

# Bài toán thỏa mãn ràng buộc

## Trí tuệ nhân tạo

HK1, 2022 - 2023

# Nội dung

- 1 Mô tả bài toán
- 2 Lan truyền ràng buộc: suy luận (inference) trong CSPs
- 3 Backtracking Search

# Tìm kiếm gì?

- Các giả định: một tác tử duy nhất, các hành động xác định, trạng thái được quan sát đầy đủ, không gian trạng thái rời rạc
- Lập kế hoạch (Planning): chuỗi các hành động
  - Quan trọng: đường đi đến đích
  - Đường đi: chi phí, độ sâu (depth) khác nhau
  - Heuristics hướng dẫn theo vấn đề
- Nhận dạng (Identification): các phép gán cho các biến
  - Quan trọng: bản thân của trạng thái đích, không phải đường đến đích
  - Tất cả các đường đi cùng độ sâu
  - CSP dùng cho các bài toán nhận dạng

# Mô tả bài toán

Bài toán thỏa mãn ràng buộc (constraint satisfaction problem): tập con của bài toán tìm kiếm. Bài toán gồm 03 thành phần

- $X$ : tập hợp các biến  $\{X_1, \dots, X_n\}$
- $D$ : tập hợp các miền giá trị  $\{D_1, \dots, D_n\}$  ứng với mỗi biến
- $C$ : tập hợp các ràng buộc.

Ví dụ:

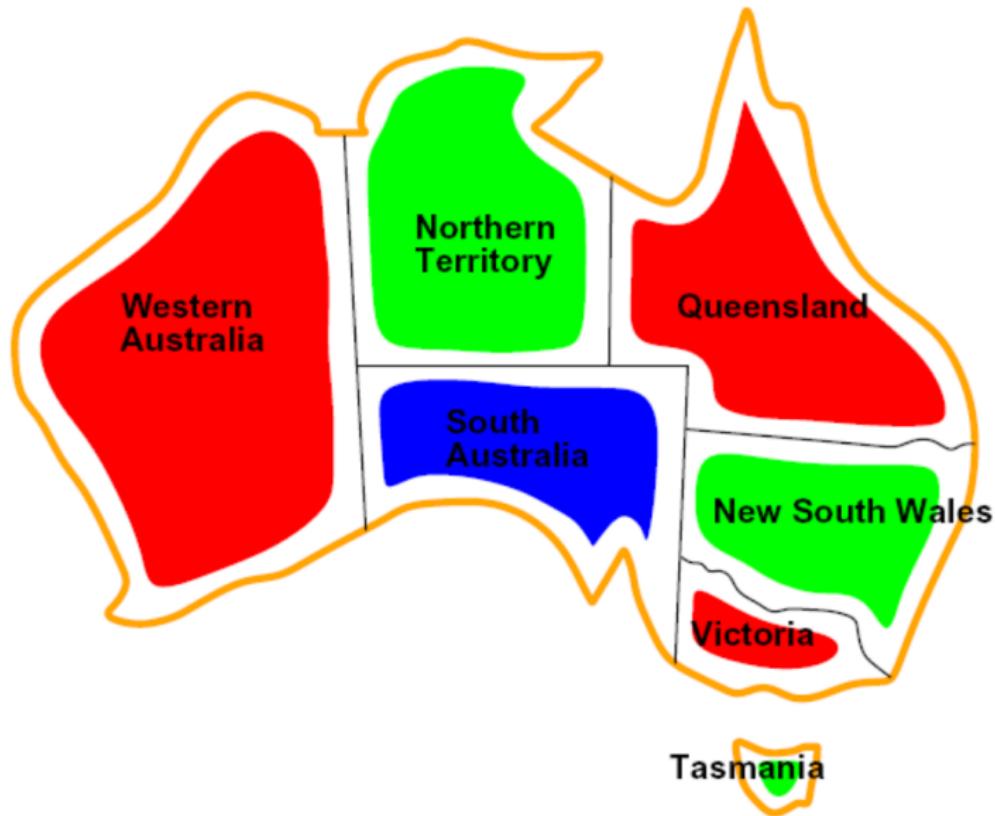
- 02 biến  $X_1, X_2$
- Miền giá trị của 02 biến:  $D_1 = D_2 = \{A, B\}$
- Ràng buộc: giá trị  $X_1$  khác với giá trị  $X_2$  thể hiện  $\langle(X_1, X_2), X_1 \neq X_2\rangle$

