

NHẬP MÔN TRÍ TUỆ NHÂN TẠO

CHƯƠNG 2

GIẢI QUYẾT VẤN ĐỀ BẰNG TÌM KIẾM

GVGD: ThS. Huỳnh Gia Khương
hgkhuong@cit.ctu.edu.vn

Nội dung

- Biểu diễn bài toán trong KGTT
- Tìm kiếm mù (uninformed search)
- Tìm kiếm heuristic (informed search)
- Cây trò chơi, cắt tỉa alpha –beta
- **Bài toán thoả mãn ràng buộc**

BÀI TOÁN THỎA MÃN RÀNG BUỘC

(CONSTRAINT SATISFACTION PROBLEMS)

Một số bài toán thoả mãn ràng buộc thực tế

Xét bản đồ các bang của nước Úc như hình bên. Cần tô màu các bang với ba màu red, green, blue sao cho hai bang cạnh nhau được tô màu khác nhau



					8			4
	8	4		1	6			
			5			1		
1		3	8			9		
6		8				4		3
		2			9	5		1
		7			2			
			7	8		2	6	
2			3					

$$\begin{array}{r} \text{S E N D} \\ \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

Một số bài toán thoả mãn ràng buộc thực tế

- Các bài toán giao nhiệm vụ
 - Ví dụ: Giáo viên nào dạy lớp nào?
- Các bài toán lập thời khóa (gian) biểu
 - Ví dụ: Lớp học nào được dạy vào thời gian nào và ở đâu?
- Các bài toán lập lịch vận tải (giao hàng) của các công ty
 - Các bài toán lập lịch sản xuất của các nhà máy

Ràng buộc

- Một ràng buộc (constraint) là một quan hệ trên một tập các biến
 - Mỗi biến có (gắn với) một tập các giá trị có thể nhận – gọi là miền giá trị (domain).
 - Trong môn học này, chúng ta chỉ xét các miền hữu hạn các giá trị rời rạc
- Một ràng buộc có thể được biểu diễn bằng
 - Một biểu thức (toán học / logic)
 - Một bảng liệt kê các phép gán giá trị phù hợp cho các biến
- Ví dụ về ràng buộc
 - Tổng các góc trong một tam giác là 180^0
 - Độ dài của từ W là 10 ký tự
 - X nhỏ hơn Y
 - Tuần có thể tham dự buổi seminar vào thứ 4 sau 14h
 - ...

Bài toán thỏa mãn ràng buộc (Constraint satisfaction problem)

- Bài toán thỏa mãn ràng buộc (CSP): sử dụng phương pháp biểu diễn có cấu trúc để biểu diễn các trạng thái, mỗi trạng thái là một tập biến, mỗi biến có một giá trị
- Bài toán được giải quyết khi mỗi biến đều được gán trị thỏa mãn tất cả ràng buộc
- Một số bài toán được giải quyết nhanh chóng khi mô hình hóa về CSP trong khi không giải quyết được bằng phương pháp tìm kiếm trong không gian trạng thái

Bài toán thỏa mãn ràng buộc (Constraint satisfaction problem)

- Một bài toán thỏa mãn ràng buộc gồm ba thành phần: X , D , và C trong đó:
 - X : tập hợp các biến $\{X_1, X_2, \dots, X_n\}$
 - D : tập hợp các miền giá trị $\{D_1, D_2, \dots, D_n\}$, $D_i = \{v_1, v_2, \dots, v_k\}$ gán vào biến X_i tương ứng
 - C : tập hợp các ràng buộc, C_i là một cặp $\langle \text{scope}, \text{rel} \rangle$ với scope là một bộ biến tham gia vào quan hệ rel (biểu diễn như một danh sách các bộ giá trị tường minh thỏa mãn ràng buộc, hoặc dưới dạng trùu tượng phép toán)
- Bài toán CSP được giải quyết khi tất cả các biến đều được gán trị hợp lệ

Các kiểu biến trong bài toán CSP

■ Các biến rời rạc

□ Các miền giá trị hữu hạn

- Với n biến và kích thước miền giá trị d , thì số lượng các phép gán đầy đủ giá trị cần xét là $O(d^n)$
- Ví dụ: Các bài toán thỏa mãn ràng buộc nhị phân (Boolean CSPs)

□ Các miền giá trị vô hạn

- Miền giá trị các số nguyên, các chuỗi, ...
- Ví dụ: Trong bài toán xếp lịch công việc, các biến là các ngày bắt đầu và kết thúc đối với mỗi công việc
- Cần một ngôn ngữ biểu diễn ràng buộc (constraint language), ví dụ:
 $StartJob_1 + 5 \leq StartJob_3$

■ Các biến liên tục

- Ví dụ: Các mốc thời gian bắt đầu và kết thúc đối với các quan sát bằng kính viễn vọng không gian Hubble
- Bài toán các ràng buộc tuyến tính có thể giải quyết được ở mức chi phí thời gian đa thức bằng phương pháp lập trình tuyến tính

Các kiểu ràng buộc trong bài toán CSP

- **Ràng buộc đơn** (unary constraint) chỉ liên quan đến 1 biến
 - Ví dụ: $SA \neq \text{green}$
- **Ràng buộc nhị phân** (binary constraint) liên quan đến 2 biến
 - Ví dụ: $SA \neq WA$
- **Ràng buộc bậc cao** (higher-order constraint) liên quan đến *nhiều hơn 2 biến*
 - Ví dụ: Các ràng buộc trong bài toán mật mã số học (trình bày ở slide tiếp theo)

Các bước giải bài toán thỏa mãn ràng buộc

Bước 1

Xác định các thành phần của bài toán CSP (các biến, miền giá trị cho các biến, tập ràng buộc trên các biến)

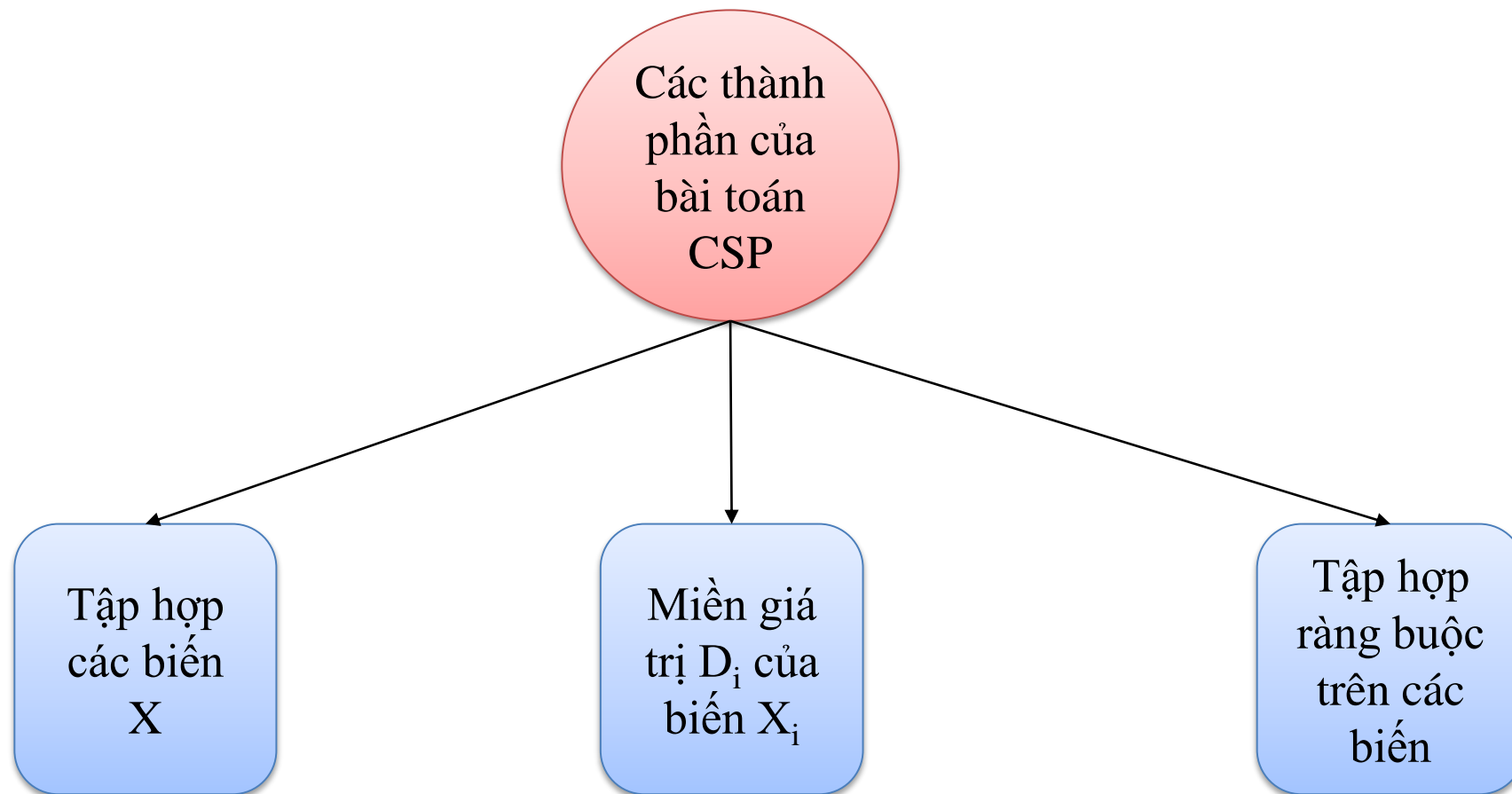
Bước 2

Vẽ đồ thị ràng buộc cho bài toán

Bước 3

Áp dụng các phương pháp giải bài toán CSP (Generate and Test, Backtracking, Constraint Propagation)

