APELLIDOS: Hidalgo Herrera, Vílchez Sánchez, Herrezuelo Barkova GRUPO: 1

NOMBRE: Lucas, Juan Carlos, Antonio Valerio NIF: 26512990Q 14276362D 79383156J Nº HOJAS: 15

LMD

Doble Grado en Ingeniería Informática y Matemáticas

14 de junio de 2024

1. Encuentre razonadamente una relación de recurrencia que para todo número natural n proporcione el número exacto a_n de cadenas de bits de longitud n que contenga al menos un par de ceros consecutivos. Seguidamente:

   a) Dé razonadamente las condiciones iniciales que definirán el problema de recurrencia.

   b) Calcule razonadamente a_{500}.

2. Para cualquier conjunto $\Gamma \cup \{\alpha, \beta, \gamma\}$ de fórmulas proposicionales, considere la siguiente igualdad:

$$\text{Con}(\Gamma, \alpha \to (\beta \to (\alpha \to \gamma))) = \text{Con}(\Gamma, (\beta \to \alpha) \to (\beta \to \gamma))$$

   Si es cierta, dé una demostración y de no serlo, demuéstrelo con un contraejemplo.

3. Demuestre que para todo conjunto de fórmulas proposicionales Γ se cumple la igualdad:

$$Con(\Gamma) = \bigcup_{\substack{\Gamma_f \subseteq \Gamma \\ \Gamma_f \text{ finito}}} Con(\Gamma_f)$$

4. Considere la función booleana dada como sigue:

$$f(a, b, c, d) = \sum m(0, 1, 4, 5, 6, 7, 9, 11, 15) + \sum d(10, 14)$$

   y haga lo siguiente:

   a) Dé razonadamente una expresión minimal de la función a condición de ser SOP.

   b) Dé razonadamente una expresión minimal de la función a condición de ser POS.

   c) Elija justificadamente la expresión de menor coste entre la SOP y la POS encontradas en los apartados anteriores.

   Para este ejercicio empleará exclusivamente el algoritmo de Quine-McCluskey y razonará, escueta pero suficientemente, los pasos en la aplicación de dicho algoritmo.

5. Considere las fórmulas de cierto lenguaje de primer orden:

   - $\varphi_0 \equiv \forall x(r(x, x) \to \exists y r(x, y))$
   - $\varphi_1 \equiv \forall x \forall y(r(x, y) \to r(y, x))$
   - $\varphi_2 \equiv \forall x(\neg r(y, x))$
   - $\varphi_3 \equiv \exists x(\neg r(x, x))$

   y diga razonadamente si son ciertas o no cada una de las siguintes afirmaciones:

   a) $\varphi_0, \varphi_1, \varphi_2 \models \varphi_3$

   b) $\varphi_0, \varphi_2, \varphi_3 \models \varphi_1$

6. De ser posible, construya razonadamente un grafo G (sin lazos ni lados paralelos) que teniendo 7 vértices: dos sean de grado 2, uno de grado 3, tres de grado 4 y no haya ninguno de grado 1.

1. Encuentre razonadamente una relación de recurrencia que para todo número natural n proporcione el número exacto a_n de cadenas de bits de longitud n que contenga al menos un par de ceros consecutivos. Seguidamente:

   a) Dé razonadamente las condiciones iniciales que definirán el problema de recurrencia.

   b) Calcule razonadamente a_{500}.

Antes de resolver concretamente alguno de los apartados, proporcionaremos el desarrollo deductivo para obtener la expresión de la recurrencia que modela el problema.

Denotaremos por A_n al conjunto de combinaciones de n bits que cumplen la propiedad del enunciado, es decir, cada elemento de A_n contiene al menos un par de ceros consecutivos. Cabe aclarar que a_n es el cardinal de A_n para todo n número natural.

Sea B_n el conjunto de combinaciones de n bits que no cumplen la propiedad del enunciado para todo n número natural, denotaremos por β_n al cardinal de este conjunto.

