Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Sistemas Inteligentes

## Tema 3.2: Búsqueda en problemas de Satisfacción de Restricciones

Curso 2024-25

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Índice

- Formulación de CSPs como redes de restricciones
- Ejemplos
- Métodos de resolución:
  - Esquema Backtracking
  - Esquema Forward Checking
  - Esquema de propagación de restricciones

Universitat d'Alacant
Universidad de Alicante

# Problemas de satisfacción de restricciones (CSPs)

Conjunto de **variables** definidas sobre **dominios** finitos y conjunto de **restricciones** definidas sobre subconjuntos de dichas variables:

$$(V, D, \rho)$$

- Un conjunto de **variables**

$$V = \{V_1, V_2, \ldots, V_n\} \equiv \{V_i\}_{i=1..n}$$

- definidas sobre **dominios** discretos Di (conjunto finito de posibles valores)

$$D = \{D_1, D_2, \ldots, D_n\} \equiv \{D_j\}_{j=1..n}$$

- un conjunto de **restricciones** definidas sobre subconjuntos de dichas variables

$$\rho = \{\rho_1, \rho_2, \ldots, \rho_n\} \equiv \{\rho_k\}_{k=1..n}$$

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# CSPs

- Un ejemplo:

$$X::\{1,2\}, Y::\{1,2\}, Z::\{1,2\}$$
$$X=Y, X\neq Z, Y>Z$$

  - Solución: encontrar asignaciones de valor a las variables que satisfagan todas las restricciones.

$$(X=2, Y=2, Z=1)$$

- Solución al problema: la relación n-aria que satisface todas las restricciones del problema
- Dependiendo de los requerimientos del problema hay que encontrar todas las soluciones o sólo una

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Redes de restricciones

- Un CSP se puede representar como un grafo.
  - Sobre el grafo se puede definir una red de restricciones
  - Quíntupla $\langle V, E, c, l, a \rangle$
    - $V$: conjunto de nodos.
    - $E$: conjunto de aristas.
    - $c$: $E \rightarrow V_k, k \leq n$ función de asignación de aristas a tuplas de nodos.
    - $l$: $E \rightarrow \rho$ ; función de asignación de aristas a restricciones
    - $a$: permutación que define el orden de selección para resolver el problema.

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# CSP binario

- Variables

$$V = \{V_1, V_2, \ldots, V_n\}$$

- Dominios discretos y finitos

$$D = \{D_1, D_2, \ldots, D_n\}$$

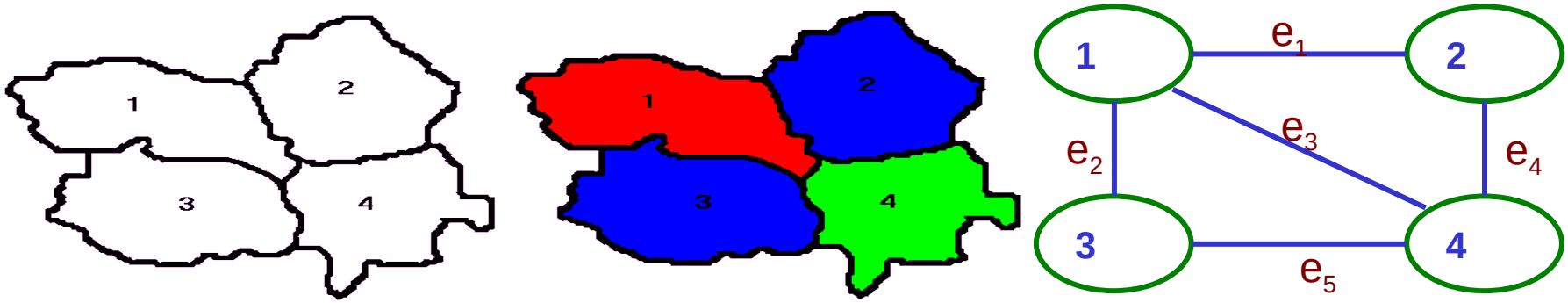
- Restricciones binarias

$$\{R_{ij}\}$$

- Todo problema **n-ario** se puede formular como un problema **binario**
- Ejemplos de CSP binarios:
  - Coloreado de mapas
  - Asignación de tareas para un robot
  - N-reinas
  - Generación de crucigramas

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo: coloreado de mapas




$$V = \{V_1, V_2, V_3, V_4\}$$

$$D_i = \{\text{rojo}, \text{azul}, \text{verde}\}, \quad \forall i, 1 \leq i \leq 4$$

$$E = \{e_1, e_2, e_3, e_4, e_5\}$$

$$\rho_k((V_i, V_j)) = \{\langle v_i, v_j \rangle \mid v_i \in D_i, v_j \in D_j, v_i \neq v_j\}, \quad \forall k, 1 \leq k \leq 5$$

$$c(e_1) = \langle V_1, V_2 \rangle, \quad c(e_2) = \langle V_1, V_3 \rangle, \quad c(e_3) = \langle V_1, V_4 \rangle, \ldots$$

$$l(e_j) = \{\langle \text{azul}, \text{verde} \rangle, \langle \text{azul}, \text{rojo} \rangle, \langle \text{verde}, \text{azul} \rangle, \langle \text{verde}, \text{rojo} \rangle, \langle \text{rojo}, \text{azul} \rangle, \langle \text{rojo}, \text{verde} \rangle\}, \quad \forall j, 1 \leq j \leq 5$$

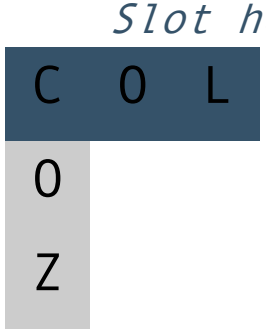
$$a = \{V_1, V_4, V_2, V_3\}$$

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo: generar crucigramas

- Dada una rejilla y un diccionario, construir un crucigrama legal




*Slot horizontal de tres letras*

```
4 palabras de 1 letra
4 palabras de 3 letras
2 palabras de 5 letras
```

- Formulación:
  - **variables:** grupo de casillas para una palabra (slots)
  - **dominios:** palabras del diccionario con la longitud adecuada
  - **restricciones:** misma letra en la intersección de dos palabras
- Características:
  - CSP binario, discreto (dominios grandes)