Xác định các thành phần của bài toán CSP

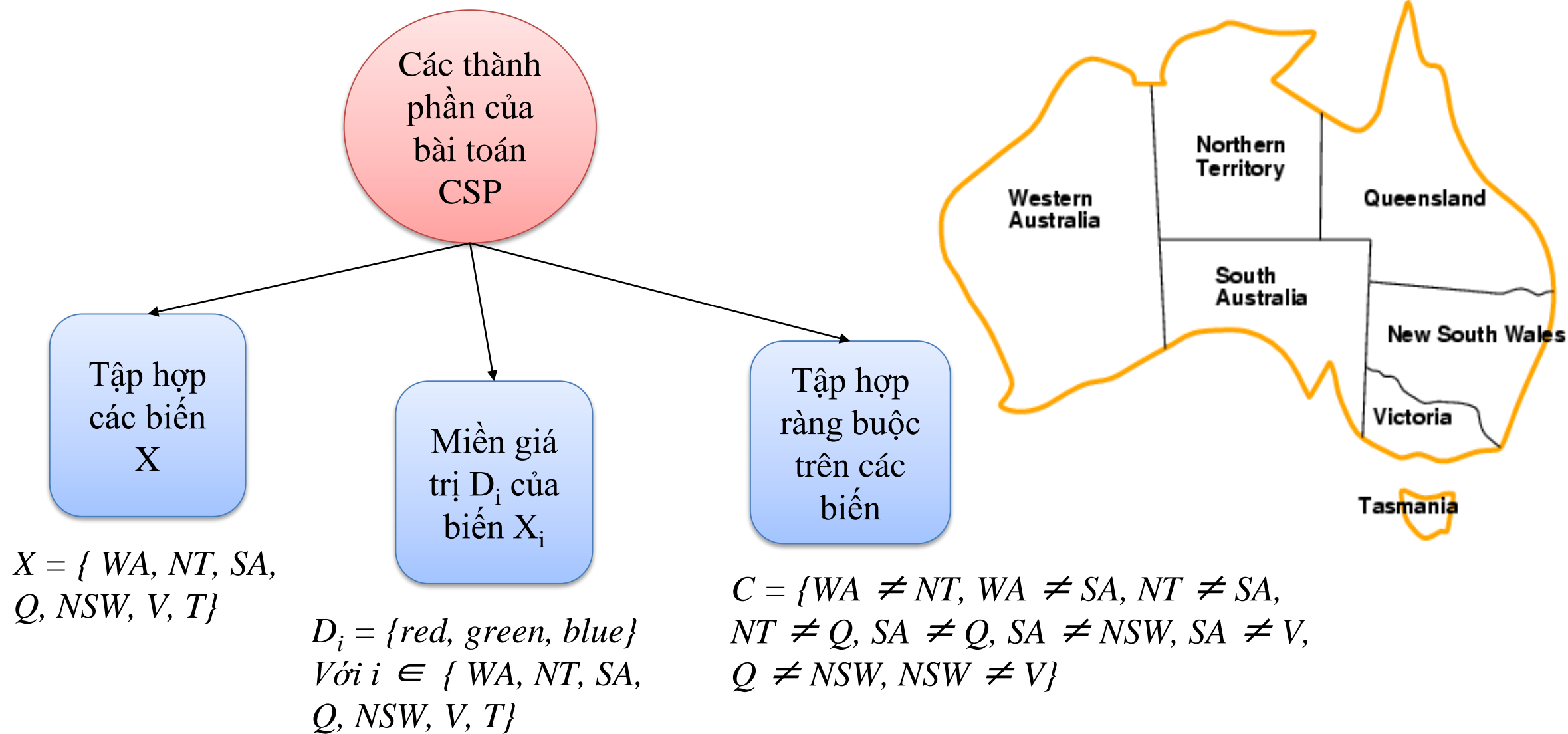


Ví dụ: Bài toán tô màu bản đồ

- Xét bản đồ các bang của nước Úc như hình bên. Cần tô màu các bang với ba màu red, green, blue sao cho hai bang cạnh nhau được tô màu khác nhau.
- Hãy xác định các thành phần của bài toán CSP này?



