Demostración de NP-Completitud de 3D-Matching

Francisco Vicente Suárez Bellón

Septiembre de 2024

Contents

1 Descripción del problema

1. Sea $M \subseteq W \times X \times Y$
2. $W \cap Y \cap X = \emptyset$(disjuntos)
3. $|W| = |X| = |Y| = q$

Se quiere conocer si existe un matching en M, osea un $M^{'} \subseteq M$ tal que:

1. $|M^{'}| = q$
2. Todos los elementos de $W \cup X \cup Y$ están en algún triplo de $M^{'}$ sin repetir ninguno.

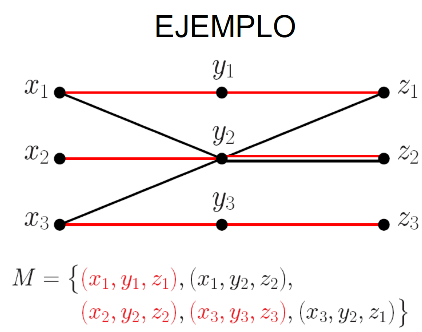


Figure 1: Ejemplo de posible entrada

2 Demostración de NP-Completitud

Pasos Principales:

1. Demostrar que 3DM $\in$ NP
2. Seleccionar un problema el cual se NP-Completo para hacer la reducción
3. Construir la transformacion del problema hacia 3DM
4. Demostrar que la transformación es en tiempo polinomial

3 3DM es NP

Dada una instancia de (M, X , Y, M) del 3DM se construye un algoritmo no determinista que genere una solución de $|W|$ triplos y compruebe en un tiempo polinomial que no hay dos tercetas con elementos comunes.

4 Problema NP-Completo para hacer la reducción

Para ello elegimos el 3-SAT

4.1 Definición de 3-SAT

1. Sea un conjunto de m cláusulas $C = c_1, \ldots, c_m$ con $|c_i| = 3$ $1 \leq i \leq m$
2. Sobre un conjunto finito de n variables booleanes $U = u_1, \ldots, u_n$

Se responde si existe alguna asignación válida de U que satisfaga todas las cláusulas

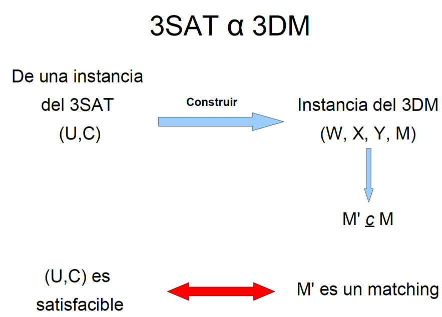


Figure 2: Reducciíon de 3-SAT a 3-DM

4.2 Traducción desde 3-SAT hacia 3-DM

3-Sat	**3-DM**
Variables: $u_1, \ldots, u_n$	Variables: $u_i(j), b_i(j), S_x(j), G_y(j)$
Literales: $u_1, \neg u_1$	Variables: $u_i, \neg u_i(j)$
Cláusulas: $C_j = (u_1, \neg u_2, u_3)$	Tercetas: $C_j\{(u_1(j), S_x(j), S_y(j)), (\neg u_2(j), S_x(j), S_y(j)), (u_3, S_x(j), S_y(j))\}$

Table 1: Comparación entre 3-Sat y 3-DM

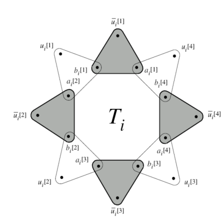
4.3 Definición de los Componentes

Para realizar la demostración contruiremos unas abstracciones llamadas componentes de las cuales hay 3 tipos:

1. Ternas de asignación
2. Ternas de satisfacción
3. Ternas de recolección

4.3.1 Definición de las Ternas de asignación

1. Para cada variable $u_i \in U$ se introduce una componente T_i
   (a) T_i depende del número de cláusulas de m en C
2. La estructura del T_i
   (a) Elementos internos: $a_i[j] \in X$, $b_i[j] \in Y$, $1 \leq j \leq m$
   No van a pertener a otras ternas de otro T_i.
   (b) Elementos externos:
   (c) $\neg u_i[j] \in W$, $1 \leq j \leq m$
   Pueden pertenecer a otras ternas.
3. El literal u_i en 3-SAT puede ser usado en varias cláusula, en el 3-DM debemos tener muchas m copias de u_i



$$T_i^t = \{(\overline{u}_i[j], a_i[j], b_i[j]) : 1 \leq j \leq m\}$$

$$T_i^f = \{ (u_i[j], a_i[j+1], b_i[j]) : 1 \leq j \leq m\} \cup \{(u_i[m], a_i[1], b_i[m]) : 1 \leq j \leq m\}$$

