



Universidad Técnica Particular de Loja  
*La Universidad Católica de Loja*

# FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL

## Problemas de Satisfacción de Restricciones

Docente:  
Jorge Cordero Zambrano

**Semana 9**

FACULTAD DE INGENIERÍAS Y ARQUITECTURA  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y ELECTRÓNICA

# Contenidos

- Introducción a los PSR
  - ▣ Aplicaciones
  - ▣ Representación
  - ▣ Variables
  - ▣ Restricciones
- Búsqueda con vuelta atrás para PSR
- Búsqueda local para PSR

# Introducción a los PSR

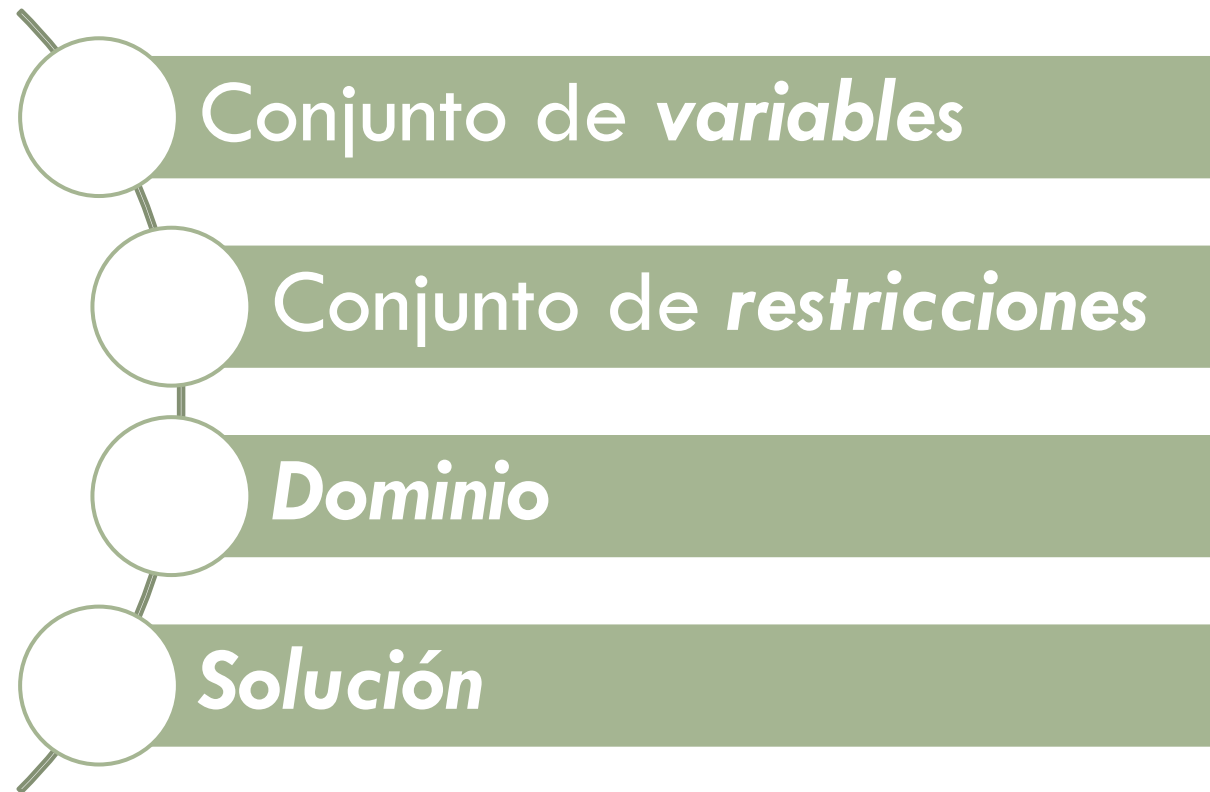
- Una **restricción** es una limitación que se produce en alguna cosa.



- La programación por restricciones es una metodología de software utilizada para la descripción y posterior resolución efectiva de cierto tipo de problemas, típicamente combinatorios y de optimización.

# Introducción a los PSR

- Formalmente un problema de satisfacción de restricciones está definido por:



# Introducción a los PSR

- Formalmente un PSR está definido por:
- Un conjunto de **variables**, cada una de las cuales tiene un dominio de valores.
- Un conjunto de **restricciones**, en la que cada una de ellas implica un subconjunto de variables.
- Un **dominio**, valores posibles para cada variable.
- La **solución**, que puede ser una cualquiera, todas, o una óptima.

Cada variable  $X_i$  tiene un **dominio** no vacío  $D_i$  de **valores** posibles.  
Cada restricción  $C_i$  implica algún subconjunto de las variables y especifica las combinaciones aceptables de valores para ese subconjunto.

# Introducción a los PSR

- Un PSR se resuelve cuando cada variable tiene un valor que satisface todas las restricciones de la variable.

# Introducción a los PSR

- **Aplicaciones. -**
- Colorear mapas
- Crucigramas
- N-reinas
- Asignación / distribución / ubicación de recursos
  - ▣ Distribución de tareas de fabricación
  - ▣ Ubicación de gasolineras, de antenas de telefonía
  - ▣ Asignación de personal
  - ▣ Generación de horarios



# Colorear mapas mediante PSR

## □ Mapa de provincias del sur de Ecuador



- Colorear cada región de este mapa usando los colores **rojo**, **verde** o **azul**, de tal forma que **dos regiones vecinas no tengan el mismo color**.



# Colorear mapas mediante PSR

## □ Mapa de provincias del sur de Ecuador



## □ Formulación como PSR:

### ▣ **Variables:** provincias al sur de Ecuador

- OR (El Oro)
- LO (Loja)
- AZ (Azuay)
- ZA (Zamora Chinchipe)

# Colorear mapas mediante PSR

## □ Mapa de provincias del sur de Ecuador



## □ Formulación como PSR:

■ **Restricciones:** requieren que regiones vecinas tengan colores distintos.

Se pueden representar como una desigualdad ( $\neq$ )

$OR \neq LO$ ,  $OR \neq AZ$ , etc.

Las combinaciones aceptables para  $OR$  y  $LO$  son los pares:

$\{(rojo, verde), (rojo, azul), (verde, rojo), (verde, azul), (azul, rojo), (azul, verde)\}$

# Colorear mapas mediante PSR

## □ Mapa de provincias del sur de Ecuador



## □ Formulación como PSR:

▣ **Dominios:** de cada variable es el conjunto de colores {*rojo*, *verde*, *azul*}.