Ví dụ: Bài toán tô màu bản đồ

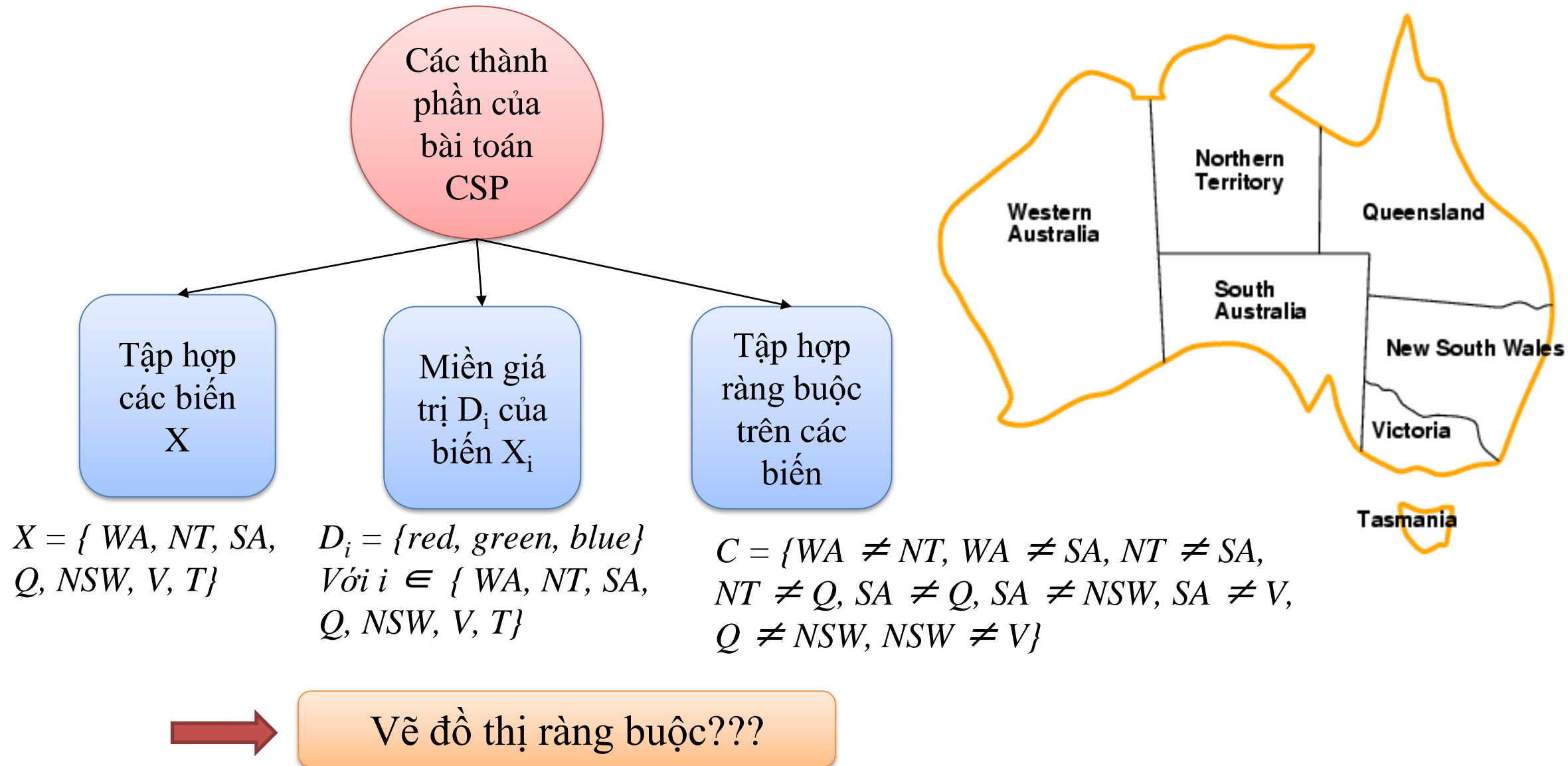


Vẽ đồ thị ràng buộc cho bài toán CSP

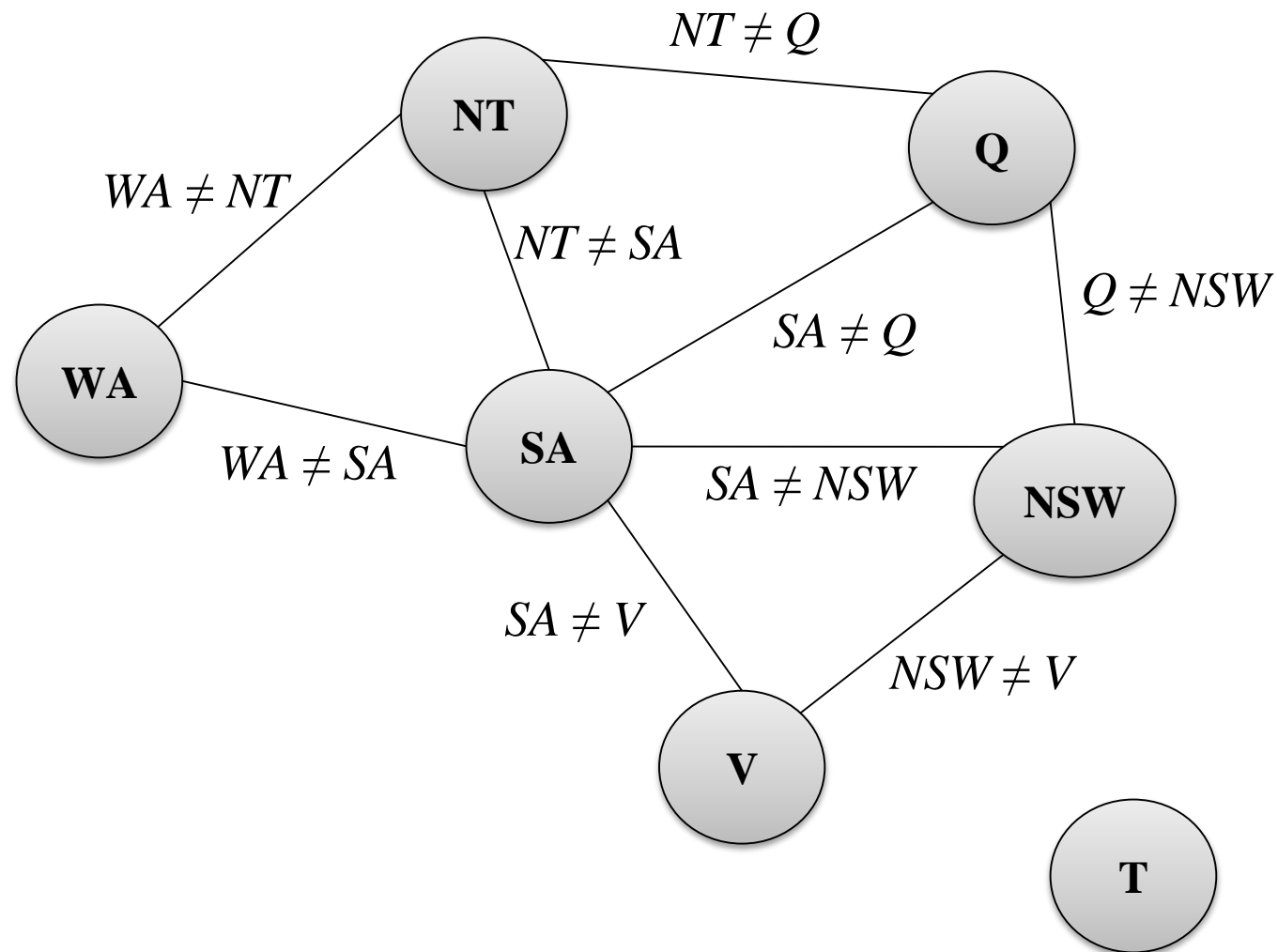
Với tập ràng buộc C được xác định sau khi xác định các thành phần của bài toán CSP. Tiến hành vẽ đồ thị ràng buộc như sau:

- Mỗi nút trên đồ thị biểu diễn một biến X của bài toán CSP
- Mỗi cạnh đồ thị thể hiện một ràng buộc i nào đó trong tập ràng buộc C

Ví dụ: Bài toán tô màu bản đồ



Vẽ đồ thị ràng buộc cho bài toán CSP



Các phương pháp giải bài toán CSP

Phương pháp Generate and Test

Phương pháp Backtracking

Phương pháp lan truyền ràng buộc
(Constraint Propagation)

Phương pháp Generate and Test

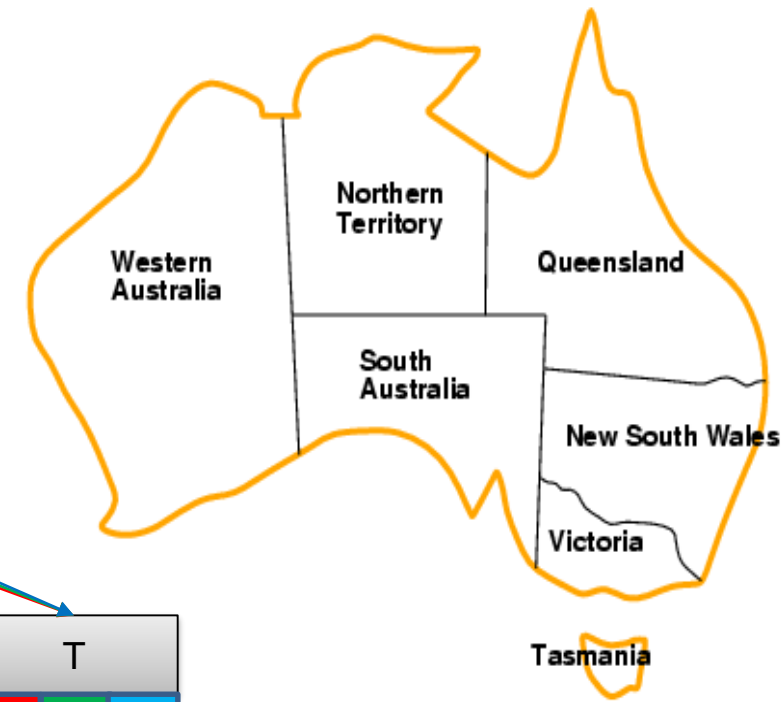
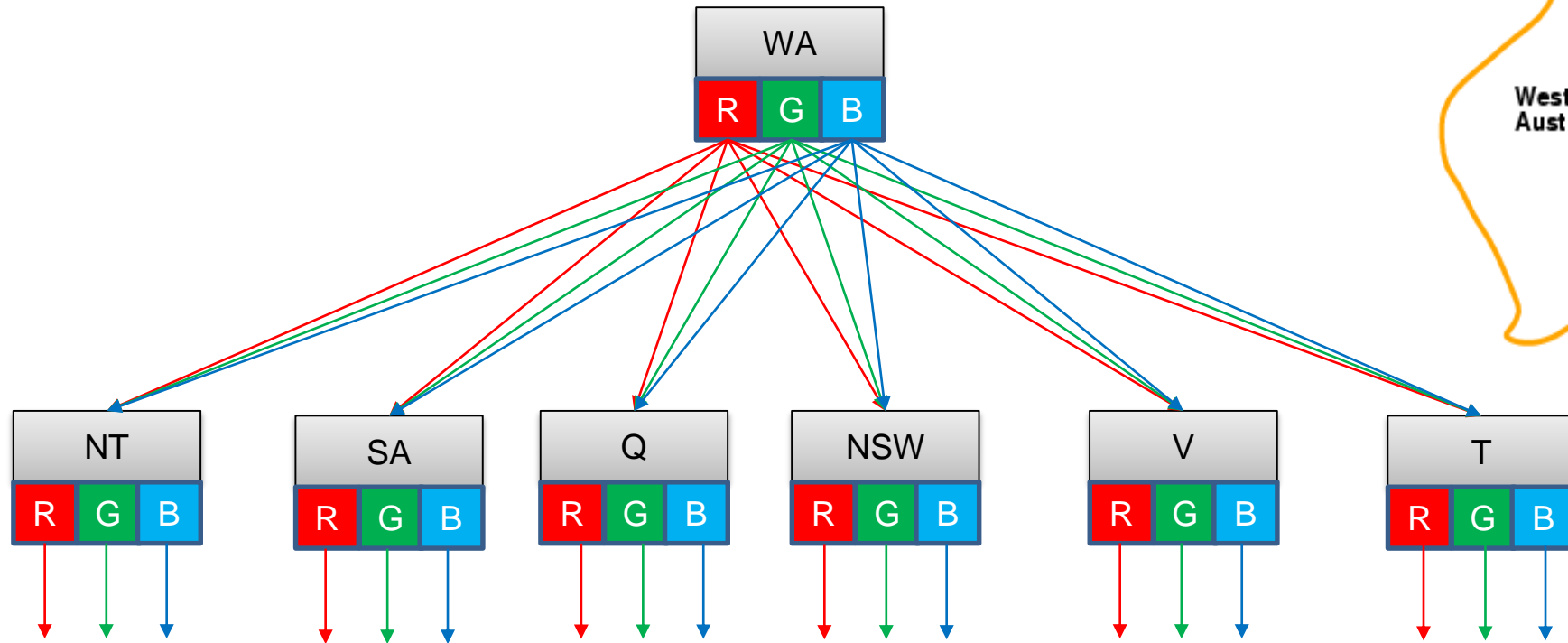
- Là phương pháp giải quyết vấn đề tổng quát nhất
 - Ý tưởng:
 - + Sinh ra một khả năng (candidate) của lời giải
 - + Kiểm tra xem khả năng này có thực sự là một lời giải
 - Áp dụng phương pháp kiểm thử đối với bài toán CSP
 - + Bước 1. Gán các giá trị cho tất cả các biến
 - + Bước 2. Kiểm tra xem tất cả các ràng buộc được thỏa mãn hay không
- Lặp lại 2 bước này cho đến khi tìm được một phép gán thỏa mãn

