Problema de la Identificación por Intersección

David Cabezas Berrido

El problema es el siguiente:

Dado un conjunto U de tamaño n, m subconjuntos $A_1, \dots, A_m \subset U$ y m números naturales $c_1, \dots, c_m$, determinar si existe un subconjunto $X \subset U$ tal que en cada intersección $X \cap A_i$ hay c_i elementos:
$|X \cap A_i| = c_i \quad \forall i = 1, \dots, m$

Denotaremos este problema como II, veremos que este problema es NP-completo.

Primero comprobaremos que II $\in$ NP, diseñando un algoritmo no determinista que lo resuelve. Ante una instancia del problema II dada por U, $A_1, \dots, A_m$ y $c_1, \dots, c_m$:

1. Para cada uno de los n elementos de U, elegimos de forma no determinista si ese elemento formará parte de X o no. $\quad O(n)$
2. Para cada uno de los subconjuntos $A_i \subset U \quad i = 1, \dots, m$): $\quad O(m)$
   - 2.1. Fijamos $d = 0$. $\quad O(\log(n))$
   - 2.2. Para cada elemento a de A_i: $\quad O(n)$
     - 2.2.1. Se comprueba si $a \in X$. $\quad O(n)$
     - 2.2.2. En caso afirmativo se incrementa d: $d = d + 1$. $\quad O(\log(n))$
   - 2.3. Si $d \neq c_i$, se **rechaza** la entrada. $\quad O(\log(n))$
3. Se **acepta** la entrada. $\quad O(1)$

Claramente es un algoritmo que termina siempre, veamos que lo hace en tiempo polinómico:

La línea 1 realiza n pasos, menos que la longitud de la entrada, que notaremos como l. La línea 2 se ejecuta $m \leq l$ veces, en cada una se recorren todos los elementos de un A_i (a lo más n elementos), se comprueba si están en X (barrido por a lo más n elementos) y se incrementa un entero que como mucho vale n, su longitud será como mucho $\log(n)$, luego para incrementarlo se realizarán $O(\log(n))$ pasos, al igual que para fijarlo a 0. La comparación de d con cada c_i también requerirá $O(\log(n))$ pasos. Luego el número de pasos para reconocer si una instancia del problema tiene solución es $O(n)+O(m)\Big(O\big(\log(n)\big)+O(n)\big(O(n)+O\big(\log(n)\big)\big)+O\big(\log(n)\big)\Big)+O(1)$, que es $O(l^3)$. Por tanto II $\in$ NP.

Para ver que es NP-completo, reducimos 3-SAT (que sabemos que es NP-completo) a II, esta reducción tendrá que ser logarítmica en espacio.

Dada una instancia de 3-SAT: un conjunto $P = \{p_1, \dots, p_k\}$ de símbolos proposicionales y una colección $C = \{\gamma_1, \dots, \gamma_q\}$ de cláusulas sobre los símboles, cada una con exactamente 3 literales. Construiremos una instancia del problema II: U, $A_1, \dots, A_m$ y $c_1, \dots, c_m$ tal que existe un subconjunto $X \subset U$ cumpliendo $|X \cap A_i| = c_i \quad \forall i = 1, \dots, m$ si y solo si existe una asignación de valores de verdad a los símbolos de P que haga verdaderas todas las cláusulas de C.

Tendremos en U todos los símbolos proposicionales y sus negaciones, en principio $U = \{p_1, \dots, p_k, \neg p_1, \dots, \neg p_k\}$. Entenderemos que seleccionar un símbolo p_i en X es equivalente a asignarle a ese símbolo el valor de True (1): $p_i \in X \Leftrightarrow p_i = 1$ y $\neg p_i \in X \Leftrightarrow \neg p_i = 1 \Leftrightarrow p_i = 0$.

Para que realmente la pertenencia al conjunto X represente a un valor de verdad, debemos forzar que para cada $i = 1, \ldots, k$ uno y sólo uno de los literales p_i y $\neg p_i$ esté en X. Esto se consigue introduciendo conjuntos $A_1, \ldots, A_k$ y naturales $c_1, \ldots, c_k$ en II tales que

$$A_i = \{p_i, \neg p_i\}, \quad c_i = 1 \qquad \forall i = 1, \ldots k$$

Notamos que para cada cláusula de C: $\gamma_i = x \vee y \vee z \quad i = 1, \ldots, q$, donde x, y, z son literales de P (símbolos de P o sus negaciones), la cláusula es cierta si y solo si alguno de sus literales está en X (es cierto), esto equivale a que $|X \cap \{x, y, z\}| \geq 1$. Pero en II tenemos que imponer que el cardinal de la intersección sea un número exacto, para ello añadimos por cada cláusula $\gamma_i \in C$, un subconjunto $B_i \subset U$ $(i = 1, \ldots, q)$ con los literales de la cláusula como elementos y dos nuevos elementos a U: u_i, v_i, que sólo aparecen en el subconjunto asociado a la cláusula B_i.

De esta forma, si $\gamma_i = x \vee y \vee z$, entonces $B_i = \{x, y, z, u_i, v_i\}$ e imponemos $|X \cap B_i| = 3$. Una asignación de valores de verdad a los literales x, y, z satisface la cláusula si y solo si al menos un literal (de x, y, z) está en X (es verdadero), esto equivale a $1 \leq |X \cap \{x, y, z\}| \leq 3$ y a que exista una forma de elegir si u_i, v_i pertenecen o no a X de tal forma que $|X \cap B_i| = 3$: si $|X \cap \{x, y, z\}| = 1$, entonces $u_i, v_i \in X$; si $|X \cap \{x, y, z\}| = 2$, entonces $u_i \in X$ y $v_i \notin X$ (o al revés); y si $|X \cap \{x, y, z\}| = 3$, entonces $u_i, v_i \notin X$.

Tomamos $U = \{p_1, \ldots, p_k, \neg p_1, \ldots, \neg p_k, u_1, \ldots, u_q, v_1, \ldots v_q\}$, luego $n = 2(k + q)$. Tomamos $m = k + q$, reindexamos los $B_i \;\; i = 1, \ldots, q$ como $A_{k+i} \;\; i = 1, \ldots, q$ y tomamos $c_i = 3 \;\; i = k + 1, \ldots, k + q$.

Tenemos ahora una instancia del problema II que cumple: existe $X \subset U$ tal que $|X \cap A_i| = c_i \quad \forall i = 1, \ldots m$ si y solo si existe una asignación de valores de verdad a los símbolos proposicionales de la instancia del problema 3-SAT que satisfagan todas las cláusulas.

De existir tal X, tenemos la asignación explícitamente consultando para cada $i = 1, \ldots, k$ si es p_i o $\neg p_i$ el que pertenece a X.

Recíprocamente, de existir tal asignación es claro que el conjunto de literales verdaderos interseca en un único elemento con cada $A_i \quad i = 1, \ldots, k$ y se pueden añadir ciertos u_i y v_i a X para que la intersección con cada $A_i \quad i = 1, \ldots, q$ tenga exáctamente 3 elementos.

Es fácil ver que esta reducción se puede hacer en espacio logarítmico:

Primero leemos P y escribimos la primera parte de U, con cada símbolo y su negación. Después leemos C y escribimos la segunda parte de U con una pareja u, v por cada cláusula.

A continuación, leemos P y escribimos los subconjuntos $A_i \quad i = 1, \ldots, k$ cada uno con un símbolo y su negación. Pasamos otra vez a leer las cláusulas y por cada una escribimos un subconjunto $A_i \quad i = k + 1, \ldots, q$ con los literales de la cláusula y los símbolos u_i y v_i correspondientes.

Finalmente leemos P y por cada elemento escribimos $c_i = 1 \quad i = 1, \ldots, k$. Y leemos C y por cada cláusula escribimos $c_i = 3 \quad i = k + 1, \ldots, k + q = m$ en la salida.

En ningún momento escribimos en la entrada ni necesitamos volver a atrás en la salida. Según el modelo, puede que necesitemos almacenar algunos índices que indiquen por qué símbolo o cláusula vamos leyendo o escribiendo, pero estos ocupan espacio $O(\log(l))$ donde l es la longitud de la entrada.

Hemos reducido 3-SAT a II usando espacio logarítmico. Por tanto II es un problema NP-completo.