

# CONSTRAINT SATISFACTION PROBLEMS

## (Thỏa mãn ràng buộc)

### Ví dụ minh họa 1: Map Coloring

- **Variables:**  $X = \{WA, NT, Q, NSW, V, SA, T\}$
- **Domains:**  $D_i = \{\text{red}, \text{green}, \text{blue}\}$
- **Constraints:** Adjacent regions must have different colors
 
$$C = \left\{ \begin{array}{l} SA \neq WA, SA \neq NT, SA \neq Q, SA \neq NSW, SA \neq V, \\ WA \neq NT, NT \neq Q, Q \neq NSW, NSW \neq V \end{array} \right\}$$

- where  $SA \neq WA$  is a shortcut of  $\{(SA, WA), SA \neq WA\}$
- $SA \neq WA$  can be fully enumerated as  $\{(\text{red}, \text{green}), (\text{red}, \text{blue}), (\text{green}, \text{red}), (\text{green}, \text{blue}), (\text{blue}, \text{red}), (\text{blue}, \text{green})\}$

- There are many possible solutions

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



### Ví dụ minh họa 2: Toy problems in CSP

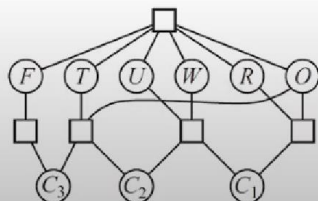
#### 4-Queens Problem

- **Variables:**  $Q1, Q2, Q3, Q4$
- **Domains:**  $D = \{1, 2, 3, 4\}$
- **Constraints**
  - $Q_i \neq Q_j$  (cannot be in the same row)
  - $Q_i - Q_j \neq i - j$  (cannot be in the same diagonal)

	Q1	Q2	Q3	Q4
1				
2				
3				
4				

#### The Cryptarithmic

$$\begin{array}{r} T W O \\ + T W O \\ \hline F O U R \end{array}$$



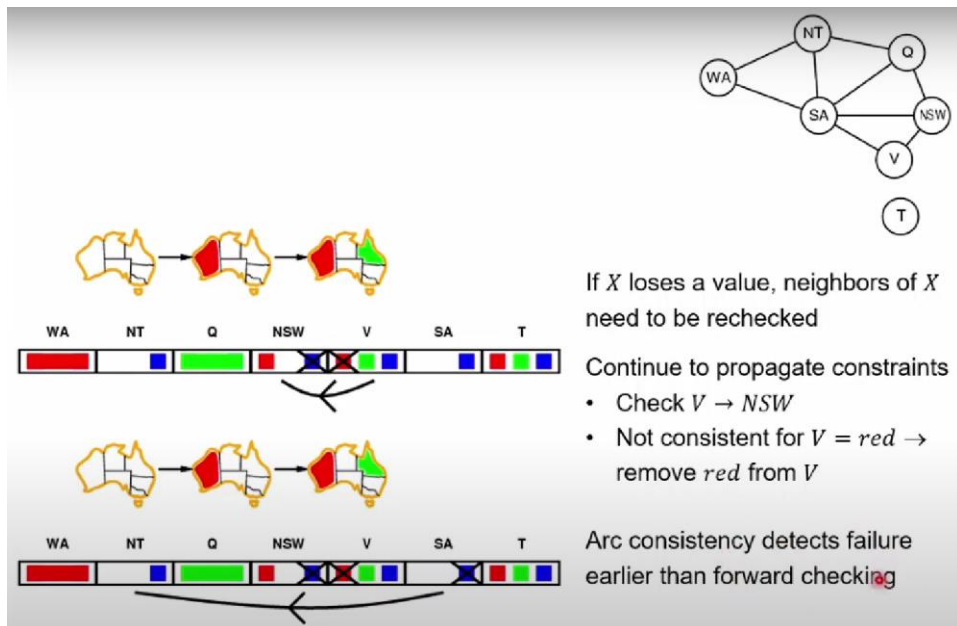
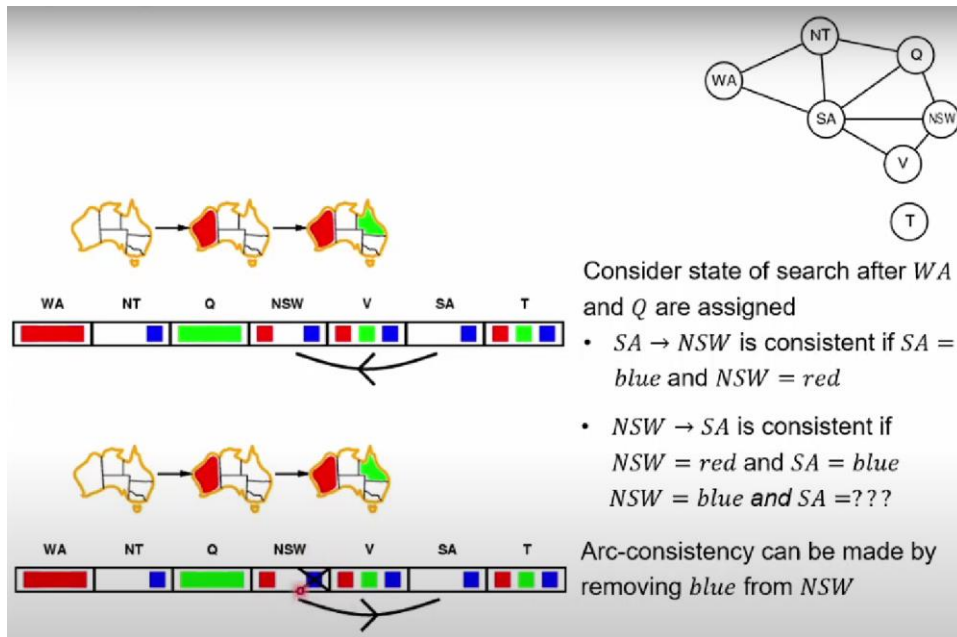
- **Variables:**  $F T U W R O C_1 C_2 C_3$
- **Domains:**  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- **Constraints:**
  - $AllDiff(F, T, U, W, R, O)$
  - $C_3 = F, T \neq 0, F \neq 0$
  - ...

