Representaciones Vectoriales
↳ "Formalismo de la satisfacción de restricciones.
CSP - "Constraint Satisfaction Problems."

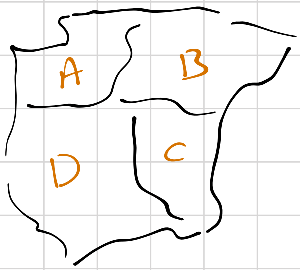
a) Conjunto finito de variables (de decisión)

b) Conjunto de dominio finitos (uno por variable) con los valores de las decisiones (opciones).

c) Conjunto finito de restricciones → Aridad: n-aria ≡ afecta a n variables.
Subconjuntos de valores compatibles de las variables → definen el objetivo.
EJ: variables: x_1, x_2, x_3
Dominio: $D_1 = D_2 = D_3 = \{1,2,3\}$
Restricciones: $x_1 < x_2$ →
$x_2 < x_3$ → } Cada una es una restricción binaria (afecta a 2 variables)

$x_3 > x_2 > x_1$ → Restricción Ternaria (y así sucesivamente).

EJ: Coloreado de mapas

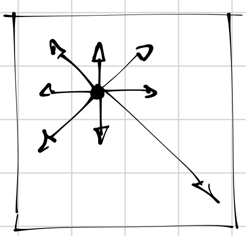


Variables: $\underbrace{A, B, C, D}_{\text{Colores de las regiones}}$

Dominios: $D_A = D_B = D_C = D_D = \{\underbrace{R, V, A}_{\text{Colores}}\}$

Restricciones: (Adyacencias)
$A \neq B$ $B \neq C$ $C \neq D$
$A \neq D$ $B \neq D$

EJ: 4-reinas



Nota: Para cada problema, tenemos que pensar que nuestras variables vengan dada en base a cómo queremos resolver nuestro problema.
EJ: ¿en qué fila coloco la reina de la columna i?
Variable: R_1, R_2, R_3, R_4 } reina en la columna i
Dominio: $D_1 = D_2 = D_3 = D_4 = \{1,2,3,4\}$
Restricciones: Distinta fila: AllDiff (R_1, R_2, R_3, R_4)
↳ Automatización: el algoritmo construye la condición de desigualdad.

Distinta Columna ← no hace falta ponerla, la consideramos en la construcción del problema

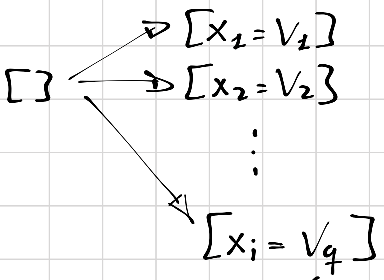
Distinta Diagonal: $\underbrace{|i-j| \neq |R_i - R_j|}_{\text{Distancia de fila} \neq \text{columna.}}$

- Resolución mediante Búsqueda.
  - Definimos un espacio de estados.
    - Estado: lista de asignaciones parciales compatibles.
      - Estado Inicial: [ ] (lista vacía).
      - Estado Final: [. . . .] (lista con todas las variables).
  - Acciones

    Establecemos un órden en las variables: $x_1, x_2, \dots x_n$.

    ↓

    Luego asignamos un valor compatible a la siguiente variable no asignada.



↳ V_i compatible con las restricciones

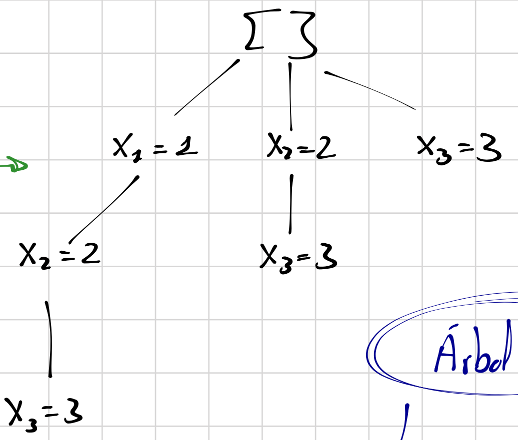
EJ: X_1, X_2, X_3

Dominio: $\{1, 2, 3\}$

Restricciones: $x_1 < x_2$

$x_2 < x_3$

} Espacio de estados →



Árbol finito

↳ Se recomienda usar algoritmos de resolución, como backtracking.