Ví dụ: Bài toán tô màu bản đồ

- Xét bản đồ các bang của nước Úc như hình bên. Cần tô màu các bang với ba màu red, green, blue sao cho hai bang cạnh nhau được tô màu khác nhau.
- Hãy liệt kê một số mẫu kết quả có thể tạo ra để test với phương pháp Generate and Test cho bài toán CSP này?



Ví dụ: Bài toán tô màu bản đồ



Solution 1: { WA = red, NT = red, SA = red, Q = red, NSW = red, V = red, T = red } → Failed

Solution 2: { WA = red, NT = green, SA = red, Q = red, NSW = red, V = red, T = red } → Failed

Solution 3: { WA = red, NT = green, SA = green, Q = red, NSW = red, V = red, T = red } → Failed

....

Solution n: { WA = red, NT = green, SA = blue, Q = red, NSW = green, V = red, T = green } → Passed

How many solutions???

Phương pháp Generate and Test

→ Điểm yếu nghiêm trọng của phương pháp tìm kiếm bằng kiểm thử là việc **phải xét quá nhiều các khả năng gán** (hiển nhiên) không thỏa mãn các ràng buộc

* Ví dụ

+ Các biến X, Y, Z lấy các giá trị {1, 2}

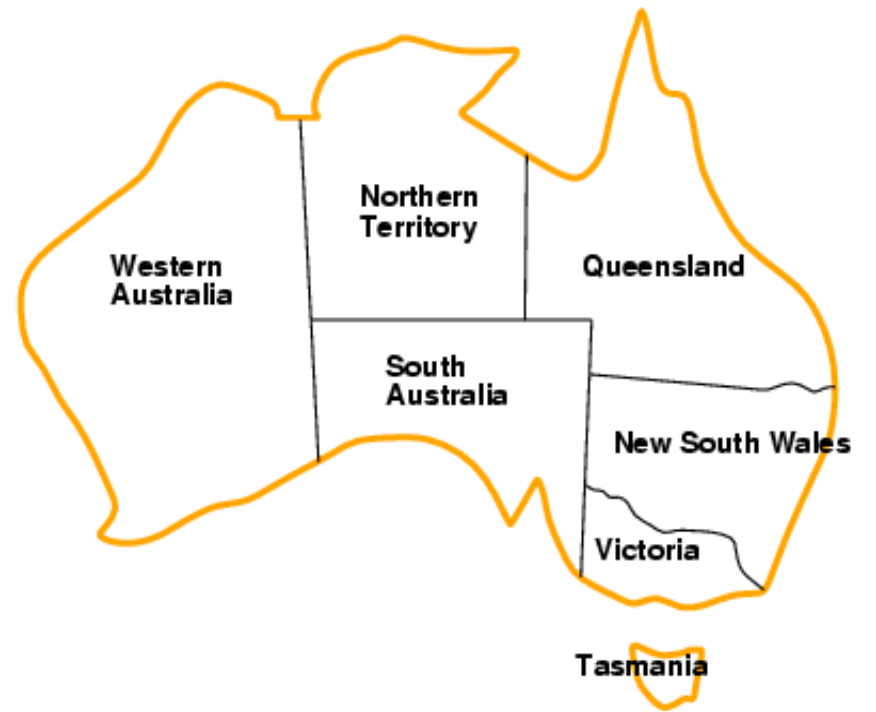
+ Các ràng buộc: $X=Y$, $X \neq Z$, $Y>Z$

+ Các phép (khả năng) gán: (1, 1, 1); (1, 1, 2); (1, 2, 1); (1, 2, 2); (2, 1, 1); (2, 1, 2); (2, 2, 1)

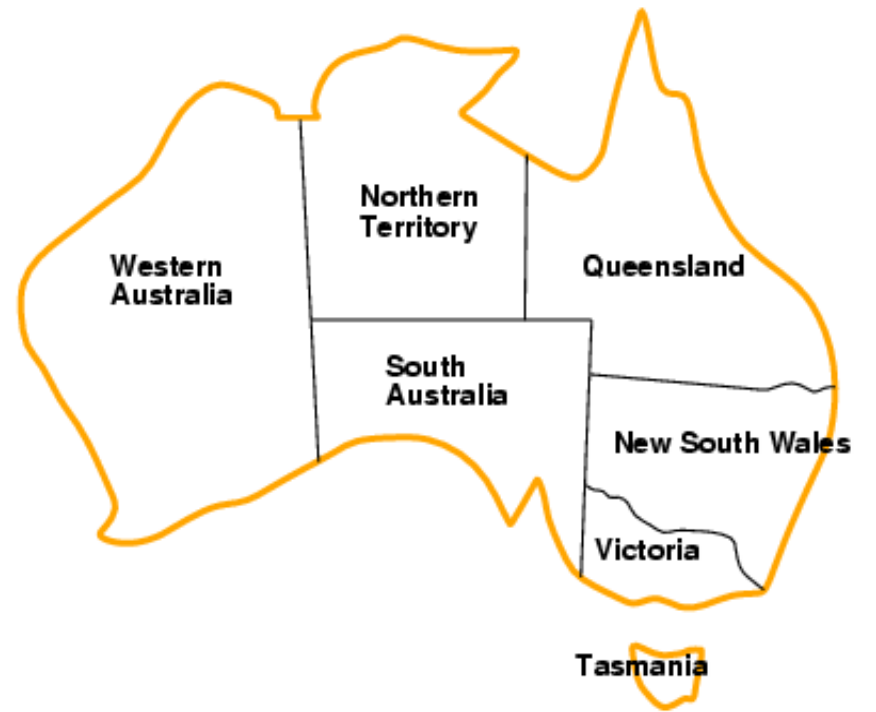
Phương pháp Backtracking

- Tìm kiếm quay lui (backtracking) là giải thuật tìm kiếm được sử dụng phổ biến nhất trong CSP.
- + Dựa trên giải thuật tìm kiếm theo chiều sâu (depth-first search)
- + Mỗi lần gán, chỉ làm việc (gán giá trị) cho một biến
(Tìm kiếm bằng kiểm thử: mỗi lần gán xác định các giá trị cho tất cả các biến)
- 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).