Por tanto, es fácil ver que la suma de la cardinalidad de ambos conjuntos es 2^n para todo n número natural; este hecho se da por conocido.

Nosotros sabemos que para cada elemento de B_n denotado por b_n distinguimos por casos:

- Si el bit menos significativo (resp más significativo) de b_{n+1} es 1; entonces, como b_n no cumple la propiedad no podrá tener dos ceros consecutivos; luego podremos considerar la cardinalidad del conjunto B_n.

- Si el bit menos significativo (resp más significativo) de b_{n+1} es 0, el bit anterior deberá ser un 1 pues en caso contrario b_{n+1} ya no pertenecería a B_{n+1}. Luego, la cadena de $n-2$ bits de b_{n+1} podrá ser cualquier cadena de longitud $n-2$ que cumpla la propiedad, es decir, hay $|B_{n-2}|$ cadenas.

Por tanto, llegamos al siguiente sistema de ecuaciones de diferencias:

$$\begin{cases} 2^n = a_n + \beta_n \;;\; a_n = 2^n - \beta_n \\ \beta_n = \beta_{n-1} + \beta_{n-2} \end{cases}$$

Estamos ya en disposición de resolver los apartados del ejercicio.

a) Gracias a la resolución del sistema en recurrencias tal y como hemos visto en clase llegamos a que nuestra recurrencia pasaría a ser un problema de recurrencias con los siguientes términos

$\beta_0 = 1$

$\beta_1 = 2$

donde b_0 denota la cadena con 0 bits y b_1 denota la cadena con 1 bit. En ambos caso es claro que de todas las combinaciones posibles con 1 bit o 0 bits ninguna cumple la propiedad luego todas pertenecerían a B_1 y B_0 respectivamente.

b) Mediante la resolución del sistema en recurrencias con un sistema iterativo según un programa aparecido en el zip entregado

Obtenemos que $a_{500} = 3'27339 \cdot 10^{150}$

2. Para cualquier conjunto $\Gamma \cup \{\alpha, \beta, \gamma\}$ de fórmulas proposicionales, considere la siguiente igualdad:

$$\mathrm{Con}(\Gamma, \alpha \to (\beta \to (\alpha \to \gamma))) = \mathrm{Con}(\Gamma, (\beta \to \alpha) \to (\beta \to \gamma))$$

Si es cierta, dé una demostración y de no serlo, demuéstrelo con un contraejemplo.

Afirmamos que esta igualdad es cierta y para ello vamos a usar los siguientes resultados que aquí enunciamos.

Corolario 3.3.5. Para cualquier conjunto de fórmulas $\Gamma \cup \{\alpha, \beta\}$, $\mathrm{Con}(\Gamma, \alpha) = \mathrm{Con}(\Gamma, \beta)$ siempre que $\alpha \equiv \beta$.

Luego, usando este corolario demostrar la certeza de la igualdad del enunciado bastará con probar que las fórmulas siguientes son lógicamente equivalentes:

$$\gamma_0 = \alpha \to (\beta \to (\alpha \to \gamma))$$

$$\gamma_1 = (\beta \to \alpha) \to (\beta \to \gamma)$$

Para ello hemos usado el siguiente resultado:

Teorema 3.4.6 Para toda fórmula φ del lenguaje existe una fórmula φ_{fnc} cumpliendo

1. φ_{fnc} está en forma normal conjuntiva
2. $\varphi \equiv \varphi_{fnc}$

Por tanto, nuestro problema ahora se reduce a comprobar que $\gamma_{0_{fnc}} \equiv \gamma_{1_{fnc}}$ y para ello calculamos la forma normal conjuntiva de cada una de ellas usando el método proporcionado en la siguiente