# Colorear mapas mediante PSR

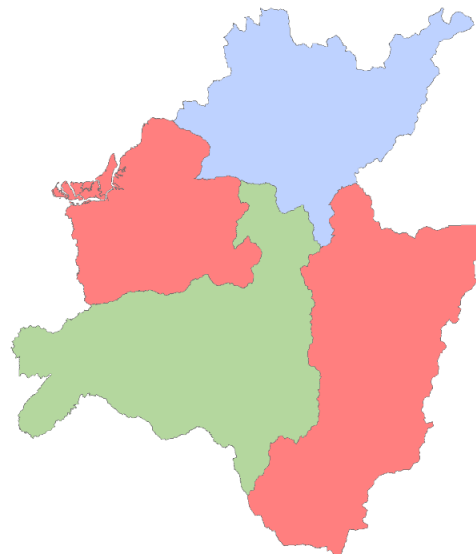
## □ Mapa de provincias del sur de Ecuador



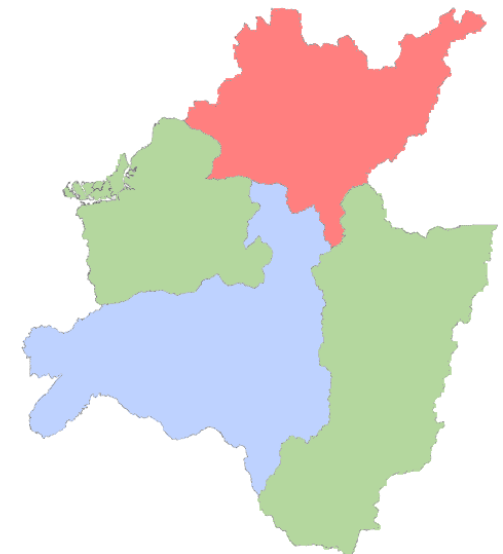
## □ Formulación como PSR:

■ **Solución:** Hay muchas soluciones posibles

$\{OR = \text{rojo}, LO = \text{verde},$   
 $AZ = \text{azul}, ZA = \text{rojo}\}$

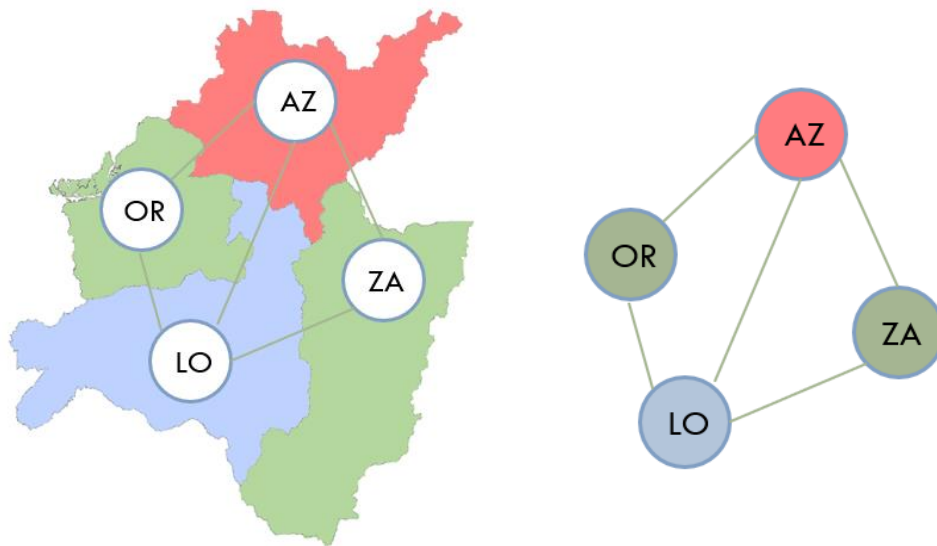


$\{OR = \text{verde}, LO = \text{azul},$   
 $AZ = \text{rojo}, ZA = \text{verde}\}$



# Colorear mapas mediante PSR

- Representación del problema de colorear un mapa como un **grafo de restricciones**.
- La estructura del grafo de restricciones puede usarse para simplificar el proceso de solución, en algunos casos produciendo una reducción exponencial de la complejidad



Los **nodos** del grafo corresponden a las **variables del problema**, y los **arcos** corresponden a las **restricciones**.

# PSR - Tipos de restricciones

- **Restricción unaria:** restringen el valor de una sola variable.
  - ▣ Ej: A los habitantes de Azuay no le gusta el color verde.  $AZ \neq \text{verde}$
- **Restricción binaria:** relaciona dos variables.
  - ▣ Ej: El Oro y Loja no pueden tener el mismo color.  $OR \neq LO$
- Un PSR **binario** tiene sólo restricciones binarias, y se puede representar con un grafo de restricciones.
- **Restricciones múltiples:** se aplica a más de dos variables. El valor asignado a una variable reduce el dominio del resto de variables.