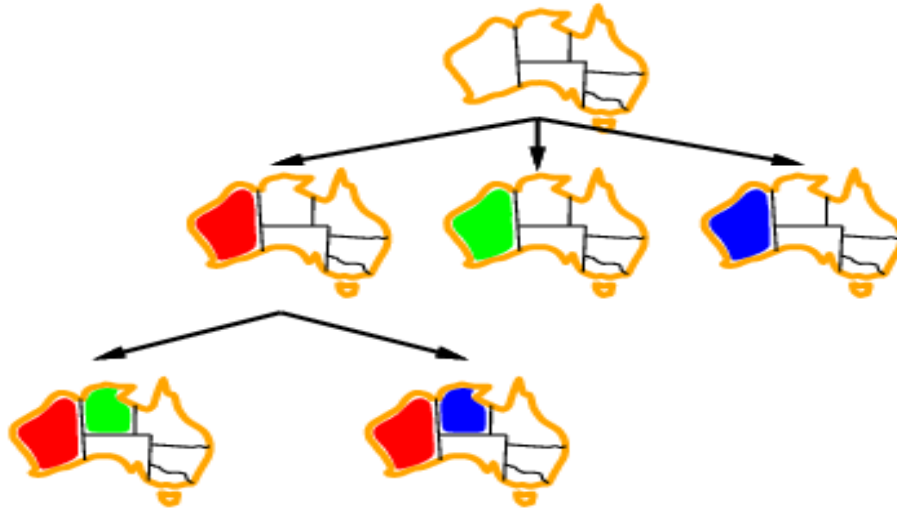
Phương pháp Backtracking



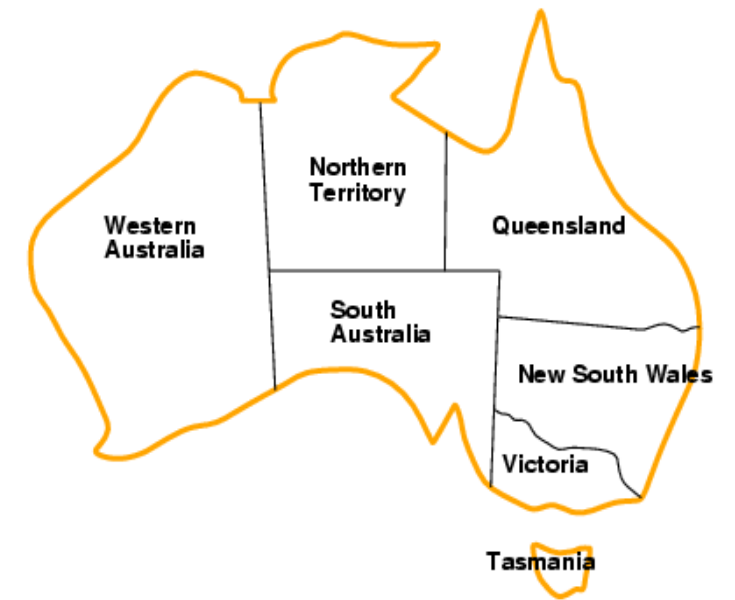
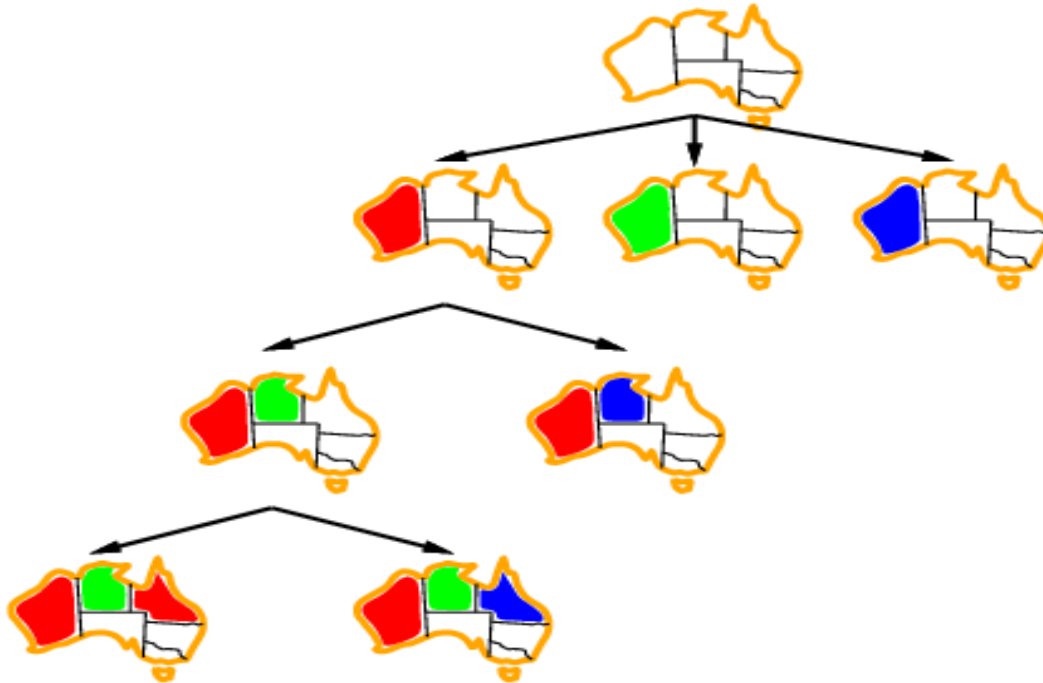
Phương pháp Backtracking