# Mô tả bài toán

## Giải quyết một CSP

- Trạng thái định nghĩa bằng một phép gán:  
 $\{\dots X_i = v_i, X_j = v_j \dots\}$
- Phép gán **nhất quán/hợp lệ** (consistent/legal): không vi phạm ràng buộc nào
- Phép gán **trọn vẹn** (complete): tất cả các biến được gán giá trị
- Giải pháp (solution): phép gán hợp lệ, trọn vẹn

# Ví dụ - Tô màu bản đồ



# Ví dụ - Tô màu bản đồ

- Biến là các bang:

$$X = \{WA, NT, Q, NSW, V, SA, T\}$$

- Miền giá trị cho mỗi biến:

$$D_i = \{red, green, blue\}$$

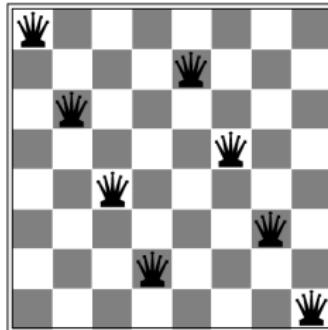
- Ràng buộc:

$$C = \{SA \neq WA, SA \neq NT, SA \neq Q, SA \neq NSW, SA \neq V, WA \neq NT, NT \neq Q, Q \neq NSW, NSW \neq V\}$$

- Giải pháp:

$$\{WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = red\}$$

# Ví dụ - 8 Queens



- Biến là vị trí các quân hậu tại các cột:  $X = \{Q_1, \dots, Q_8\}$
- Miền giá trị cho mỗi biến:  $D_i = \{1, \dots, 8\}$
- Ràng buộc:  $Q_1 \neq Q_2, Q_1 \neq Q_2 - 1, Q_1 \neq Q_2 + 1, Q_1 \neq Q_3, Q_1 \neq Q_3 - 2, Q_1 \neq Q_3 + 2, \dots$
- Giải pháp:  $\{Q_1 = 1, Q_2 = 3, Q_3 = 5, Q_4 = 7, Q_5 = 2, Q_6 = 4, Q_7 = 6, Q_8 = 8\}$

# CSP như bài toán tìm kiếm

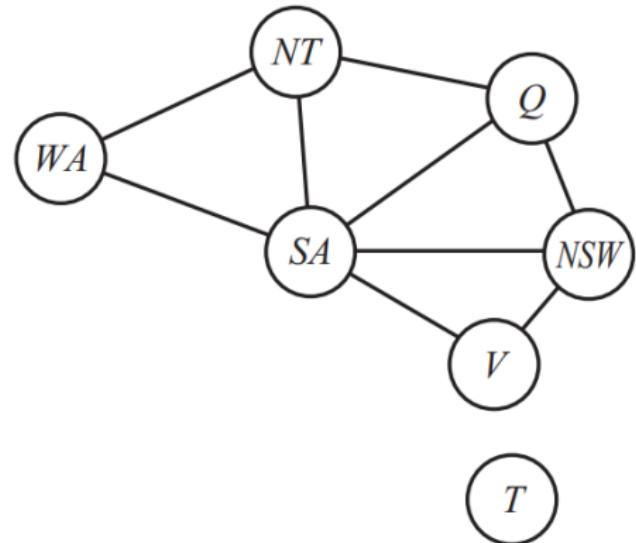
- **trạng thái (states)**: phép gán nhất quán
- **trạng thái bắt đầu (initial state)**: phép gán rỗng
- **hàm trạng thái kế tiếp**: gán một biến chưa được gán sao cho phép gán vẫn là nhất quán
- **kiểm tra đích**: phép gán trọn vẹn
- **chi phí đường đi**: hằng chi phí = 1 cho mỗi bước