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo: generar crucigramas

## Formalización:

$$V = \{V_1, \ldots, V_n\},\ \text{slot para palabras}$$

$$D_i = \{\text{palabras de longitud } |v_i|\}, \forall i \in [1, n]$$

$$E = \{e_1, \ldots, e_m\},\ \text{intersecciones}$$

$$c : E \to V \times V,\ \text{asigna intersecciones a pares de palabras}$$

$$l : E \to \{\text{misma letra}\},\ \text{restricción de intersección}$$

$$\rho((v_i, v_j)) = \{(w_i, w_j) \in D_i \times D_j : w_i[k] = w_j[l] \text{ para } (k, l) = pos\_inter(v_i, v_j)\}$$

$$a : \text{permutación en } [1, n]$$

Universitat d'Alacant
Universidad de Alicante

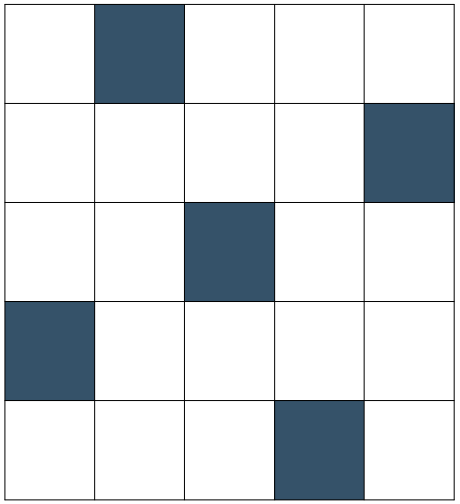
Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo: n-reinas

- Posicionar n reinas en un tablero de ajedrez n×n, de forma que no se ataquen

n = 5

(4,1,3,5,2)



- Formulación: (1 reina por columna)
  - **variables:** reinas, Xi reina en la columna i-ésima
  - **dominios:** filas posibles {1, 2, …, n}
  - **restricciones:** no colocar dos reinas en
    - la misma columna
    - la misma diagonal
- Características:
  - Dominios discretos y restricciones binarias

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo: n-reinas

- Formalización:

$$V = \{X_1, X_2, ..., X_n\}, X_i : \text{ reina en la columna i-ésima}$$
$$D_i = \{1, 2, ..., n\}, \forall i \in [1, n], \text{ filas posibles}$$
$$E = \{e_{ij} : 1 \leq i < j \leq n\}, \text{ conexiones entre columnas}$$
$$c : E \to V \times V, \ \mathrm{c}(e_{ij}) = (X_i, X_j)$$
$$l : E \to \{\text{no ataque}\}, \text{ restricción de no ataque entre reinas}$$
$$\rho((X_i, X_j)) = \{(r_i, r_j) \in D_i \times D_j : r_i \neq r_j \wedge |i - j| \neq |r_i - r_j|\}$$
$$a : \text{permutación en } [1, n] \text{ (orden de colocación de reinas)}$$

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo: criptoaritmética

- Sustituir cada letra por un dígito distinto (distinta cifra, distinta letra) de manera que la suma sea correcta
- Formulación:
  - **variables**: G, O, T, A, U, $C_1$, $C_2$, $C_3$

    ($C_1$, $C_2$, $C_3$ variables de acarreo)
  - **dominios**: O, T, U ∈ {0, …, 9}
    - G, A ∈ {1, …, 9}
    - $C_1$, $C_2$, $C_3$ ∈ {0, …, 4}
  - **restricciones**:
    - letras distintas G ≠ O, G ≠ T, …, A ≠ U
    - suma correcta
      - (unidades) $5A = 10C_1+A$
      - (decenas) $5T+C_1 = 10C_2+U$
      - (centenas) $5O+C_2 = 10C_3+G$
      - (unidad. millar) $5*G+C_3 = A$
- Características: dominios discretos y relaciones múltiples

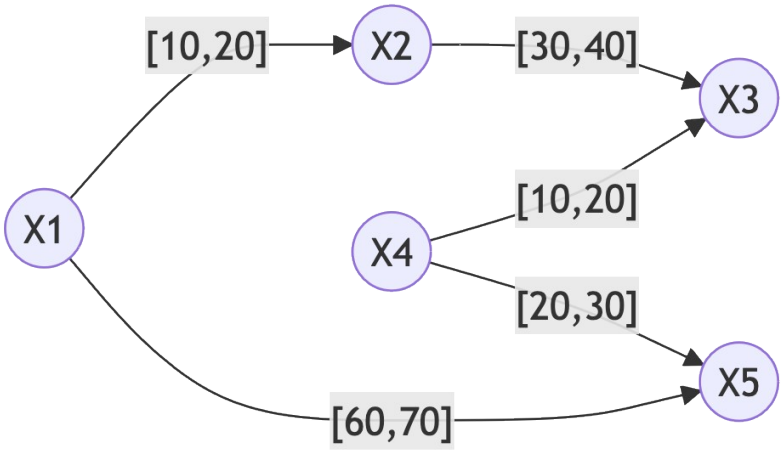
```
  GOTA
+ GOTA
  GOTA
  GOTA
  GOTA
 ------
 AGUA
```

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Ejemplo: restricciones temporales

- Dado un conjunto de sucesos que ocurren en intervalos temporales con ciertas relaciones, encontrar un asignación temporal consistente