- Consistencia en Arcos (2-consistente)
  - Suponemos restricciones binarias.

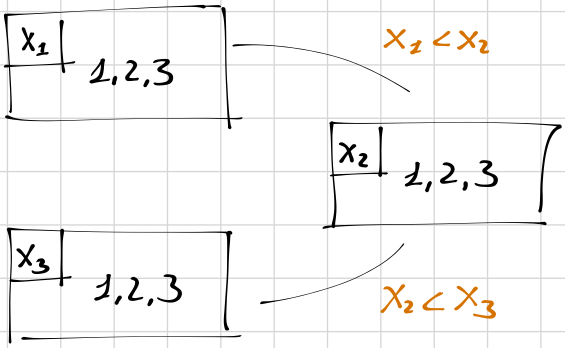
Problema → $\boxed{\text{Consistencia en arcos}}$ → Problema → $\boxed{}$ → Solución

Resolución Mediante grafo de restricciones.

• Algoritmo "AC1" (más sencillo - para resolver problemas de grafos de restricciones).

Pseudocódigo

> Para cada variable x_i
>   Para cada restricción (x_i, x_j)
>   Para cada valor $v \in D_i$
>   ¿existe un valor compatible
>   ¿En caso contrario, eliminar v de D_i



$\boxed{x_1}$ • $x_1 < x_2$

Para $x_1 = 1$ ¿hay valor compat. de x_2? Si
Para $x_1 = 2$ ¿hay valor compat. de x_2? Si
Para $x_1 = 3$ ¿hay valor compat. de x_2? No

• $x_1 < x_2$

Para $x_2 = 1$ ¿hay valor compat. de x_1? No
Para $x_2 = 2$ ¿hay valor compat. de x_1? Si
Para $x_2 = 3$ ¿hay valor compat. de x_1? Si

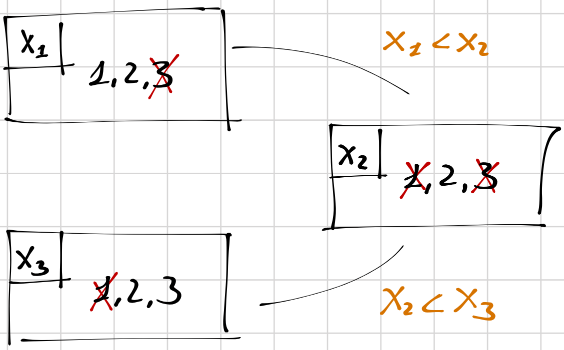
$\boxed{x_2}$ • $x_2 < x_3$

Para $x_2 = 2$ ¿hay valor compat. de x_3? Si
Para $x_2 = 3$ ¿hay valor compat. de x_3? No

• $x_2 < x_3$

Para $x_3 = 1$ ¿hay valor compat. de x_2? No
Para $x_3 = 2$ ¿hay valor compat. de x_2? No
Para $x_3 = 3$ ¿hay valor compat. de x_2? Si

Todos estos procedimientos se pueden representar con restricciones binarias.



Aun ya habiendo simplificado, podemos seguir haciéndolo para optimizar el grafo.

$\boxed{X_1}$ $\cdot\ X_1 < X_2$

Si $X_1 = 1 \Rightarrow$ Si

Si $X_1 = 2 \Rightarrow$ No

$\cdot\ X_1 < X_2$

Si $X_2 = 2 \Rightarrow$ Si

$\boxed{X_2}$ $X_2 < X_3$

Si $X_2 = 2 \Rightarrow$ Si

$X_2 < X_3$

Si $X_3 = 3 \Rightarrow$ Si

EJ: 4 números del 2 al 9

1 • Diferentes.

2 • El primero y el tercero iguales módulo 5.

3 • El cuarto es mayor que el segundo en 1 unidad.

4 • El primero y el segundo no son primos entre sí.

a) Variables: X_1, X_2, X_3, X_4

b) Dominios: $D_1 = D_2 = D_3 = D_4 = \{2,3,4,5,6,7,8,9\}$

c) Restricciones:

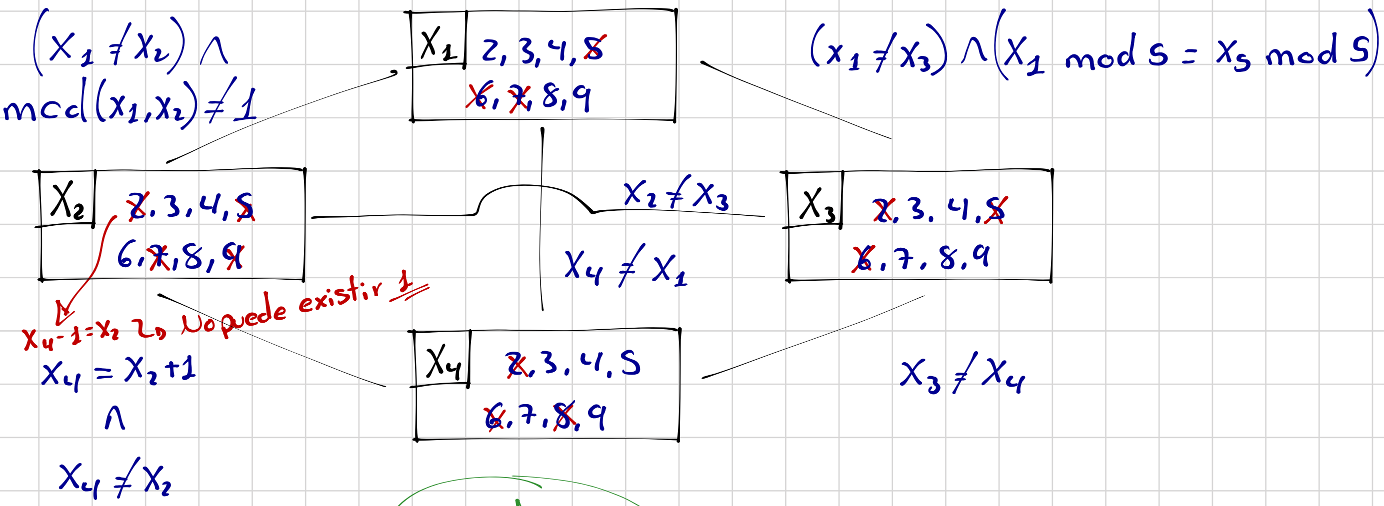
1 $AllDiff(X_1, X_2, X_3, X_4)$

2 $X_1 \bmod 5 = X_3 \bmod 5$

3 $X_4 = X_2 + 1$

4 $\underbrace{mcd}(X_1, X_2) \neq 1$

Máximo Común Divisor



(Una vez eliminado los valores de un dominio, para futuros casos seguirán eliminados.)

1º Iteración

	X_1	X_2	X_3	X_4
X_1	////	5,7	6	✓
X_2	5,7	////	✓	9
X_3	2,5,6	✓	////	✓
X_4	✓	2,6,8	✓	////

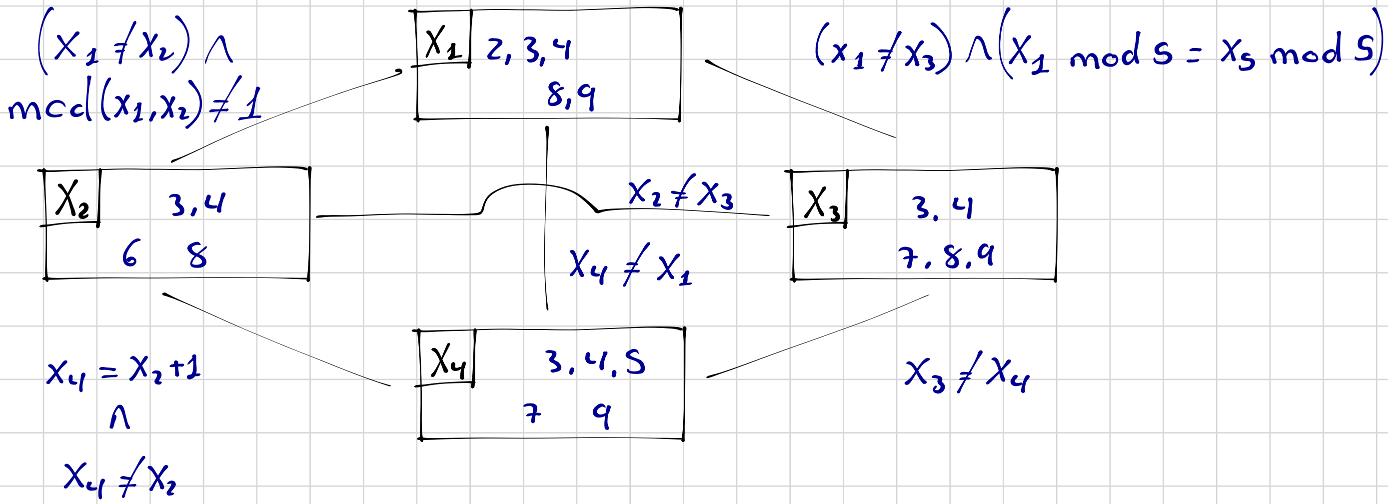
[✓ = Cumple todo]

