3-DIMENSIONAL MATCHING

E. T. S. INGENIERÍA INFORMÁTICA
Análisis de algoritmos

Ubay Diaz Machín (alu2238@etsii.ull.es)
José Ismael Galindo Bonilla (alu2621@etsii.ull.es)

Universidad de La Laguna

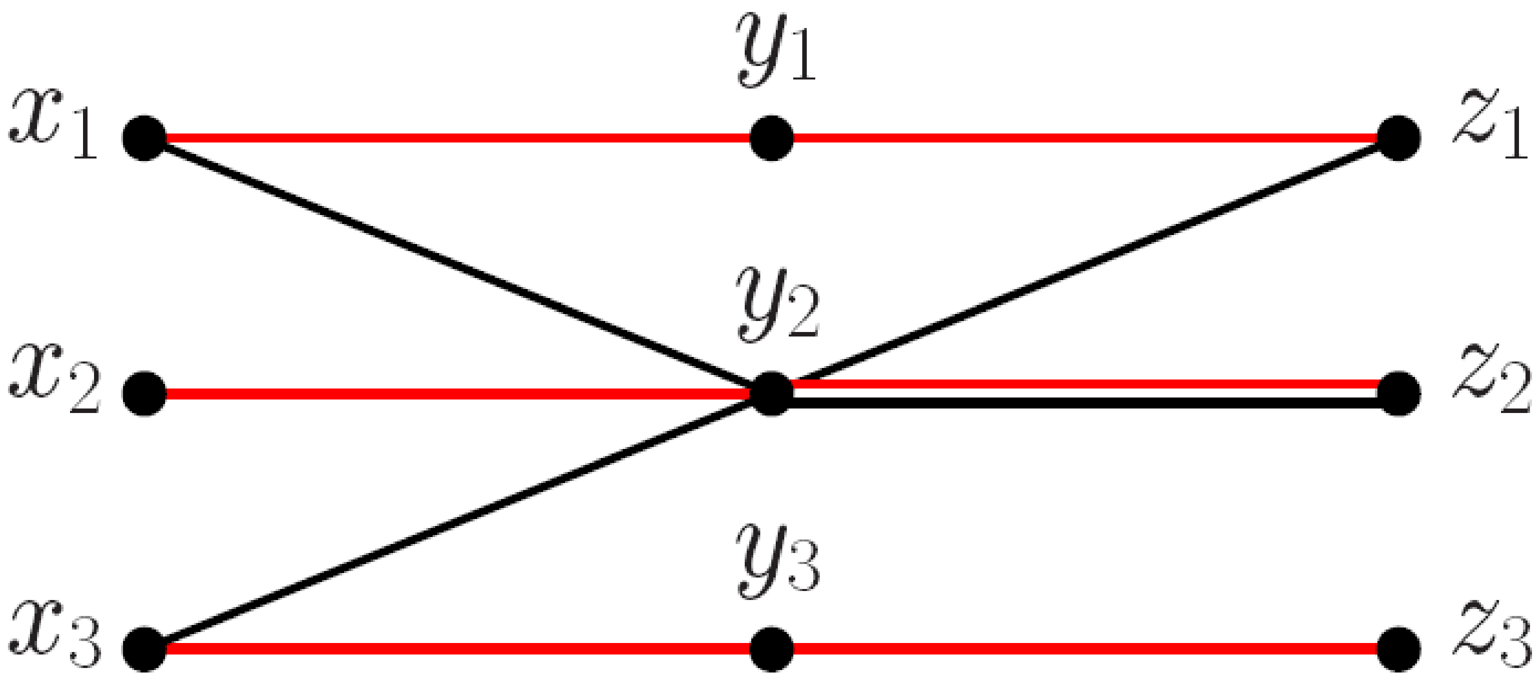
ÍNDICE

- Descripción del problema 3DM
- 3DM es NP Completo
  - 3DM es NP
  - Transformación polinomial del 3SAT al 3DM
  - Demostración.

