Cuadrado Mágico 3x3

Lógica para Ciencias de la Computación

Nicolas Otero
Santiago Alvarez Barbosa

Universidad del Rosario | Escuela de Ingeniería, Ciencia y Tecnología
MACC Matemáticas Aplicadas y Ciencias de la Computación

Planteamiento y Formulación del problema

Un Cuadrado Mágico se obtiene cuando se coloca una serie de ciertos números naturales dentro de una matriz tal que todas las filas, columnas y diagonales sumen exactamente al mismo número. Como se ve en la figura

En nuestro caso, usaremos un Cuadro Mágico 3x3, donde se usarán únicamente los números 1-9 sin repetición para llenar la matriz

Problema: Consideremos un matriz 3x3 vacía. El problema consiste en ubicar los números 1-9 en las casillas para así lograr que se forme un de los 8 Cuadrados Mágicos posibles

4	9	2
3	5	7
8	1	6

Representación en Lógica Proposicional y Letras Proposicionales

Consideremos la siguiente matriz de 9 casillas. En esta matriz, vamos a identificar las casillas con la letra A y los números del 1 al 9

Ahora, cada casilla tiene un conjunto de números específicos que se pueden encontrar ahí. Entonces:

- A1, A3, A7, A9 ={ 2 ,4 ,6, 8}
- A2, A4, A6, A8 = {1 ,3 ,7 ,9}
- A5 = {5}

A1	A2	A3
A4	A5	A6
A7	A8	A9

Representación en Lógica Proposicional y Letras Proposicionales

Como se van usar los números 1-9, tenemos 9 letras proposicionales, estas son p,q,r,s,t,u,v,w & x respectivamente

Para mejor entender el uso de las letras proposicionales hay que tener en cuenta:

- p_i es verdad sii 1 está en la casilla i
- q_i es verdad sii 2 está en la casilla i
- r_i es verdad sii 3 está en la casilla i
- s_i es verdad sii 4 está en la casilla i
- Así para el resto de letras proposicionales

q_1	s_1	p_2	r_2	q_3	s_3
u_1	w_1	v_2	x_2	u_3	w_3
p_4	r_4	t_5		p_6	r_6
v_4	x_4			v_6	x_6
q_7	s_7	p_8	r_8	q_9	s_9
u_7	w_7	v_8	x_8	u_9	w_9

Regla 1

<u>En cada casilla debe haber un solo número</u>

Para hacer esta regla, tuvimos en cuenta los posibles números que pueden aparecer dentro de cada casilla.

Ejemplo:

[$(w_1 \wedge \neg u_1 \wedge \neg s_1 \wedge \neg q_1) \vee (\neg w_1 \wedge u_1 \wedge \neg s_1 \wedge \neg q_1) \vee (\neg w_1 \wedge \neg u_1 \wedge s_1 \wedge \neg q_1) \vee (\neg w_1 \wedge \neg u_1 \wedge \neg s_1 \wedge q_1)$

$\wedge$

$(p_2 \wedge \neg r_2 \wedge \neg v_2 \wedge \neg x_2) \vee (\neg p_2 \wedge r_2 \wedge \neg v_2 \wedge \neg x_2) \vee (\neg p_2 \wedge \neg r_2 \wedge v_2 \wedge \neg x_2) \vee (\neg p_2 \wedge \neg r_2 \wedge \neg v_2 \wedge x_2)$

$\wedge$

… Así respectivamente con las restantes letras proposicionales

Regla 2

No se pueden repetir números dentro de la matriz

Para esta regla, se tuvo en cuenta en qué casillas pueden aparecer ciertos números

Ejemplo:

$[(w_1 \wedge \neg w_3 \wedge \neg w_7 \wedge \neg w_9) \vee (\neg w_1 \wedge w_3 \wedge \neg w_7 \wedge \neg w_9) \vee (\neg w_1 \wedge \neg w_3 \wedge w_7 \wedge \neg w_9) \vee (\neg w_1 \wedge \neg w_3 \wedge \neg w_7 \wedge w_9)$

$\wedge$

$(p_2 \wedge \neg p_4 \wedge \neg p_6 \wedge \neg p_8) \vee (\neg p_2 \wedge p_4 \wedge \neg p_6 \wedge \neg p_8) \vee (\neg p_2 \wedge \neg p_4 \wedge p_6 \wedge \neg p_8) \vee (\neg p_2 \wedge \neg p_4 \wedge \neg p_6 \wedge p_8)$

$\wedge$

… Así respectivamente con las restantes letras proposicionales

Regla 3

Cada fila, columna y diagonal debe sumar a 15

Para esta regla dividimos la matriz en 8 diferentes regiones. Que son 3 filas, 3 columnas y 2 diagonales.