2º Iteración

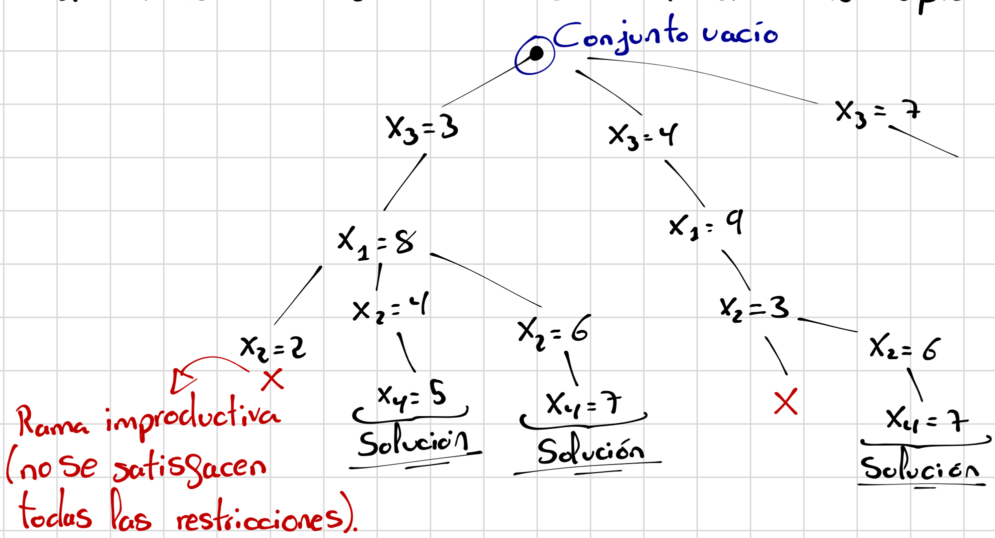
	x_1	x_2	x_3	x_4
x_1		✓	✓	✓
x_2	✓		✓	✓
x_3	✓	✓		✓
x_4	✓	✓	✓	

↳ Se sigue iterando hasta que no hayan diferencias.

Consistencia en árbol



· Podemos deducir que el orden de elección de variables importa, puesto que dependiendo del mismo obtendremos el resultado más rápido o no.



[4-reinas]

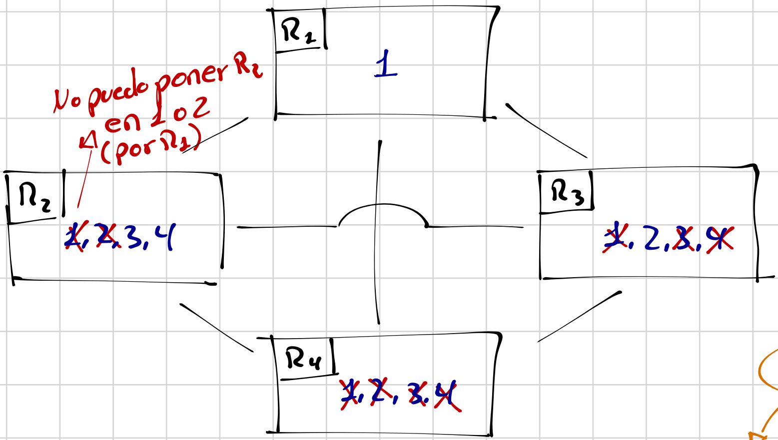
Variable: R_i = fila de la reina en la columna i, $i = 1,2,3,4$
Dominios: $D_i = \{1,2,3,4\}$
Restricciones: $AllDiff(R_1, R_2, R_3, R_4)$