DESCRIPCIÓN DEL PROBLEMA

- Instancia:
  - Un conjunto $M \subseteq W \times X \times Y$
    - $W \cap Y \cap X = \Phi$ (disjuntos)
    - $|W| = |X| = |Y| = q$
- Pregunta:
  - Hay un matching en M.
  - Es decir, un subconjunto $M' \subseteq M$ tal que:
    - $|\ M'\ | = q$
    - Todos los elementos $W \cup X \cup Y$ están en alguna terceta de M' sin repetir ninguno.

EJEMPLO



$$M = \{(x_1, y_1, z_1), (x_1, y_2, z_2), \\ (x_2, y_2, z_2), (x_3, y_3, z_3), (x_3, y_2, z_1)\}$$

3DM ES NP COMPLETO

- Pasos Principales:
  - Demostrar que 3DM es NP
  - Seleccionar un problema π' que sea NP Completo
  - Construir una transformación (f) π' α 3DM
  - Comprobar que la transformación f se hace en tiempo polinomial.

3DM ES NP

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

ALGORITMO π'

- 3 SATISFABILITY (3SAT)
  - Instancia:
    - Conjunto de ***m*** cláusulas C = {c_1, ..., cm}
      - | c_i | = 3 , 1 ≤ i ≤ m
    - Sobre un conjunto finito de ***n*** variables booleanas
      - U = {u_1, ..., u_n}
  - Pregunta:
    - Existe aluna asignación válida de U que satisfaga todas las cláusulas de C.

3SAT α 3DM

De una instancia del 3SAT (U,C) $\xrightarrow{\text{Construir}}$ Instancia del 3DM (W, X, Y, M)

$\downarrow$

$M' \subseteq M$

(U,C) es satisfacible $\Longleftrightarrow$ M' es un matching

3SAT α 3DM

- **Técnica para la demostración de NP Completitud:**
  - Diseño de Componentes
- **Notación:**

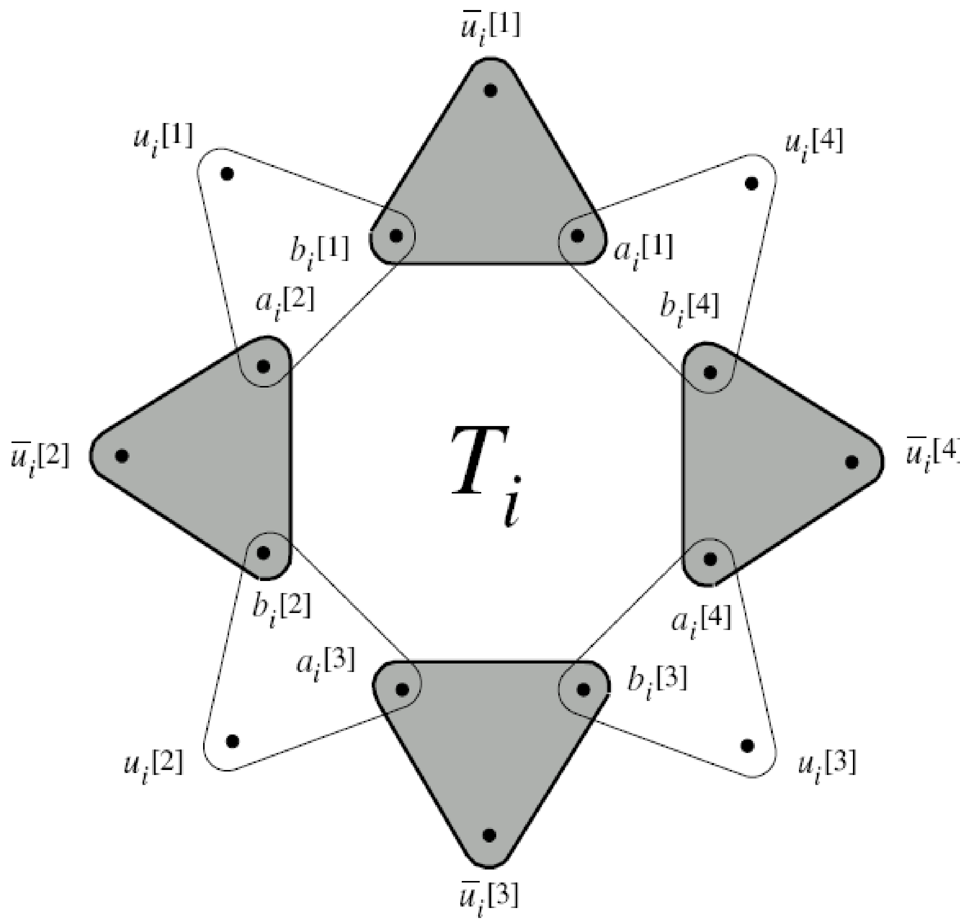
3SAT		3DM
Variables: $u_1, \ldots, u_n$	$\longrightarrow$	**Variables:** $u_1(j), a_i(j), b_i(j), s_x(j), g_y(j)$
Literales: $u_1, \neg u_1$	$\longrightarrow$	**Variables:** $u_1(j), \neg u_1(j)$
Cláusulas: $c_j = (u_1, \neg u_2, u_3)$	$\longrightarrow$	**Tercetas:** $C_j = \{(u_1(j), s_x(j), s_y(j)),$ $(\neg u_2(j), s_x(j), s_y(j)),$ $(u_3(j), s_x(j), s_y(j))\}$