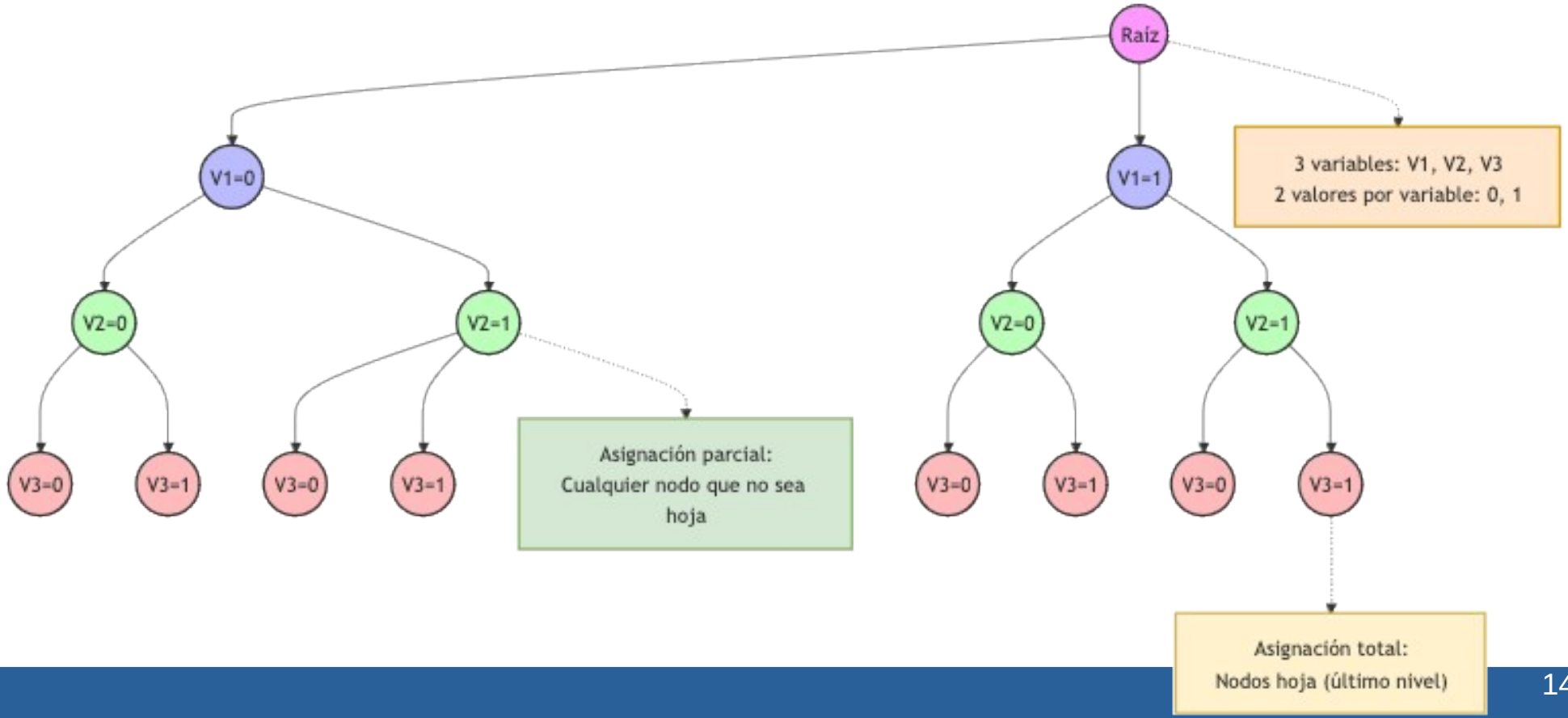



- Formulación:
  - **variables:** sucesos
  - **dominios:** intervalo temporal para cada suceso
  - **restricciones:** distancia temporal permitida entre sucesos; relaciones temporales antes, después, solapado, etc.
- Características : Dominios continuos y restricciones binarias

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Árbol de interpretaciones

- Partimos de un nodo raíz que supervisa el proceso.
- Cada nivel corresponde a una asignación de valor para una característica de datos. El orden de descenso viene especificado por a.
- Cada nodo identifica una posibilidad de asignación (Variable, valor).
- La solución se construye de forma incremental de tal forma que cada hoja es una interpretación.



Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Métodos de resolución

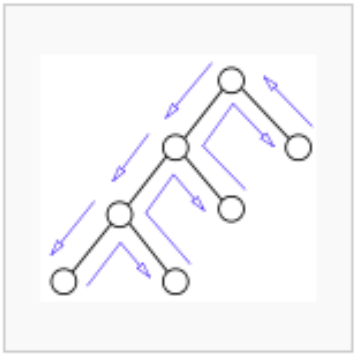
- Tres estrategias
  - **Búsqueda**
    - Explorar el espacio de estados hasta encontrar una solución, demostrar que no existe o agotar los recursos
  - **Inferencia**
    - Deducir un  problema equivalente que sea más fácil de resolver
  - **Hibridas**
    - Combinación de las aproximaciones anteriores. Sobre un esquema de búsqueda se incorporan métodos de inferencia

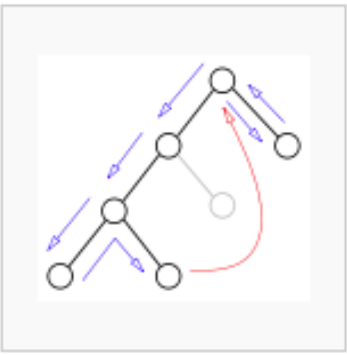
Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Métodos de resolución

- **Búsqueda**
  - *Generación y test:*
    - Generar de forma sistemática y exhaustiva cada una de las posibles asignaciones a las variables y comprobar si satisfacen todas las restricciones.
    - Hay que explorar el espacio definido por el producto cartesiano de los dominios de las variables.
  - *Backtracking:* se trata de construir la solución de forma gradual, instanciando variables en el orden definido por la permutación dada
  - *Backjumping:* parecido al BT pero el retroceso no se hace a la variable instanciada anteriormente sino a la variable más profunda que está en conflicto con la variable actual.





Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Métodos de resolución

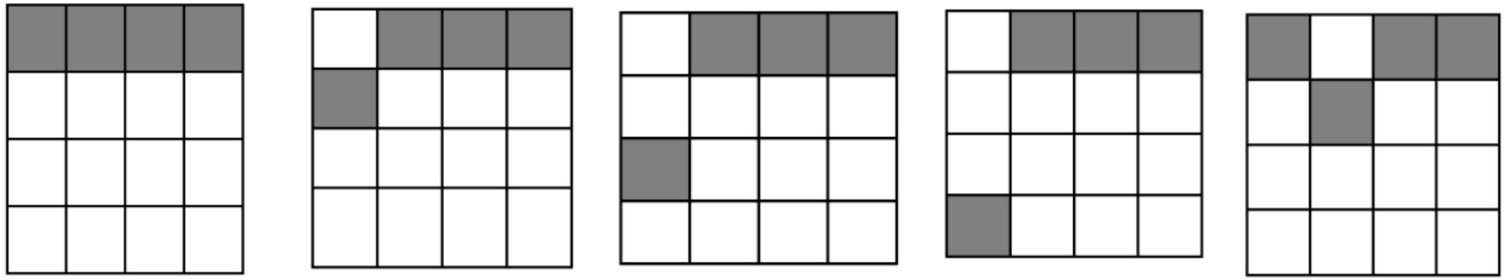
- **Inferencia**
  - Consistencia de arco
  - Consistencia de caminos
  - K-consistencia
- **Algoritmos híbridos**
  - Forward Checking
  - Maintaining Arc Consistency
  - Heurísticas

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Generación y test

- Estrategia:
  - Generación y test de todas las asignaciones totales posibles
    - 1. Generar una asignación de todas las variables
    - 2. Comprobar si es solución. Si es, stop, sino ir a 1
  - Eficiencia:
    - Es muy poco eficiente
    - Genera muchas asignaciones que violan la misma restricción