$$\forall i \neq j \quad |i-j| \neq \underbrace{|R_i - R_j|}_{\text{No se encuentran en la misma diagonal}}$$

a) Aplicar consistencia en arcos.

· Suponiendo $[R_1 = 1]$.

	R_1	R_2	R_3	R_4
1	O	X	X	X
2	X	X		X
3	X		X	X
4	X		X	X



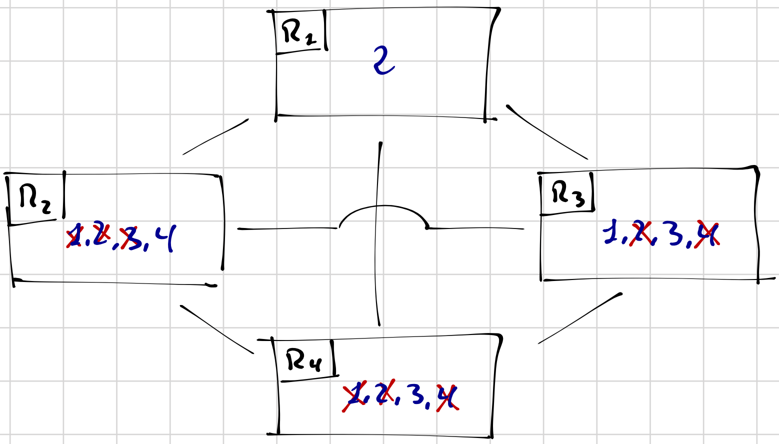
	R_1	R_2	R_3	R_4
R_1	///	✓	✓	✓
R_2	1,2	///	✓	✓
R_3	1,3	4	///	✓
R_4	1,4	✓	2,3	///

Dominio Vacío

Por tanto, el problema NO tiene solución, para $R_1 = 1$.

· Suponiendo $[R_1 = 2]$.

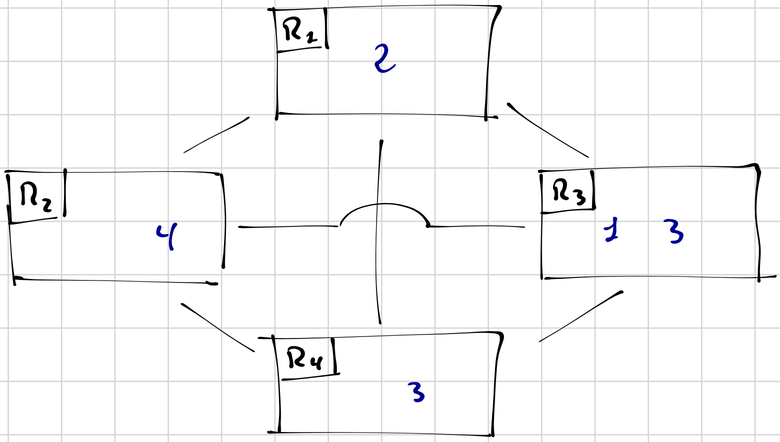
	R_1	R_2	R_3	R_4
1	X	X		X
2	O	X	X	X
3	X	X	X	
4	X		X	X



	R_1	R_2	R_3	R_4
R_1	///	✓	✓	✓
R_2	1,2,3	///	✓	✓
R_3	2,4	3	///	✓
R_4	2	4	1	///

2º Iteración

	R1	R2	R3	R4
1	X	X		X
2	O	X	X	X
3	X	X	X	
4	X		X	X



	R1	R2	R3	R4
R1		✓	✓	✓
R2	✓		✓	✓
R3	✓	✓		✓
R4	✓	✓	✓	