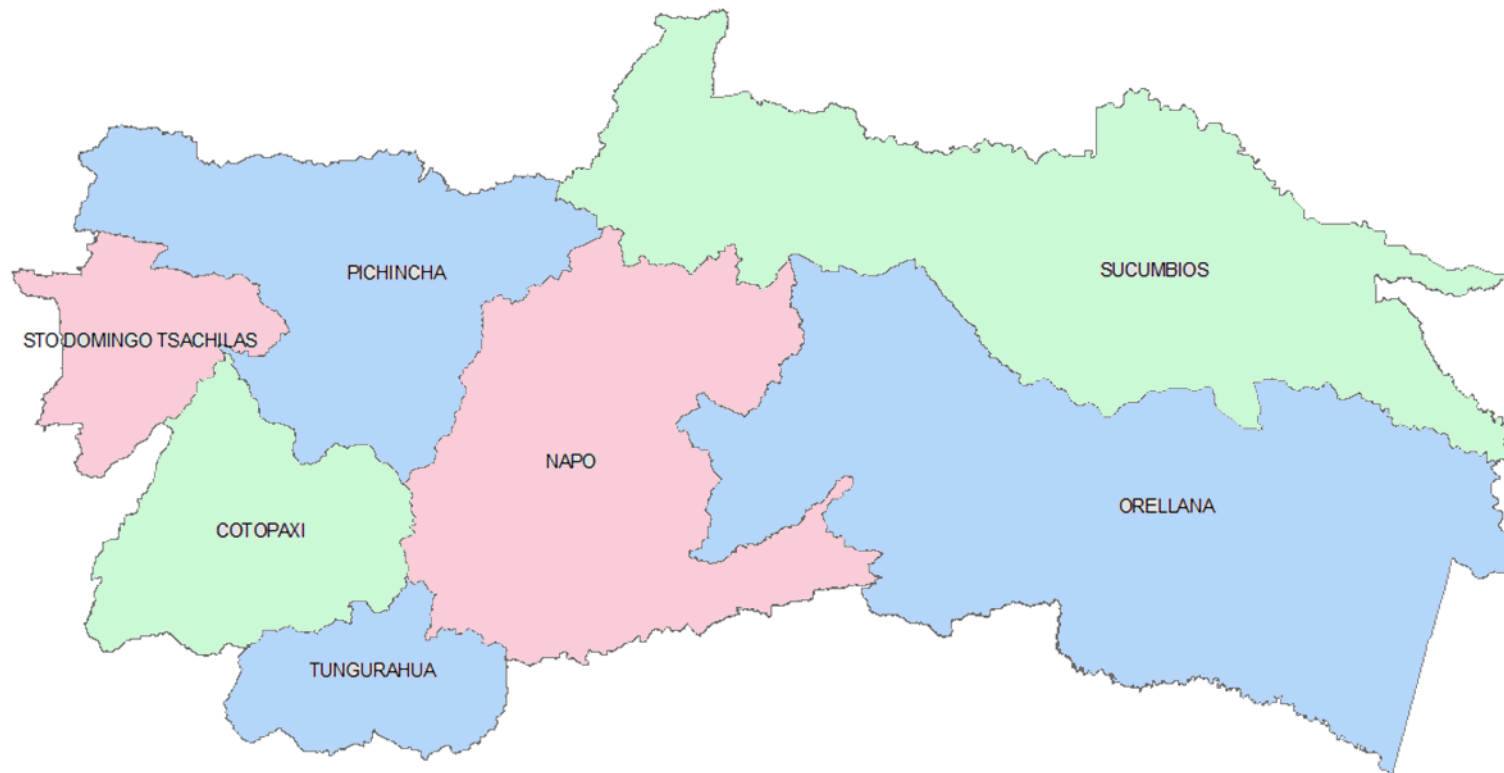
# Ejercicio

- Colorear cada región de este mapa, usando los colores rojo, verde o azul, de tal forma que dos regiones vecinas no tengan el mismo color.



# Ejercicio

- Colorear cada región de este mapa, usando los colores rojo, verde o azul, de tal forma que dos regiones vecinas no tengan el mismo color.





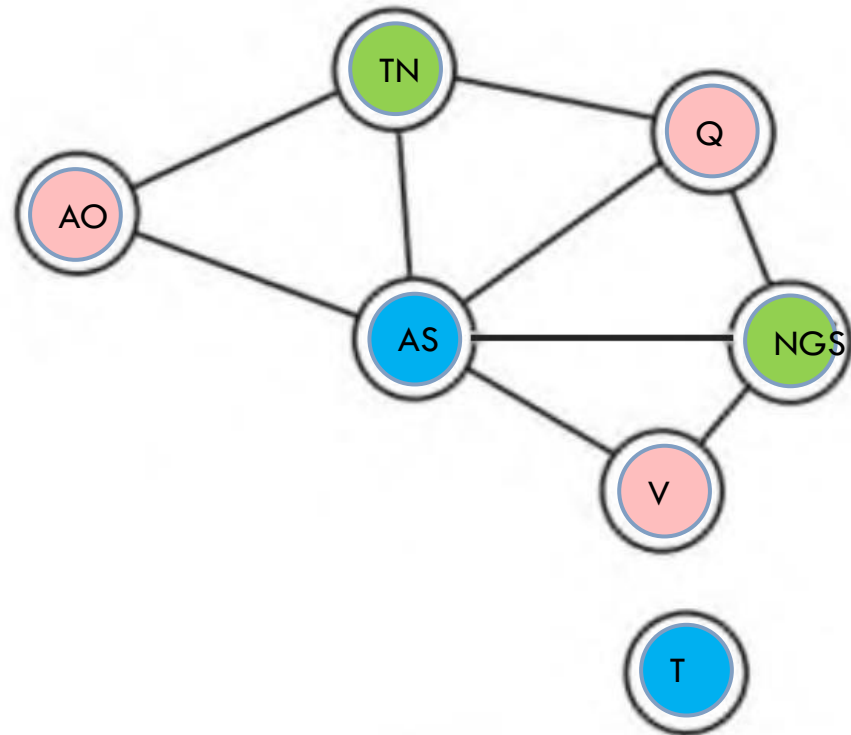
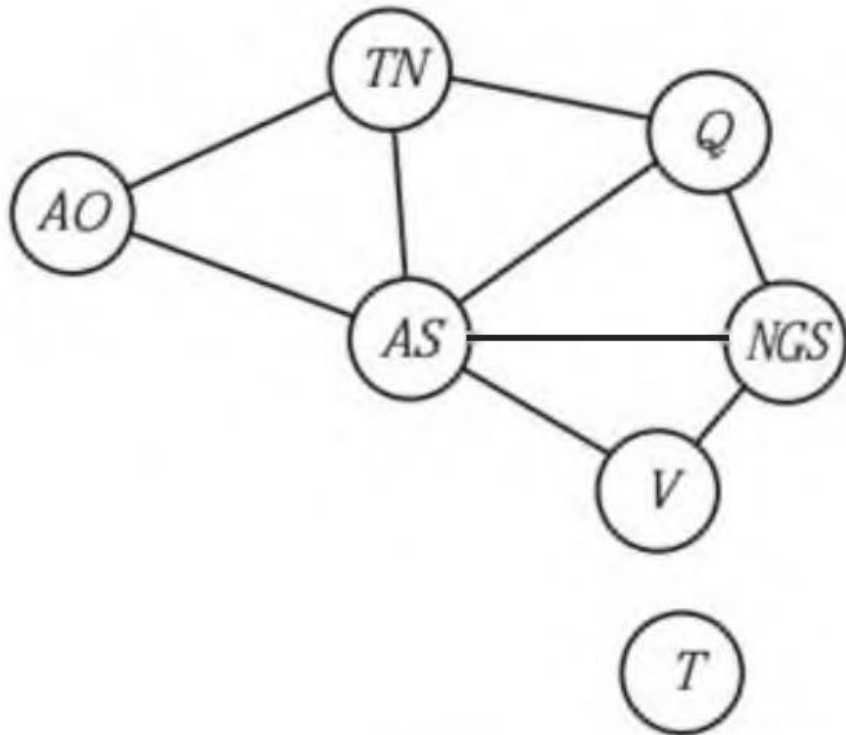
# Ejercicio

- Colorear cada región de este mapa, usando los colores rojo, verde o azul, de tal forma que dos regiones vecinas no tengan el mismo color.



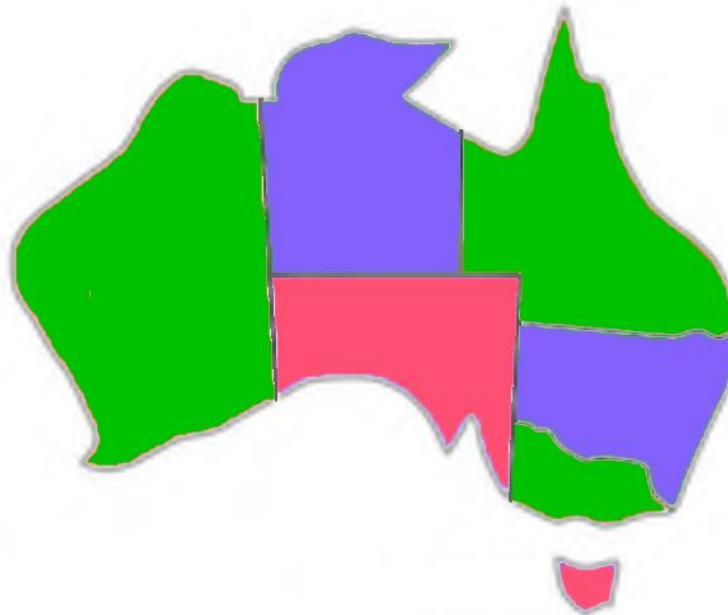
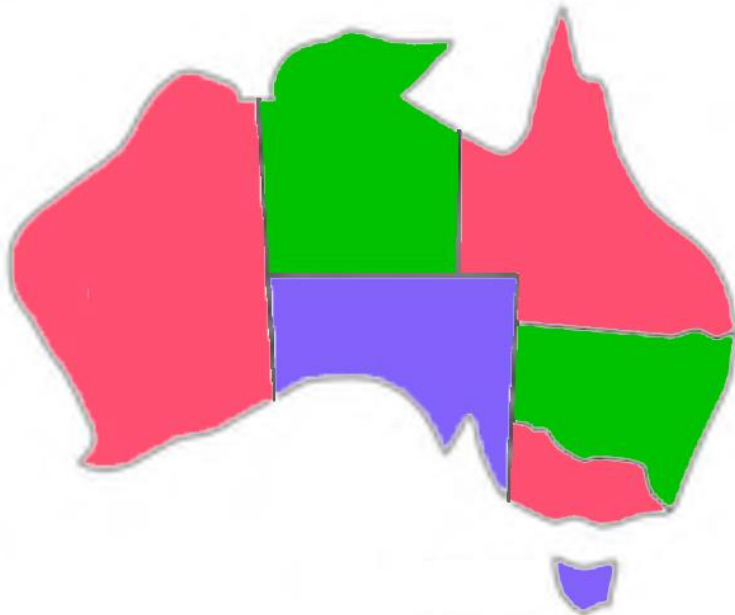
# Ejercicio

- Colorear cada región de este mapa, usando los colores rojo, verde o azul, de tal forma que dos regiones vecinas no tengan el mismo color.



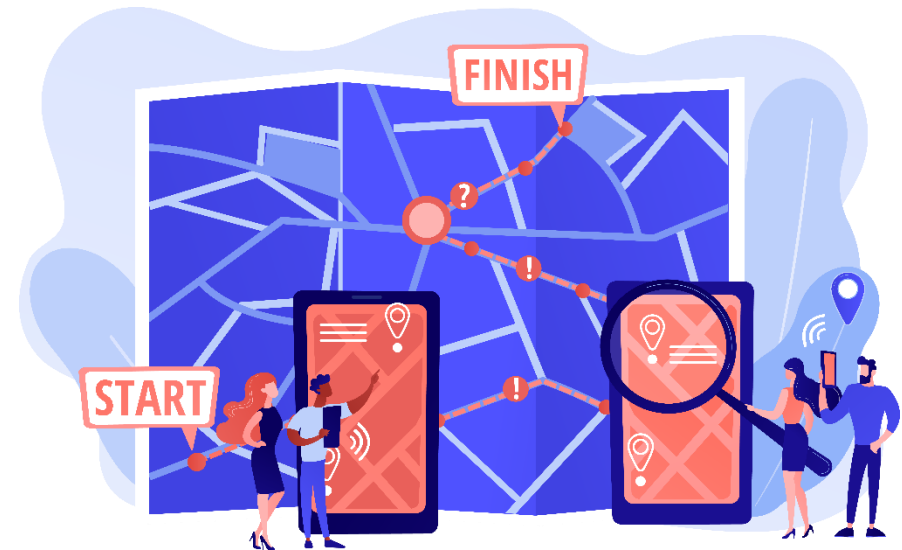
# Ejercicio

- Colorear cada región de este mapa, usando los colores rojo, verde o azul, de tal forma que dos regiones vecinas no tengan el mismo color.



# Búsqueda con vuelta atrás

- El término **búsqueda con vuelta atrás** (Backtracking) es una estrategia para encontrar soluciones a problemas que satisfacen restricciones.
- La búsqueda con vuelta atrás utiliza la búsqueda *primero en profundidad* para elegir valores para una variable y vuelve atrás cuando una variable no tiene ningún valor legal para asignarle.



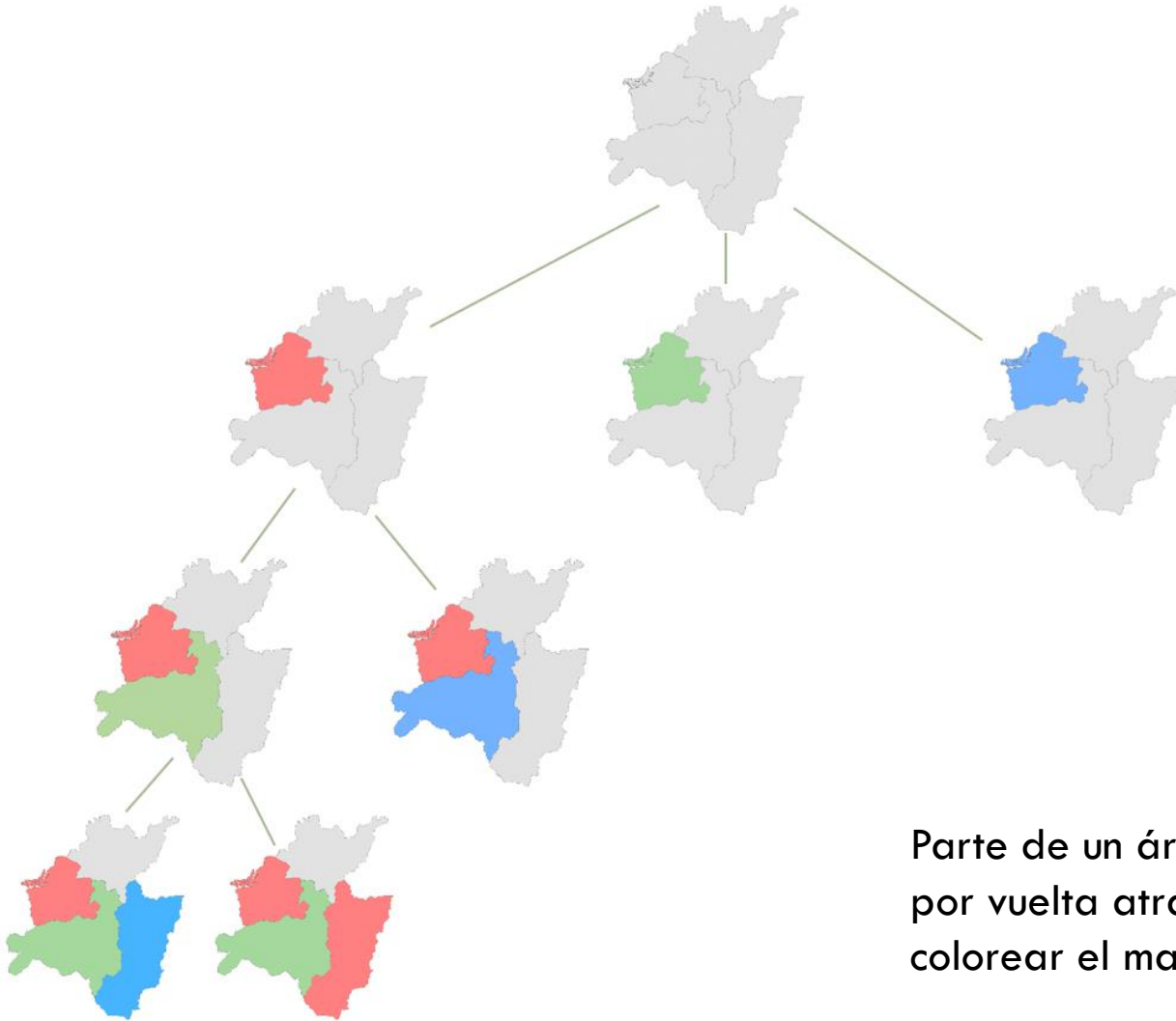
# Búsqueda con vuelta atrás

## □ Funcionamiento:

- ▣ Se escoge variable sin valor.
- ▣ Se expande los valores que cumplan las restricciones.
- ▣ Vuelta atrás si el dominio de una variable se queda vacío.

# Búsqueda con vuelta atrás

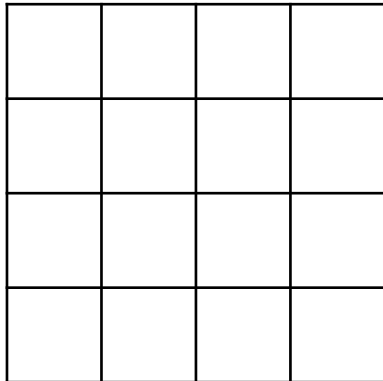
Para resolver este problema utilizando el algoritmo de vuelta atrás, es necesario construir un árbol de búsqueda



Parte de un árbol de búsqueda generado por vuelta atrás para el problema de colorear el mapa

# Ejercicio

- Colocar 4 reinas, una en cada fila de un tablero 4x4, sin que se maten.



Identificar las restricciones  
Diseñar el árbol de búsqueda

# Ejercicio

- Colocar 4 reinas, una en cada fila de un tablero 4x4, sin que se maten.
- Grupos 1 y 3

Posición de la reina

Supuesto 1

R1=1

○			

Supuesto 2

R1=2

	○		



# Ejercicio

- Colocar 4 reinas, una en cada fila de un tablero 4x4, sin que se maten.
- Grupos 2 y 4

Posición de la reina

Supuesto 3

R1=3

		o	

Supuesto 4

R1=4

			o

# Ejercicio

- Colocar 4 reinas, una en cada fila de un tablero 4x4, sin que se maten.

	o		

R1=2

	o		
x	x	x	o

R2=1 no  
R2=2 no  
R2=3 no  
R2=4 si

	o		
			o
o			

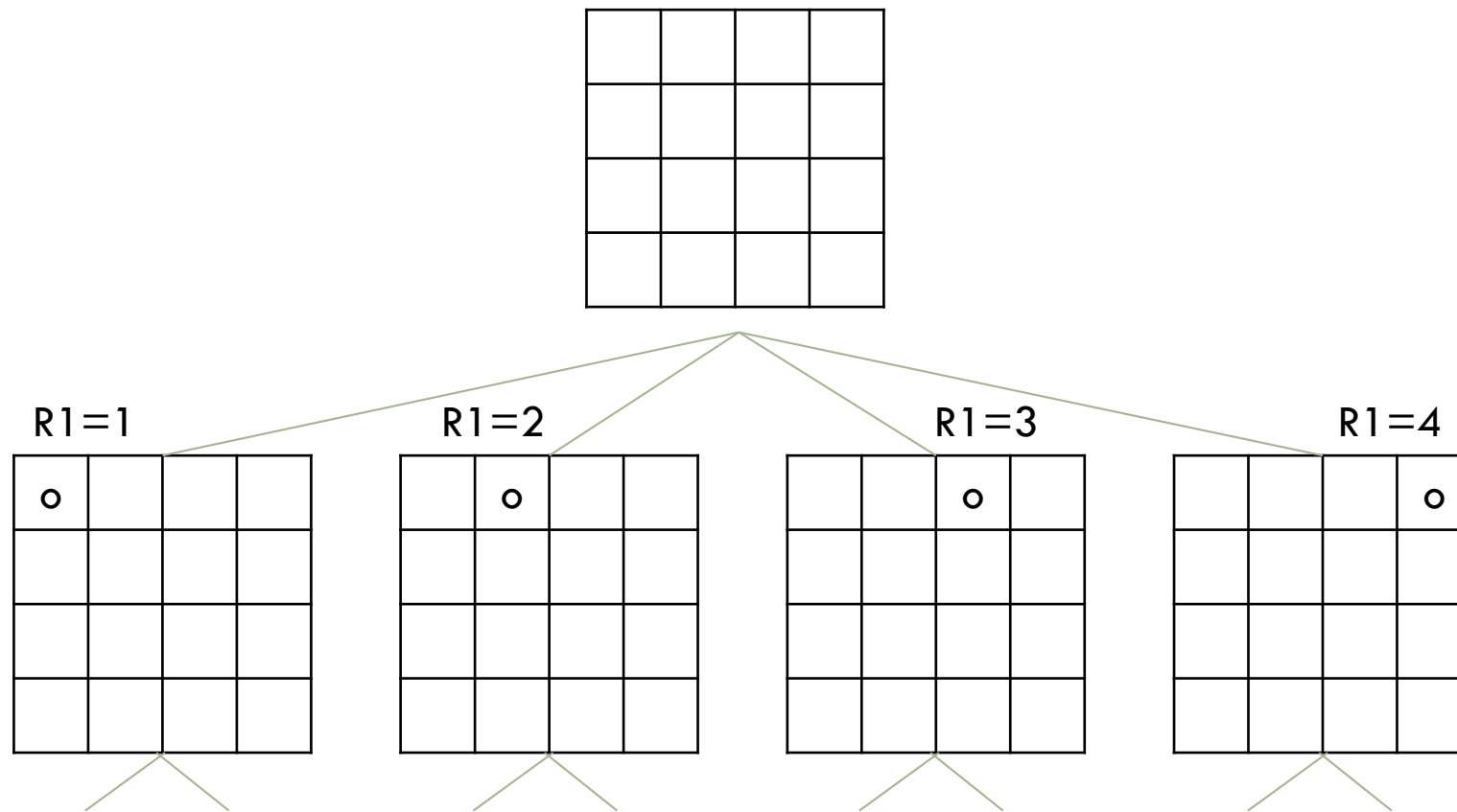
R3=1 si

	o		
			o
o			
x	x	o	

R4=1 no  
R4=2 no  
R4=3 si

# Ejercicio

- Colocar 4 reinas, una en cada fila de un tablero 4x4, sin que se maten.



# Ejercicio

- Colocar 4 reinas, una en cada fila de un tablero 4x4, sin que se maten.

R1=1

○			

R1=4

			○

○			
x	x	○	

○			
x	x		○

			○
○		x	x
x	x	○	x

			○
	○	x	x

○			
		○	
x	x	x	x

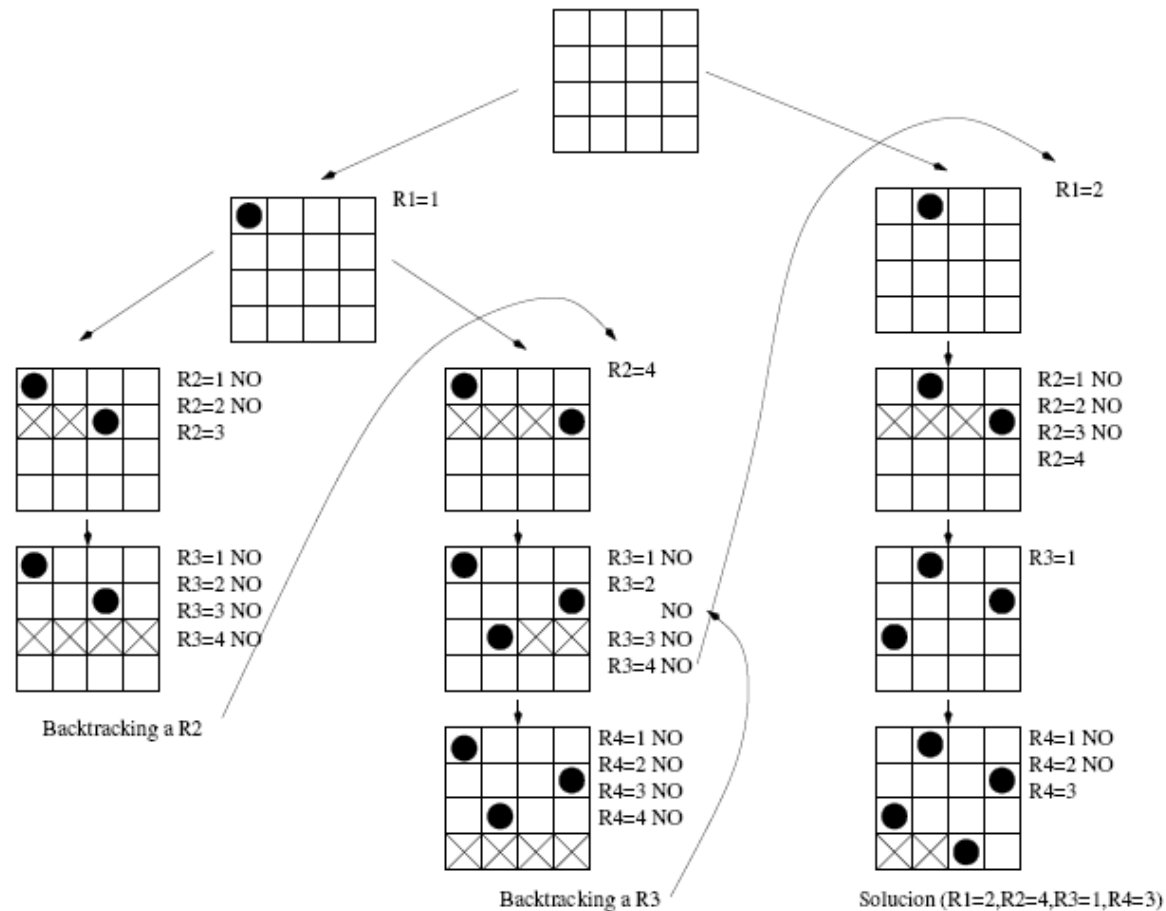
○			
			○
x	○	x	x
x	x	x	x

			○
○		x	x
x	x	○	x
x	x	x	x

			○
	○		
x	x	x	x

# Ejercicio

- Colocar 4 reinas, una en cada fila de un tablero 4x4, sin que se maten.