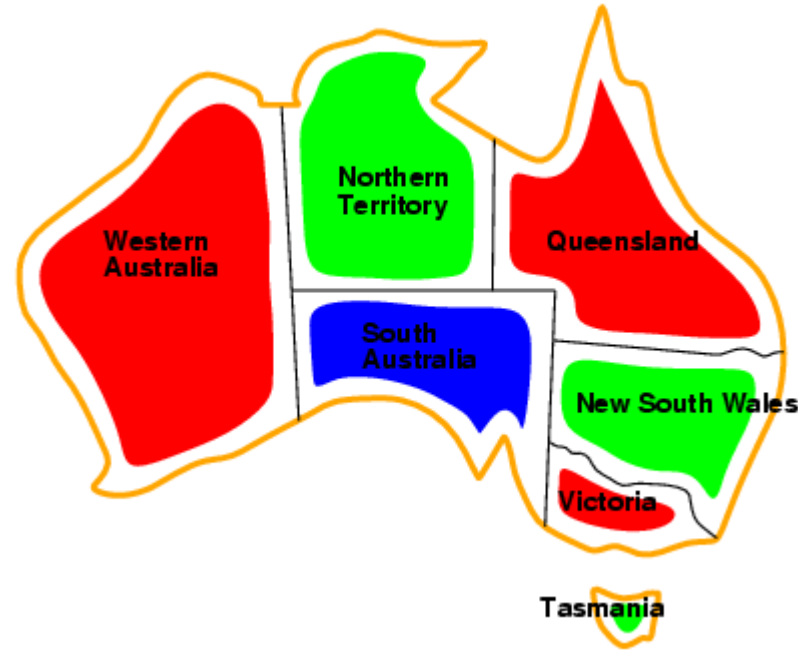
Phương pháp Backtracking



Phương pháp Backtracking

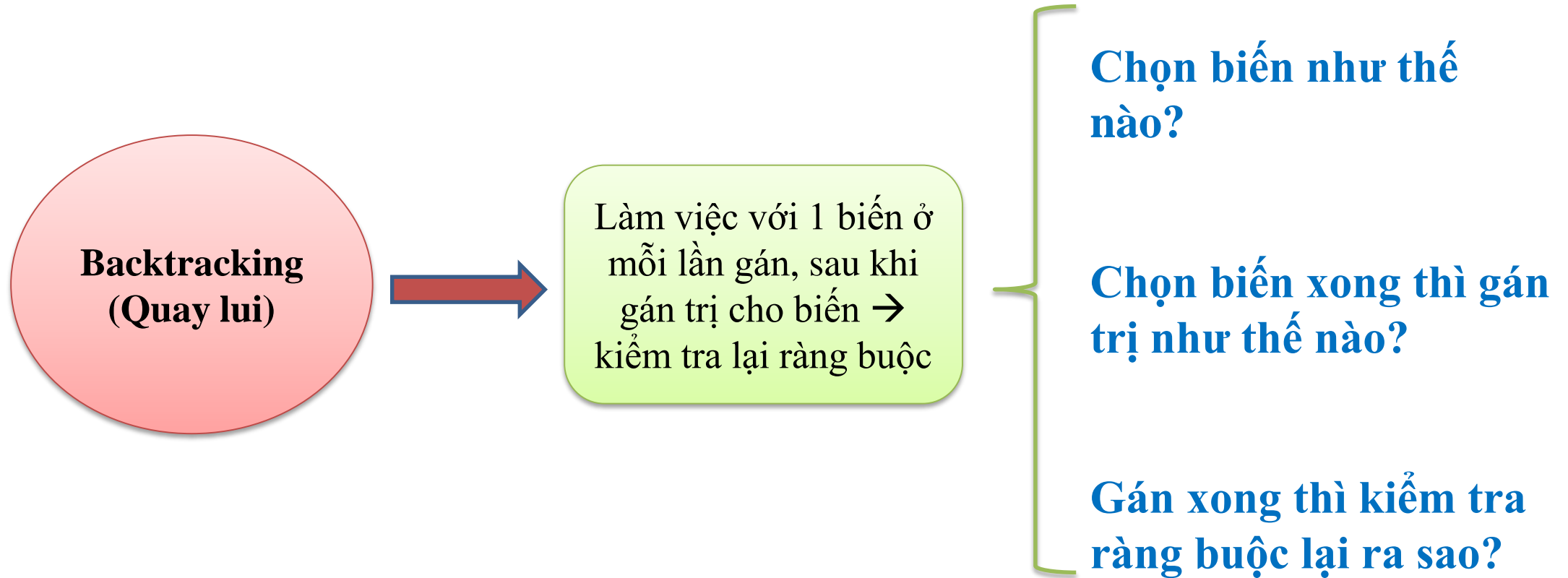


Phương pháp Backtracking



→ **Giải pháp:** WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green

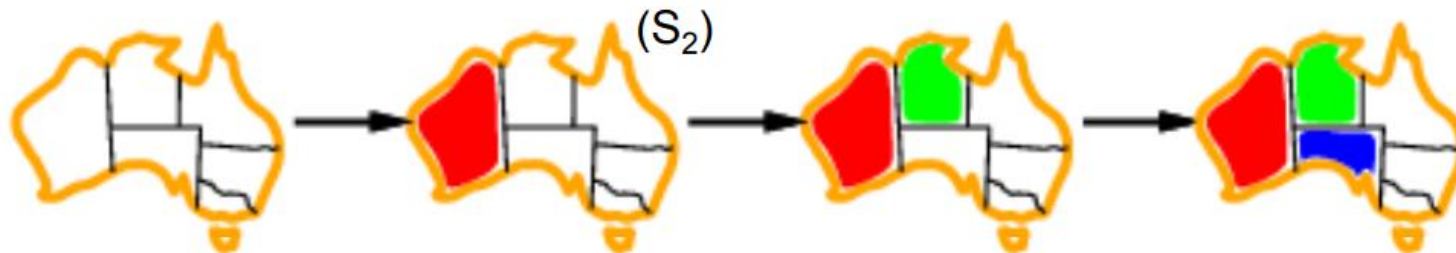
Phương pháp Backtracking



Phương pháp Backtracking

Biến bị ràng buộc nhiều nhất

- Quy tắc lựa chọn thứ tự xét các biến: Ưu tiên biến bị ràng buộc nhiều nhất (most constrained variable)
 - Chọn biến có số lượng các giá trị hợp lệ ít nhất
 - Ví dụ: Tại bước S_2 , biến NT được chọn vì nó có số lượng các giá trị hợp lệ ít nhất (2)

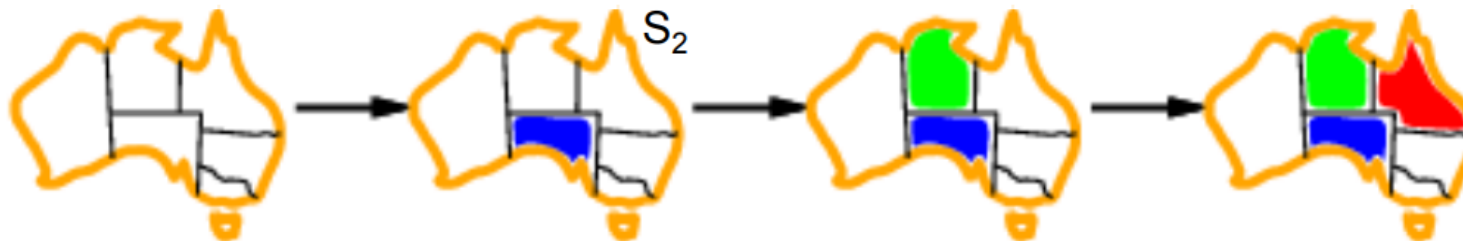


- Còn được gọi là quy tắc ưu tiên các biến có tập giá trị hợp lệ nhỏ nhất (Minimum Remaining Values – MRV)

Phương pháp Backtracking

Biến ràng buộc các biến khác nhiều nhất

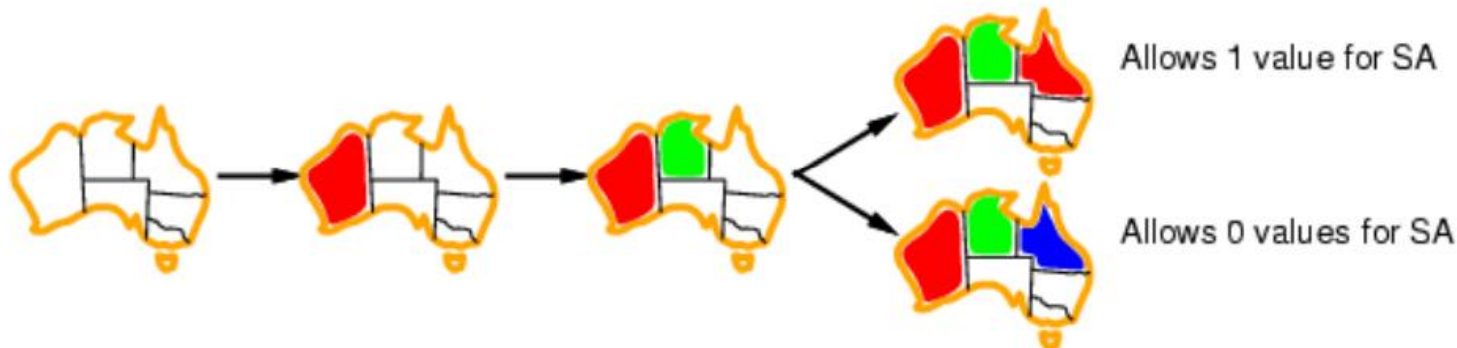
- Khi có ≥ 2 biến có như nhau số lượng giá trị hợp lệ ít nhất, thì chọn biến nào?
 - Ví dụ: Trong ví dụ trước, 2 biến NT và SA có cùng số lượng giá trị hợp lệ ít nhất (2)
- Chọn biến ràng buộc (khống chế) các biến khác (chưa được gán giá trị) nhiều nhất
 - Ví dụ: Tại bước S_2 , tuy cùng mức độ bị ràng buộc, nhưng biến SA nên được xét trước biến NT – vì SA ràng buộc 5 biến khác, còn NT chỉ ràng buộc 3 biến khác



Phương pháp Backtracking

Giá trị ràng buộc các biến khác ít nhất

- Đối với một biến, các giá trị được xét (để gán) theo thứ tự nào?
- Chọn giá trị ràng buộc (khống chế) các biến khác (chưa được gán giá trị) ít nhất
 - Giá trị này gây ra hạn chế tối thiểu đối với các khả năng gán giá trị của các biến khác



Phương pháp Backtracking

Kiểm tra tiến (Forward checking)

- Mục đích: Tránh các thất bại, bằng kiểm tra trước các ràng buộc
- Kiểm tra tiến đảm bảo sự phù hợp (consistency) giữa biến đang được xét gán giá trị và các biến khác **có liên quan (ràng buộc) trực tiếp** với nó
- Ý tưởng:
 - Ở mỗi bước gán giá trị, theo dõi các giá trị hợp lệ (có thể được gán) đối với các biến chưa được gán giá trị
 - Loại bỏ (dừng) hướng tìm kiếm hiện tại khi có bất kỳ một biến (chưa được gán giá trị) nào đó không còn giá trị hợp lệ

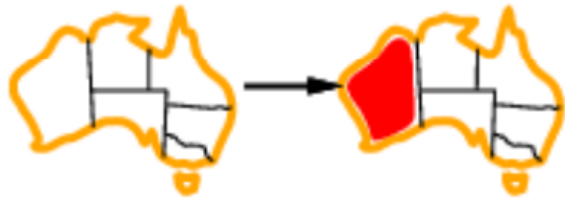
Phương pháp Backtracking

Kiểm tra tiến – Ví dụ (1)



