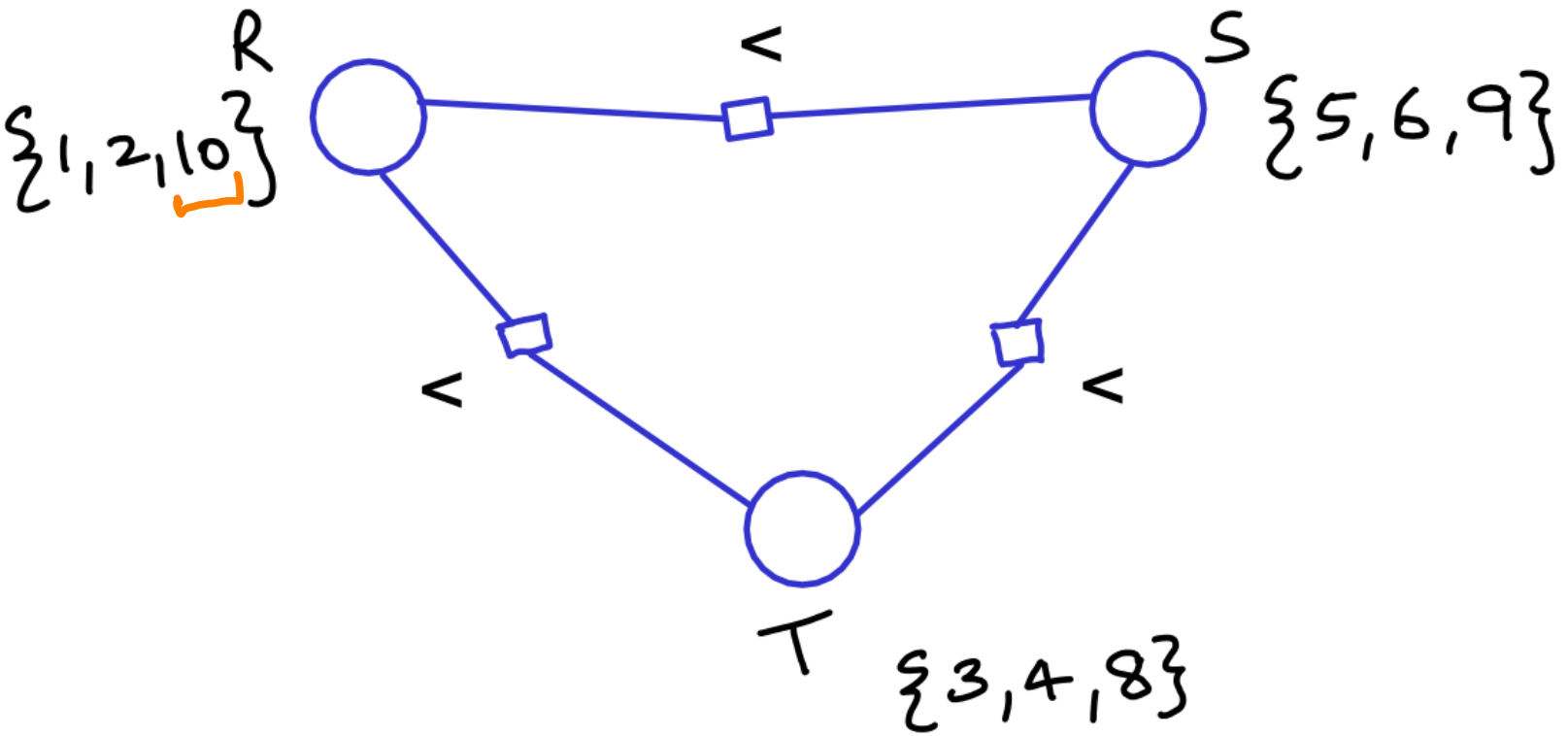
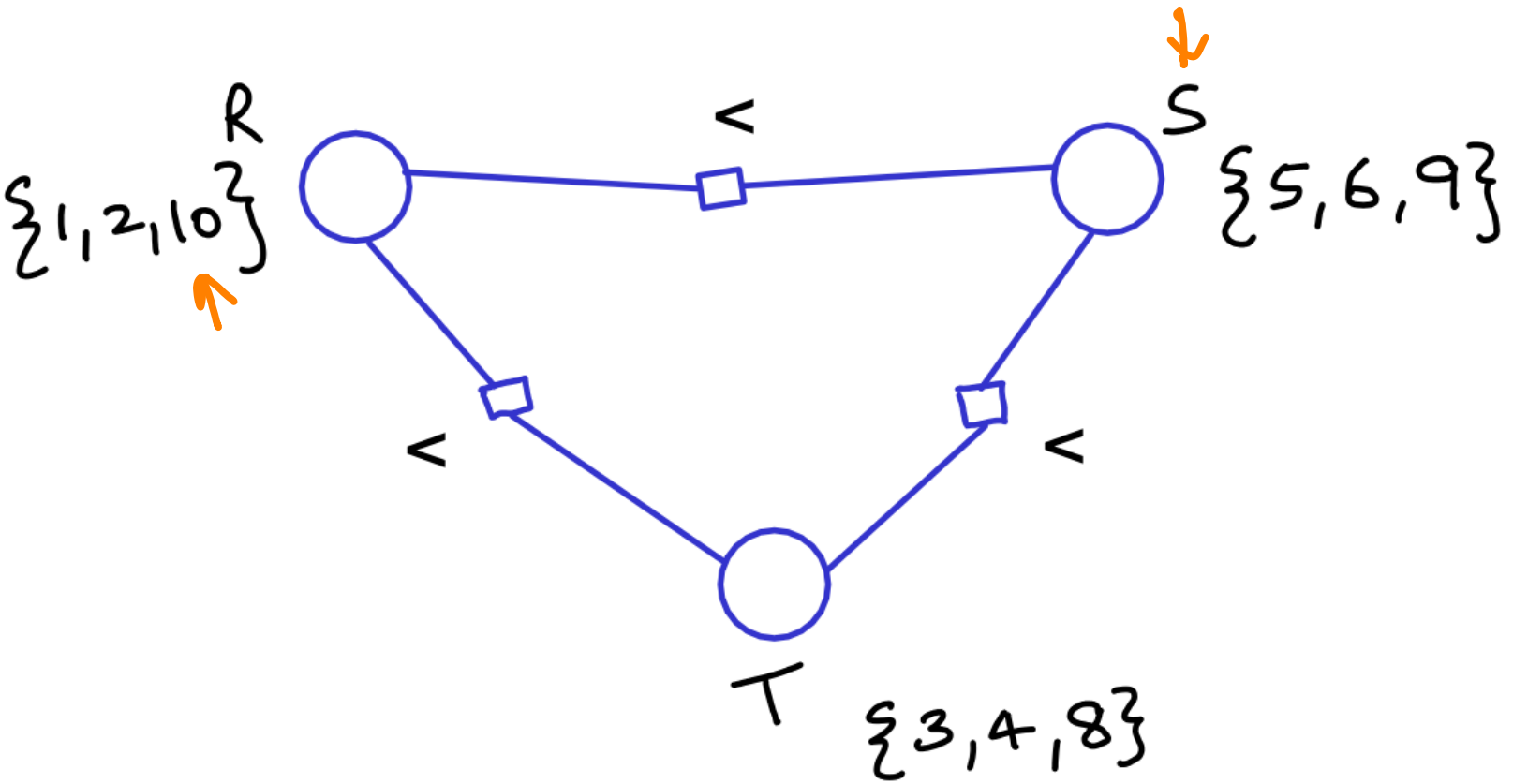
k-consistency