- **La demostración se basa en la construcción de tres tipos de componentes:**
  - "Truth setting and fan out" (Tercetas de asignación)
  - "Satisfaction testting" (Tercetas de satisfacción)
  - "Garbage collection" (Tercetas de relleno)

3SAT α 3DM

- ***Componentes "Truth setting and fan out"***
  - Para cada variable ***u_i Є U*** se introduce una componente ***T_i***.
    - ***Ti*** depende del número de cláusulas *m* de *C*
  - Estructura de ***T_i***:
    - Elementos internos:
      - a_i [j] Є X, 1 ≤ j ≤ m ; b_i [j] Є Y, 1 ≤ j ≤ m
        - No van a pertenecer a otras tercetas de otro Ti
    - Elementos externos:
      - u_i [j] , ⌐u_i [j] Є W, 1 ≤ j ≤ m
        - Pueden pertenecer a otras tercetas
  - **El literal *ui* en 3SAT puede ser usado en varias cláusulas, en el 3DM debemos tener muchas m copias de *ui*.**

3SAT α 3DM

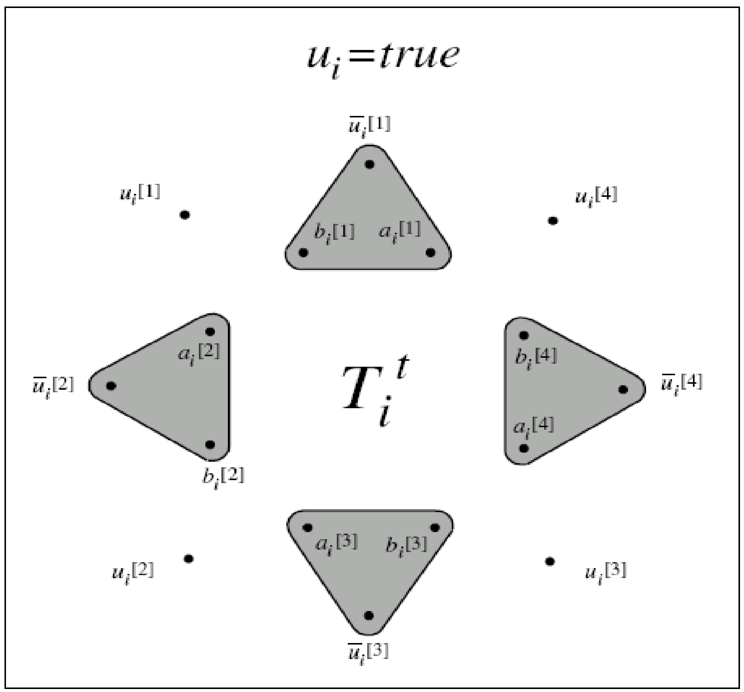


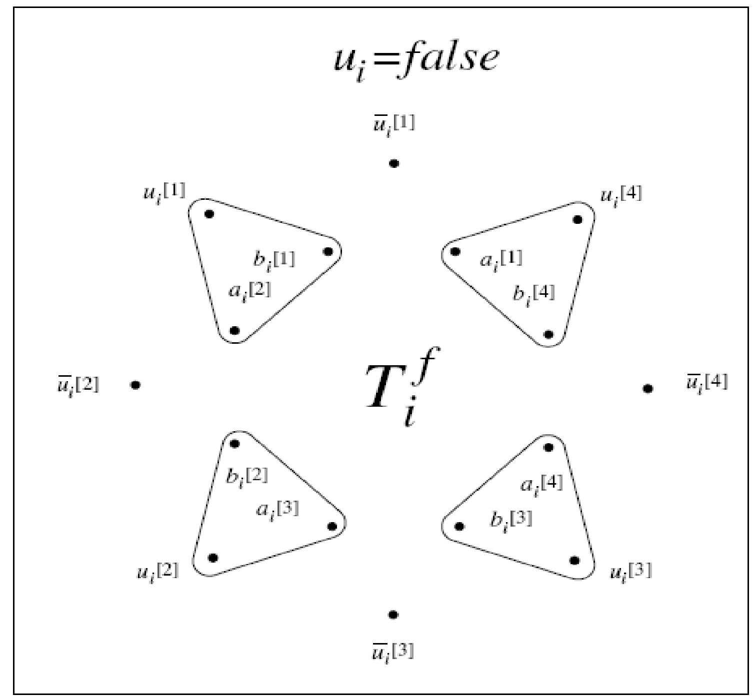
$$T_i^t = \{(\overline{u}_i[j], a_i[j], b_i[j]) : 1 \leq j \leq m\}$$