- Tener en cuenta el Teorema 3.4.3
- Usar que

$$\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$$

$$\varphi \to \psi \equiv \neg\varphi \vee \psi$$

y en ese orden

- Usar repetidamente que:

$$\neg\neg\varphi \equiv \varphi$$

las leyes de De Morgan que establecen que:

$$\neg(\varphi \vee \psi) = \neg\varphi \wedge \neg\psi$$

$$\neg(\varphi \wedge \psi) = \neg\varphi \vee \neg\psi$$

· Absorque

$$\varphi \vee (\Phi \wedge \psi) = (\varphi \vee \Phi) \wedge (\varphi \vee \psi)$$

Luego obtenemos que:

$$\gamma_0 = \alpha \longrightarrow (\beta \to (\alpha \to \gamma)) = \neg\alpha \vee (\beta \to (\alpha \to \gamma)) = \neg\alpha \vee (\neg\beta \vee (\alpha \to \gamma))$$

$$= \neg\alpha \vee (\neg\beta \vee (\neg\alpha \vee \gamma)) = \neg\alpha \vee (\neg\beta \vee \neg\alpha \vee \gamma) = \neg\beta \vee \neg\alpha \vee \gamma = \gamma_{0_{fuc}}$$

$$\gamma_1 = (\beta \to \alpha) \longrightarrow (\beta \to \gamma) = \neg(\beta \to \alpha) \vee (\beta \to \gamma) = \neg(\neg\beta \vee \alpha) \vee (\neg\beta \vee \gamma)$$

$$= (\beta \wedge \neg\alpha) \vee (\neg\beta \vee \gamma) = (\beta \vee (\neg\beta \vee \gamma)) \wedge (\neg\alpha \vee (\neg\beta \vee \gamma)) = (\beta \vee \neg\beta \vee \gamma) \wedge (\neg\alpha \vee \neg\beta \vee \gamma)$$

$$\overset{(1)}{=} \neg\alpha \vee \neg\beta \vee \gamma = \gamma_{1_{fuc}}$$

Luego, como $\gamma_{0_{fuc}} = \gamma_{1_{fuc}}$ tenemos que la igualdad inicial es cierta

En (1) hemos usado que $(\beta \vee \neg\beta \vee \gamma)$ es tautología por ser $(\beta \vee \neg\beta)$ tautología

3. Demuestre que para todo conjunto de fórmulas proposicionales Γ se cumple la igualdad:

$$Con(\Gamma) = \bigcup_{\substack{\Gamma_f \subseteq \Gamma \\ \Gamma_f \text{ finito}}} Con(\Gamma_f)$$

Procedemos por doble inclusión. Notaremos por $\bigcup Con(\Gamma_f)$ al conjunto $\bigcup_{\substack{\Gamma_f \subseteq \Gamma \\ \Gamma_f \text{finito}}} Con(\Gamma_f)$

Veamos que $\bigcup Con(\Gamma_f) \subseteq Con(\Gamma)$, para ello, sea γ una fórmula cualquiera tal que $\gamma \in \bigcup Con(\Gamma_f)$ entonces existe un Γ_f finito tal que $\gamma \in Con(\Gamma_f)$ luego, como $\Gamma_f \subseteq \Gamma$, aplicando la propiedad (2), $Con(\Gamma_f) \subseteq Con(\Gamma)$ luego $\gamma \in Con(\Gamma)$ obteniendo la primera inclusión

(2) Para cualesquiera conjuntos de fórmulas Γ y Δ se cumple:

Si $\Gamma \subseteq \Delta$, entonces $Con(\Gamma) \subseteq Con(\Delta)$

Veamos ahora la otra inclusión, es decir, $Con(\Gamma) \subset \bigcup Con(\Gamma_f)$ Sea α una fórmula cualquiera tal que $\alpha \in Con(\Gamma)$ entonces $\Gamma \models \alpha$ luego para toda valoración v fija pero arbitraria tal que $v(\gamma) = 1$ para toda $\gamma \in \Gamma$; luego, en particular, $v(\gamma) = 1$ para toda $\gamma \in \Gamma_f$, es decir, $\Gamma \models \Gamma_f$ por tanto $\Gamma \models \Gamma_f \wedge \alpha$

4. Considere la función booleana dada como sigue:

$$f(a,b,c,d) = \sum m(0,1,4,5,6,7,9,11,15) + \sum d(10,14)$$

y haga lo siguiente:

a) Dé razonadamente una expresión minimal de la función a condición de ser SOP.

b) Dé razonadamente una expresión minimal de la función a condición de ser POS.

c) Elija justificadamente la expresión de menor coste entre la SOP y la POS encontradas en los apartados anteriores.