Árbol de búsqueda parcial para ubicar 4 reinas.

# Actividad recomendada

- Estimado estudiante, realice la formulación del PSR para el mapa
  - ▣ ¿Cuál es el conjunto de variables?
  - ▣ ¿Cuál es el conjunto de restricciones? Definir al menos dos ejemplos.

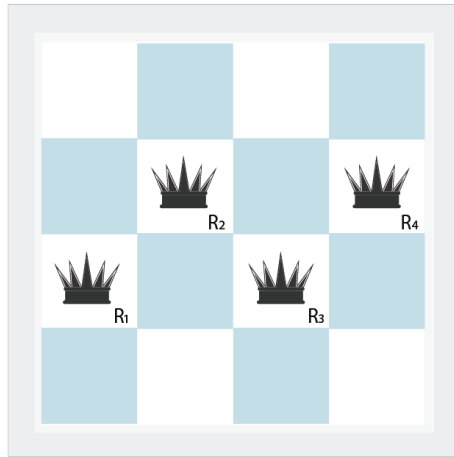


# Búsqueda local para PSR

- Los algoritmos de **búsqueda local (BL)** son muy eficientes para la resolución de muchos PSR.
- La búsqueda local utiliza una **formulación de estados completa** donde cada estado asigna un valor a cada variable, y la búsqueda **cambia** el valor de la variable hasta alcanzar un estado desde el cual no se puede alcanzar ningún estado mejor.
- Para la asignación de un nuevo valor para la variable se utiliza la heurística de **mínimo-conflicto**.

# Búsqueda local para PSR

## □ Problema 4-reinas para mínimo-conflicto

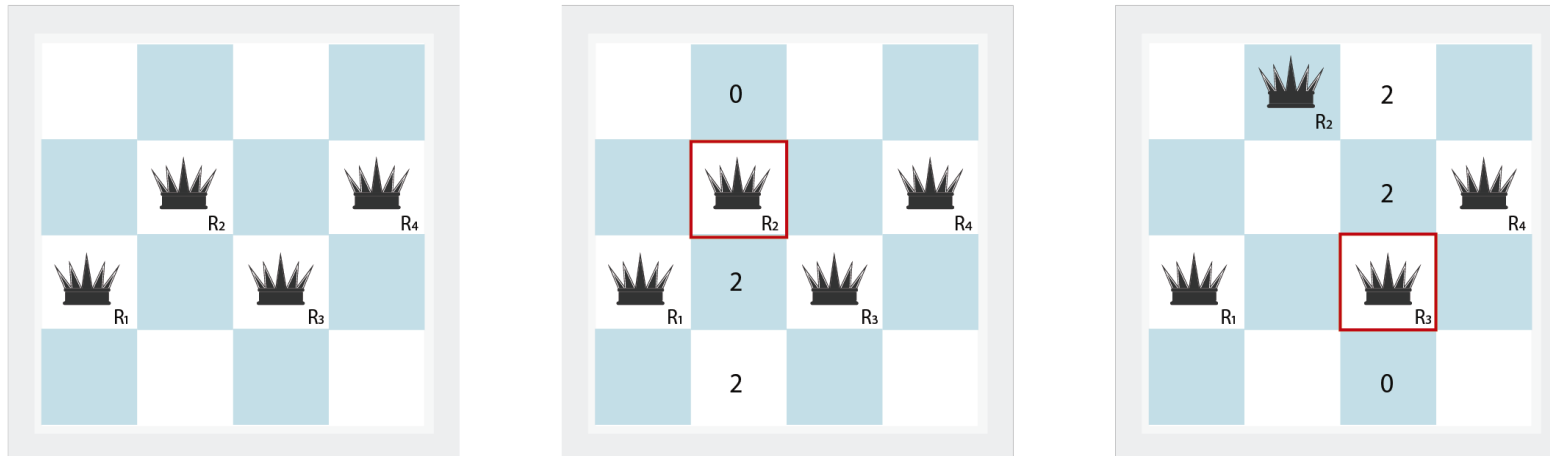


El estado inicial es una asignación completa de las 4 variables, que por lo general incumple la restricción de no atacarse entre sí.



# Búsqueda local para PSR

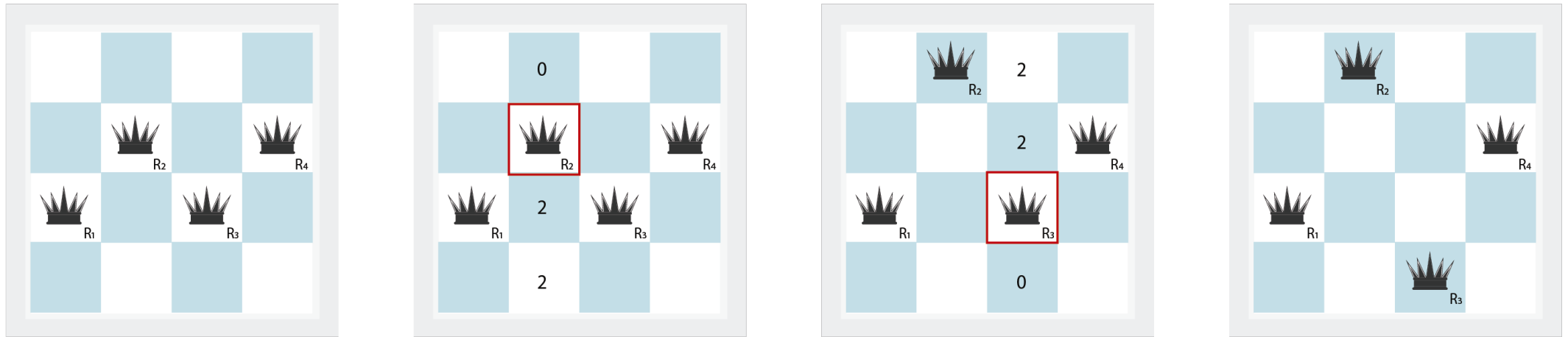
## □ Problema 4-reinas para mínimo-conflicto



En cada etapa, una reina es elegida para su reasignación en su columna. Se elige aleatoriamente una variable en conflicto (R2). Para cambiar la variable a algo que nos acerque a una solución, el enfoque es seleccionar el valor con mínimo conflicto. El número de conflictos se muestra en cada casilla.