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

3SAT α 3DM

- Si ningún elemento interno de la componente ***Ti*** aparece en otra ***Th*** (i ≠ h)
    - M' será un matching con m elementos de Ti





- Si ui = true se elegirá como M' las tercetas grises, dejando libre el resto para poder utilizarlas en la construcción del resto de componentes.

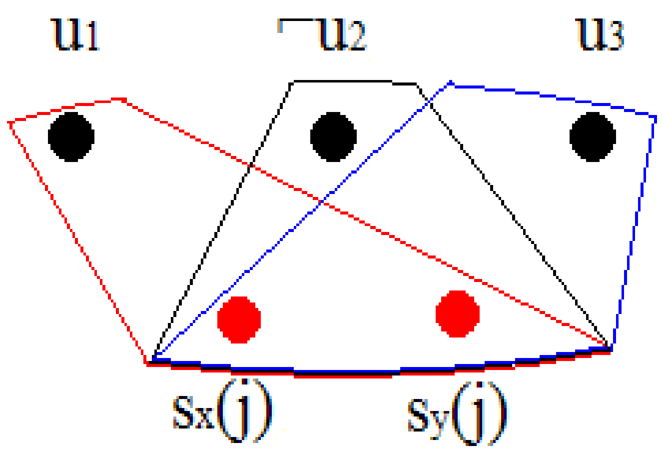
3SAT α 3DM

- ***<u>Componentes "Satisfaction testting"</u>***
  - Para cada cláusula c_j Є C introducimos una componente C_j.
  - Estructura:
    - Elementos Internos: $s_x[j]$ Є X, $s_y[j]$ Є Y : $1 \le j \le m$
    - Elementos externos: $u_i[j]$, $\neg u_i[j]$ Є W : $1 \le i \le n; 1 \le j \le m$
  - $C_j = \{(u_i[j], s_x[j], s_y[j])$: si el literal u_i Є $c_j\}$

    U $\{(\neg u_i[j], s_x[j], s_y[j])$: si el literal $\neg u_i$ Є $c_j\}$

3SAT α 3DM

$c_j = (u_1, \neg u_2, u_3) \longrightarrow C_j = \{(u_1(j), s_x(j), s_y(j)),$
$(\neg u_2(j), s_x(j), s_y(j)),$
$(u_3(j), s_x(j), s_y(j))\}$