Nos encontramos ante una función de conmutación de cuatro variables con las siguientes características.

- Minterms {0, 1, 4, 5, 6, 7, 9, 11, 15}
- Términos no importa {10, 14}

a) Con lo ya expuesto, queda claro que, para obtener una expresión de la función de conmutación como suma de productos, podemos usar el algoritmo de Quine-McClusky.

Para ello seguiremos los pasos enumerados desde 1 hasta 4

1 Generación de los implicantes primos; este proceso, se encuentra reflejado en la siguiente tabla donde en la primera columna colocamos los implicantes agrupados por el número de unos de su codificación en binario. Para cada una de las columnas siguientes obtenemos los implicantes primos comparando cada uno de los grupos de la columna anterior con su siguiente. Una vez que un implicante obtenido no puede ser comparado, vemos que ese implicante es primo (marcado por *) y será considerado para el siguiente paso:

columna 1	columna 2	columna 3
0 0000 ✓	{0,1} 000- ✓	{0,1,4,5} 0-0- *
1 0001 ✓	{0,4} 0-00 ✓	{0,4,1,5} 0-0-
4 0100 ✓	{1,5} 0-01 ✓	{4,5,6,7} 01-- *
5 0101 ✓	{1,9} -001 *	{4,6,5,7} 01--
6 0110 ✓	{4,5} 010- ✓	{6,7,14,15} -11- *
9 1001 ✓	{4,6} 01-0 ✓	{6,14,7,15} -11-
10 1010	{5,7} 01-1 ✓	{10,11,14,15} 1-1- *
7 0111 ✓	{6,7} 011- ✓	{10,14,11,15} 1-1-
11 1011 ✓	{6,14} -110 ✓	
14 1110 ✓	{9,11} 10-1 *	
15 1111	{10,11} 101- ✓	
	{10,14} 1-10 ✓	
	{7,15} -111 ✓	
	{11,15} 1-11 ✓	
	{14,15} 111- ✓	

Como no hay más posibilidades de combinación, los implicantes primos son:

{{0,1,4,5}, {4,5,6,7}, {1,9}, {6,7,14,15}, {10,11,14,15}, {9,11}}

2 Generación de la tabla de implicantes primos; esta tabla simplemente recoge los minterms que cada implicante primo cubre para facilitar la resolución de la misma. Marcaremos por * a los implicantes primos esenciales en caso de que los haya, estos son esenciales siempre y cuando ellos, y sólo ellos, cubran un minterm en concreto. La tabla es la siguiente

implicante		0	1	4	5	6	7	9	11	15
* {0,1,4,5}	0_0_	O	O	O	O					
{4,5,6,7}	01__			O	O	O	O			
{1,9}	_001		O					O		
{6,7,14,15}	_11_					O	O			O
{10,11,14,15}	1_1_								O	O
{9,11}	10_1							O	O	
		O								

Al estudiarla encontramos que al objeto de cubrir todos los minterms el implicante primo e_1 = {0,1,4,5} es esencial.

3 Reducción de la tabla de implicantes primos; eliminando e_1 vemos que la tabla queda como sigue. Con colores se refleja el proceso seguido estrictamente en ese orden

implicante		6	7	9	11	15
~~{4,5,6,7}~~	~~01__~~	O	O			
~~{1,9}~~	~~_001~~			O		
{6,7,14,15}	_11_	O	O			O
{10,11,14,15}	1_1_				O	O
{9,11}	10_1			O	O	

— El implicante primo {9,11} domina por filas al implicante primo {1,9}

— La columna del minterm 7 domina a la columna del minterm 6.

