Artificial Intelligence

By Dr. Naglaa fathy
Lecturer, Computer Science Department
Faculty of Computers and Artificial Intelligence
Benha University

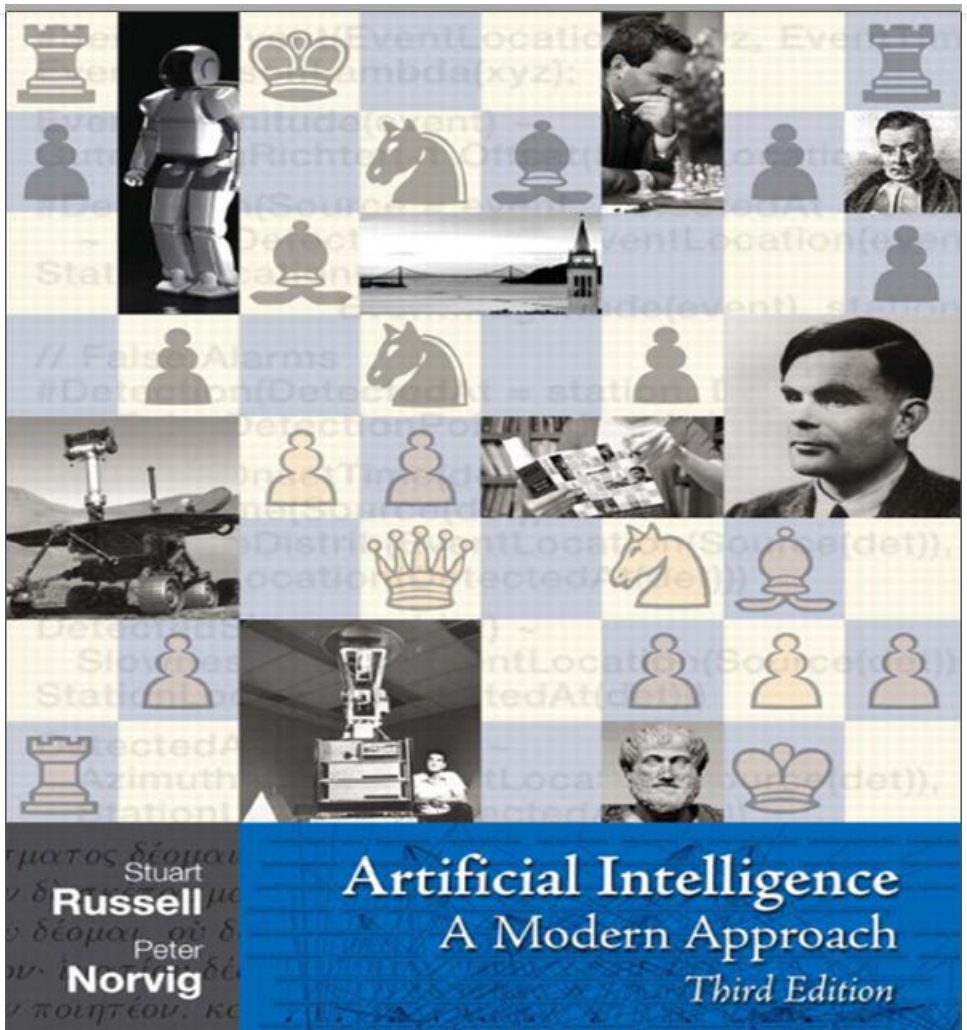
Lecture 8
Constraint Satisfaction Problems(CSP)