3SAT α 3DM

- Cualquier matching *M' <u>c</u> M* debe contener una terceta de *Cj* para emparejar los elementos internos s_x [j] y s_y [j]:

  - S_x [j] y Sy [j] pueden ser emparejados, sí sólo sí, al menos uno de los literales (u_i) de c_j no ha sido emparejado en alguna componente “Truth seeting” T_i ($T_i \cap M'$)

  - Si tenemos una 3SAT-Instancia satisfacible, entonces las variables Sx[j] y Sy[j] pueden ser emparejadas

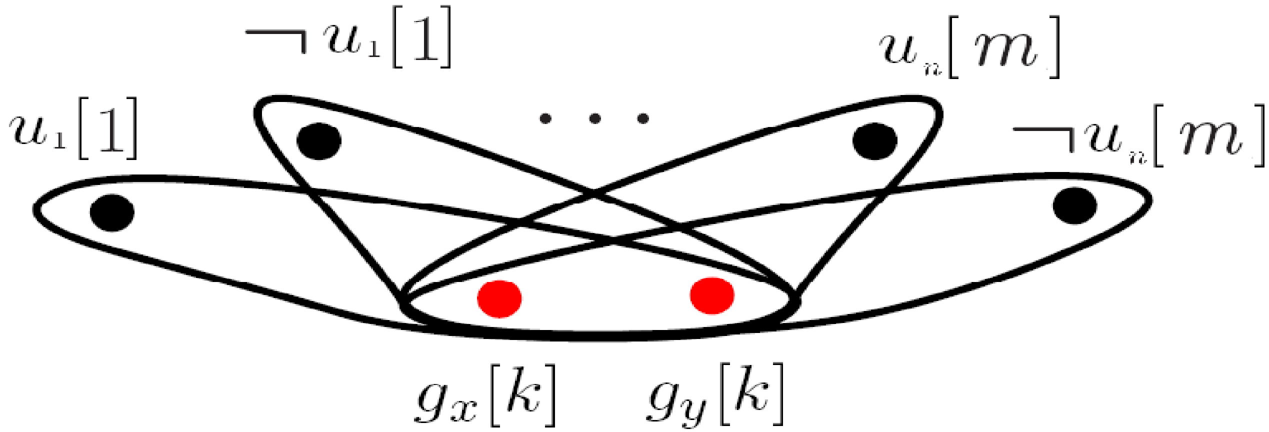
  - Si tenemos una 3SAT-Instancia no satisfacible, entonces las variables Sx[j] y Sy[j] no pueden ser emparejadas.

3SAT α 3DM

- ***Componentes "Garbage collection"***
  - Hay muchos $u_i[j]$ que no se emparejan con componentes "truth setting" ni con componentes "satisfaction testing"
  - Introducimos m (n -1) variables nuevas:
    - $g_x[k] \in X$, $g_y[k] \in Y$: $1 \leq k \leq m(n-1)$
  - ¿ Por qué m (n-1) variables?
    - Hay $m \times n$ variables u sin emparejar después de calcular las tercetas de asignación.
    - Si todas las m cláusulas se satisfacen se han emparejado m variables.
    - Finalmente quedan sin emparejar $(m \times n) - m = m(n-1)$

3SAT α 3DM

– Cada pareja (gx [k], gy [k]) se enlazará con una única variable ui [j] o ⌐ui [j] que no estén en las tercetas que se han formado con las componentes anteriores:



$$G = \{(u_i[j], g_x[k], g_y[k]), (\overline{u}_i[j], g_x[k], g_y[k])$$

$$: 1 \leq k \leq m(n-1), 1 \leq i \leq n, 1 \leq j \leq m\}$$

3SAT α 3DM

- Resumiendo:

$$W = \{(u_i[j], \overline{u}_i[j] : 1 \leq i \leq n, 1 \leq j \leq m\} \quad (2mn)$$

$$X = A \cup S_X \cup G_X \quad (2mn)$$

$$A = \{a_i[j] : 1 \leq i \leq n, 1 \leq j \leq m\}$$
$$S_X = \{s_x[j] : 1 \leq j \leq m\}$$
$$G_X = \{g_x[j] : 1 \leq j \leq m(n-1)\}$$