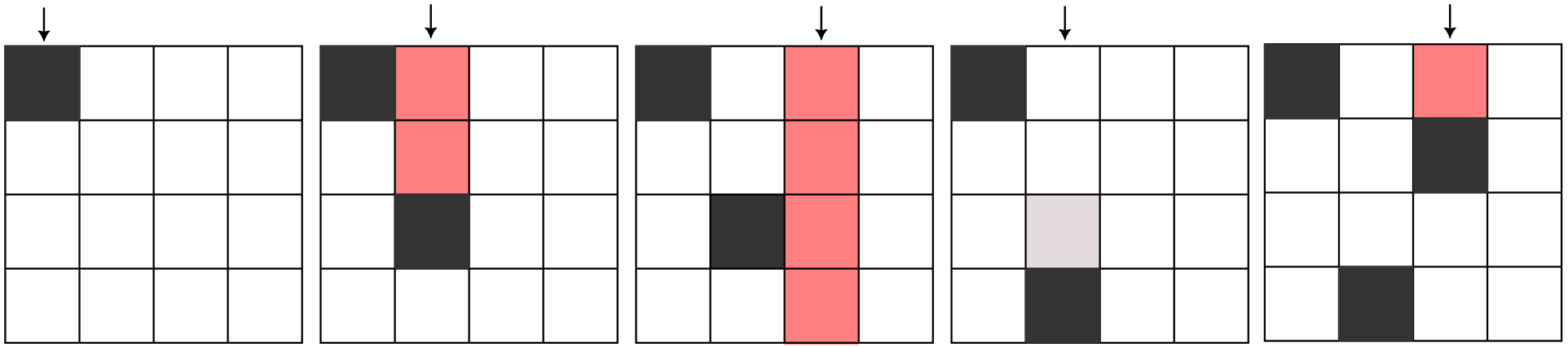
Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Backtracking

- Estrategia:
  - Construir una solución parcial: asignación parcial que satisface las restricciones de las variables involucradas
  - Extender la solución parcial, incluyendo una variable cada vez hasta llegar una solución total
  - Si no se puede extender ejecutamos vuelta a atrás:
    - cronológico: se elimina la última decisión
    - no cronológico : se elimina una decisión anterior

Universitat d'Alacant
Universidad de Alicante

# Backtracking



Backtracking

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Backtracking

```
procedimiento Backtracking(k, V[n])    ; Llamada inicial: Backtracking(1, V[n])
inicio
V[k] = Selección(d_k)    ; Selecciona un valor de d_k para asignar a x_k
si Comprobar(k, V[n]) entonces
   si k = n entonces
      devolver V[n]    ; Es una solución
   si no
      Backtraking(k + 1, V[n])
   fin si
si no
   si quedan_valores(d_k) entonces
      Backtraking(k, V[n])
   si no
      si k = 1 entonces
         devolver ∅    ; Fallo
      si no
         Backtraking(k − 1, V[n])
      fin si
   fin si
fin si
fin Backtracking
```

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Limitaciones del backtracking

- Trashing e inconsistencia de nodo
  - Relacionado con las **restricciones unarias**. Sucede cuando un dominio contiene un valor que no satisface una restricción unaria.
- Inconsistencia de arista
  - Relacionado con las **restricciones binarias**. Sucede cuando existe una restricción binaria entre dos variables de tal forma que para un determinado valor de la primera variable no existe ninguna asignación posible para la segunda.
- Dependencia de la ordenación
  - El orden de selección de las variables es un factor crítico. Se han desarrollado diversas heurísticas de selección de variable y de valor.
    - **Variable:** Orden estático y Orden dinámico
    - **Valor:** p.e. los que conducen a un CPS más simple

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Forward Checking (FC)

- En cada etapa de la búsqueda, FC comprueba hacia delante la asignación actual con todos los valores de las futuras variables que están restringidas con la variable actual.
- Los valores de las variables futuras que son inconsistentes con la asignación actual son temporalmente eliminados de sus dominios.
- Si el dominio de una variable futura se queda vacío, la instanciación de la variable actual se deshace y se prueba con un nuevo valor. Si ningún valor es consistente, entonces se lleva a cabo el backtracking cronologico.

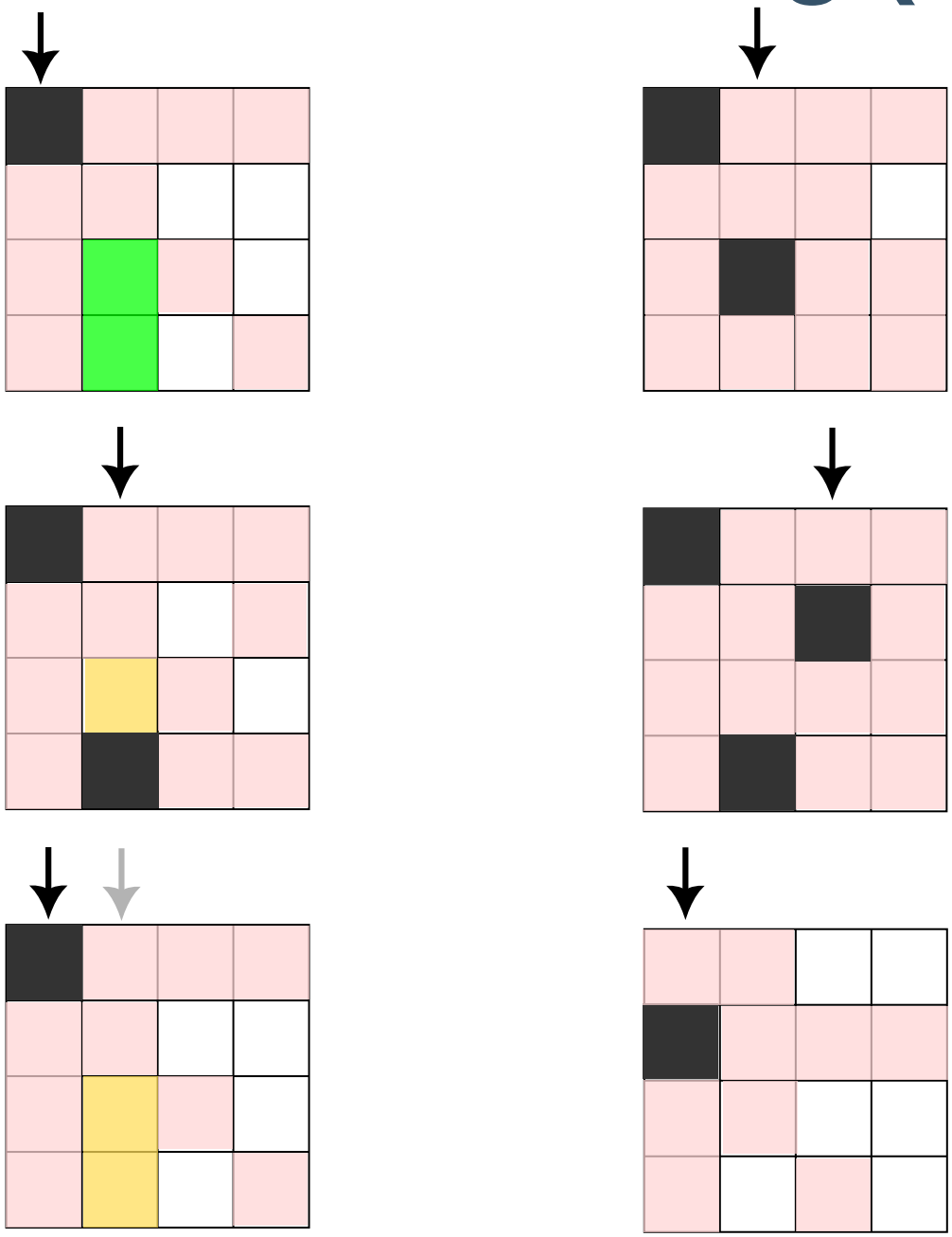
Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# FC