Lectures References

Artificial Intelligence
A Modern Approach

Third Edition

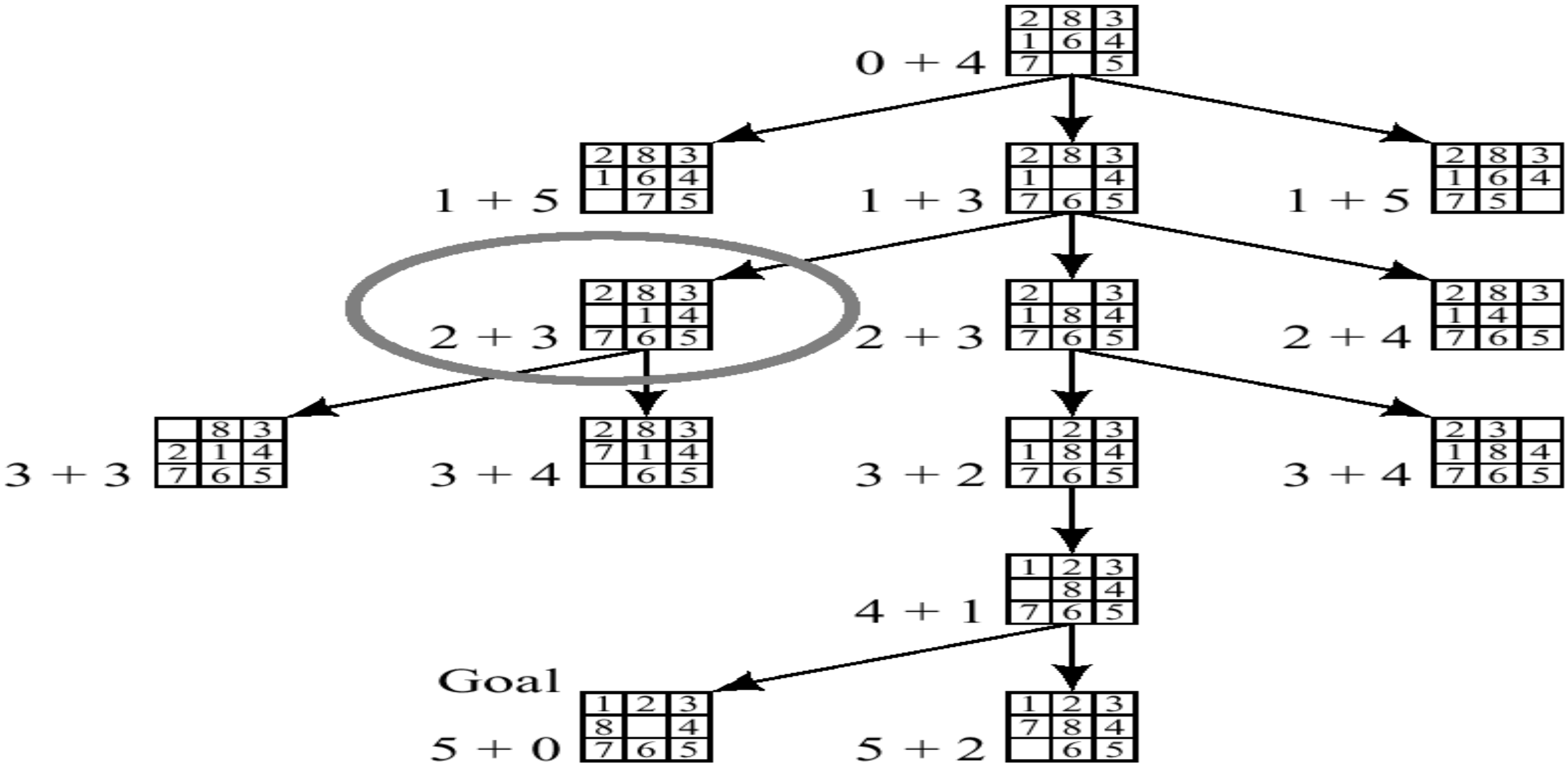
Stuart J. Russell and Peter Norvig



Agenda

- Constraint Satisfaction Problem (CSP)
- Backtracking of CSP
- Improving backtracking efficiency

A* on 8-puzzle with h(n)



Constraint Satisfaction Problem(CSP)

- A constraint satisfaction problem consists of three components, X, D, and C:
  - X is the set of variables, { X1, . . . , Xn }
  - D is a set of domains, { D1, . . . , Dn }, one for each variable.
  - C is a set of constraints that specify allowable combinations of values.
  - Goal test is a set of constraints specifying allowable combinations of values for subsets of variables.
  - A solution to a CSP an assignment of a domain value to each variable
  - Such that no constraints are broken

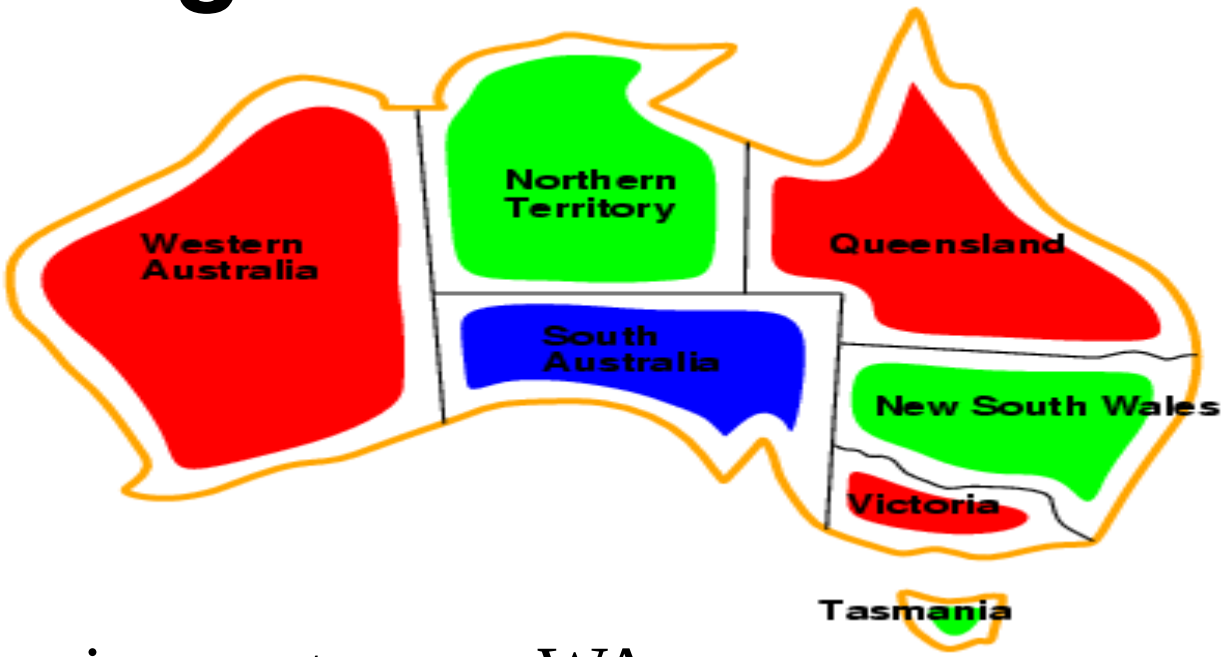
Example

- Suppose we have two variables: x and y
- x can take values {1,2,3}
- y can take values {2,3}
- Then the constraint x=y is written:
- {(2,2), (3,3)}
- The constraint x < y is written:
  - {(1,2), (1,3), (2,3)}
- The constraint x > y is written:
  - {(3,2)}
- The constraint x!= y is written:
  - {(1,2), (1,3), (2,3)}

Example: Map-Coloring

- Variables *WA, NT, Q, NSW, V, SA, T*
- Domains D_i = {red,green,blue}
- Constraints: adjacent regions must have different colors.
- e.g., WA ≠ NT, NT ≠ SA, SA ≠ Q,Q ≠ NSW,
- WA ≠ SA,SA ≠ NSW,SA ≠ V,NSW ≠ V

Example: Map-Coloring



- Solutions are complete and consistent assignments, e.g., WA = red, NT = green,Q = red,NSW = green,V = red,SA = blue,T = green.

Example: Cryptarithmetic

- Variables: $F\ T\ U\ W\ R\ O\ X_1\ X_2\ X_3$
- Domains: $\{0,1,2,3,4,5,6,7,8,9\}$
- Constraints: *Alldiff (F,T,U,W,R,O).*
  - $O + O = R + 10 \cdot X_1$
  - $X_1 + W + W = U + 10 \cdot X_2$
  - $X_2 + T + T = O + 100 \cdot X_3$
  - $X_3 = F,\ T \neq 0,\ F \neq 0$
- Solution : T=9,W=3,O=8

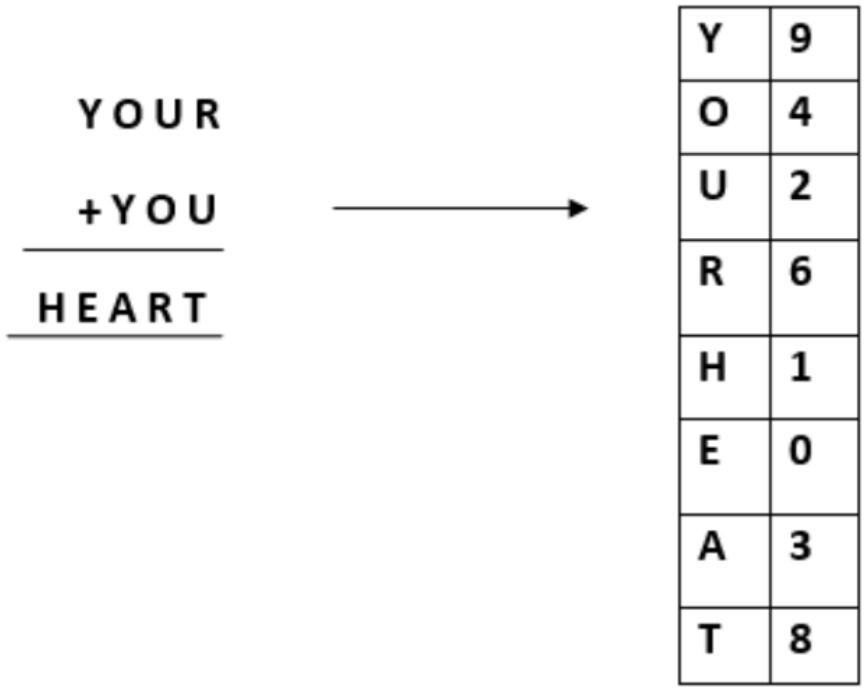
  F=1,U=7,R=6

```
    T W O
+   T W O
-------
F O U R
```

Example: Cryptarithmetic

- Variables: $S\ E\ N\ D\ M\ O\ R\ X_1\ X_2\ X_3\ X_4$
- Domains: $\{0,1,2,3,4,5,6,7,8,9\}$
- Constraints: $Alldiff\ (S\ ,\ E\ ,N\ ,D\ ,M\ ,O\ ,R\ ).$
  - $D+ E = E + 10 \cdot X_1$
  - $X_1 + N + R = E + 10 \cdot X_2$
  - $X_2 + O+ E= N + 10 \cdot X_3$
  - $X_3 + S+ M= O + 10 \cdot X_4$
  - $X_4 = M,\ S\neq 0,\ M \neq 0$
- Solution : S=9,E=5,N=6
  D=7,M=1,O=0
  R=8,Y=2

Example: Cryptarithmetic



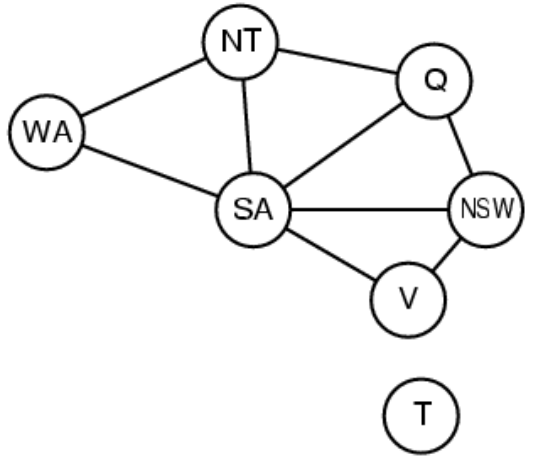
Y	9
O	4
U	2
R	6
H	1
E	0
A	3
T	8

Example: Cryptarithmetic

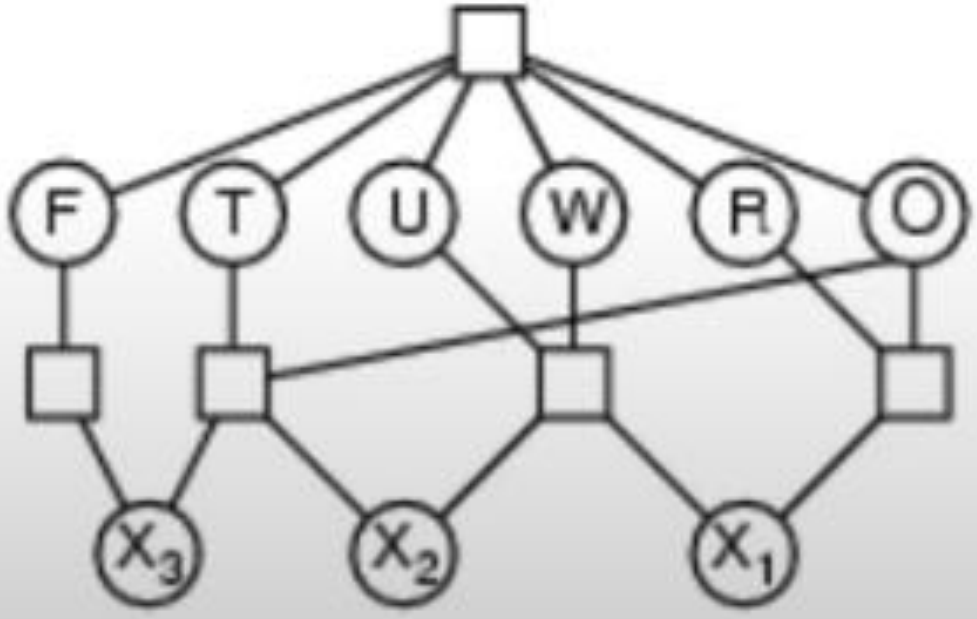
B	7
A	4
S	8
E	3
L	5
G	1
M	9

Constraint graph

- Binary CSP: each constraint relates two variables.
- Constraint graph: nodes are variables, arcs are constraints.



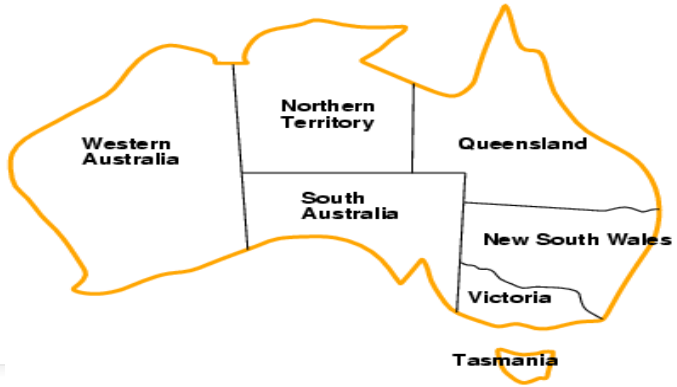
Constraint graph



Varieties of constraints

- Unary constraints involve a single variable,
  - e.g., SA ≠ green.
- Binary constraints involve pairs of variables,
  - e.g., SA ≠ WA.
- Higher-order constraints involve 3 or more variables,
  - e.g., cryptarithmetic column constraints.

Backtracking search



- Variable assignments are commutative, i.e.,

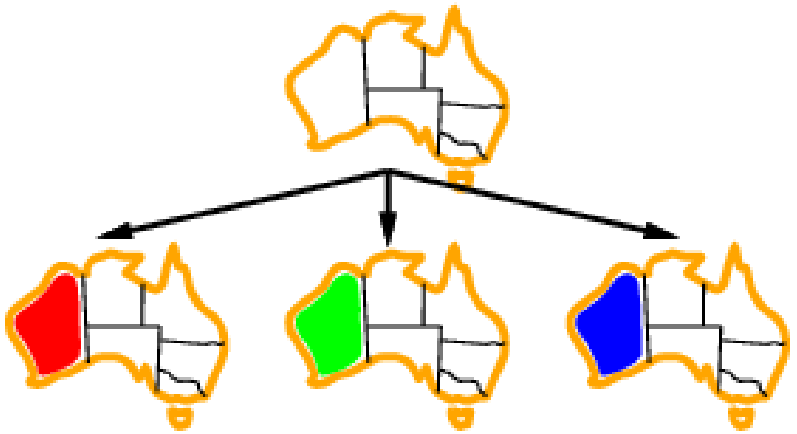
[WA = red then NT = green] same as [NT = green then WA = red]

- Depth-first search for CSPs with single-variable assignments is called backtracking search.

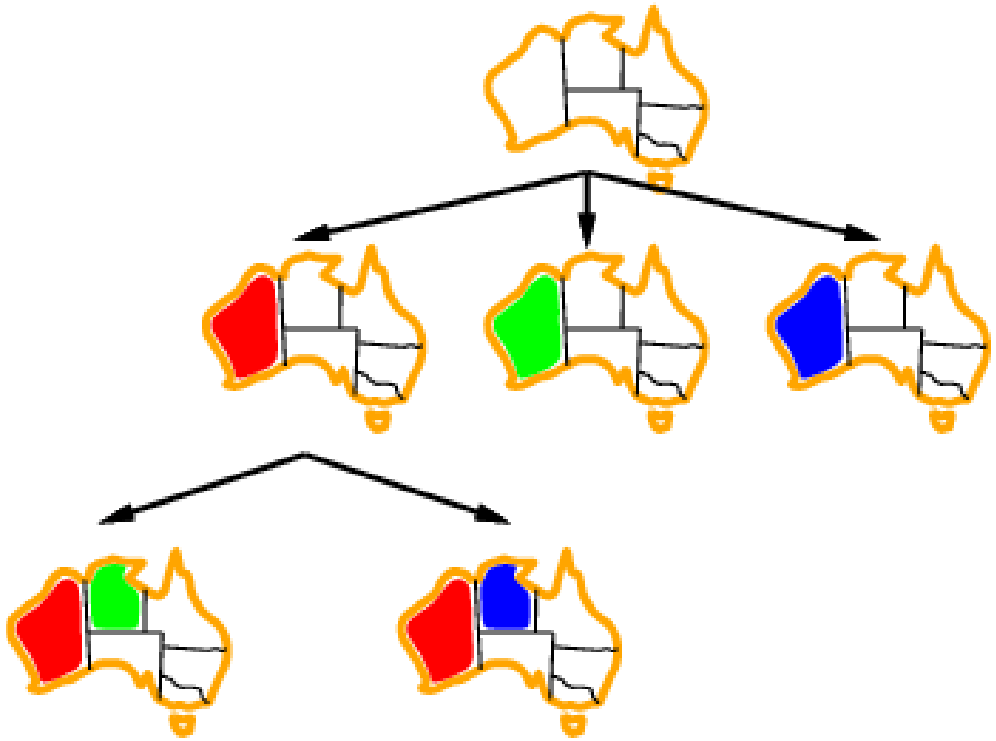
- Backtracking search is the basic uninformed algorithm for CSPs.

Backtracking example

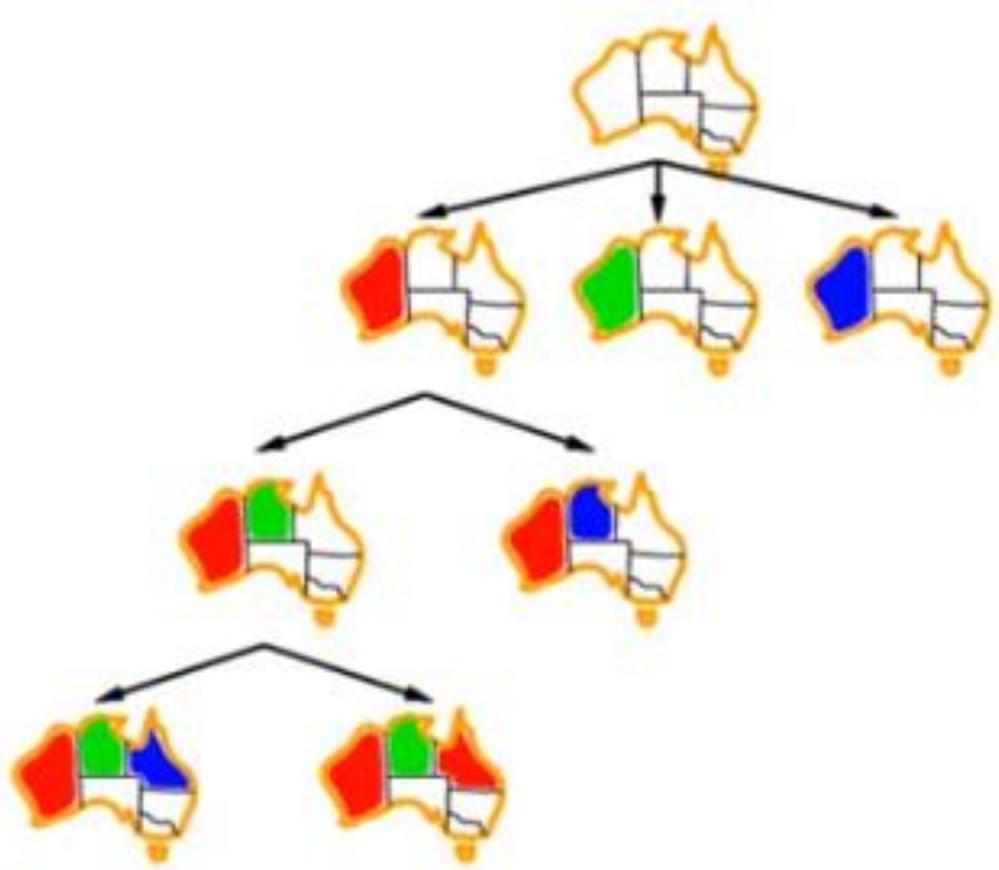
Backtracking example



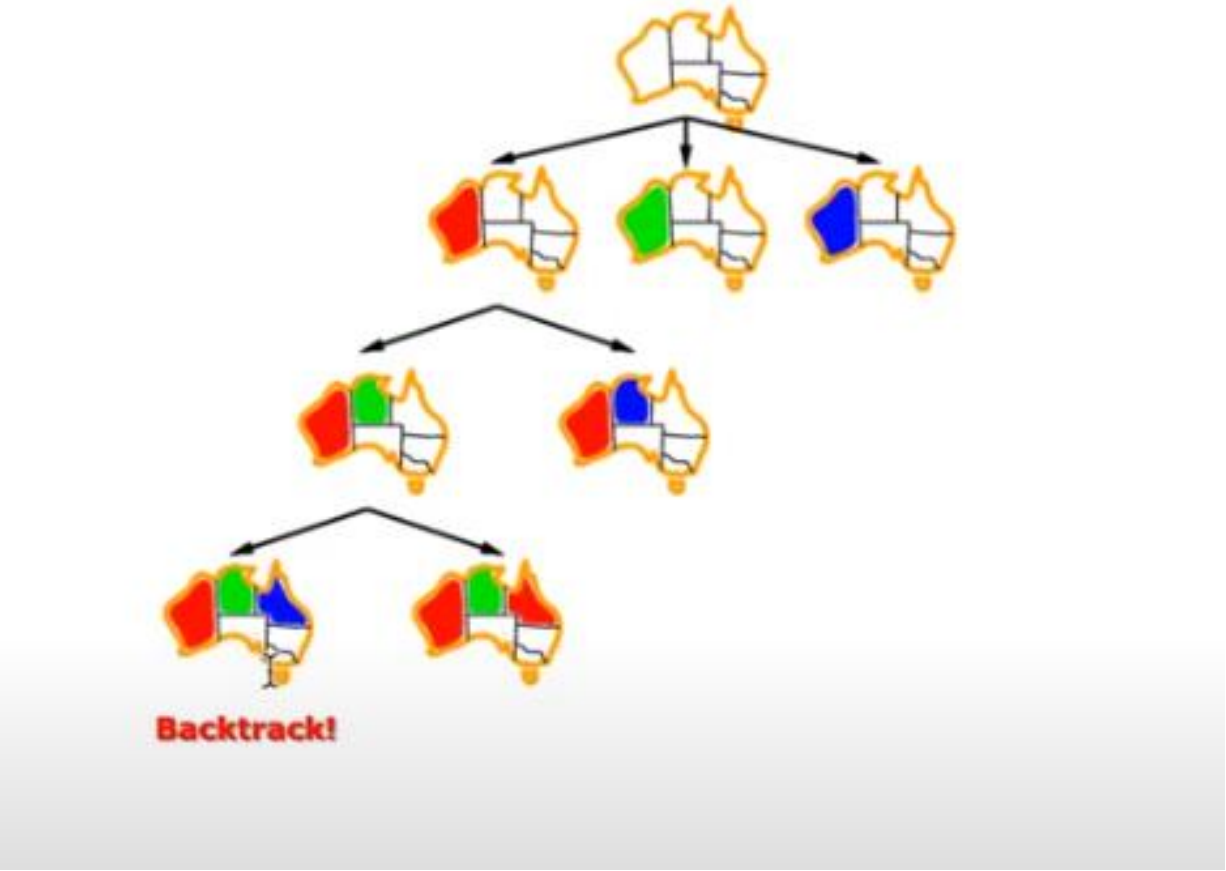
Backtracking example



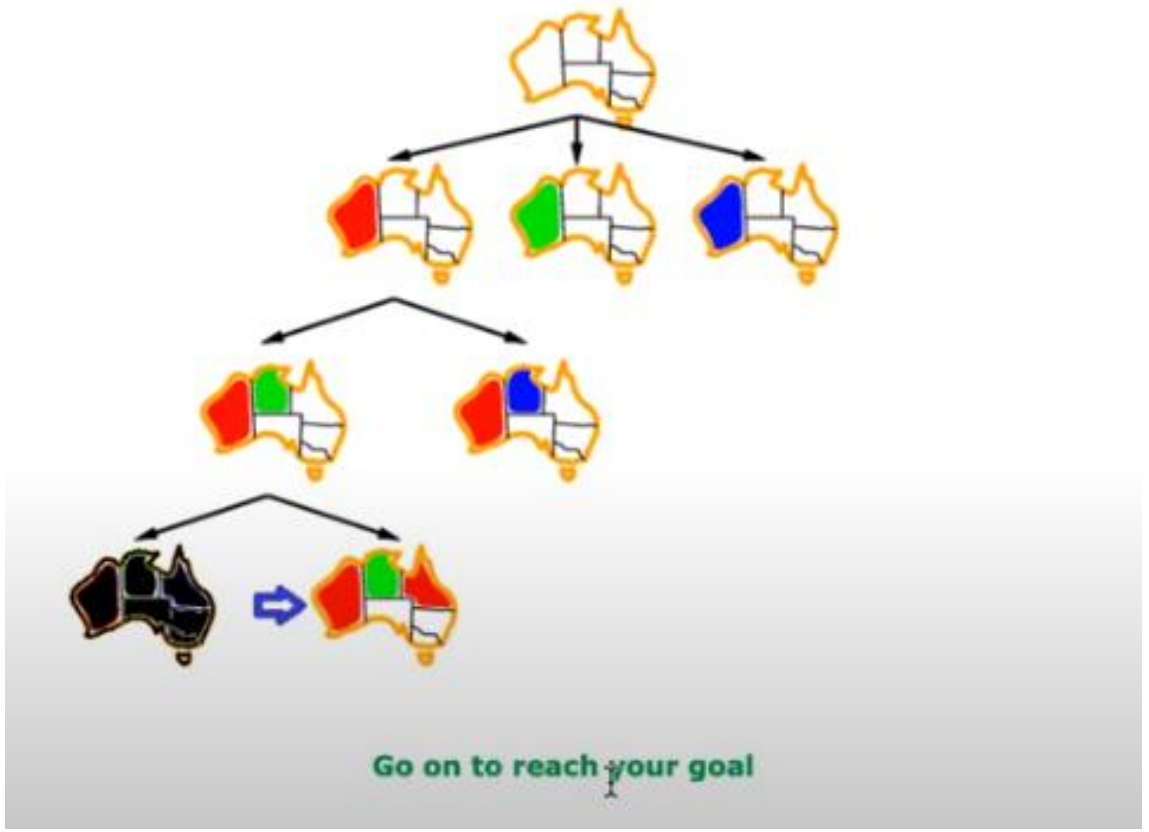
Backtracking example



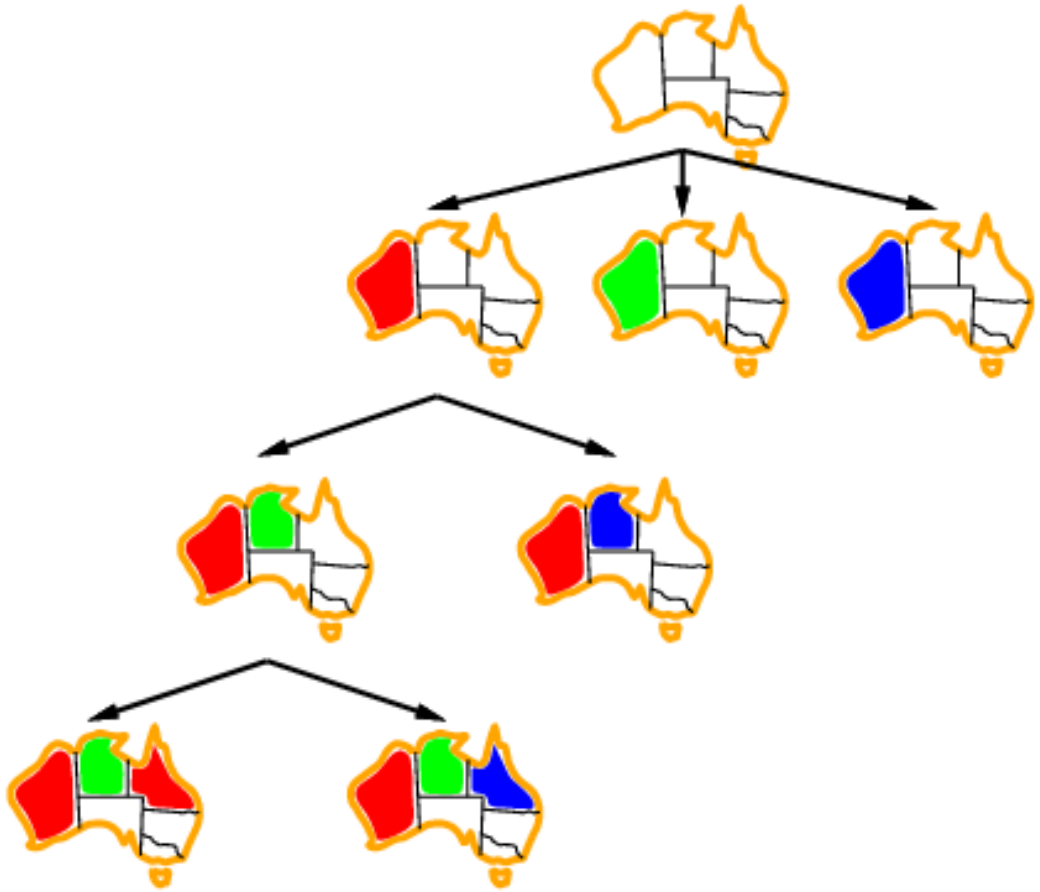
Backtracking example



Backtracking example



Backtracking example



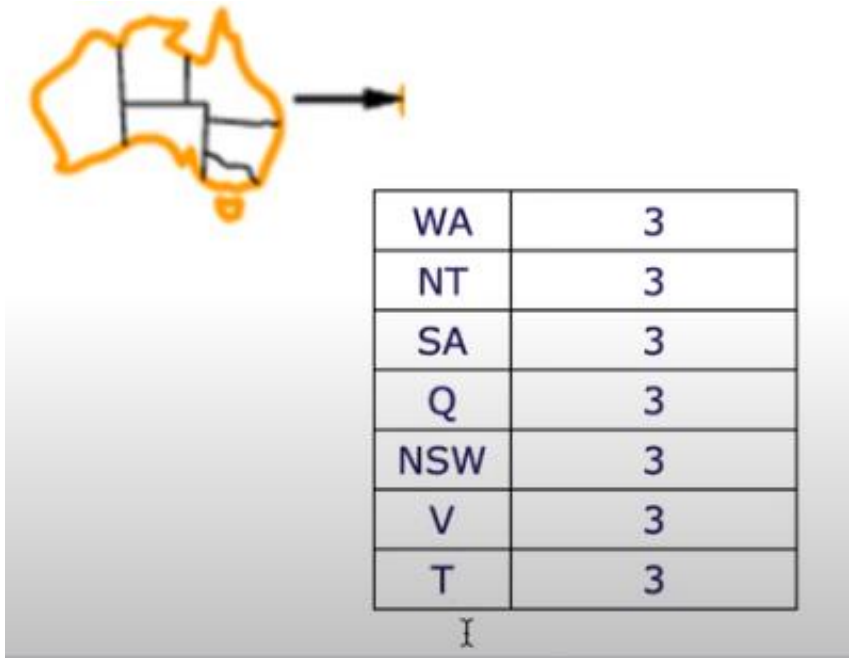
Improving backtracking efficiency

- General-purpose methods can give huge gains in speed:
  - Which variable should be assigned next?
  - In what order should its values be tried?
  - Can we detect inevitable failure early?

Which variable should be assigned next?

- Minimum Remaining Values (MRV)

  choose the variable with the fewest legal values.

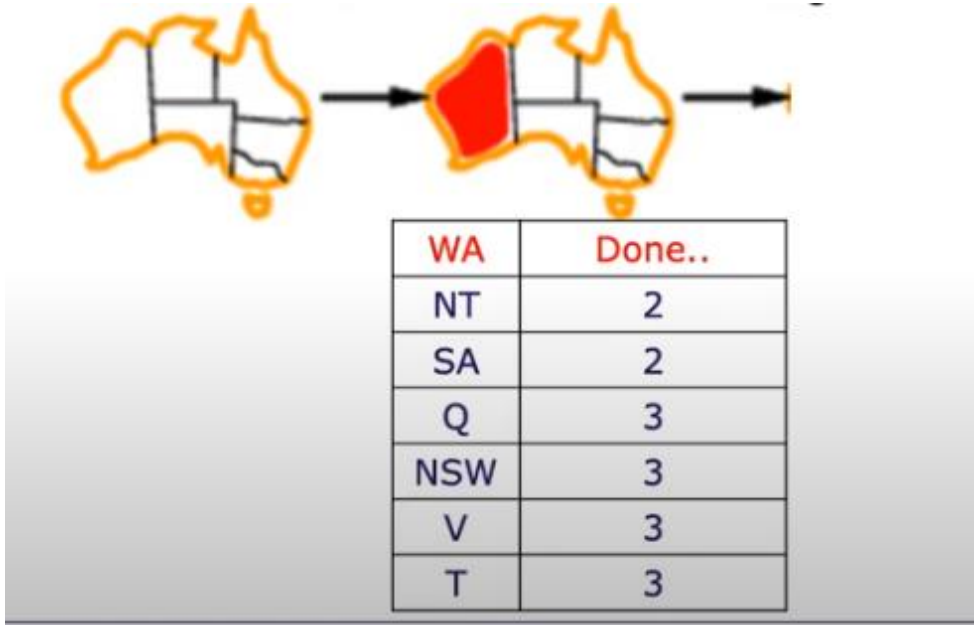


WA	3
NT	3
SA	3
Q	3
NSW	3
V	3
T	3

Which variable should be assigned next?