$$Y = B \cup S_Y \cup G_Y \quad (2mn)$$

$$B = \{b_i[j] : 1 \leq i \leq n, 1 \leq j \leq m\}$$
$$S_Y = \{s_y[j] : 1 \leq j \leq m\}$$
$$G_Y = \{g_y[j] : 1 \leq j \leq m(n-1)\}.$$

$$M = \left(\bigcup_{i=1}^{n} T_i\right) \cup \left(\bigcup_{j=1}^{m} C_j\right) \cup G. \quad (2mn+3m+2m^2n(n-1))$$

3SAT α 3DM

Significato	Enumerazione
Numero di variabili in $\langle U, C\rangle$	n
Numero di clausole $\langle U, C\rangle$	m
Componente "Truth setting and fan out": triple in M	$2mn$
Componente "Truth setting and fan out": triple in M'	mn
Componente "Satisfaction testing": triple in M	$3m$
Componente "Satisfaction testing": triple in M'	m
Componente "Garbage collection": triple in M	$2m^2n(n-1)$
Componente "Garbage collection": triple in M'	$m(n-1)$
Dimensione del perfect matching: $\|M'\| = q = \|W\| = \|X\| = \|Y\|$	$2mn$
Dimensione M	$2mn + 3m + 2m^2n(n-1)$

3SAT α 3DM

- **Ejemplo de X, Y, W para n = 4 y m=2**

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

3SAT α 3DM

- Se ha observado que las tercetas resultantes M son el producto cartesiano de W x X x Y

- Esta forma de definir las tercetas:
  - Desde su definición en términos de una instancia (U,C) del 3SAT
  - M se construye en tiempo polinomial

DEMOSTRACIÓN

- Para completar la demostración de NP Completitud falta por demostrar:

M contiene un matching M' $\Longleftrightarrow$ (U,C) es satisfacible

DEMOSTRACIÓN

(U,C) es satisfacible $\longrightarrow$ **M' $\underline{c}$ M es un matching**

- Sea t: U → { T, F } el dominio de valores para U que satisface las cláusulas C.

- Se construye un matching M' $\underline{c}$ M del modo siguiente:
  - Para cada cláusula cj ϵ C:
    - Z_j ϵ $\{u_i, \neg u_i : 1 \leq i \leq n\} \cap c_j$
      - Literales con asignación verdadera.
      - Debe de existir al menos uno, ya que t satisface a cj.

DEMOSTRACIÓN

- Se construye la M':

$$M' = \left(\bigcup_{t(u_i)=T} T_i^t \right) \cup \left(\bigcup_{t(u_i)=F} T_i^f \right) \cup \left(\bigcup_{j=1}^{m} \{(z_j[j], s_x[j], s_y[j])\} \right) \cup G'$$

- G': conjunto de m(n-1) tercetas de G que incluyen:
  - Todos los $g_x[k] \in X$, $g_y[k] \in Y$
  - 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 un matching.

DEMOSTRACIÓN

M' $\underline{c}$ M es un matching $\longrightarrow$ **(U,C) es satisfacible**

- Se ha visto que para cada ui є U, M' incluía exactamente m tercetas de T_i: T^t_i o T^f_i.
- Sea t: U → { T, F } donde $t(u_i) = T \leftrightarrow M' \cap T_i = T^t_i$
  - t será una asignación correcta que satisface C.
- Consideremos una cláusula arbitraria c_j є C
  - Para cubrir los elementos internos de la componente C_j (de la componente de testting)
    - Se necesita al menos una terceta de C_j contenida en M'.
    - Esta terceta contiene un literal de c_j є C, que no estará en $M' \cap T_i$

DEMOSTRACIÓN

- Como $t(u_i) = T \leftrightarrow M' \cap T_i = T^t_i$
  - Entonces t satisface la cláusula c_j

- Si todas las cláusulas $c_j \in C$ se satisfacen
  - (U,C) es satisfacible

3-Dimensional Matching
es
NP COMPLETO