- F1 = {A1, A2 ,A3}
- F2 = {A4, A5 ,A6}
- F3 = {A7, A8 ,A9}

- C1 = {A1, A4 ,A7}
- C2 = {A2, A5 ,A8}
- C3 = {A3, A6 ,A9}

- D1 = {A1, A5 ,A9}
- D2 = {A3, A5 ,A7}

Ejemplo:

$[(w_1 \wedge p_2 \wedge u_3) \vee (w_1 \wedge r_2 \wedge s_3) \vee (u_1 \wedge v_2 \wedge q_3) \vee (s_1 \wedge x_2 \wedge q_3) \vee (q_1 \wedge s_2 \wedge x_3) \vee (q_1 \wedge v_2 \wedge u_3) \vee (s_1 \wedge r_2 \wedge w_3) \vee (u_1 \wedge p_2 \wedge w_3)] \rightarrow 15$

Sin embargo, decidimos generalizar esta regla para evitar problemas con el código y decidimos implementarlas de la siguiente manera:

$(q9 \wedge x8 \wedge s7 \wedge v6 \wedge t5 \wedge r4 \wedge u3 \wedge p2 \wedge w1) \vee (s9 \wedge r8 \wedge w7 \wedge x6 \wedge t5 \wedge p4 \wedge q3 \wedge v2 \wedge u1) \vee$ … así de acuerdo con los otros ocho posibilidades

Representación gráfica de soluciones

Supongamos que al finalizar la ejecucion del codigo, se presenta la siguiente interpretación:

{p2:1, p4:0, p6:0, p8:0, r2:0, r4:1, r6:0, r8:0, v2:0, v4:0, v6:1, v8:0, x2:0, x4:0, x6:0, x8:1, t5:1, q1:0, s1:0, u1:0, w1:1, q3:0, s3:0, u3:1, w3:0, q7:0, s7:1, u7:0, w7:0, q9:1, s9:0, u9:0, w9:0}

En este caso, todas las letras proposicionales con un valor de 1 representan que en esa casilla, se encuentra ese número en específico. Entonces, la representación gráfica sería la siguiente

8	1	6
3	5	7
4	9	2

Resolución del problema

Los pasos que seguimos:

1. Antes de empezar el código, se reemplazaron las letras proposicionales por caracteres para facilitar el uso de ciertas funciones
2. Se crearon las 3 reglas en lógica proposicional y se unieron para crear una única regla. Como no eran muy extensas, optamos por hacerlo de forma manual
3. Usamos Tseitin para así obtener una fórmula en FNC y que esta sea igual de buena que nuestra regla
4. Después usamos nuestro algoritmo DPLL para obtener una interpretación que haga verdadera la regla
5. Esta interpretación fue muy extensiva, entonces recorrimos el diccionario y solo nos enfocamos en nuestras letras proposicionales
6. Representación gráfica de la solución

Solución

Al correr nuestro código, nos dio la siguiente interpretación:

{'Ġ': 1, 'ē': 1, 'ė': 0, 'ě': 0, 'ğ': 0, 'Đ': 0, 'đ': 0, 'Ē': 0, 'Ā': 0, 'ā': 1, 'ĕ': 1, 'Ą': 1, 'ċ': 1, 'Ě': 1, 'Ď': 1, 'Ĝ': 1, 'Ĉ': 0, 'Č': 0, 'Ĕ': 0, 'Ę': 0, 'ą': 0, 'ĉ': 0, 'č': 0, 'ę': 0, 'ĝ': 0, 'Ă': 0, 'Ć': 0, 'Ċ': 0, 'Ė': 0, 'Ğ': 0, 'ă': 0, 'ć': 0, 'ď': 0}

Que, al pasarla en nuestro código de visualización, nos da la siguiente figura

Como se puede evidenciar, obtuvimos un cuadro mágico ya que si se suman cada fila, columna o diagonal, siempre se obtiene 15

8	3	4
1	5	9
6	7	2

Conclusiones y Demostración

Habiendo realizado este proyecto, podemos hacer ciertas conclusiones.

Nuestro código puede ser usado como base para solucionar otros tipos de cuadros mágicos ya que las reglas son las mismas con excepción de la suma en la filas, columnas o diagonales. Aunque, ya no se podría hacer de una forma manual como lo hicimos en este caso y tendríamos que crear funciones para las reglas.

El próximo paso de nuestro proyecto podría ser agregar condiciones iniciales y que después se resuelva el cuadro mágico y modificar el código para resolver cuadros mágicos de más casillas, no únicamente 3x3