# Các kiểu ràng buộc

Ràng buộc có thể được phân loại bằng số lượng các biến mà nó phụ thuộc:

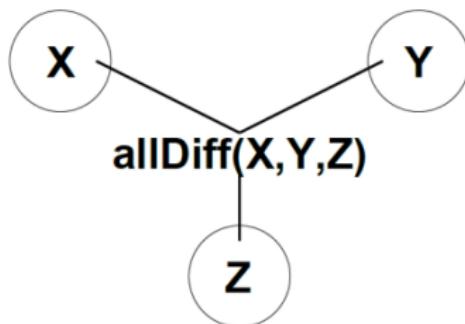
- **Ràng buộc đơn (unary constraint)**: phụ thuộc vào một biến  $X_i$
- **Ràng buộc nhị phân (binary constraint)**: phụ thuộc vào 02 biến  $X_i$  và  $X_j$ , có thể biểu diễn bằng đồ thị ràng buộc
- **Ràng buộc bậc cao ( $n$ -ary constraint)**: phụ thuộc *nhiều hơn 02 biến*

# Đồ thị ràng buộc - Tô màu bản đồ

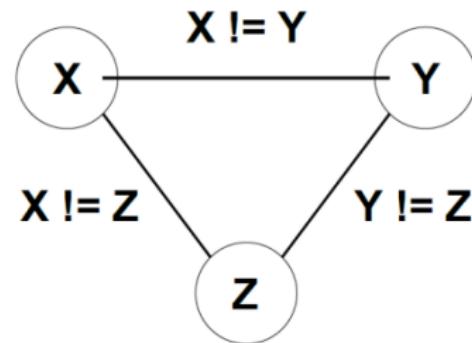


# Ràng buộc bậc cao

Có thể giảm thành ràng buộc nhị phân



constraint hypergraph



binarized constraint graph

# Lan truyền ràng buộc



WA	NT	Q	NSW	V	SA	T
Red	Green	Blue	Red	Green	Blue	Red
Red		Green	Blue	Red	Green	Blue
Red			Blue	Red	Green	Blue

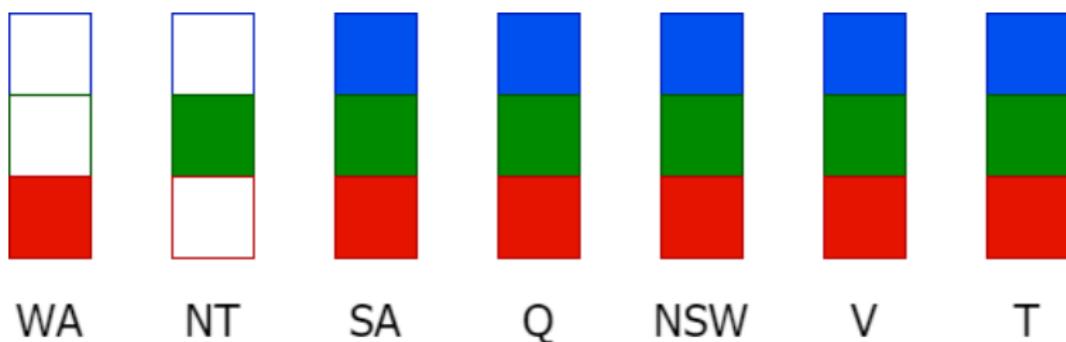
- NT và SA không thể cùng màu xanh (blue)
- Có cách kiểm tra sớm không?