- Pseudocódigo intuitivo:

1. Seleccionar $x_i$.
2. Instanciar $x_i \leftarrow a_i : a_i \in D_i$.
3. Razonar hacia adelante (forward-check):
   - Eliminar de los dominios de las variables $(x_{i+1}, \ldots, x_n)$ aún no instanciadas, aquellos valores inconsistentes con respecto a la instanciación $(x_i, a_i)$, de acuerdo al conjunto de restricciones.
4. Si quedan valores posibles en los dominios de todas las variables por instanciar, entonces:
   - Si $i < n$, incrementar $i$, e ir al paso (1).
   - Si $i = n$, salir con la solución.
5. Si existe una variable por instanciar, sin valores posibles en su dominio, entonces retractar los efectos de la asignación $x_i \leftarrow a_i$. Hacer:
   - Si quedan valores por intentar en $D_i$, ir al paso (2).
   - Si no quedan valores:
     - Si $i > 1$, decrementar $i$ y volver al paso (2).
     - Si $i = 1$, salir sin solución.

# Forward Checking (FC)



Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# FC

- Algoritmo

**Algoritmo** Forward Checking

```
1:  function FC(i : variable)                                  ▷ Retorna booleano
2:      for all a ∈ dominio[i] do
3:          X_i ← a
4:          if i = N then return VERDADERO                     ▷ ¿ ← todas variables ?
5:          else
6:              if FORWARD(i, a) then
7:                  if FC(i + 1) = VERDADERO then return VERDADERO
8:              RESTAURAR(i)
        return FALSO
9:
10: function FORWARD(i : variable, a : valor)                  ▷ Retorna booleano
11:     dominio_vacio ← FALSO
12:     for j ← i + 1 to N do
13:         for all b ∈ dominio[j] do
14:             if not ES_CONSISTENTE(i, a, j, b) then
15:                 Eliminar b de dominio[j]
16:                 Añadir b a podado[j]
17:         if dominio[j] está vacío then
18:             dominio_vacio ← VERDADERO
19:             break
20:     if dominio_vacio then return FALSO
        return VERDADERO
21:
22: procedure RESTAURAR(i : variable)
23:     for j ← i + 1 to N do
24:         for all b ∈ podado[j] do
25:             if X_i es responsable del filtrado de b then
26:                 Eliminar b de podado[j]
27:                 Añadir b a dominio[j]
28:
29: function ES_CONSISTENTE(i : variable, a : valor, j : variable, b : valor)  ▷
    Retorna booleano
30:                                  ▷ ¿'a' en X_i es consistente con 'b' en X_j?
31:                                                  ▷ Depende del problema
```

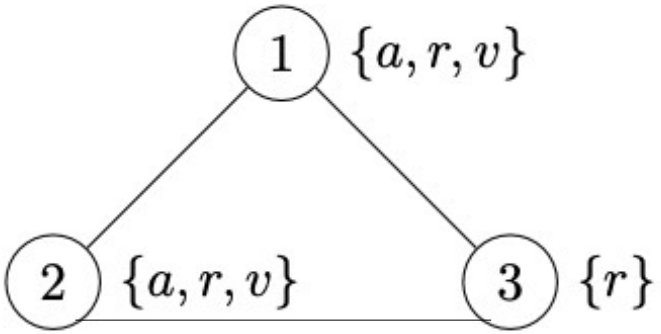
Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Ejemplo de FC

- Configuración inicial:
  - **Variables:** $x,y$
  - **Dominios:** $D_x = D_y = \{1,2,3,4,5\}$
  - **Restricción:** $x < y - 1$
- Inicialmente:
  - $CD_x = CD_y = \{1,2,3,4,5\}$
- Proceso:
  - **Caso 1**: Si asignamos x = 2
    - Los únicos valores que puede tomar y son 4, 5
    - Por tanto: $CD_y = \{4,5\}$
    - ...
  - **Caso 2**: Asignamos x = 4
    - No hay asignación posible compatible con la restricción
    - Por tanto: $CD_y = \{\}$
    - Acción: Deshacer x = 4 y realizar backtracking
    - ...

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Otro ejemplo de FC




1: $\{a, r, v\}$
2: $\{a, r, v\}$
3: $\{r\}$

- 1: {**a**,r,v} → 2: {a,r,v}
- 2:{**r**,v} → 3:{ } (por tanto volvemos hacia atrás)
- 2:{r,**v**} → 3:{r}
- 3:{r}

Solución: 1=a, 2=v, 3=r

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Propagación de restricciones

- Transformar el problema en otro más sencillo sin inconsistencias de arco.
- Propiedad de consistencia de arista:
  - **Una arista dirigida c(ep) = <Vi, Vj> es consistente si y sólo si para todo valor asignable a Vi existe al menos un valor en Vj que satisface la restricción asociada a la arista.**
- Un CSP puede transformarse en una red consistente mediante un algoritmo sencillo (AC3) que examina las aristas, eliminando los valores que causan inconsistencia del dominio de cada variable.
- Después del proceso:
  - No hay solución
  - Hay más de una solución
  - Hay una única solución