- minimum remaining values (MRV)

  choose the variable with the fewest legal values.



WA	Done..
NT	2
SA	2
Q	3
NSW	3
V	3
T	3

Which variable should be assigned next?

- minimum remaining values (MRV)

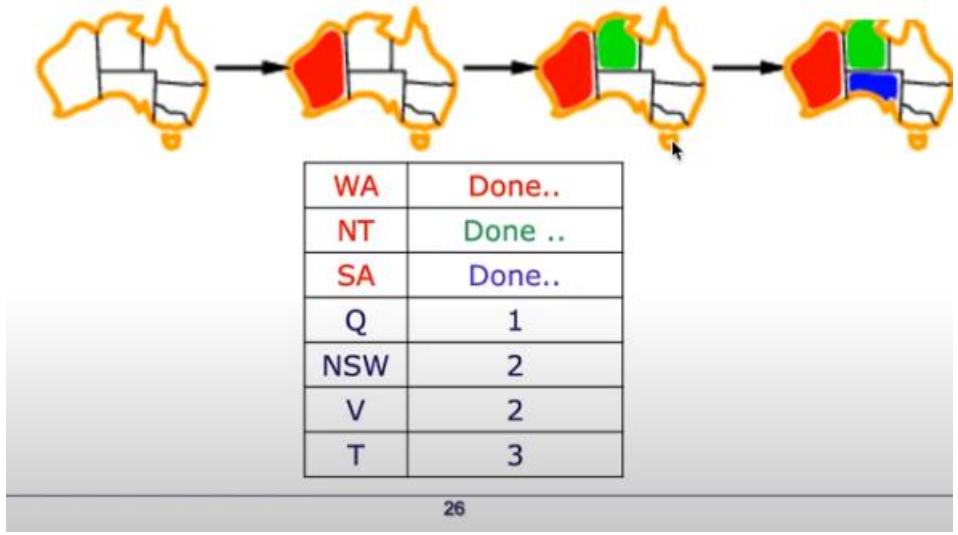
  choose the variable with the fewest legal values.

WA	Done..
NT	Done ..
SA	1
Q	2
NSW	3
V	3
T	3

Which variable should be assigned next?

- minimum remaining values (MRV)

  choose the variable with the fewest legal values.



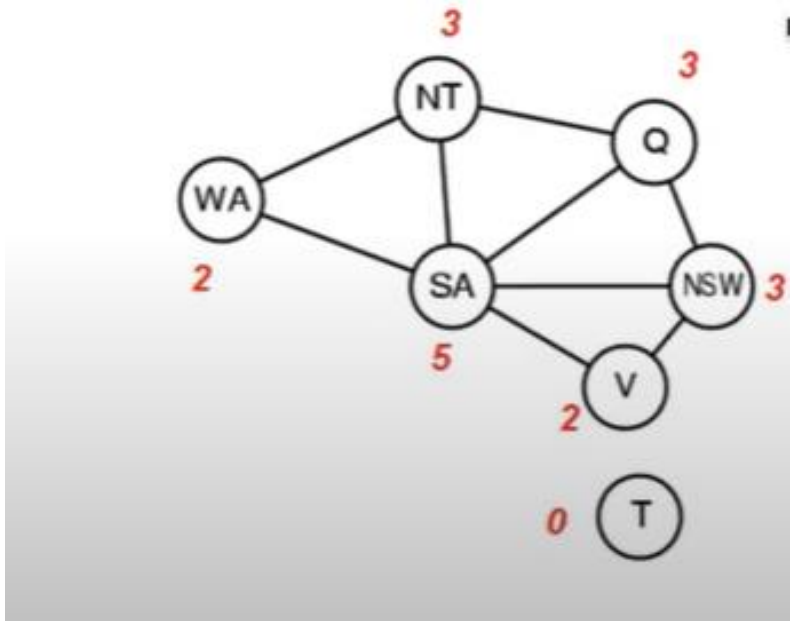
WA	Done..
NT	Done ..
SA	Done..
Q	1
NSW	2
V	2
T	3

26

Which variable should be assigned next?

- Most Constrain Variable (MCV)

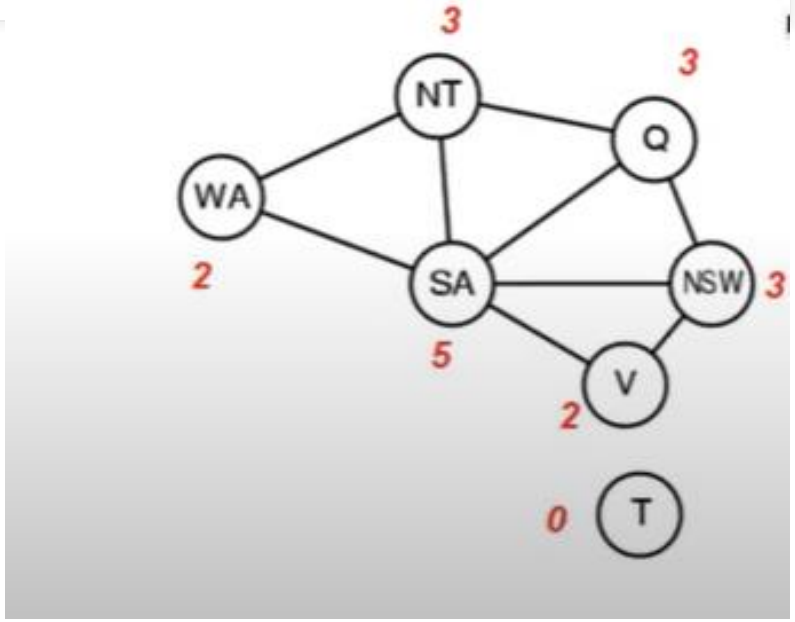
  choose the variable with the most constraint on



Which variable should be assigned next?