# Lan truyền ràng buộc

- Lan truyền ràng buộc (constraint propagation): sử dụng ràng buộc để *giảm số lượng giá trị hợp lệ của biến*
  - Đan xen với tìm kiếm
  - Tiền xử lý trước khi bắt đầu tìm kiếm
- Ý tưởng từ tính nhất quán cục bộ (local consistency): xem mỗi biến là một nút và mỗi ràng buộc nhị phân là một cung thì quá trình thực thi tính nhất quán cục bộ trong mỗi phần của đồ thị sẽ loại bỏ các giá trị không nhất quán

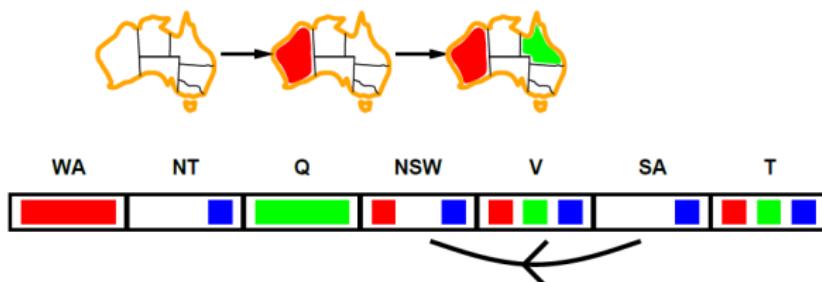
# Nhất quán nút (node consistency)

- Một biến đơn gọi là nhất quán về nút (node-consistent) nếu:
  - tất cả giá trị trong miền giá trị thỏa mãn ràng buộc đơn của biến
  - một mạng ràng buộc được gọi là node-consistent nếu mọi biến là node-consistent
- Ví dụ: bổ sung thêm ràng buộc cho bài toán tô màu  
 $WA = \text{red} \wedge NT = \text{green}$

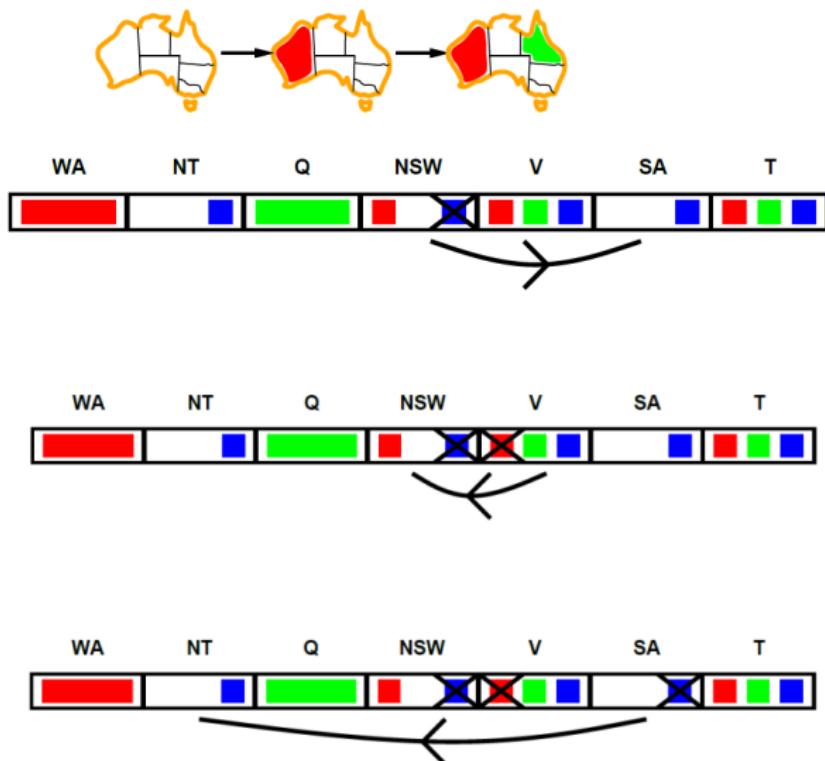


# Nhất quán cung (arc consistency)

- Một biến đơn gọi là nhất quán về cung (arc-consistent) nếu:
  - mọi giá trị trong miền giá trị của nó thỏa mãn các ràng buộc của biến
  - $X_i$  là arc-consistent đối với  $X_j$  nếu với mọi giá trị trong miền giá trị  $D_i$  sẽ có một giá trị trong miền  $D_j$  thỏa mãn cung  $(X_i, X_j)$
  - Để  $(X_i, X_j)$  đảm bảo arc-consistent  $\Rightarrow$  loại bỏ giá trị trong  $D_i$  làm ràng buộc không thỏa mãn