# Arc consistency

Liên quan đến ràng buộc nhị phân.

E.g.  $\langle (X, Y), \{(0, 0), (1, 1), (2, 4), (3, 9)\} \rangle \rightarrow X \text{ domain } \{0, 1, 2, 3\} \text{ and } Y \{0, 1, 4, 9\}$

Có những tình huống không hiệu quả.



## AC-3 Algorithm (Vở ghi)

# Backtracking Search (Lý thuyết vở ghi)

```
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 ← 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 ← INFERENCE(csp, var, value)
```

```
    if inferences ≠ failure then
```

```
      add inferences to assignment
```

```
      result ← BACKTRACK(assignment, csp)
```

```
      if result ≠ failure then
```

```
        return result
```

```
    remove {var = value} and inferences from assignment
```

```
return failure
```

Which variable should  
be assigned next?

In what order should  
its values be tried?

What inferences  
should be performed?

## Phương pháp tìm kiếm quay lui đối với bài toán CSP

- Gán giá trị lần lượt cho các biến – Việc gán giá trị của biến này chỉ được làm sau khi đã hoàn thành việc gán giá trị của biến khác
- Sau mỗi phép gán giá trị cho một biến nào đó, kiểm tra các ràng buộc có được thỏa mãn bởi tất cả các biến đã được gán giá trị cho đến thời điểm hiện tại – Quay lui (backtrack) nếu có lỗi (không thỏa mãn các ràng buộc)

và quay lại khi biến <sup>ko</sup> có giá trị hợp lệ.

- Thuật giải:

+ Đầu tiên các phép gán rỗng h/

+ Nếu các phép gán đầy đủ thì trả về các phép gán  
(if assignment is complete then return assignment)

+ Chọn 1 biến chưa được gán ghi để gán giá trị.

Sd heuristic để tìm ra biến nào nên được gán tiếp theo:

$var \leftarrow \text{SELECT\_UNASSIGNED\_VARIABLE}(csp)$

( $\text{ORDER\_DOMAIN\_VALUES}(var, assignment, csp)$ ): Thứ tự  
hệ mã các giá trị của nó nên thử)

+ for each  $va$  value in  $\text{ORDER\_DOMAIN\_VALUES}(var, assignment, csp)$  do

→ Nếu như giá trị phù hợp với phép gán thì

. thêm  $hvar = value$  vào phép gán.

.  $inferences \leftarrow \text{INFERENCE}(csp, var, value)$ : Thuộc

hiện việc suy diễn, lan truyền ràng buộc ntn.

. Nếu phép suy diễn ko thất bại thì bổ sung vào phép

gán của mình.