— La fila del implicante primo {6,7,14,15} domina por filas a la del implicante primo {4,5,6,7}

Tras aplicar los pasos anteriores llegamos a la situación reflejada en la tabla siguiente.

implicante	6	9	11	15
✱ ~~{6,7,14,15} -11-~~	O			O
{10,11,14,15} 1-1-			O	O
✱ ~~{9,11} 10-1~~		O	O	
	O			

✱ Como el minterm 6 sólo es cubierto por el implicante primo {6,7,14,15}, este último es esencial. A este implicante primo lo denotaremos por $e_2 = \{6,7,14,15\}$

✱ Tras el paso ✱ llegamos a la misma situación con el minterm 9 y el implicante primo {9,11}, que denotaremos por e_3 al ser esencial.

4. Resolución de la tabla de implicantes primos; al realizar estos pasos terminamos con la tabla vacía luego la trascripción de cada implicante primo es la siguiente.

pseudominterm	implicante	patrón	representa
e_1	{0,1,4,5}	0-0-	$\bar{a}\bar{c}$
e_2	{6,7,14,15}	-11-	bc
e_3	{9,11}	10-1	$a\bar{b}d$

Por tanto, la función de conmutación expresada como suma de productos (SOP) es

$$f(a,b,c,d) = \bar{a}\bar{c} + bc + a\bar{b}d$$

b) Como el álgebra de Boole del problema es $\mathcal{B}_2 = \langle B_2, +, \cdot, \bar{\ }, 0, 1 \rangle$, el álgebra de Boole de los números binarios, y $f : \mathcal{B}^4 \longrightarrow \mathcal{B}$ la función de conmutación, obtenemos que deberemos estudiar la siguiente función de conmutación para obtener dicha expresión como producto de sumas

$$f(a,b,c,d) = \prod M(2,3,8,12,13) \cdot \prod d(10,14)$$

Para ello, aplicaremos de nuevo el algoritmo de Quine-McClusney, pero esta vez de forma más esquemática, siguiendo la misma notación que en el apartado a).

1 Generación de los implicantes primos.

columna 1	columna 2	columna 3
2 0010 ✓	{2,3} 001– *	{8,10,12,14} 1––0 *
8 1000 ✓	{2,10} –010 *	{8,12,10,14} 1––0
3 0011 ✓	{8,10} 10–0 ✓	
10 1010 ✓	{8,12} 1–00 ✓	
12 1100 ✓	{10,14} 1–10 ✓	
13 1101	{12,13} 110– *	
14 1110	{12,14} 11–0 ✓	

Como no hay más posibilidades de combinación, los implicantes primos son:

$$\{\{2,3\}, \{2,10\}, \{8,10,12,14\}, \{12,13\}\}$$

2 Construcción de la tabla de implicantes primos; la tabla sería la siguiente:

implicantes	2	3	8	12	13
* {2,3} 001–	O	O			
{2,10} –010	O				
* {8,10,12,14} –100			O	O	
* {12,13} 110–				O	O
		O	O		O

Al estudiarla encontramos que al objeto de cubrir todos los minterms los implicantes primos $e_1 = \{2,3\}$, $e_2 = \{8,10,12,14\}$ y $e_3 = \{12,13\}$ son esenciales.