Phương pháp Backtracking

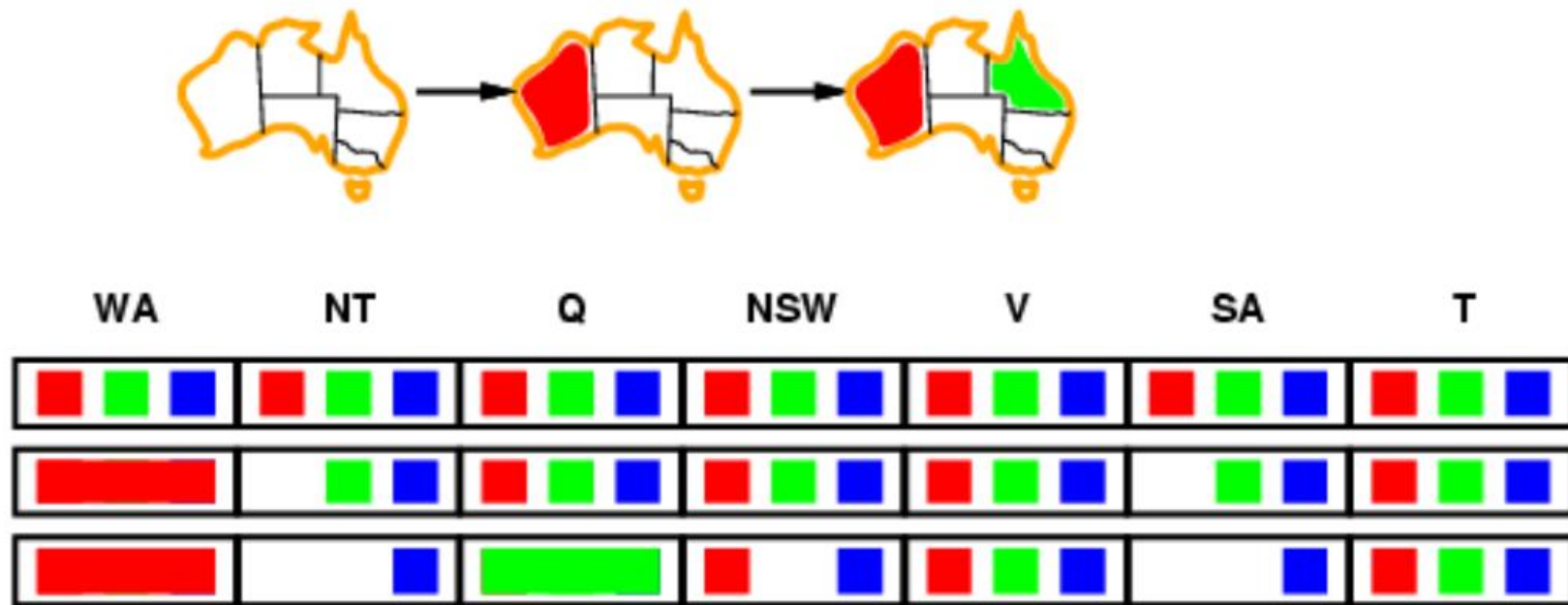
Kiểm tra tiến – Ví dụ (2)



WA	NT	Q	NSW	V	SA	T
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>

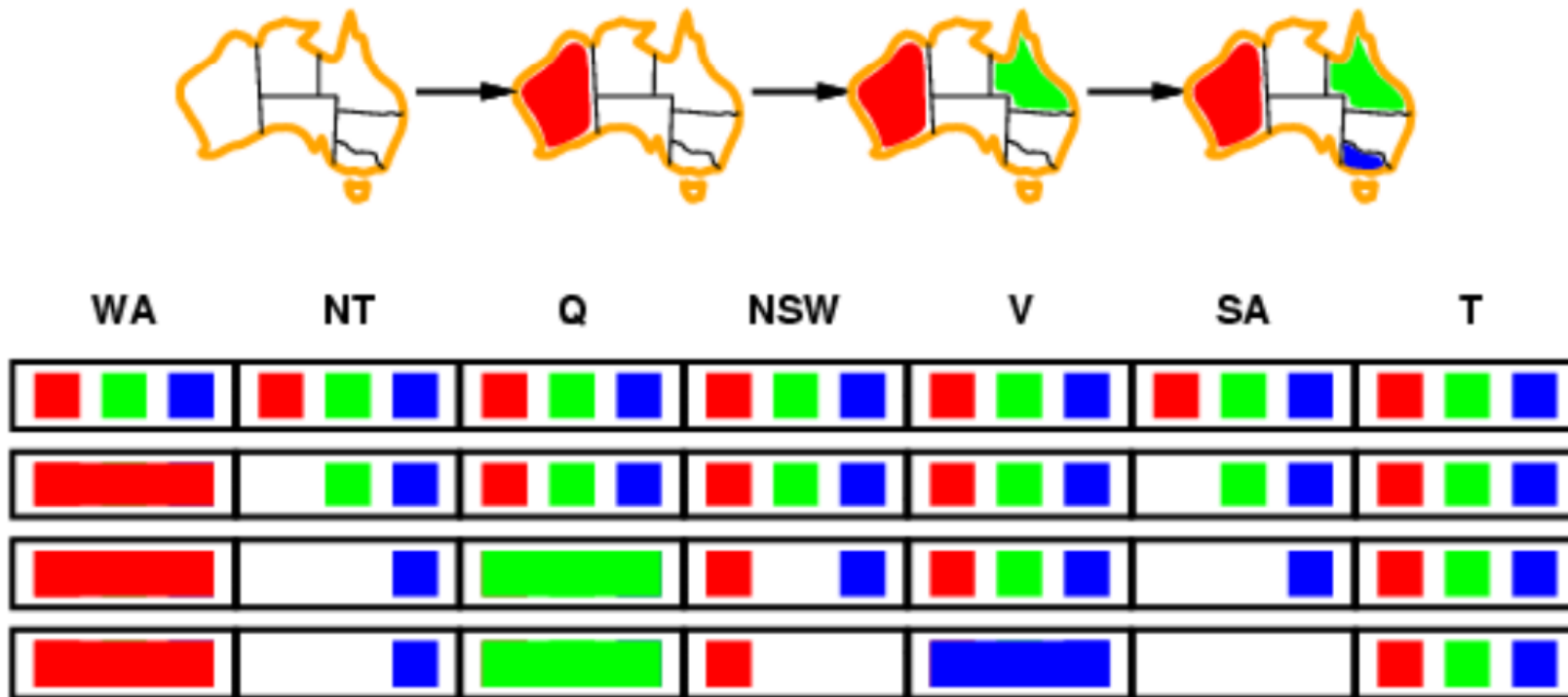
Phương pháp Backtracking

Kiểm tra tiến – Ví dụ (3)



Phương pháp Backtracking

Kiểm tra tiến – Ví dụ (4)



Phương pháp lan truyền ràng buộc (Constraint Propagation)

- Trạng thái khởi đầu chứa các ràng buộc được cho trong mô tả vấn đề.
- Một trạng thái đích là trạng thái đã bị ràng buộc « đủ », « đủ » được định nghĩa tùy thuộc vấn đề. Ví dụ, trong câu đố « mật mã số học » « đủ » có nghĩa là mỗi chữ được gán một chữ số duy nhất.

* Thỏa mãn ràng buộc là quá trình hai bước:

+ Đầu tiên, các ràng buộc được phát hiện và được **lan truyền** xa nhất có thể.

+ Sau đó, nếu vẫn chưa có lời giải, bắt đầu tìm kiếm. **Đoán** một sự kiện, thêm vào ràng buộc mới, lan truyền ràng buộc ...

Phương pháp lan truyền ràng buộc (Constraint Propagation)

1. Lan truyền các ràng buộc sẵn có: Đặt $OPEN$ = tập chứa các đối tượng cần phải gán trị (trong lời giải). Tiến hành các bước sau đến tận khi gặp mâu thuẫn hoặc $OPEN$ rỗng:

a. Chọn một đối tượng X trong $OPEN$. Tăng cường tập các ràng buộc trên X .

b. Nếu tập này khác với tập đã được gán cho X trong lần kiểm tra trước hoặc X lần đầu tiên được kiểm tra, thêm vào $OPEN$ tất cả các đối tượng chia sẻ các ràng buộc với X .

c. Xóa X khỏi $OPEN$.