Universitat d'Alacant
Universidad de Alicante

# Algoritmo AC3

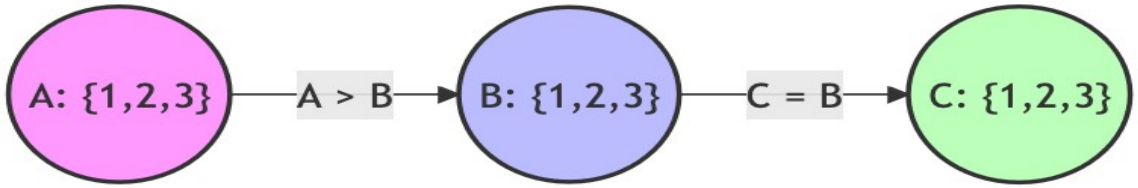
**Algoritmo** Algoritmo AC3

```
1:  Q ← {c(e_p) = ⟨V_i, V_j⟩ | e_p ∈ E, i ≠ j}
2:  while Q ≠ ∅ do
3:      ⟨V_k, V_m⟩ ← seleccionar_y_borrar(Q)
4:      cambio ← falso
5:      for all v_k ∈ D_k do
6:          if no_consistente(v_k, D_m) then
7:              borrar(v_k, D_k)
8:              cambio ← cierto
9:      if D_k = ∅ then
10:         salir_sin_solución
11:     if cambio = cierto then
12:         Q ← Q ∪ {c(e_r) = ⟨V_i, V_k⟩ | e_r ∈ E, i ≠ k, i ≠ m}
```

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo AC3




## Dominios

- A={1,2,3}
- B={1,2,3}
- C={1,2,3}

## Q

## Aristas

A>B

B<A

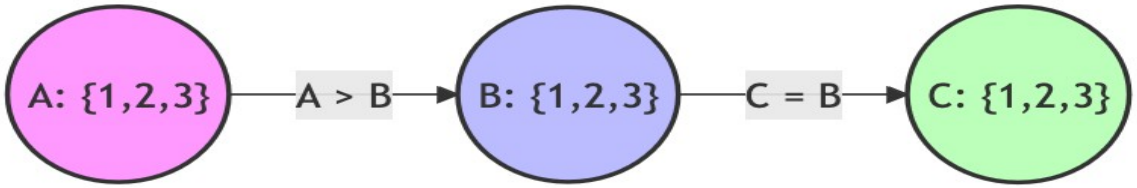
B=C

C=B

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Ejemplo AC3




| Dominios | Q | Aristas |
|---|---|---|
| • A={1,2,3} | • A>B | A>B |
| • B={1,2,3} | • B<A | B<A |
| • C={1,2,3} | • B=C | B=C |
| | • C=B | C=B |

- Aplicamos la línea 1

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo AC3

| Dominios | Q | Aristas |
|---|---|---|
| • A={1,2,3} | • A>B | A>B |
| • B={1,2,3} | • B<A | B<A |
| • C={1,2,3} | • B=C | B=C |
| | • C=B | C=B |

- Cogemos la primera restricción (línea 3)
  - Para cada valor de A (parte izquierda de la regla) ¿se cumple? Sino eliminar

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Ejemplo AC3

| Dominios | Q | Aristas |
|---|---|---|
| • A={1,2,3} | • A>B | A>B |
| • B={1,2,3} | • B<A | B<A |
| • C={1,2,3} | • B=C | B=C |
| | • C=B | C=B |

- Como **A** cambia añadimos a **Q** cualquier arista que tenga **A** en la parte derecha si no esta ya.
- ¡En esta caso no se añade **B<A** porqué ya está! (línea 12)

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo AC3

| Dominios | Q | Aristas |
|---|---|---|
| • A={1,2,3} | • ~~A>B~~ | A>B |
| • B={1,2,3} | • B<A | B<A |
| • C={1,2,3} | • B=C | B=C |
| | • C=B | C=B |

- Cogemos la segunda restricción (línea 3)
- Para cada valor de B ¿se cumple? Sino eliminar

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Ejemplo AC3

| Dominios | Q | Aristas |
|---|---|---|
| • A={1,2,3} | • ~~A>B~~ | A>B |
| • B={1,2,3} | • B<A | B<A |
| • C={1,2,3} | • B=C | B=C |
| | • C=B | C=B |

- B ha cambiado, por lo que añadimos las restricciones que tengan en su parte derecha a B si no estaban en Q...

# Ejemplo AC3

## Dominios

- A={ ,2,3}
- B={1,2, }
- C={1,2,3}

## Q

- ~~A>B~~
- B<A
- B=C
- C=B
- A>B

## Aristas

A>B

B<A

B=C

C=B

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Ejemplo AC3

## Dominios

- A={ ,2,3}
- B={1,2, }
- C={1,2,3}

## Q

- ~~A>B~~
- ~~B<A~~
- B=C
- C=B
- A>B

## Aristas

A>B

B<A

B=C

C=B

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo AC3

## Dominios

- A={ ,2,3}
- B={1,2, }
- C={1,2,3}

## Q

- ~~A>B~~
- ~~B<A~~
- B=C
- C=B
- A>B

## Aristas

A>B

B<A

B=C

C=B

- Ningún cambio al aplicar esta regla...

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo AC3

## Dominios

- A={1,2,3}
- B={1,2,3}
- C={1,2,3}

## Q

- ~~A>B~~
- ~~B<A~~
- ~~B=C~~
- C=B
- A>B

## Aristas

A>B

B<A

B=C

C=B

- Seleccionamos la siguiente y continuamos el mismo proceso...

Universitat d'Alacant
Universidad de Alicante

# Ejemplo AC3

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

## Dominios

- A={ ,2,3}
- B={1,2, }
- C={1,2, }

## Q

- ~~A>B~~
- ~~B<A~~
- ~~B=C~~
- C=B
- A>B

## Aristas

A>B

B<A

B=C

C=B

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo AC3

## Dominios

- A={ ,2,3}
- B={1,2, }
- C={1,2, }

## Q

- ~~A>B~~
- ~~B<A~~
- ~~B=C~~
- C=B
- A>B
- B=C

## Aristas

A>B

B<A

B=C

C=B

- Como se modifico el domino de C incluimos las restricciones con ella en la parte derecha...