Figure 3: Ejemplo terna de asignación

4. Si ningún elemento interno de la componente T_i aparece en otra T_h con $h \neq i$
5. $M^{'}$ será matching con m elementos de T_i

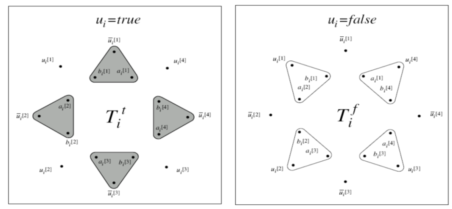


Figure 4: Ejeplo para cuando $u_i = False$ y $u_i = True$

6. Si $u_i = True$ se elegirá como $M^{'}$ las ternas en gris, dejando libre el resto para poder utilizarlas en la contrucción del resto de componentes

4.3.2 Definición ternas de satisfacción

Para cada cláusula $c_j \in C$ introducimos una componente C_j

La estructura será la siguiente:

- Elementos internos: $s_x[j] \in X$, $s_y[j] \in Y$, $1 \leq j \leq m$
- Elementos externos: $u_i[j], \neg u_i[j] \in W$,$1 \leq j \leq m$

Donde tendremos que $C_j = \{(u_i[j], s_x[j], s_y[j])$: si el literal $u_i \in c_j \cup (\neg u_i[j], s_x[j], s_y[j])$: si el literal $\neg u_i \in c_j\}$

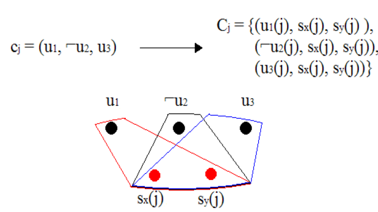


Figure 5: Ejemplo de la componente de satisfacción

Además se tiene que cumplir que: Para cualquier matching $M^{'} \subseteq M$ debe contener una terna de C_j para emparejar los elementos internos $s_x[j]$ y $s_y[j]$ bajo esta condición:

- $s_x[j]$ y $s_y[j]$ pueden estar emparejados, ssi, al menos uno de los literales u_i de c_j no ha sido emparejado en alguna componente de asignación T_i,$(T_i \cap M^{'})$
- Si tenemos una 3-SAT instancia factible, entonces las variables $s_x[j]$ y $s_y[j]$ pueden ser emparejadas.
- Si tenemos una 3-SAT instancia no factible, entonces las variables $s_x[j]$ y $s_y[j]$ no pueden ser emparejadas.

4.4 Componentes de recolección (Garbage collection)

Al existir muchos $u_i[j]$ que no se emparejan con componentes de asignación ni con los componentes de satisfacción Introducimos $m(n-1)$ nuevas varibles. $g_x[k] \in X, g_y \in Y : 1 \leq k \leq m(n-1)$ dado que hay $m \times n$ variables dse asignación u sin emparejar después de calcularr las tercetas de asignación.Además si todas las m cláusulas se satisfacen se han emparejado m variables, por lo tanto quedan sin emparejar $(m \times n) - m = m(n-1)$

Finalmente cada pareja $(g_x[k], g_y[k])$ se enlazará con una única variable $u_i[j]$ o $\neg u_i[j]$ que no estén en las tercetas que se han formado con las componentes anteriores:

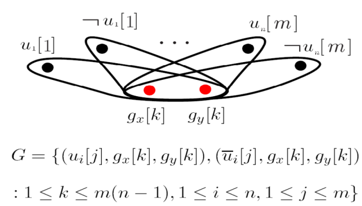


Figure 6: Ejemplo componente de recolección

$W = \{u_i\,[j]\,, \neg u_i\,[j] : 1 \leq i \leq n, 1 \leq j \leq m\}$

- $X = A \cup S_x \cup G_x \quad (2mn)$
  - $A = \{a_i\,[j] : 1 \leq i \leq n, 1 \leq j \leq m\}$
  - $S_x = \{s_x\,[j] : 1 \leq j \leq m\}$
  - $G_x = \{g_x\,[j] : 1 \leq j \leq m(n-1)\}$
- $Y = B \cap S_y \cup G_y \quad (2mn)$
  - $B = \{b_i\,[j] : 1 \leq i \leq n, 1 \leq j \leq m\}$
  - $S_y = \{s_y\,[j] : 1 \leq j \leq m\}$
  - $G_y = \{g_y\,[j] : 1 \leq j \leq m(n-1)\}$
- $M = \bigcup_{i=1}^{n} T_i \cup \bigcup_{j=1}^{m} C_j \cup G.\ 2mn + 3m + 2m^2 n(n-1)$