3. Reducción de la tabla de implicantes primos; teniendo en cuenta los implicantes primos esenciales obtenemos una tabla vacía; por tanto, ya estaría reducida

4 Resolución de la tabla de implicantes primos; al obtener la tabla vacía vemos que los implicantes primos tienen la siguiente trascripción:

Pseudobinario	implicante	patrón	representa
e_1	$\{2,3\}$	001-	$a+b+\bar{c}$
e_2	$\{8,10,12,14\}$	1--0	$\bar{a}+d$
e_3	$\{12,13\}$	110-	$\bar{a}+\bar{b}+c$

Por tanto, la función de conmutación expresada como producto de sumas (POS) es:

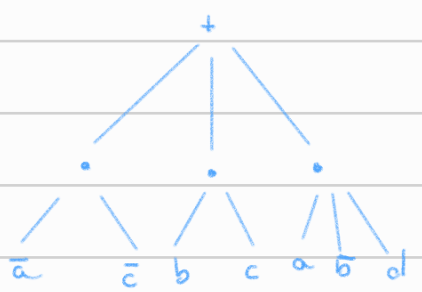
$$f(a,b,c,d) = (a+b+\bar{c})(\bar{a}+d)(\bar{a}+\bar{b}+c)$$

c) Para este apartado debemos hacer una comparación del análisis de costes de la función expresada como suma de productos y como producto de sumas:

SOP $\rightarrow$ $f(a,b,c,d) = \bar{a}\bar{c} + bc + a\bar{b}d$

POS $\rightarrow$ $f(a,b,c,d) = (a+b+\bar{c})(\bar{a}+d)(\bar{a}+\bar{b}+c)$

Para el primer caso, el análisis se refleja en el siguiente árbol:



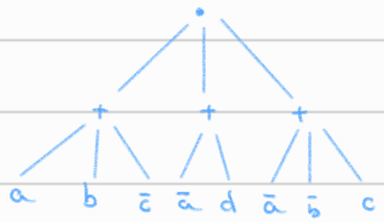
$\rightarrow$ or: 1

$\rightarrow$ and: 3

$\rightarrow$ entradas: 10

Coste total: 14

Para el segundo caso, el análisis se refleja en el siguiente árbol:



-> or : 3
-> and : 1
-> Entradas : 11

Coste total : 15

Luego expresando la función como suma de productos obtenemos un menor coste y por ende, es nuestra expresión elegida.

6. De ser posible, construya razonadamente un grafo G (sin lazos ni lados paralelos) que teniendo 7 vértices: dos sean de grado 2, uno de grado 3, tres de grado 4 y no haya ninguno de grado 1.

El problema se reduce a encontrar una sucesión finita de números que cumpla las propiedades del problema. La sucesión del problema es la siguiente

5 4 4 4 3 2 2

Veamos que esta sucesión finita es una sucesión gráfica y apliquemos el algoritmo de Havel-Hakimi a la inversa para construir el grafo pedido.

Como primera comprobación, aunque no es necesario, veremos que la suma de los elementos es par pues, en caso de no serlo, la sucesión de partida no sería una sucesión gráfica.

$5+4+4+4+3+2+2=24$

Luego seguimos con el estudio.

Veremos que nuestra sucesión es gráfica aplicando el Teorema de Havel-Hakimi:

5	4	4	4	3	2	2	
	3	3	3	2	1	2	Reordenamos
	3	3	3	2	2	1	
		2	2	1	2	1	Reordenamos
		2	2	2	1	1	
			1	1	1	1	
				0	1	1	Reordenamos
				1	1	0	
					0	0	

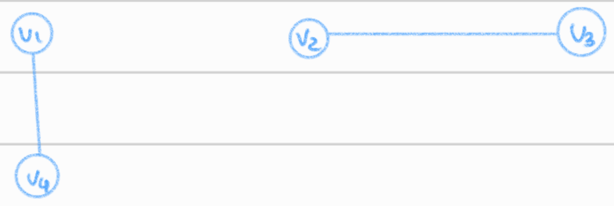
Como hemos llegado a la sucesión 0 0 y sabemos que esta sucesión es gráfica vemos que nuestra sucesión de partida es una sucesión gráfica.