# Nhất quán cung (arc consistency)



# Nhất quán cung (arc consistency)

**function** AC-3(*csp*) **returns** false if an inconsistency is found and true otherwise

**inputs:** *csp*, a binary CSP with components (*X*, *D*, *C*)

**local variables:** *queue*, a queue of arcs, initially all the arcs in *csp*

**while** *queue* is not empty **do**

$(X_i, X_j) \leftarrow \text{REMOVE-FIRST}(\text{queue})$

**if** REVISE(*csp*, *X<sub>i</sub>*, *X<sub>j</sub>*) **then**

**if** size of *D<sub>i</sub>* = 0 **then return** false

**for each** *X<sub>k</sub>* **in** *X<sub>i</sub>.NEIGHBORS - {X<sub>j</sub>}* **do**

            add (*X<sub>k</sub>*, *X<sub>i</sub>*) to *queue*

**return** true

---

**function** REVISE(*csp*, *X<sub>i</sub>*, *X<sub>j</sub>*) **returns** true iff we revise the domain of *X<sub>i</sub>*

*revised*  $\leftarrow$  false

**for each** *x* **in** *D<sub>i</sub>* **do**

**if** no value *y* in *D<sub>j</sub>* allows (*x,y*) to satisfy the constraint between *X<sub>i</sub>* and *X<sub>j</sub>* **then**

            delete *x* from *D<sub>i</sub>*

*revised*  $\leftarrow$  true

**return** *revised*

# Tìm kiếm quay lui (Backtracking Search)

- Giải thuật tìm kiếm được sử dụng phổ biến trong CSP
- dựa trên giải thuật tìm kiếm theo chiều sâu (DFS)
- Vấn đề đặt ra:
  - Thứ tự biến
  - Thứ tự giá trị

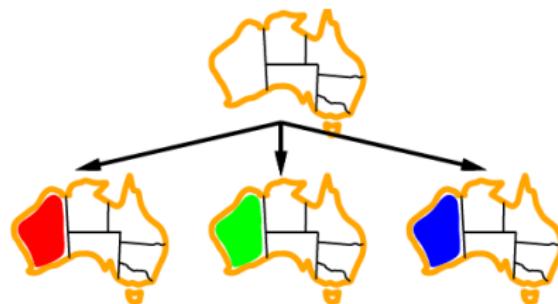
# Tìm kiếm quay lui (Backtracking Search)

```
function BACKTRACKING-SEARCH(csp) returns a solution, or failure
    return BACKTRACK({ }, csp)
function BACKTRACK(assignment, csp) returns a solution, or failure
    if assignment is complete then return assignment
    var  $\leftarrow$  SELECT-UNASSIGNED-VARIABLE(csp)
    for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
        if value is consistent with assignment then
            add {var = value} to assignment
            inferences  $\leftarrow$  INFERENCE(csp, var, value)
            if inferences  $\neq$  failure then
                add inferences to assignment
                result  $\leftarrow$  BACKTRACK(assignment, csp)
                if result  $\neq$  failure then
                    return result
                remove {var = value} and inferences from assignment
    return failure
```