Significado	**Enumeración**
Cantidad de variables en $< U, C >$	n
Cantidad de cláusulas en $< U, C >$	m
Cantidad de componentes de **asignación** triple en M	$2mn$
Cantidad de componentes de **asignación*** triple en $M^{`}$	mn
Cantidad de componentes de **satisfacción** triple en M	$3m$
Cantidad de componentes de **satisfacción** triple en $M^{`}$	m
Cantidad de componentes **recolección** en M	$2m^2n(n-1)$
Cantidad de componentes **recolección** en $M^{`}$	$m(n-1)$
Cardinalidad del emparejamiento perfecto	$2mn$
Cardinalidad de M	$2mn = 3m = 2m^2n(n-1)$

4.5 Ejemplo de X, Y, W

W	X	Y
$u_1[1]$	$a_1[1]$	$b_1[1]$
$\overline{u}_1[1]$	$a_1[2]$	$b_1[2]$
$u_1[2]$	$a_2[1]$	$b_2[1]$
$\overline{u}_1[2]$	$a_2[2]$	$b_2[2]$
$u_2[1]$	$a_3[1]$	$b_3[1]$
$\overline{u}_2[1]$	$a_3[2]$	$b_3[2]$
$u_2[2]$	$a_4[1]$	$b_4[1]$
$\overline{u}_2[2]$	$a_4[2]$	$b_4[2]$
$u_3[1]$	$s_x[1]$	$s_y[1]$
$\overline{u}_3[1]$	$s_x[2]$	$s_y[2]$
$u_3[2]$	$g_x[1]$	$g_y[1]$
$\overline{u}_3[2]$	$g_x[2]$	$g_y[2]$
$u_4[1]$	$g_x[3]$	$g_y[3]$
$\overline{u}_4[1]$	$g_x[4]$	$g_y[4]$
$u_4[2]$	$g_x[5]$	$g_y[5]$
$\overline{u}_4[2]$	$g_x[6]$	$g_y[6]$

Figure 7: $n = 4$ y $m = 2$

Se puede observar que las ternas resultantes M son el producto cartesiano de $W \times X \times Y$

Esta es la forma de definir las ternas desde su definición en términos de una instancia (U, C) de 3-SAT además M se construye en tiempo polinomial

5 Demostración $\exists M^{'} \subseteq M \Leftrightarrow (U, C)$ es satisfacible

5.1 (U, C) satisfacible $\implies$ $M^{'} \subseteq M \Leftrightarrow$ es un matching

Sea $t : U \rightarrow \{T, F\}$ el dominio de los valores en U que satisfacen las cláusulas C Para ello se construye un matching $M^{'} \subseteq M$ donde:

1. Para cada cláusula $c_j \in C$: $Z_j \in \{u_i \neg u_i : 1 \leq i \leq n\} \cap c_j$

2. Construyendo $M' = \bigcup_{t(u_i)=T} T_i^t \cup \bigcup_{t(u_i)=F} T_i^f \cup \left(\bigcup_{j=1}^{m} \{(Z_j[j], S_x[j], S_y[j])\} \right)$

   Ahora definamos $G^{'}$ conjunto de m(n-1) ternas de G las incluyen todos los $g_x[k] \in X, g_y[k]\ inY$ y los $u_i[j], \neg u_i[j] \in W$ que no se han emparejado.

   Es fácil de verificar que siempre se puede construir un $G^{'}$ para que el resultado del conjunto $M^{'}$ sea una matching.

5.2 $M^{'} \subseteq M \Leftrightarrow$ es un matching $\implies$ (U, C) satisfacible $\implies$

Se ha visto que para cada $u_i \in U$ $M^{'}$ incluía exactamente m ternas de $T_i : T_i^t$ o T_i^f. Ahora, sea $t \rightarrow \{T, F\}$ donde $t(u_i) = T \leftrightarrow M^{'} \cap T_i = T_i^t$ donde t será una asignación correcta que satisface a C. Consideremos ahora una cláusula cualquiera $c_j \in C$, para cubrir todos los elementos internos de la componente C_j(de la componente de satisfacción) Necesitamos al menos una terna de C_j contenida en $M^{'}$, esta terna contiene un literal $c_j \in C$ que no estará en $M^{'} \cap T_i$. Además como $t(u_i) = T \leftrightarrow M^{'} \cap T_i = T_i^t$ entonces t satisface la cláusula c_j y como todas las cláusulas $c_j \in C$ se satisfacen (U, C) es satisfacible. Se cumplen los pasos previstos entonces 3-DM es NP-Completo