# Búsqueda local para PSR

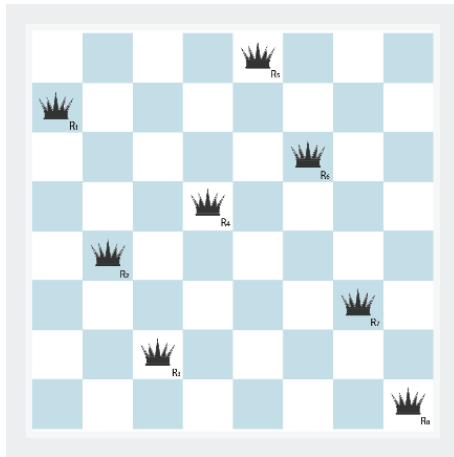
## □ Problema 4-reinas para mínimo-conflicto



El algoritmo mueve a la reina a la posición de mínimo conflicto, deshaciendo los empates de manera aleatoria. El mismo procedimiento se realiza en el siguiente paso con R3, y al final no hay más variables conflictivas, llegando a la solución.

# Búsqueda local para PSR

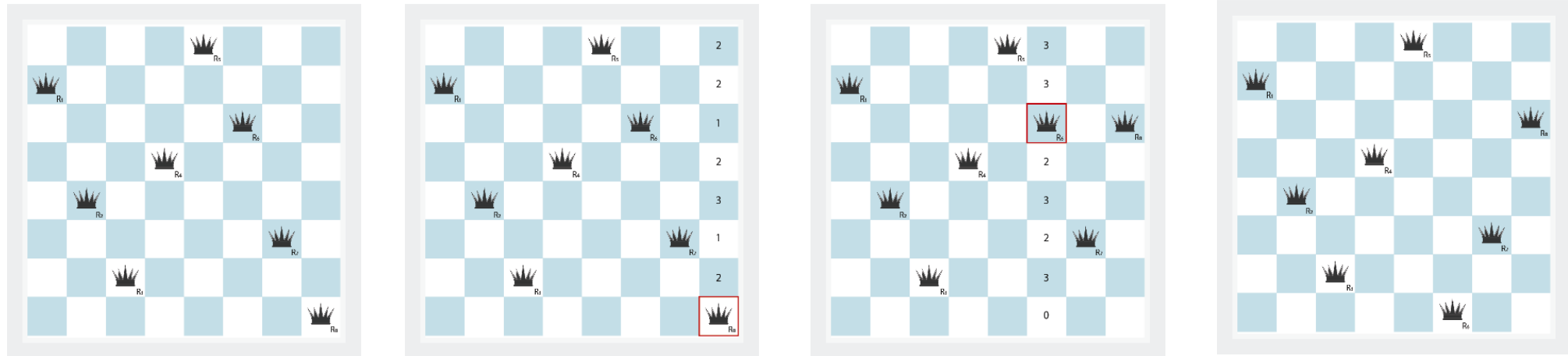
## □ Problema 8-reinas para mínimo-conflicto



El estado inicial es una asignación completa de las 8 variables, que por lo general incumple la restricción de no atacarse entre sí.

# Búsqueda local para PSR

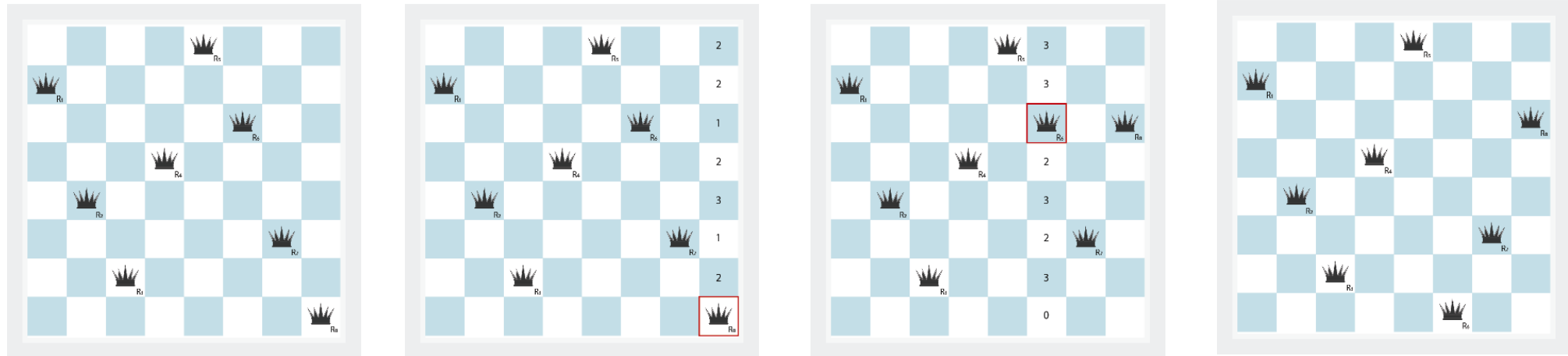
## □ Problema 8-reinas para mínimo-conflicto



En cada etapa, una reina es elegida para su reasignación en su columna. Se observa que las reinas R8 y R6 no cumplen la restricción de no atacarse con otra reina. Se elige aleatoriamente una variable en conflicto (R8). El número de conflictos se muestra en cada casilla. El algoritmo mueve a la reina a la posición de mínimo conflicto, en este caso existe dos opciones con mínimo conflicto y se opta por mover R8 a la casilla 3.8.

# Búsqueda local para PSR

## □ Problema 8-reinas para mínimo-conflicto



El mismo procedimiento se realiza con  $R_6$ , y al final se tienen las reinas ubicadas sin atacarse entre sí

# Recursos educativos

- Problemas de satisfacción de restricciones
- Explicando Backtracking | Fuerza Bruta Pero Elegante
- Backtracking en PSR
- Búsqueda local
- Problemas de satisfacción de restricciones

# Referencias

- Cordero, J. (2023). *Texto-Guía de Inteligencia Artificial*. Loja, Ecuador. EdiLoja.
- Barba-Guaman, L. (2023). *Texto Guía: Inteligencia Artificial Avanzada*. Loja. Ecuador: Editorial UTPL.
- Russel, S., & Norving, P. (2004). *Inteligencia Artificial. Un enfoque Moderno*. Madrid: PEARSON.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*, 4th edition. Pearson Education.
- García S., A. (2017). *Inteligencia Artificial: Fundamentos, práctica y aplicaciones*. México. Alfaomega
- Bejar, J.(2007). *Inteligencia Artificial: Resolución de problemas, algoritmos de búsqueda*. Barcelona, España.
- Imágenes tomadas de [freepink](https://www.freepink.com/).