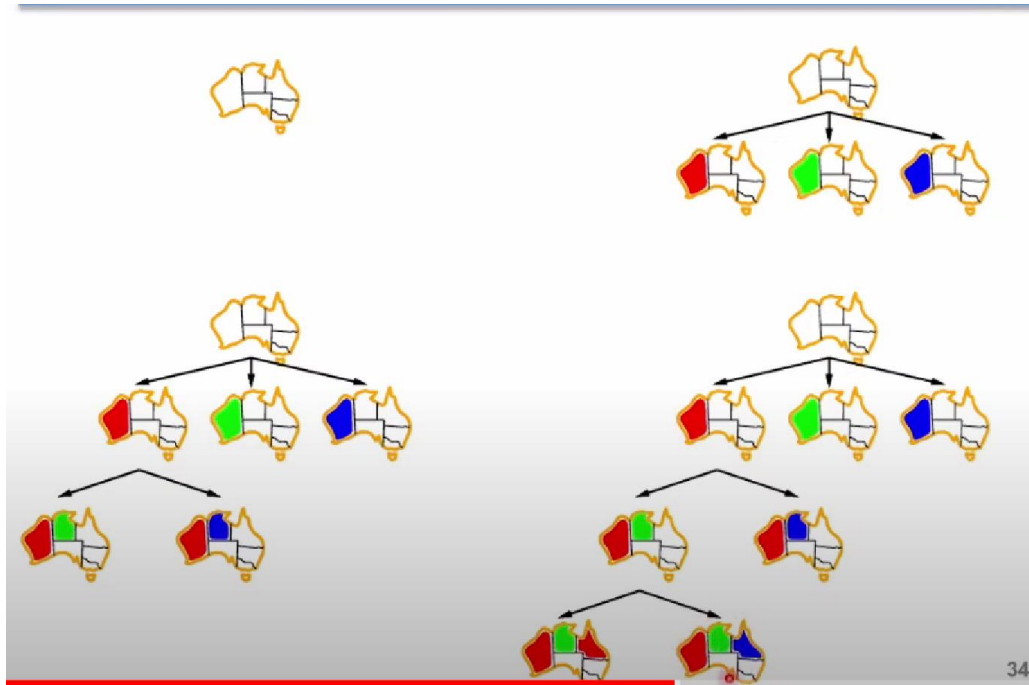
$result \leftarrow \text{BACKTRACK}(assignment, va, csp)$ : Suy

dẫn cho biến kế tiếp.

. Nếu phép suy diễn thất bại thì bỏ qua bỏ  $hvar = value$

và tiếp tục suy diễn ra khỏi phép gán.

→ Nếu sau vòng lặp ko tìm được giá trị thì trả về failure



## Các Heuristic giúp đánh giá lựa chọn đường đi

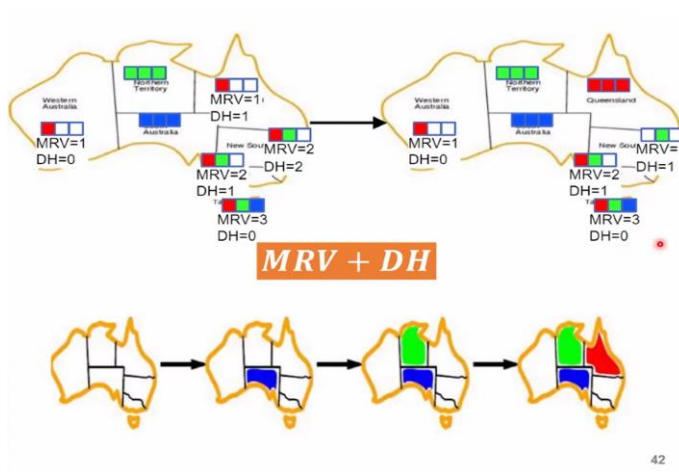
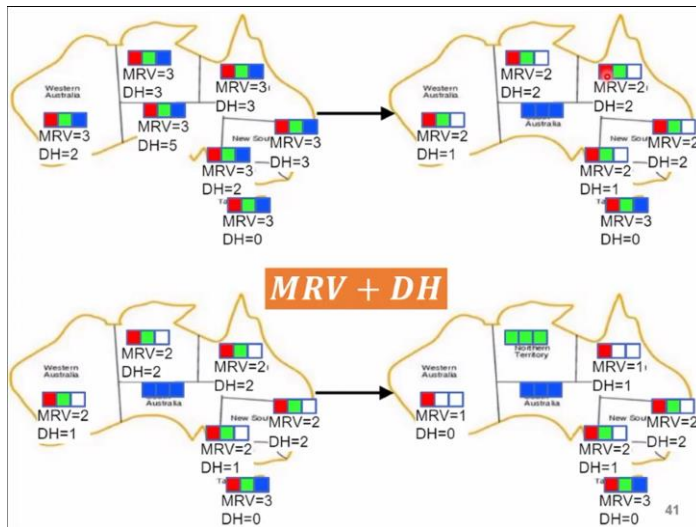
**Minimum-remaining-values(MRV) heuristic:** Chọn biến nào có ít giá trị hợp lệ nhất

**Degree heuristic (DH):** (Thường sử dụng khi xét MRV bằng nhau): Chọn biến nào có số ràng buộc lớn nhất với các biến chưa được gán. Cần cập nhật sau mỗi lần.

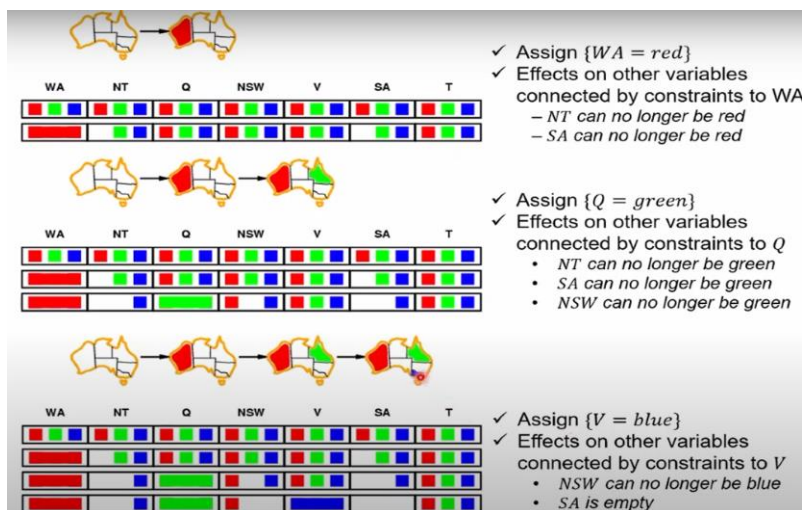
**Least constraining value (LCV) heuristic:** Chọn giá trị cho biến sao cho mà trong tương lai mình có nhiều khả năng gán giá trị đó nhất. Sắp xếp ngàm định các giá trị chưa được sử dụng.

**MRV thấp nhất và DH cao nhất, MRV ưu tiên hơn DH.**





## Forward checking:



FC has detected that partial assignment is *inconsistent* with the constraints and backtracking can occur.

## Quiz: Lập thời khóa biểu môn học

### 1. Lập thời khóa biểu môn học

Bạn được giao nhiệm vụ lập thời khóa biểu cho các lớp Khoa học máy tính được tổ chức vào Thứ hai, Thứ tư, và Thứ sáu. Có 5 lớp tổ chức vào những ngày này và 3 giáo sư sẽ giảng dạy các lớp đó. Mỗi giáo sư chỉ có thể giảng một lớp tại một thời điểm.

Các lớp bao gồm

1. Lớp 1 – Nhập môn lập trình: 08:00 – 09:00 sáng
2. Lớp 2 – Nhập môn Trí tuệ nhân tạo: 08:30 – 09:30 sáng
3. Lớp 3 – Xử lý ngôn ngữ tự nhiên: 09:00 – 10:00 sáng
4. Lớp 4 – Thị giác máy tính: 09:00 – 10:00 sáng
5. Lớp 5 – Học máy: 10:30 – 11:30 sáng

Các giáo sư bao gồm

1. Giáo sư A: đủ tiêu chuẩn dạy Lớp 1, 2 và 5
  2. Giáo sư B: đủ tiêu chuẩn dạy Lớp 3, 4 và 5
  3. Giáo sư C: đủ tiêu chuẩn dạy Lớp 1, 3 và 4
1. Phát biểu bài toán thỏa mãn ràng buộc (CSP). Giả sử mỗi lớp là một biến, chỉ ra các miền giá trị và ràng buộc. Ràng buộc cần được đặc tả một cách quy chuẩn và chính xác, và ràng buộc có thể ở dạng tiềm ẩn thay vì tường minh.
  2. Vẽ đồ thị ràng buộc tương ứng với CSP đang cần giải quyết.
  3. CSP của bạn gần như có cấu trúc cây. Giải thích ngắn gọn (trong vòng 1 câu) vì sao chúng ta ưa thích giải CSP có cấu trúc cây hơn?

$X = \{1, 2, 3, 4, 5\}$

$D1 = \{A, C\}, D2 = \{A\}, D3 = \{B, C\}, D4 = \{B, C\},$   
 $D5 = \{A, B\}$

Ràng buộc: Các lớp có giờ trùng nhau  
hoặc gởi đầu sẽ không được cùng giáo  
sư

$C = \{1! = 2, 2! = 3, 3! = 4, 2! = 4\}$

## Quiz: Timetable Scheduling

- You are scheduling for computer science classes that meet on Mondays, Wednesdays and Fridays .
- There are 5 classes and 3 professors who will be teaching these classes.
- You are constrained that each professor can only teach one class at a time.
- The classes are:
  - Class 1 - Intro to Programming: meets from 8:00-9:00am
  - Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30am
  - Class 3 - Natural Language Processing: meets from 9:00-10:00am
  - Class 4 - Computer Vision: meets from 9:00-10:00am
  - Class 5 - Machine Learning: meets from 9:30-10:30am
- The professors are:
  - Professor A, who is available to teach Classes 3 and 4.
  - Professor B, who is available to teach Classes 2, 3, 4, and 5.
  - Professor C, who is available to teach Classes 1, 2, 3, 4, and 5.