- A CSP is k-consistent if, for any consistent assignment to $k-1$ variables, there is a consistent assignment for the k^{th} variable.
- 2-consistency is same as arc consistency.
- 3-consistency is same as path consistency.
- A CSP is strongly-k-consistent if it is k-consistent and is also $(k-1)$-consistent, $(k-2)$-consistent, ... all the way down to 1-consistent.

Does k-consistent CSP $\Rightarrow$ strongly-k-consistent CSP?

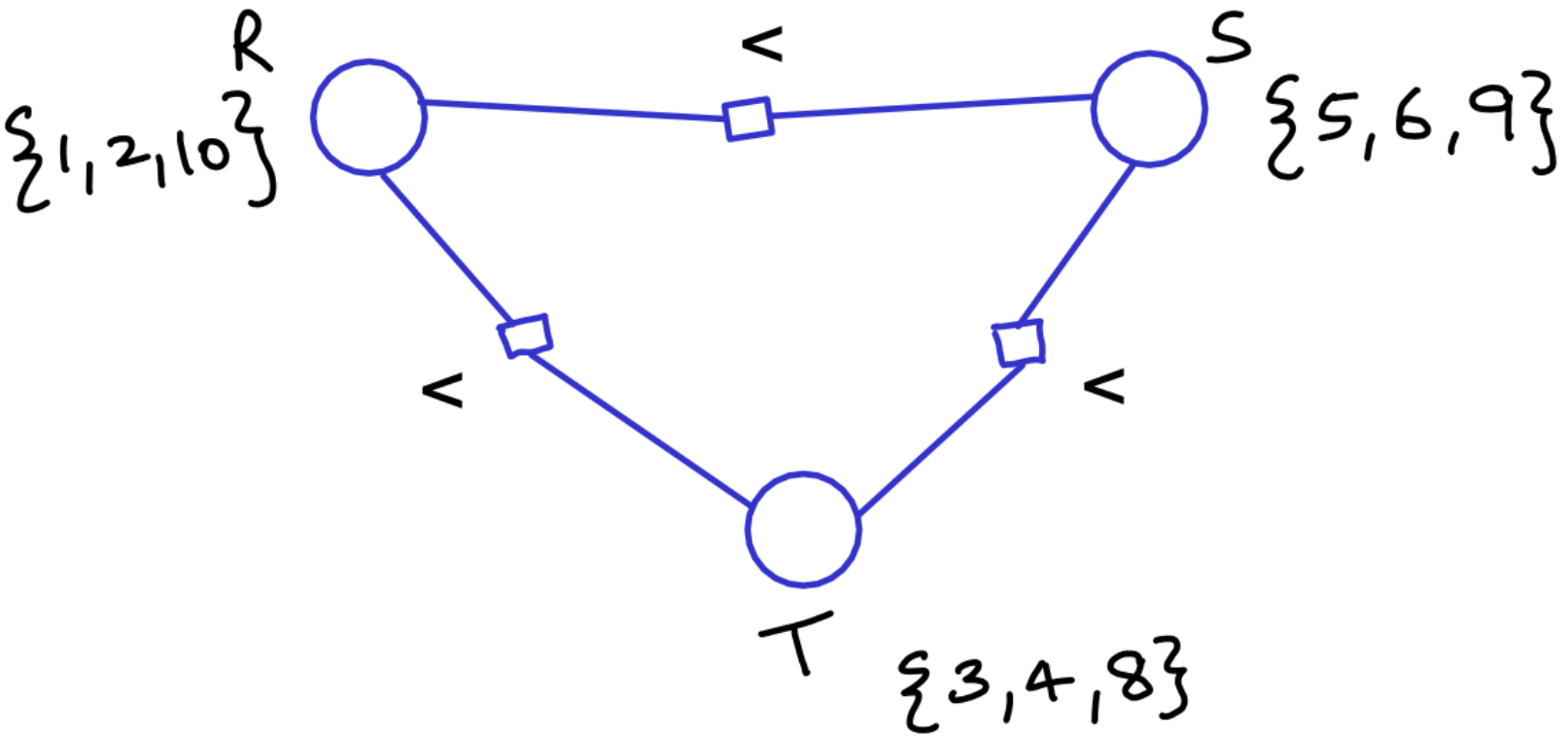


Does k-consistent CSP $\Rightarrow$ strongly-k-consistent CSP?



- Is the above CSP 3-consistent?

Does k-consistent CSP $\Rightarrow$ strongly-k-consistent CSP?



- Is the above CSP 3-consistent?
- Is the above CSP strongly-3-consistent?

Strongly-n-consistent

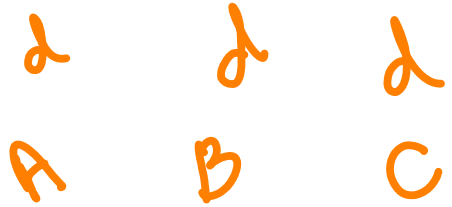
- Making a CSP strongly-n-consistent can take time exponential in n.

Strongly-n-consistent

- Making a CSP strongly-n-consistent can take time exponential in n.
- But, suppose that we have a strongly-n-consistent CSP.

Strongly-n-consistent

- Making a CSP strongly-n-consistent can take time exponential in n.
- But, suppose that we have a strongly-n-consistent CSP.
- We can find a solution in polynomial time.



Sudoku

	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		

$E6 = \{1, 2, \ldots 9\}$

$E6 = \{4\}$

	1	2	3	4	5	6	7	8	9
A	4	8	3	9	2	1	6	5	7
B	9	6	7	3	4	5	8	2	1
C	2	5	1	8	7	6	4	9	3
D	5	4	8	1	3	2	9	7	6
E	7	2	9	5	6	4	1	3	8
F	1	3	6	7	9	8	2	4	5
G	3	7	2	6	8	9	5	1	4
H	8	1	4	2	5	3	7	6	9
I	6	9	5	4	1	7	3	8	2

Sudoku: Alldiff constraints

$$Alldiff(A1, A2, A3, A4, A5, A6, A7, A8, A9)$$
$$Alldiff(B1, B2, B3, B4, B5, B6, B7, B8, B9)$$
$$\ldots$$
$$Alldiff(A1, B1, C1, D1, E1, F1, G1, H1, I1)$$
$$Alldiff(A2, B2, C2, D2, E2, F2, G2, H2, I2)$$
$$\ldots$$
$$Alldiff(A1, A2, A3, B1, B2, B3, C1, C2, C3)$$
$$Alldiff(A4, A5, A6, B4, B5, B6, C4, C5, C6)$$
$$\ldots$$

