciones descritas del problema. Además, es exacta ya que no admite pequeñas modificaciones (es rígida).

EXACT-COVER pertenece a NP.

- Demostración con verificador

Con entrada <<S,V>, S*> :

- **Para cada conjunto (solución) de S* :**
    1) Comprobar que únicamente contiene elementos de S
        - *Complejidad Temporal del paso:* ***O(n)***
- **Para cada uno de sus elementos (que pertenecen a S):**
    2) Comprobar que los valores del conjunto V que contiene no han sido elegidos por otro. Para ello, si un conjunto escoge un elemento, este queda eliminado de V para los siguientes.
        - *Complejidad Temporal del paso:* $O(n^2)$
    3) Repetir para cada conjunto (solución) de S*
        - *Complejidad Temporal del paso:* ***O(n)***
- **Solo aceptamos si todas las comprobaciones se cumplen.**
    - Remarcar que es necesario que todos los conjuntos de S* pasen las comprobaciones (propenso a ataques de inyección).

- Demostración con mTnD

N = “Con la entrada <S,V> :

1. Crear el conjunto S* de posibles soluciones inicialmente vacío.
    - *Complejidad Temporal del paso:* ***O(1)***
2. De forma no determinista, seleccionar un subconjunto X compuesto por elementos de S (candidato a ser añadido a S*).
    - *Complejidad Temporal del paso:* ***O(n)***
3. Comprobar para todo elemento de X si los valores del conjunto V que contienen se no se repiten con los elegidos por otro.
    - *Complejidad Temporal del paso:* $O(n^2)$
4. Si se cumple la condición 3, introducimos el conjunto X en el conjunto de soluciones posibles S* (consecuencia del estado de aceptación q_acc de nuestra máquina de Turing). Si por el contrario no se cumple, lo ignoramos (consecuencia del estado de rechazo q_rej de nuestra máquina de Turing).
5. Sabremos que hemos terminado de completar el conjunto S* una vez hayamos realizado el paso 2 para cada combinación posible. "

3-COLOR $\leq_p$ EXACT-COVER.

Objetivo.

Construir un universo *U* y un conjunto de conjuntos *S* tales que una cobertura exacta:

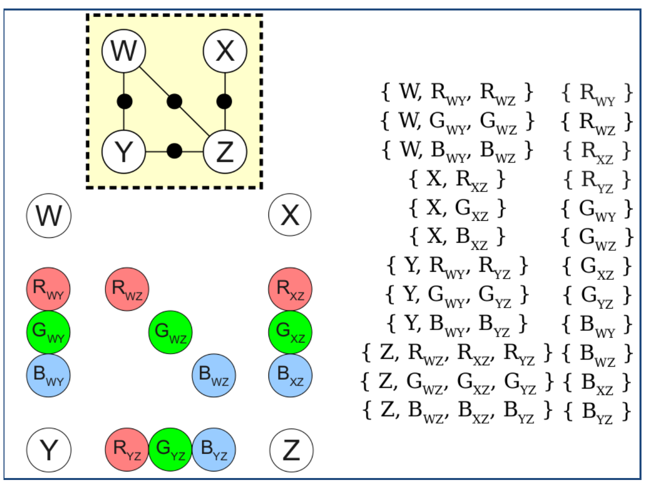
- Asigne a cada nodo de G uno de los tres colores.
- Nunca asigne el mismo color a dos nodos adyacentes.

Metodología.

Para cada nodo *v* y cada arista {*u*, *v*} en el grafo G construir en el universo U:

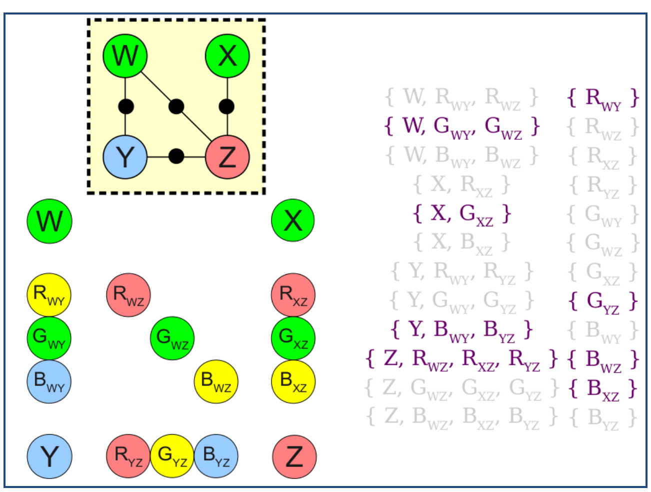
- El elemento *v* → Identificador único para el nodo.
- Los elementos R_{uv} , G_{uv} , B_{uv} → Consecuentes de instanciar un nodo v a uno de los tres colores posibles.

Explicación del gadget empleado.



En la columna de la izquierda encontramos las posibles instancias de cada uno de los nodos (RGB) y la consecuencia “señal” que se transmitirá por la arista hacia sus vecinos.

En la de la derecha, tenemos un listado (sin ninguna asociación con su predecesora) de restricciones que pueden llegar a darse en nuestro grafo por circunstancias del diseño y asignación de los colores a los nodos.



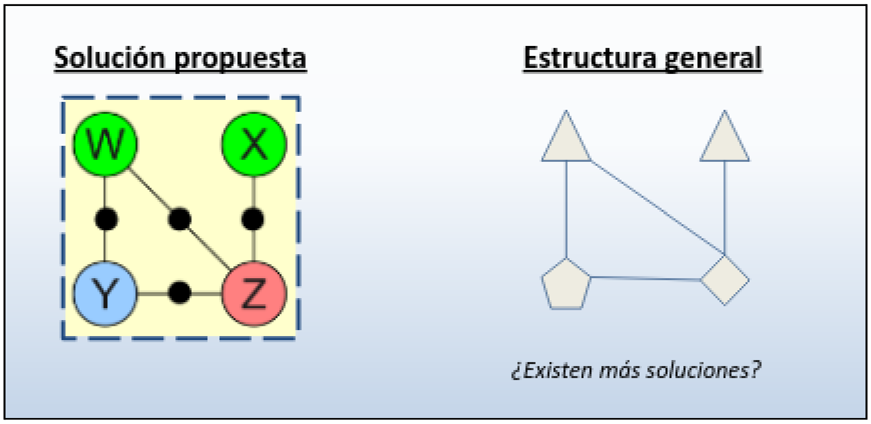
Se muestra una posible solución a la instancia del problema, en el que se han elegido las filas destacadas de la primera columna (una por cada nodo).

Se condiciona que una serie de nodos (indicados en la columna de la derecha) no puedan tomar determinados colores debido a que otro tercer nodo vecino ya lo eligió.

Demostración de la doble implicación.

1) Si $\omega \in$ 3-COLOR $\Longleftrightarrow$ f(ω) $\in$ EXACT-COVER

Para este problema tan restrictivo, tener en cuenta que la instanciación de una parte de la solución (colorear un nodo, colocar un número en el sudoku …) condiciona en gran medida al resto de elementos de S (no repetir color, no repetir números …)

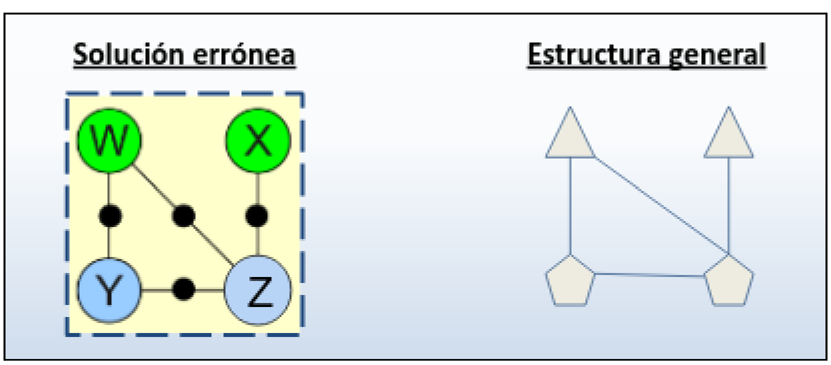


Una vez vista la solución propuesta al problema de coloreado, deberemos abstraer la estructura en la que se sustenta para así poder comprobar que ‘lo que funciona en uno, funciona en el otro’. Esto lo representaremos mediante distintas figuras, pero puede ser interpretado mediante números, barcos de Hundir la Flota, reinas de n-reinas, etc.

2) Si $\omega \notin$ 3-COLOR $\Longleftrightarrow$ f(ω) $\notin$ EXACT-COVER

Al no cumplirse la condición que impone 3-COLOR (ya sea por el no uso de los tres colores o por la incompatibilidad en el coloreado), podemos ver cómo en EXACT-COVER encontramos dos ‘figuras’ iguales conectadas ➔ *representa colisión*

Esto equivaldría a decir que hemos cogido dos veces el mismo número en sudoku, o que hemos colocado en la misma casilla dos reinas en n-reinas.



Función de transformación computable en poli-t

Sea *V* el conjunto de nodos → Contamos cada nodo 3 veces (RGB).

Sea *E* el conjunto de conexiones → Contamos cada conexión 2·3 = 6 veces.

El tamaño total del conjunto *U* es **O(|V|+|E|)**

- Tendremos que recorrer el conjunto universal en su totalidad.
- Se requiere de tiempo polinómico para recorrer O(n) celdas.

Queda demostrado que la función es computable en poli-t

Referencias bibliográficas.

- Resumen profesorado Stanford sobre EXACT-COVER.

  https://web.stanford.edu/class/archive/cs/cs103/cs103.1132/lectures/27/Small27.pdf

- Información adicional sobre algoritmos para EXACT-COVER.

  https://hal.archives-ouvertes.fr/hal-02507186/document

- Idea general para la reducibilidad 3-COLOR ≤p EXACT-COVER.

  http://www.cs.umd.edu/~gasarch/BLOGPAPERS /npcproblems.pdf