Vamos ahora a seguir el proceso a la inversa para construir el grafo de dicha sucesión gráfica

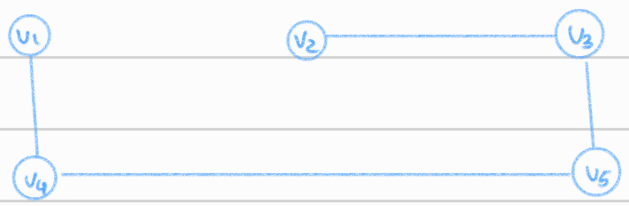
Partimos de un grafo con dos vértices de grado cero (0 0)

Pasamos a la sucesión (1 1 0) luego, debemos añadir un vértice v_3, de grado 1 y conectar v_2 con algún otro vértice, por ejemplo, lo siguiente:

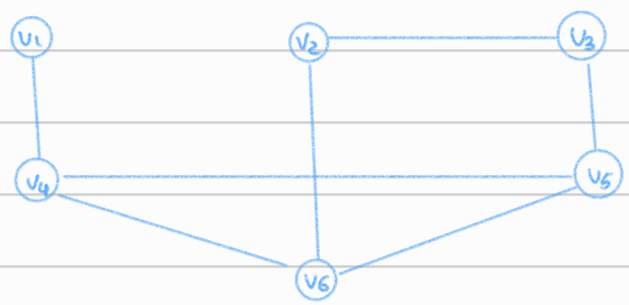
Pasamos ahora a la sucesión (1 1 1 1) luego, debemos añadir un vértice v_4 de grado 1 y conectar v_1 con otro vértice:



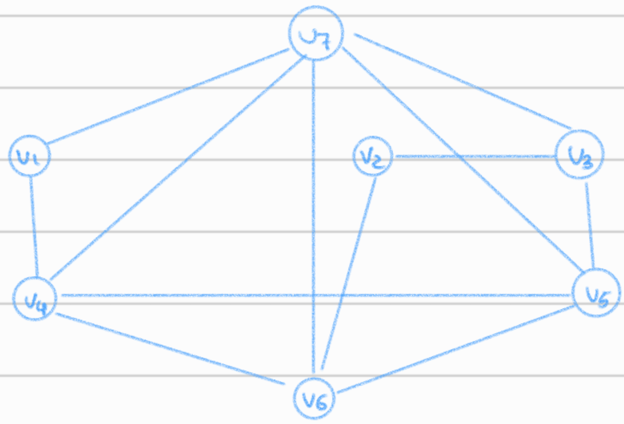
Procedemos con la sucesión (2 2 2 1 1) añadiendo un vértice v_5 de grado 2 uniéndolo de manera que se cumpla el grado de los vértices



Procedemos con la sucesión (3 3 3 2 2 1) añadiendo un vértice v_6 de grado 3 uniéndolo de manera que se cumpla el grado de los vértices



Por último, la última sucesión, la de partida, (5 4 4 4 3 2 2) añadiendo un vértice v_7 de grado 5 donde se cumpla el grado de los vértices.



Veamos que si ponemos un número mayor que 5 (pues sólo nos pide que haya 3 vértices de grado 4) obtenemos una sucesión que no es una sucesión gráfica.
Si sustituimos el 5 por un número par obtenemos una suma de la sucesión impar luego no sería una sucesión gráfica.
Si sustituimos el 5 por un número impar, bastará probar que si lo sustituimos por 7 nuestra sucesión no sería una sucesión gráfica.

7 4 4 4 3 2 2
3 3 3 2 1 1
2 2 1 1 1
1 0 1 1 Reordenamos
1 1 1 0
0 1 0 Reordenamos
1 0 0
-1 0

Luego he llegado a una sucesión que no es gráfica pues no existe ningún grafo en el que alguno de sus vértices tenga grado negativo. Esto es válido para cualquier número impar superior a 5 gracias al orden de los números naturales y a que la suma es un operador estrictamente creciente en ω.
Es claro que, si sustituimos el 5 por un número menor que él incumplimos las restricciones del enunciado.