Alldiff constraint

E5 E1 E6

- $\{1, 8\}, \{3, 8\}, \{1, 3, 8\}$

Backtracking Search for CSPs

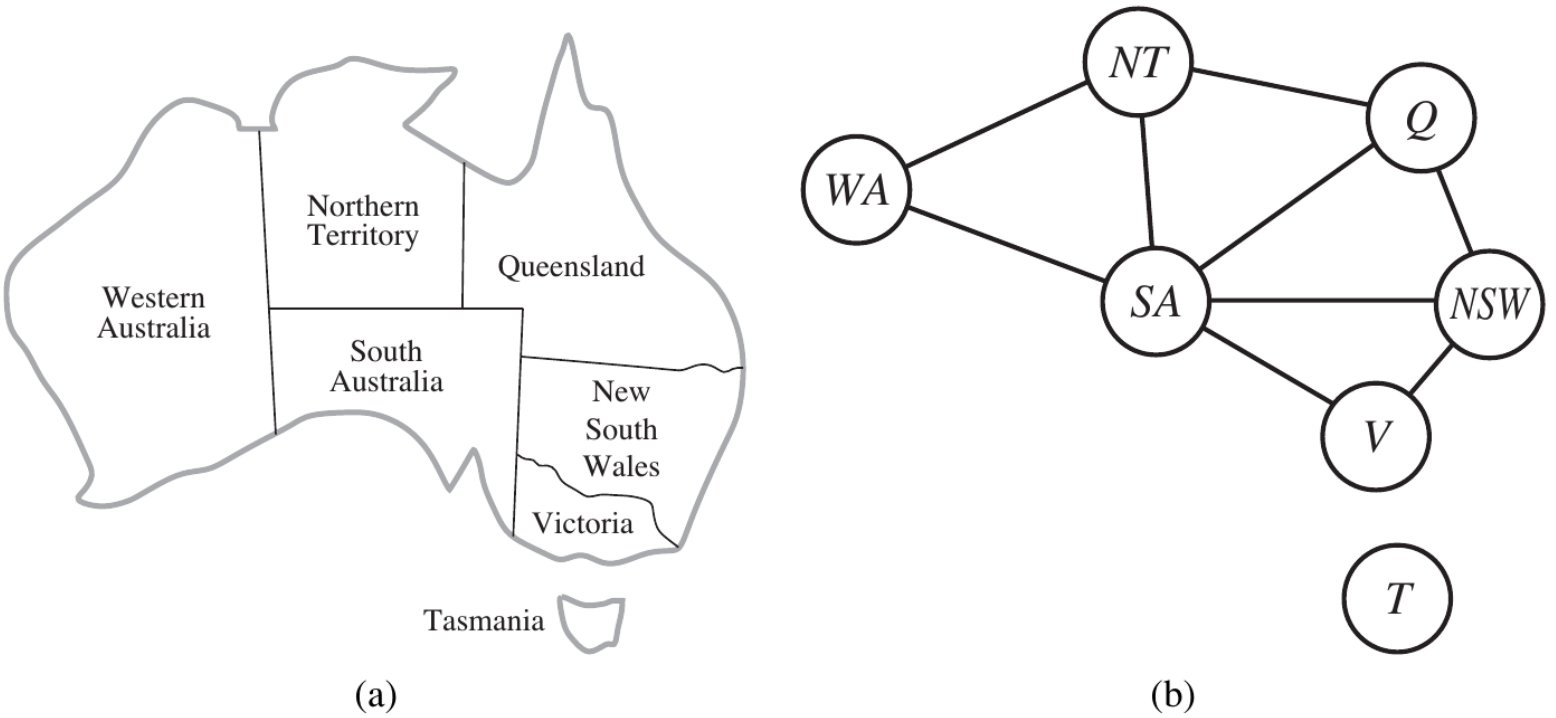
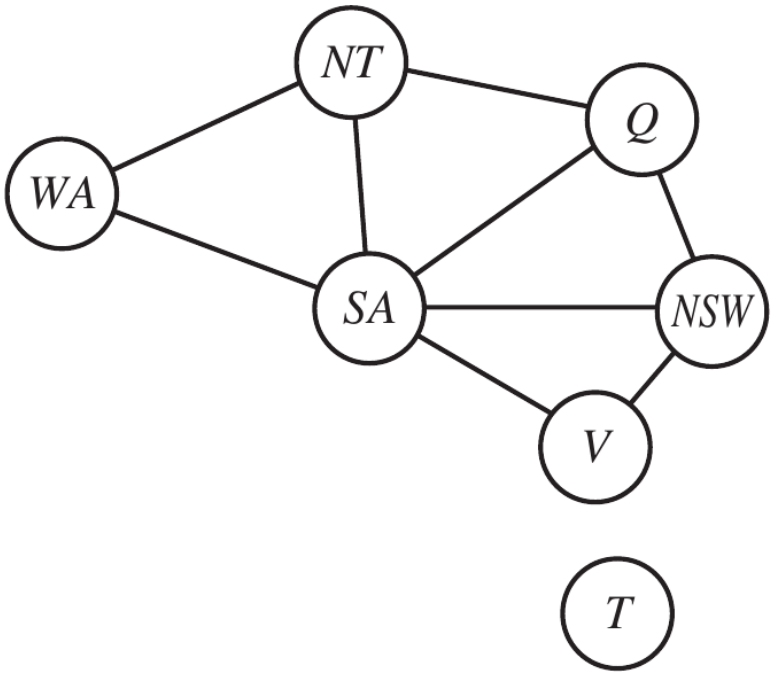
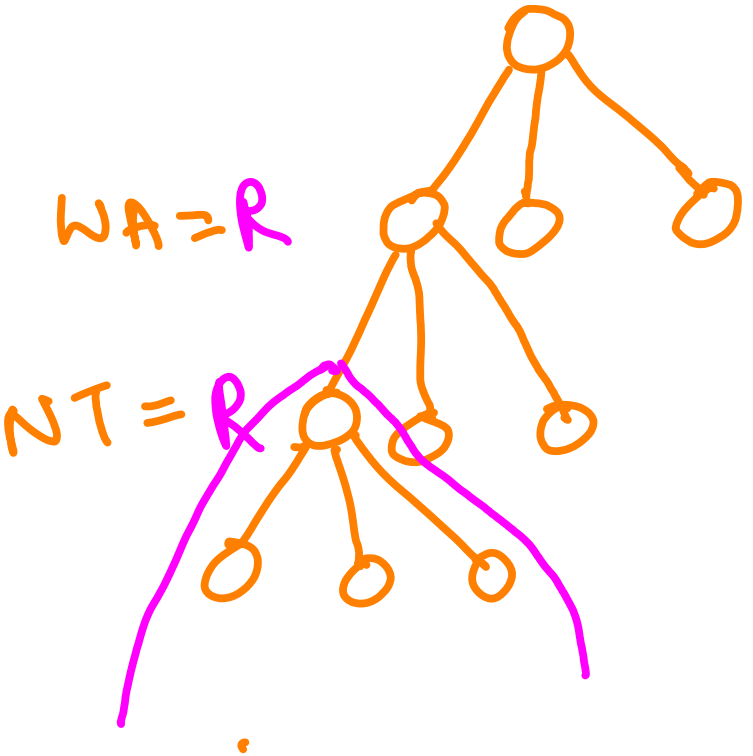


Figure 6.1 (a) The principal states and territories of Australia. Coloring this map can be viewed as a constraint satisfaction problem (CSP). The goal is to assign colors to each region so that no neighboring regions have the same color. (b) The map-coloring problem represented as a constraint graph.

- Commutativity

Backtracking Search for CSPs





n - level

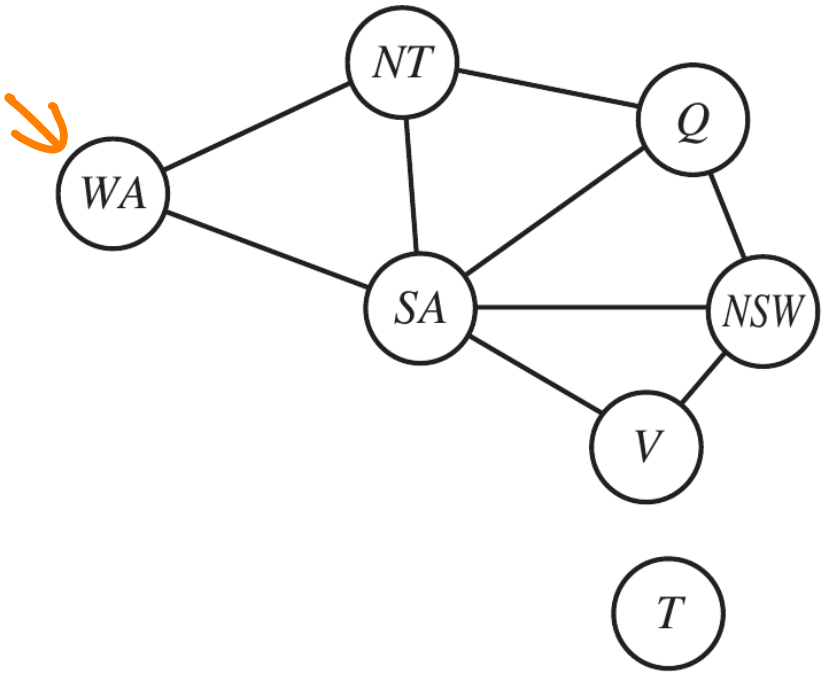
d^n

Backtracking Search for CSPs

```
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
```

Inference: Forward checking, Maintaining arc consistency



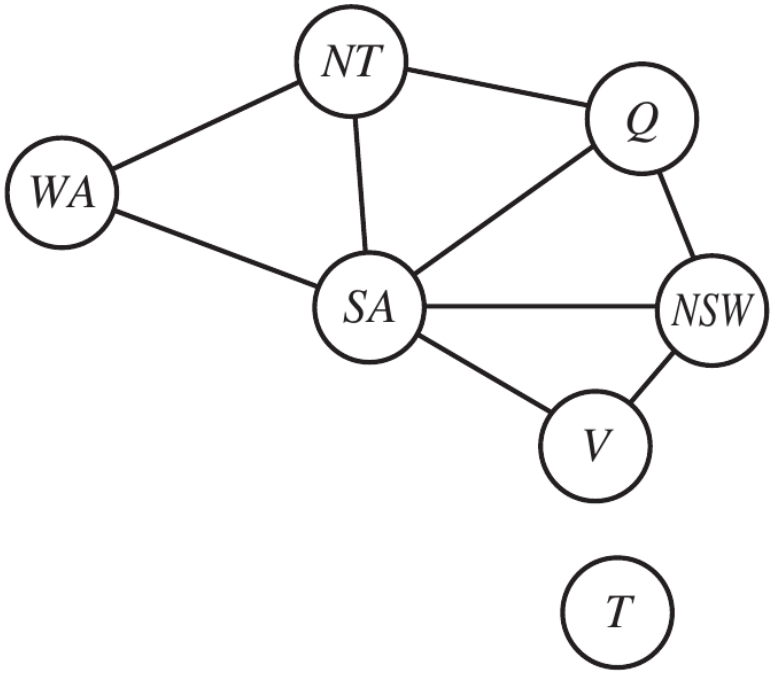
WA = red

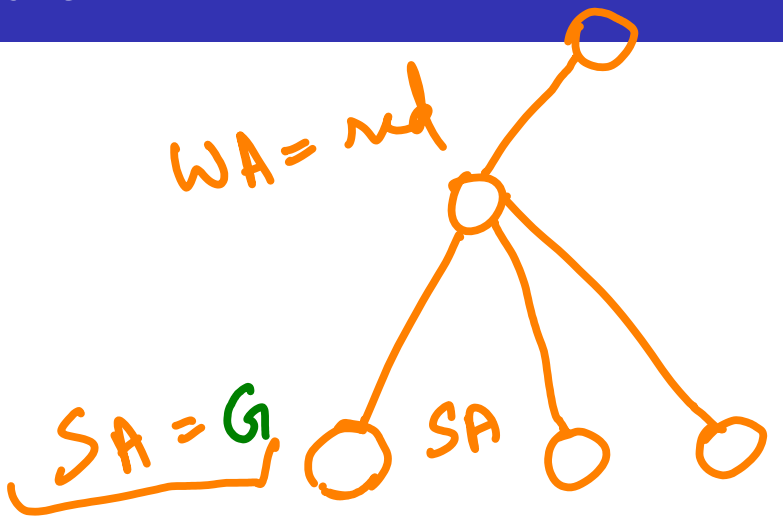
WA = {red}

→ (NT, WA) ←

→ (SA, WA) ←

Selecting an unassigned variable

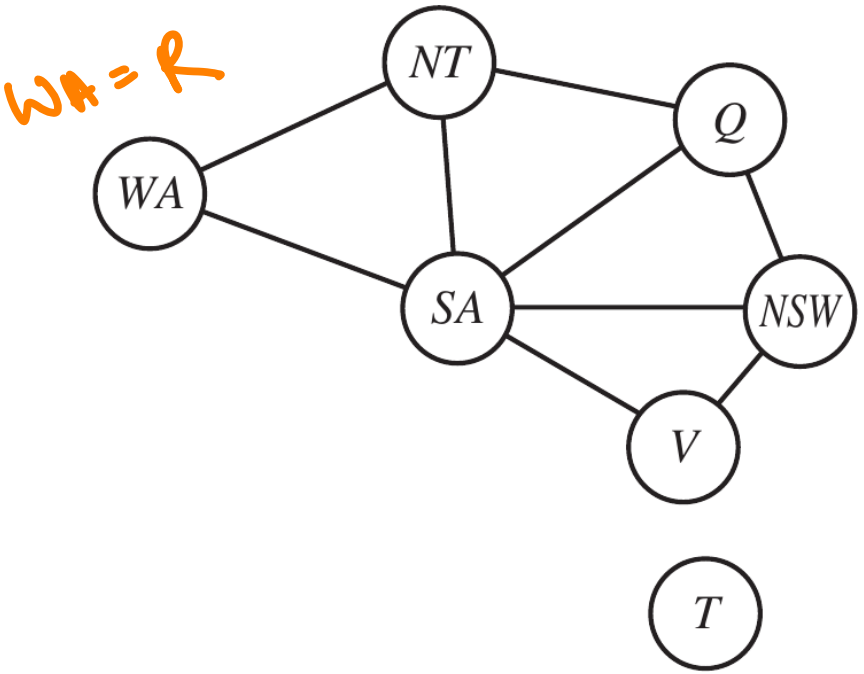




WA = red, SA = ____

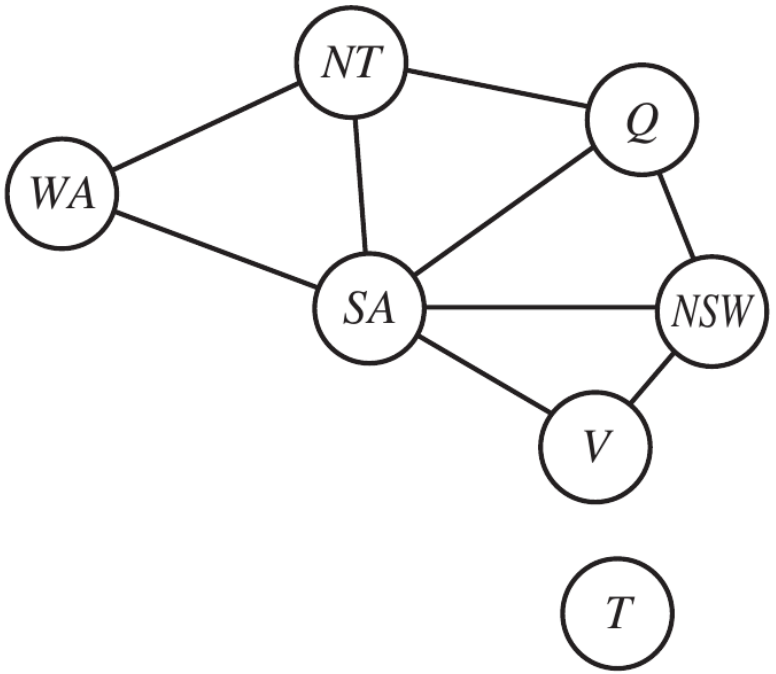
WA = red, SA = ____

Selecting an unassigned variable



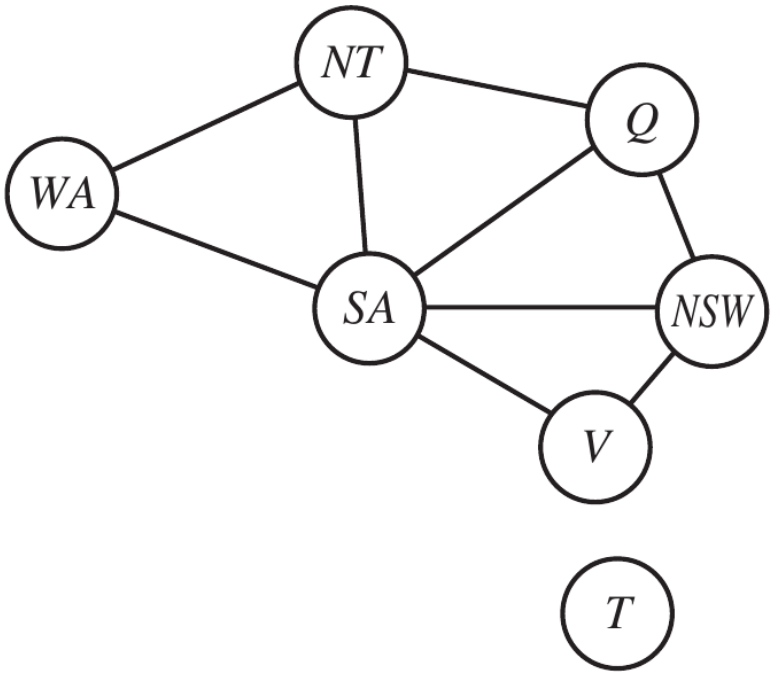
- Minimum-remaining-values heuristic

Selecting an unassigned variable



- Minimum-remaining-values heuristic
- Degree-heuristic

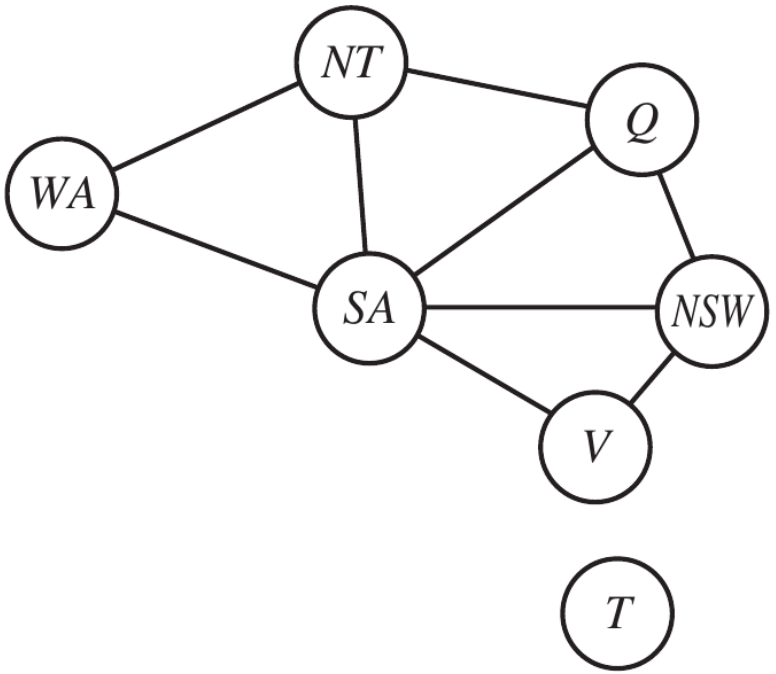
Domain values ordering



WA

SA =

Domain values ordering



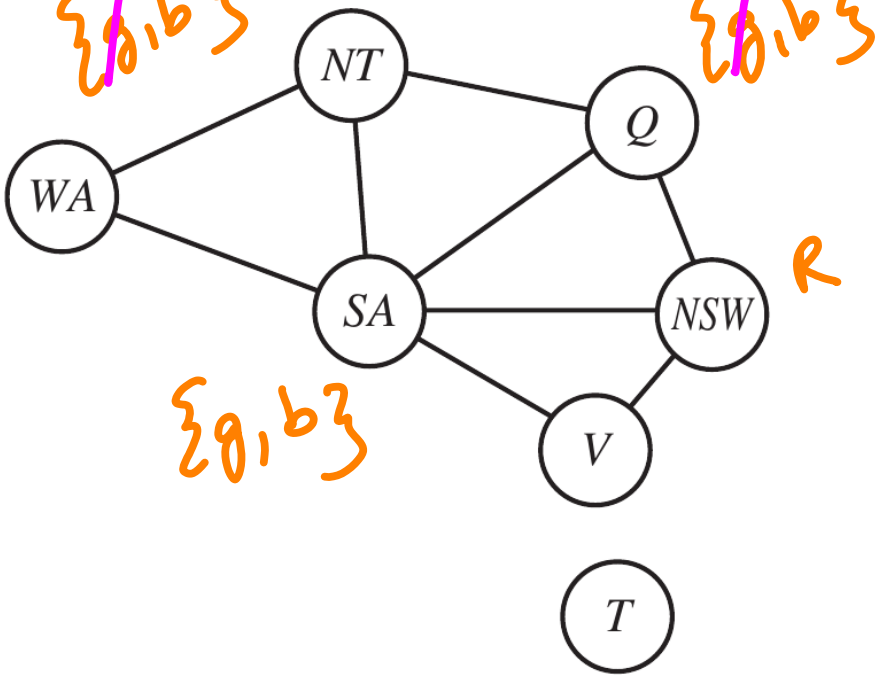
- Least-constraining-value heuristic

WA = red

Q = {blue, red}

NT = {blue, green}

Intelligent Backtracking



{R,B}

{R,B}

R

R

{R,B}

WA = Red

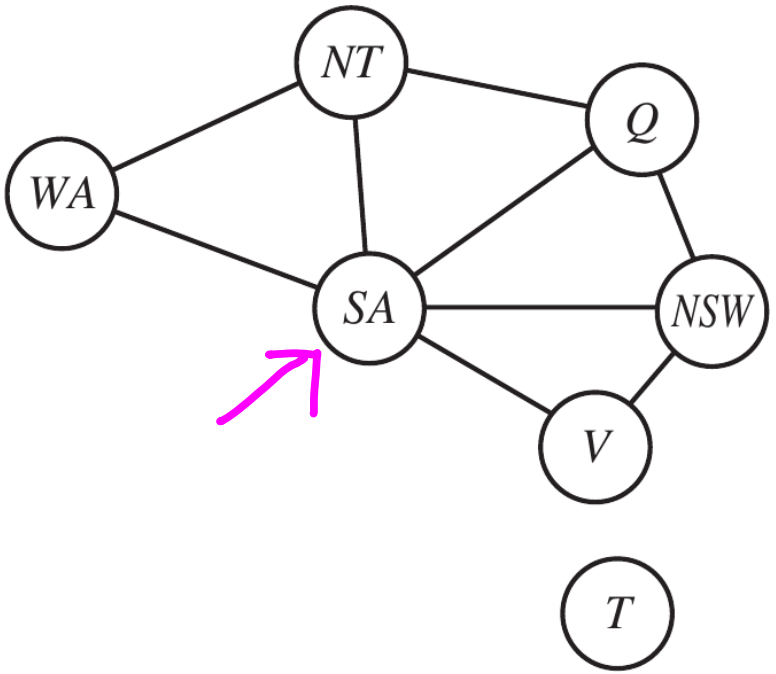
NSW = Red

T = {R}

SA = {R}

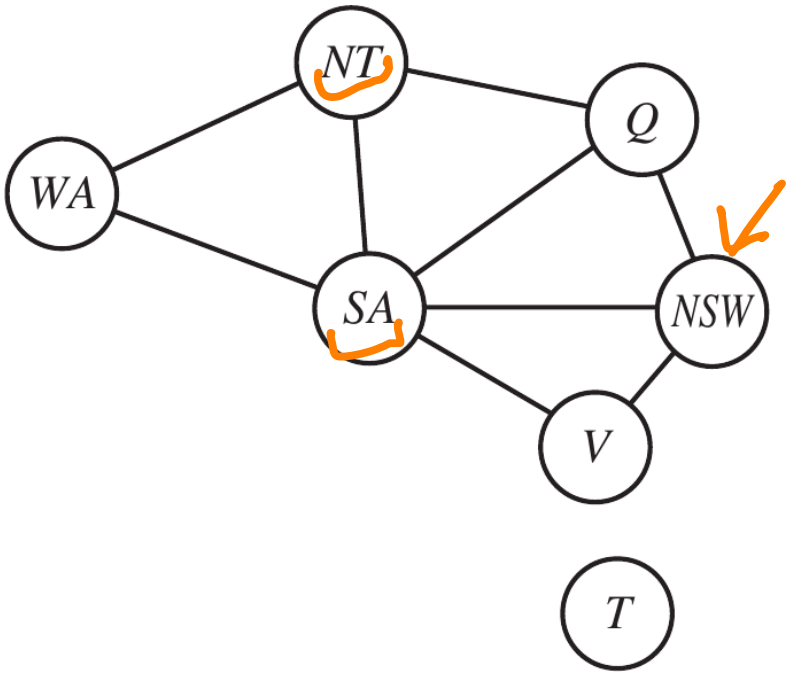
conf(SA) = { WA = R, NSW = R }

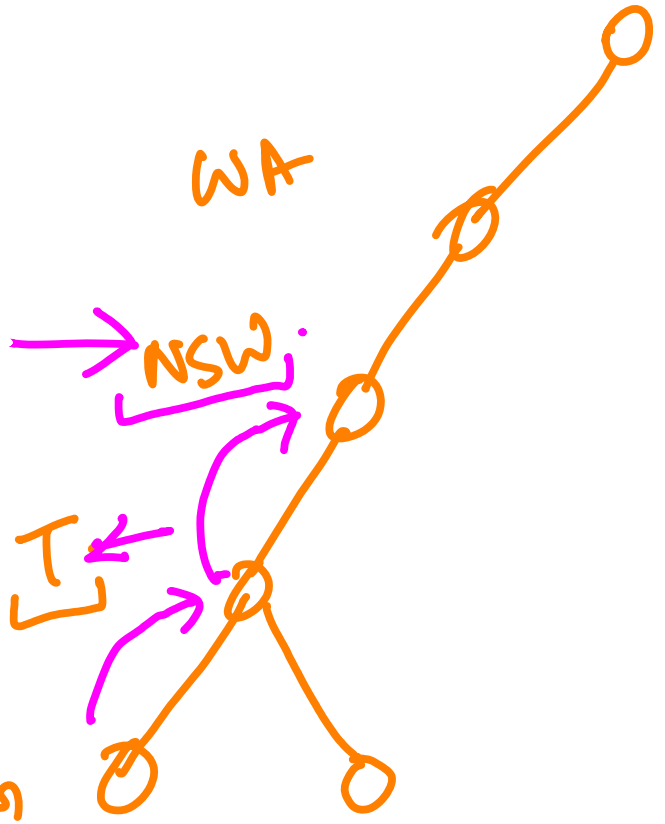
Intelligent Backtracking



- Chronological backtracking
- Conflict-directed backjumping

Intelligent Backtracking

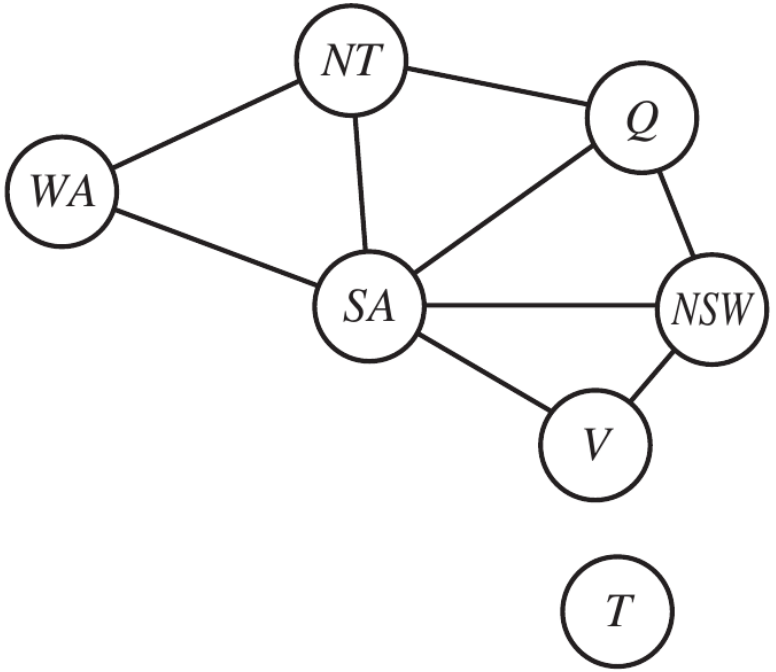




WA = red

NSW = red

Constraint Learning



[4, 4]