Universitat d'Alacant
Universidad de Alicante

# Ejemplo AC3

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

## Dominios

- A={ ,2,3}
- B={1,2, }
- C={1,2, }

## Q

- ~~A>B~~
- ~~B<A~~
- ~~B=C~~
- ~~C=B~~
- A>B
- B=C

## Aristas

A>B

B<A

B=C

C=B

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo AC3

## Dominios

- A={ ,2,3}
- B={1,2, }
- C={1,2, }

## Q

- ~~A>B~~
- ~~B<A~~
- ~~B=C~~
- ~~C=B~~
- ~~A>B~~
- B=C

## Aristas

A>B

B<A

B=C

C=B

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

# Ejemplo AC3

## Dominios

- A={ ,2,3}
- B={1,2, }
- C={1,2, }

## Q

- ~~A>B~~
- ~~B<A~~
- ~~B=C~~
- ~~C=B~~
- ~~A>B~~
- ~~B=C~~

## Aristas

A>B

B<A

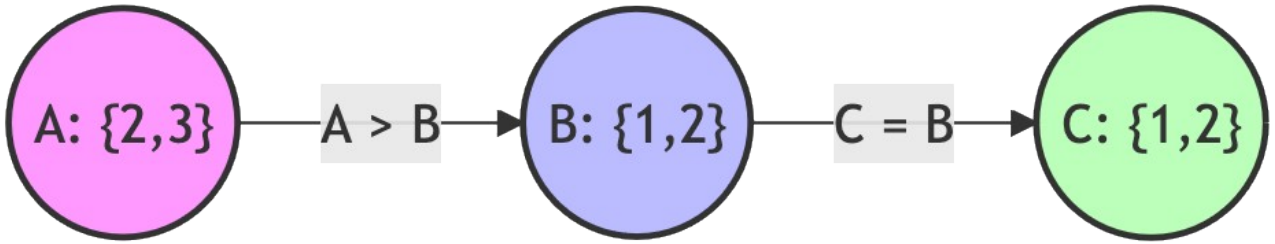
B=C

C=B

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Ejemplo AC3




## Dominios

- A={[redacted],2,3}
- B={1,2,[redacted]}
- C={1,2,[redacted]}

## Q

- ~~A>B~~
- ~~B<A~~
- ~~B=C~~
- ~~C=B~~
- ~~A>B~~
- ~~B=C~~

## Aristas

A>B

B<A

B=C

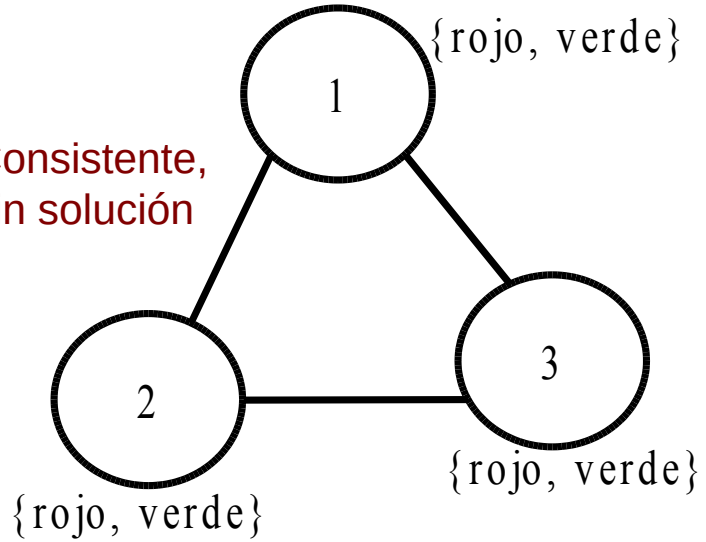
C=B

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

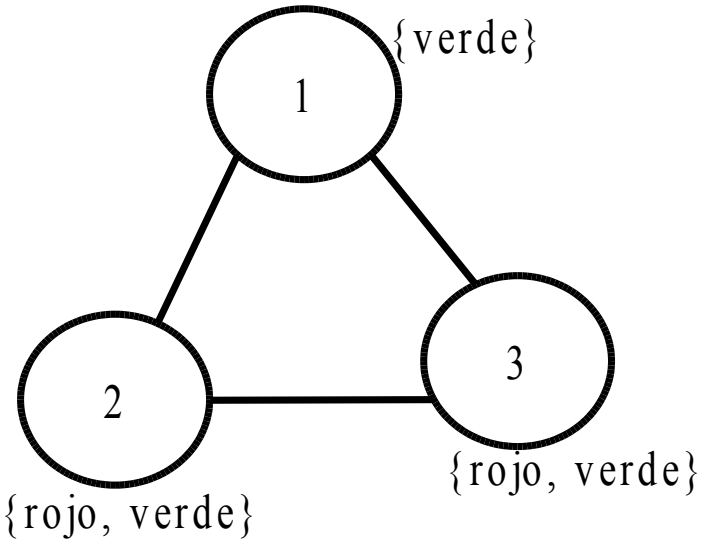
# Ejemplos de grafos

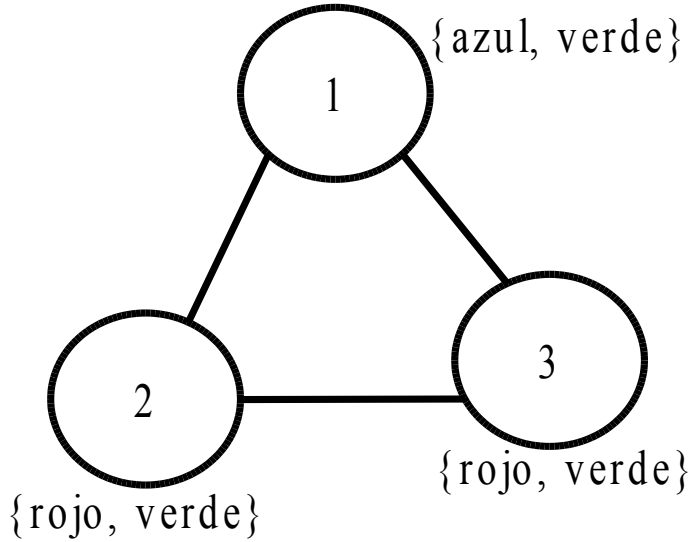
**Ejercicio: aplicar AC3**
(coloreado de mapas)