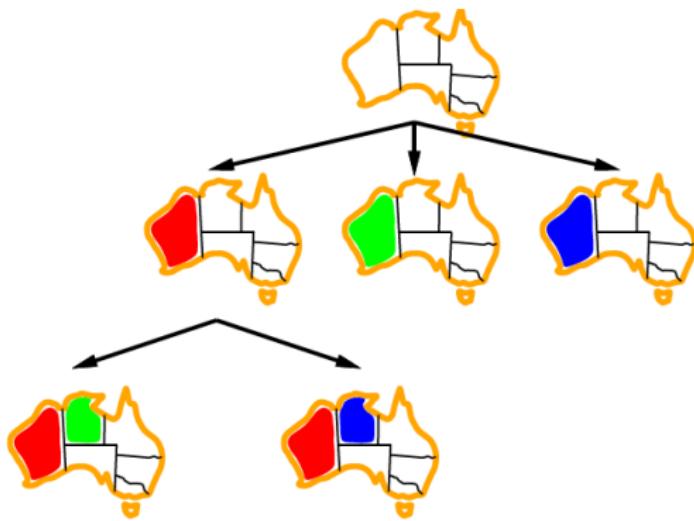
# Ví dụ



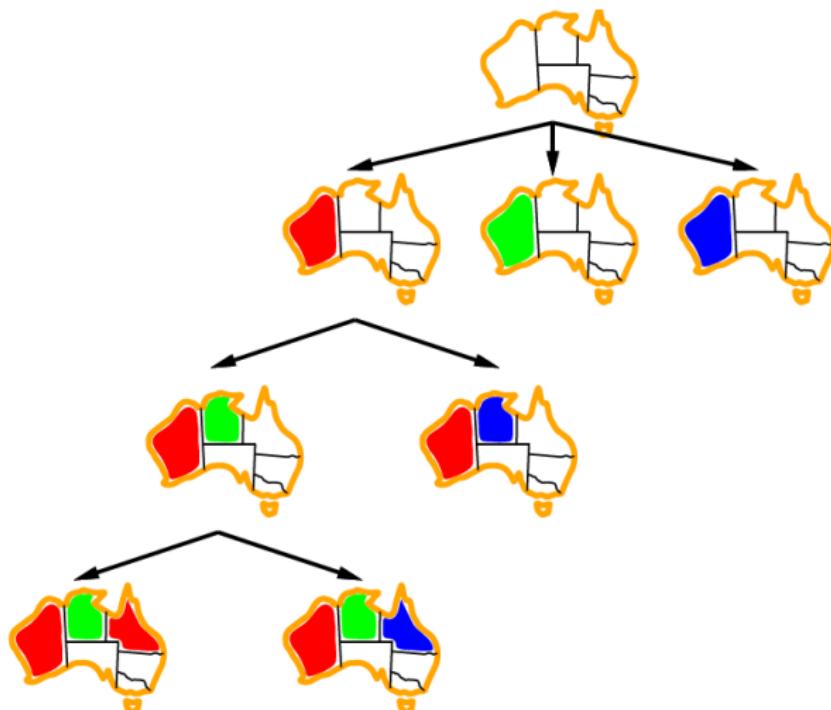
# Ví dụ



# Ví dụ

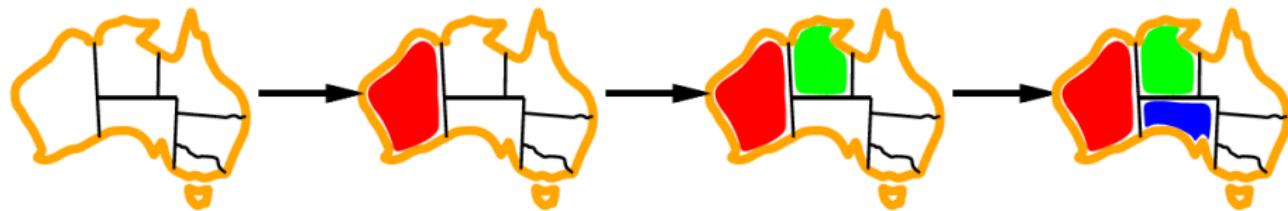


# Ví dụ



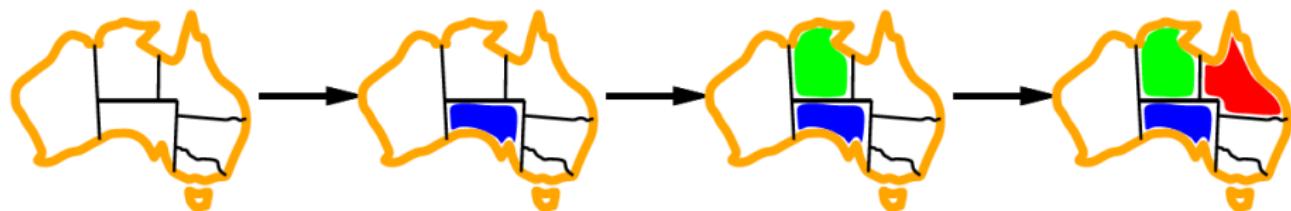
# Thứ tự biến

**Quy tắc ưu tiên các biến có tập giá trị hợp lệ nhỏ nhất - minimum remaining values (MRV):** chọn biến với số lượng tùy chọn giá trị ít nhất



# Thứ tự biến

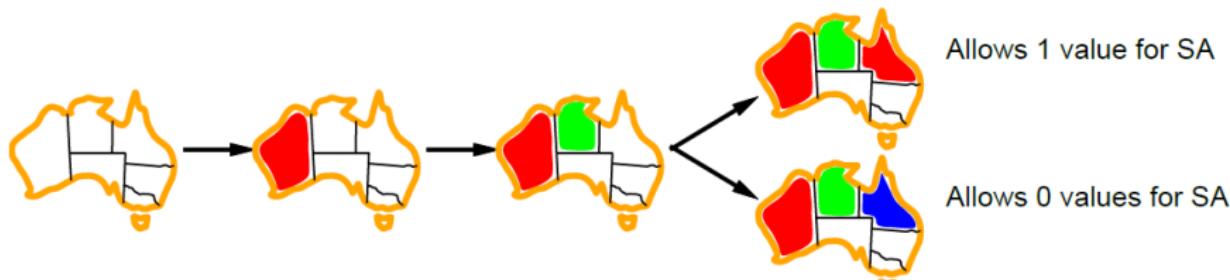
**degree heuristics:** chọn biến tham gia vào nhiều ràng buộc chưa được giải quyết nhất



# Thứ tự biến

## Giá trị ràng buộc ít nhất (least constraining value):

- chọn một giá trị loại trừ ít giá trị nhất trong các biến chưa được gán còn lại



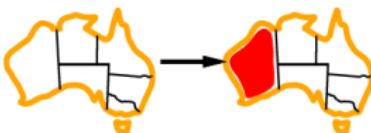
# Loại bỏ các lỗi sớm: Kiểm tra tiến (Forward checking)

- khi một giá trị được gán cho một biến, giá trị này sẽ không hợp lệ đối với biến khác sẽ bị xóa
- triển khai những sắp xếp heuristics tính toán ngầm định
- Ví dụ: WA = red thì NT không thể là red
- nếu tất cả giá trị của một biến bị xóa → DỪNG

# Forward checking



WA	NT	Q	NSW	V	SA	T
Red	Green	Blue	Red	Green	Blue	Red



WA	NT	Q	NSW	V	SA	T
Red	Green	Blue	Red	Green	Blue	Red
Red	Green	Blue	Red	Green	Blue	Red

# Forward checking



WA	NT	Q	NSW	V	SA	T
Red Green Blue						
Red		Green	Red	Red	Green	Red



WA	NT	Q	NSW	V	SA	T
Red Green Blue						
Red		Green	Red	Red	Green	Red
Red		Green	Red		Green	Red
Red		Green	Red		Green	Red

# Bài tập - 4 Queens

- biến?
- giá trị?
- ràng buộc?
- đồ thị ràng buộc?

# Bài tập - Mạng ràng buộc nhị phân

Cho 4 biến:  $X_1, X_2, X_3, X_4$

Miền giá trị:  $D_1 = \{1, 2, 3, 4\}$ ,  $D_2 = \{3, 4, 5, 8, 9\}$ ,  $D_3 = \{2, 3, 5, 6, 7, 9\}$ ,  $D_4 = \{3, 5, 7, 8, 9\}$

Ràng buộc:  $X_1 \geq X_2$ ,  $X_2 > X_3$  or  $X_3 - X_2 = 2$ ,  
 $X_3 \neq X_4$

- Vẽ đồ thị ràng buộc
- Mạng arc-consistent? Nếu không, thực hiện việc chuyển đổi để thành arc-consistent.
- Đưa ra một giải pháp