2. Nếu hợp các ràng buộc được phát hiện ở trên xác định lời giải, thông báo lời giải và thoát

3. Nếu hợp các ràng buộc được phát hiện ở trên xác định một mâu thuẫn, thông báo thất bại

Phương pháp lan truyền ràng buộc (Constraint Propagation)

4. Nếu 2. và 3. không xảy ra, Lặp lại đến tận khi tìm thấy một lời giải hoặc tất cả các lời giải có thể bị loại bỏ:
 - a) Chọn một đối tượng chưa được xác định giá trị, chọn một phương pháp tăng cường ràng buộc trên đối tượng*
 - b) Gọi đệ quy thỏa mãn ràng buộc với tập hiện hành các ràng buộc được tăng cường thêm qua bước a)*

Phương pháp lan truyền ràng buộc (Constraint Propagation)

Bài toán "số học mật mã": Gán các chữ cái bởi các chữ số, thỏa mãn điều kiện: hai chữ cái khác nhau được gán với hai chữ số khác nhau, một chữ cái được gán với đúng một chữ số, sao cho phép tính sau là đúng.

$$\begin{array}{rcccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

Phương pháp lan truyền ràng buộc (Constraint Propagation)

$$\begin{array}{r} \text{S E N D} \\ \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

- Các biến:

- ✓ S, E, N, D, M, O, R, Y

- ✓ C_i ($i = 1, 2, 3, 4$) (các nhớ của phép cộng)

- Miền giá trị: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ đối với 8 biến S, E, N, D, M, O, R, Y, và $\{0, 1\}$ đối với 4 biến C_1, C_2, C_3, C_4 .

- Ràng buộc:

- ✓ Alldiff(S, E, N, D, M, O, R, Y)

- ✓ $D + E = Y + 10 * C_1$

- ✓ $C_1 + N + R = E + 10 * C_2$

- ✓ $C_2 + E + O = N + 10 * C_3$

- ✓ $C_3 + S + M = O + 10 * C_4$

- ✓ $C_4 = M$. $S \neq 0$; $M \neq 0$.

Phương pháp lan truyền ràng buộc (Constraint Propagation)

S	E	N	D
M	O	R	E
<hr/>			
M	O	N	E
Y			

$M = (0, 1, \dots, 9) ?$

Phương pháp lan truyền ràng buộc (Constraint Propagation)

Các quy tắc để lan truyền ràng buộc sinh ra các ràng buộc sau:

- $M = 1$ ($S + M + C_3 \leq 19$)
- $S = ?$

S	E	N	D
M	O	R	E
<hr/>			
M	O	N	E
	Y		

S	E	N	D
1	O	R	E
<hr/>			
1	O	N	E
	Y		

Phương pháp lan truyền ràng buộc (Constraint Propagation)

Các quy tắc để lan truyền ràng buộc sinh ra các ràng buộc sau:

- $M = 1 \ (S + M + C_3 \leq 19)$
- $S = 8 \text{ hoặc } 9 \ (S + M + C_3 > 9, C_3 \leq 1, M = 1)$
- $O = ?$

$$\begin{array}{r} \text{S E N D} \\ \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

$$\begin{array}{r} \text{S E N D} \\ \text{1 O R E} \\ \hline \text{1 O N E Y} \end{array}$$

Phương pháp lan truyền ràng buộc (Constraint Propagation)

Các quy tắc để lan truyền ràng buộc sinh ra các ràng buộc sau:

- $M = 1$ ($S + M + C_3 \leq 19$)
- $S = 8$ hoặc 9 ($S + M + C_3 > 9$, $C_3 \leq 1$, $M = 1$)
- $O = 0$ ($S + M + C_3 = 10 + O$, với $M = 1$, $S \leq 9$, $C_3 \leq 1$ nên $O \leq 1$)
- $S = ?$

S	E	N	D
M	O	R	E
<hr/>			
M	O	N	E
Y			

S	E	N	D
1	0	R	E
<hr/>			
1	0	N	E
Y			

Phương pháp lan truyền ràng buộc (Constraint Propagation)

Các quy tắc để lan truyền ràng buộc sinh ra các ràng buộc sau:

- $M = 1$ ($S + M + C_3 \leq 19$)
- $S = 8$ hoặc 9 ($S + M + C_3 > 9$, $C_3 \leq 1$, $M = 1$)
- $O = 0$ ($S + M + C_3 = 10 + O$, với $M = 1$, $S \leq 9$, $C_3 \leq 1$ nên $O \leq 1$)
- $S = 9$, $C_3 = 0$ (vì nếu $C_3 = 1$ thì $E = 9$ và $N = 0$ mà $O = 0$)
- $N = ?$

S	E	N	D
M	O	R	E
<hr/>			
M	O	N	E
Y			
9	E	N	D
1	0	R	E
<hr/>			
1	0	N	E
Y			

Phương pháp lan truyền ràng buộc (Constraint Propagation)

Các quy tắc để lan truyền ràng buộc sinh ra các ràng buộc sau:

- $M = 1$ ($S + M + C_3 \leq 19$)
- $S = 8$ hoặc 9 ($S + M + C_3 > 9$, $C_3 \leq 1$, $M = 1$)
- $O = 0$ ($S + M + C_3 = 10 + O$, với $M = 1$, $S \leq 9$, $C_3 \leq 1$ nên $O \leq 1$)
- $S = 9$, $C_3 = 0$ (vì nếu $C_3 = 1$ thì $E = 9$ và $N = 0$ mà $O = 0$)
- $N = E + 1$ (vì $N = E + O + C_2 = E + C_2$ nên $N = E + 1$)
- $C_2 = 1$
- $R = ?$

$$\begin{array}{r}
 \begin{array}{cccc}
 S & E & N & D \\
 M & O & R & E \\
 \hline
 M & O & N & E & Y
 \end{array} \\
 \\
 \begin{array}{cccc}
 9 & E & N & D \\
 1 & 0 & R & E \\
 \hline
 1 & 0 & N & E & Y
 \end{array}
 \end{array}$$

Phương pháp lan truyền ràng buộc (Constraint Propagation)

Các quy tắc để lan truyền ràng buộc sinh ra các ràng buộc sau:

- $M = 1$ ($S + M + C_3 \leq 19$)
- $S = 8$ hoặc 9 ($S + M + C_3 > 9$, $C_3 \leq 1$, $M = 1$)
- $O = 0$ ($S + M + C_3 = 10 + O$, với $M = 1$, $S \leq 9$, $C_3 \leq 1$ nên $O \leq 1$)
- $S = 9$, $C_3 = 0$ (vì nếu $C_3 = 1$ thì $E = 9$ và $N = 0$ mà $O = 0$)
- $N = E + 1$ (vì $N = E + O + C_2 = E + C_2$ nên $N = E + 1$)
- $C_2 = 1$
- $C_1 = 1$ và $R = 8$ (nếu $C_1 = 0 \Rightarrow R = 9$)
- D từ 7 và E từ 5 (vì $D + E \geq 12$)

Không có thêm ràng buộc !

S	E	N	D
M	O	R	E
<hr/>			
M	O	N	E
Y			

9	E	N	D
1	0	8	E
<hr/>			
1	0	N	E
Y			

Phương pháp lan truyền ràng buộc (Constraint Propagation)

- Để tiến triển thực hiện « phương pháp đoán »:

$$E = 5;$$

.....

- Kết quả:

$$S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2$$

$$\begin{array}{r} 9\ 5\ 6\ 7 \\ +\ 1\ 0\ 8\ 5 \\ \hline 1\ 0\ 6\ 5\ 2 \end{array}$$

$$\begin{array}{r} S\ E\ N\ D \\ M\ O\ R\ E \\ \hline M\ O\ N\ E\ Y \end{array}$$

Bài tập CSP: Bài toán xếp lịch

Cần xếp lịch dạy của 3 giảng viên cho 5 lớp học có lịch gần nhau, bao gồm:

- Lớp 1: Phân tích thiết kế thuật toán, lịch học 8:00 - 09:00AM
- Lớp 2: Nhập môn AI, lịch học 8:30 - 09:30AM
- Lớp 3: Cơ sở dữ liệu, lịch học 9:00 - 10:00AM
- Lớp 4: Hệ điều hành, lịch học 9:00 - 10:00AM
- Lớp 5: Nguyên lý máy học, lịch học 9:30 - 10:30AM

Biết rằng giảng viên chỉ có thể dạy 1 lớp tại một thời điểm và thông tin về giảng viên lần lượt là:

- GV A có thể dạy lớp 3 và 4
- GV B có thể dạy lớp 2, 3, 4, và 5
- GV C có thể dạy lớp 1, 2, 3, 4, 5

Hãy giải quyết vấn đề trên sử dụng giải thuật thỏa mãn ràng buộc với gợi ý các biến lần lượt là các lớp học, xác định miền giá trị cho tập biến và các ràng buộc cần thiết

The End