Consistente, sin solución



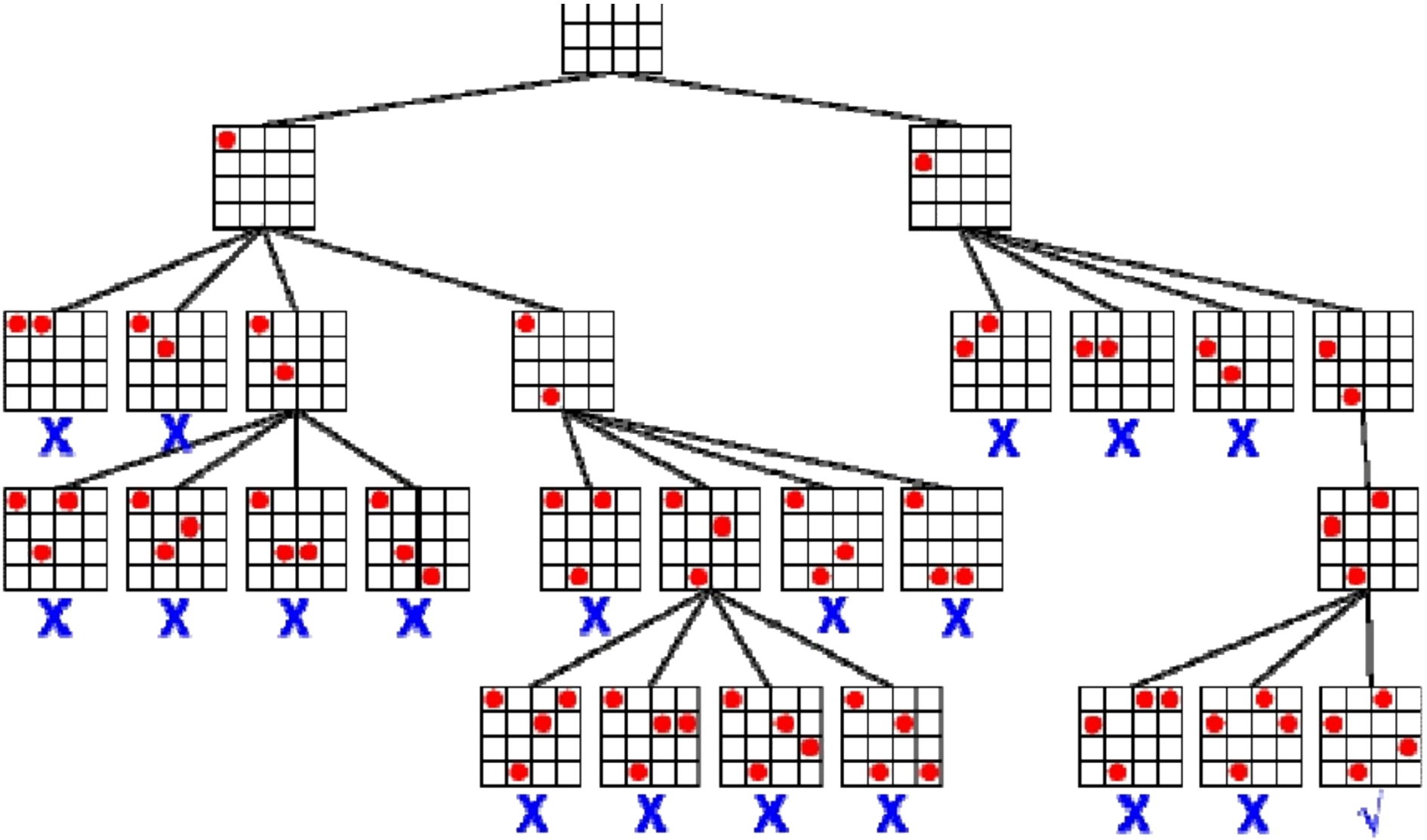

Inconsistente, sin solución




Consistente, con dos soluciones

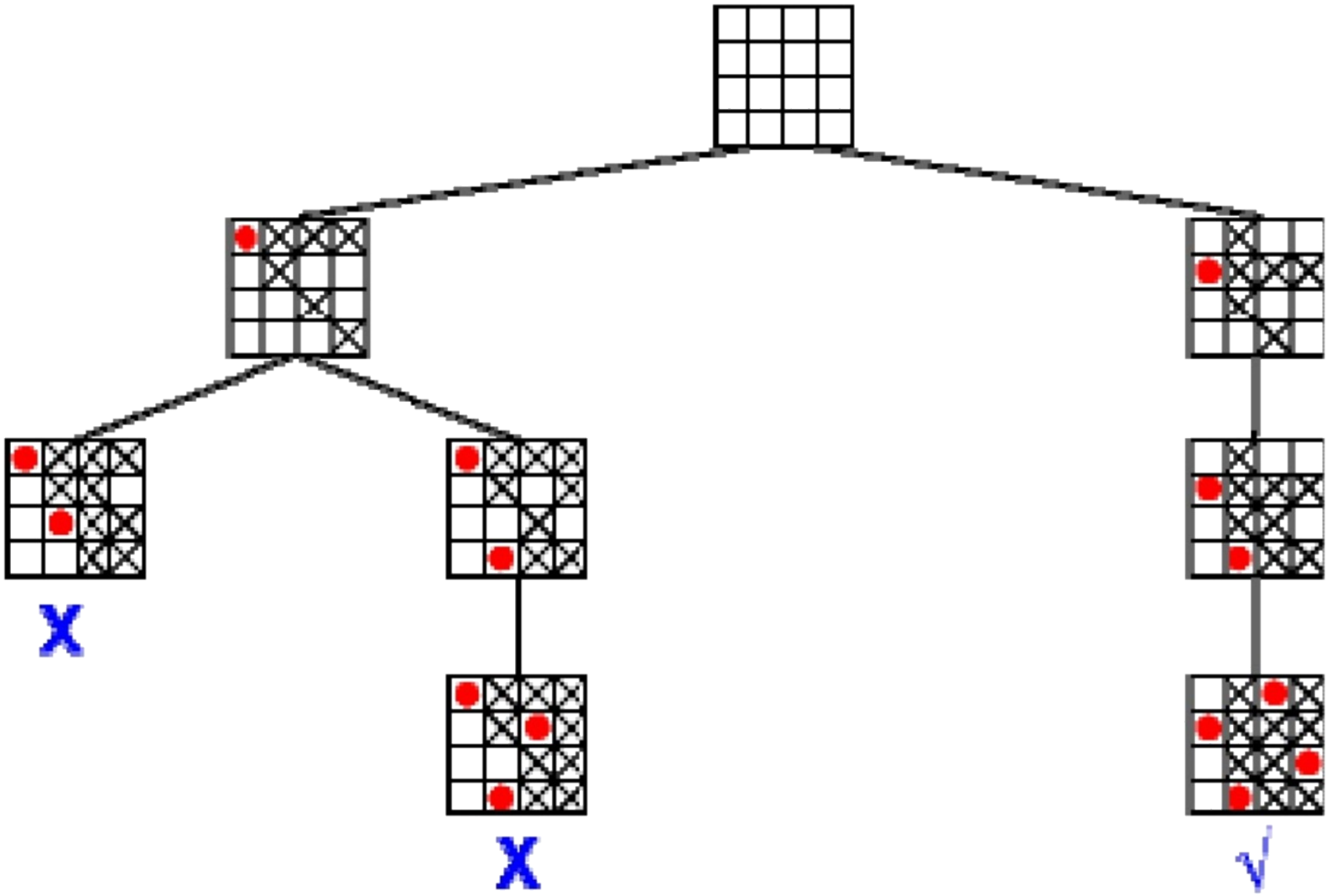
# Ejemplo: AC3 + backtracking



Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

# Ejemplo: AC + FC



# Bibliografía


- Stuart Russell, Peter Noving. “Artificial Intelligence: A Modern Approach, Global Edition ” Ed. Pearson. Prentice Hall.  2021.


Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante