

Exact cover

Claudia Puentes

December 2024

El problema *Exact Cover* consiste en determinar, dado un conjunto universo X y una colección de subconjuntos S , si existe un subcolector $S' \subseteq S$ tal que cada elemento de X aparece exactamente una vez en los subconjuntos de S' .

Para demostrar que *Exact Cover* es NP-completo, debemos: probar que el problema pertenece a la clase NP y reducir un problema conocido como NP-completo (en este caso, SAT) al problema *Exact Cover*, mostrando que es NP-Hard. Un problema pertenece a NP si una solución candidata puede ser verificada en tiempo polinomial.

En el caso de *Exact Cover*, una solución candidata es un subconjunto $S' \subseteq S$. Para verificar si S' es un Exact Cover, comprobamos que cada elemento de X aparece exactamente una vez en los subconjuntos de S' . Esto puede lograrse recorriendo S' y verificando las apariciones de cada elemento de X . La verificación toma tiempo $O(|X| \cdot |S'|)$, que es polinomial en el tamaño de la entrada. Por lo tanto, *Exact Cover* pertenece a NP.

Demostraremos que *Exact Cover* es NP-Hard mediante una reducción polinomial desde el problema SAT, que es NP-completo. Dado un conjunto de variables booleanas $\{x_1, x_2, \dots, x_n\}$ y una fórmula en forma normal conjuntiva (CNF) $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_m$, donde cada cláusula C_j es una disyunción de literales (x_i o $\neg x_i$), el problema SAT consiste en determinar si existe una asignación de verdad para las variables que haga verdadera la fórmula ϕ . Dada una instancia de SAT, construiremos una instancia de *Exact Cover*.

El conjunto universo X incluye tres tipos de elementos:

x_i , un elemento para cada variable;

C_j , un elemento para cada cláusula;

$p_{i,j}$, un elemento para cada ocurrencia de una variable x_i como literal (positivo o negativo) en una cláusula C_j .

Así:

$$X = \{x_1, x_2, \dots, x_n\} \cup \{C_1, C_2, \dots, C_m\} \cup \{p_{i,j} \mid x_i \text{ aparece en } C_j\}.$$

Para el conjunto de subconjuntos S :

Para cada variable x_i , creamos dos subconjuntos: $S_{\text{true},i} = \{x_i\} \cup \{p_{i,j} \mid x_i \text{ aparece como } \neg x_i \text{ en } C_j\}$, que cubre el elemento x_i , indicando que se asigna True a la variable y asegura que todas las cláusulas donde x_i aparece negado están representadas. $S_{\text{false},i} = \{x_i\} \cup \{p_{i,j} \mid x_i \text{ aparece como } x_i \text{ en } C_j\}$, que cubre el elemento x_i , indicando que se asigna False a la variable y asegura que todas las cláusulas donde x_i aparece como literal positivo están representadas. Para cada literal $p_{i,j}$, creamos: $\{C_j, p_{i,j}\}$, representa que el literal satisface la cláusula C_j . Esto garantiza que cada cláusula tiene al menos un literal verdadero que la satisface. $\{p_{i,j}\}$, un conjunto auxiliar para cubrir las ocurrencias no utilizadas, asegurando que todos los elementos de X sean cubiertos exactamente una vez.

Si existe una asignación de verdad que satisface la fórmula SAT, seleccionamos: $S_{\text{true},i}$ si $x_i = \text{True}$, o $S_{\text{false},i}$ si $x_i = \text{False}$. Esto garantiza que el elemento x_i está cubierto y consistente con la asignación booleana. Para cada cláusula C_j , seleccionamos un subconjunto $\{C_j, p_{i,j}\}$, donde el literal correspondiente

satisface C_j . Esto asegura que la cláusula está satisfecha. Los subconjuntos $\{p_{i,j}\}$ se seleccionan para cubrir las ocurrencias restantes de los literales no utilizados. Si existe un subcolector S' que cubre exactamente X , para cada x_i , seleccionamos $S_{\text{true},i}$ o $S_{\text{false},i}$, lo que define la asignación de verdad para x_i . Para cada C_j , seleccionamos $\{C_j, p_{i,j}\}$, lo que implica que el literal correspondiente satisface la cláusula C_j . Hemos demostrado que *Exact Cover* pertenece a NP, ya que una solución candidata puede verificarse en tiempo polinomial, y *Exact Cover* es NP-Hard, mediante una reducción polinomial desde SAT. Por lo tanto, *Exact Cover* es NP-Completo.