- Most constrain variable (MCV)

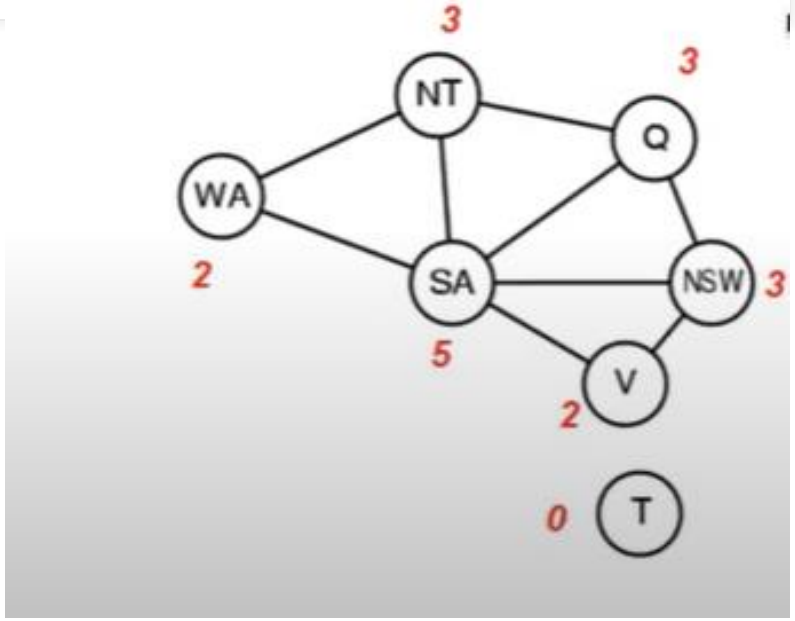
  choose the variable with the most constraint on



Which variable should be assigned next?

- Most constrain variable (MCV)

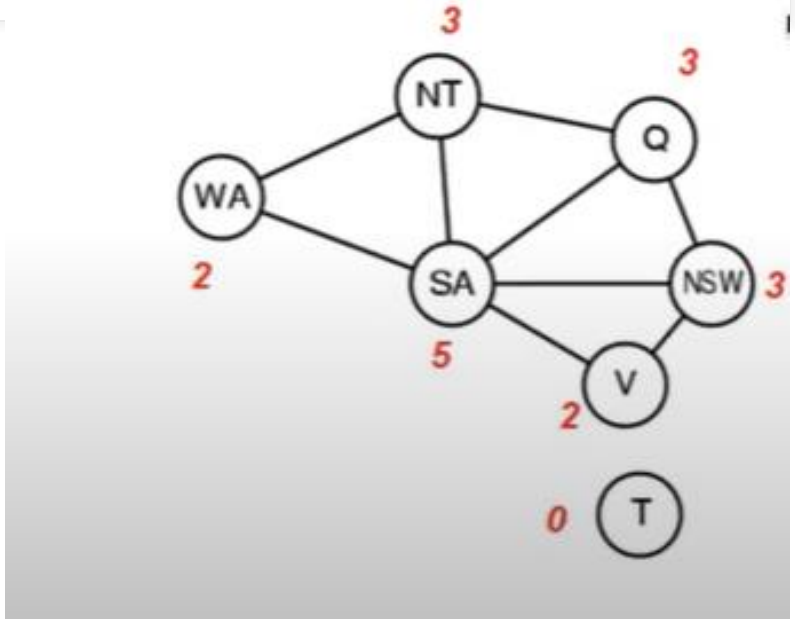
  choose the variable with the most constraint on



Which variable should be assigned next?

- Most constrain variable (MCV)

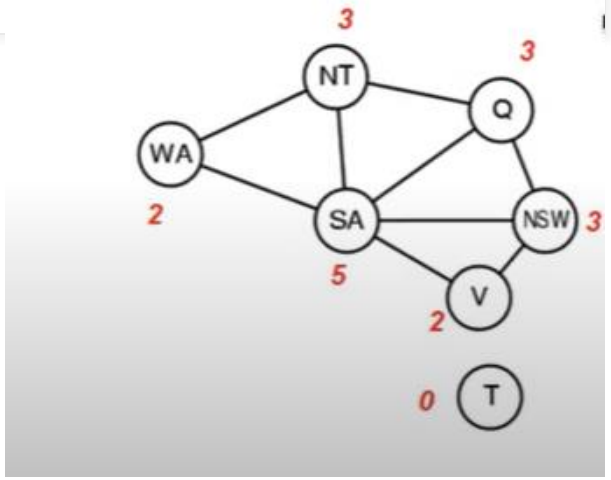
  choose the variable with the most constraint on

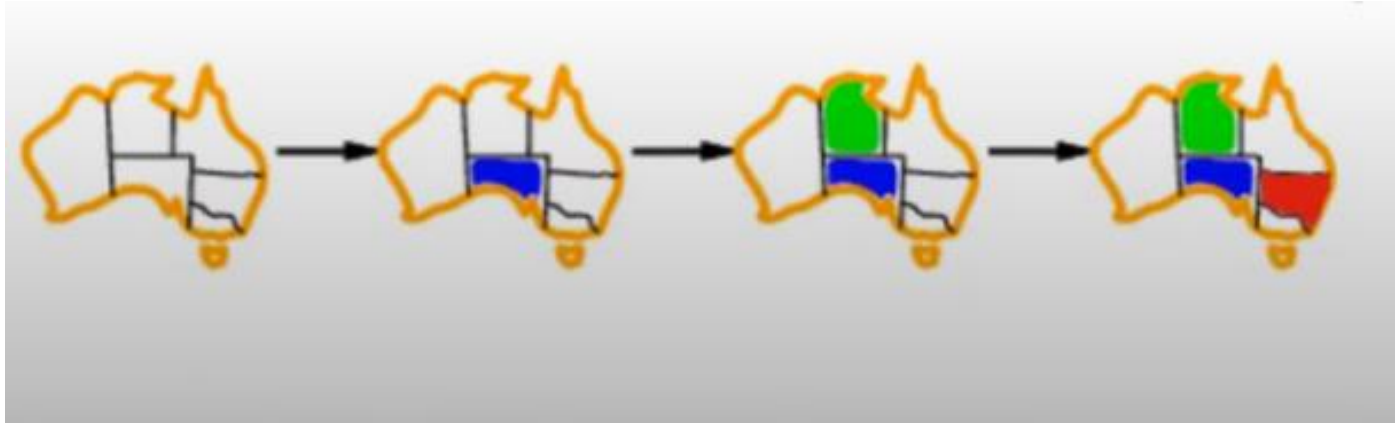


Which variable should be assigned next?

- Most constrain variable (MCV)

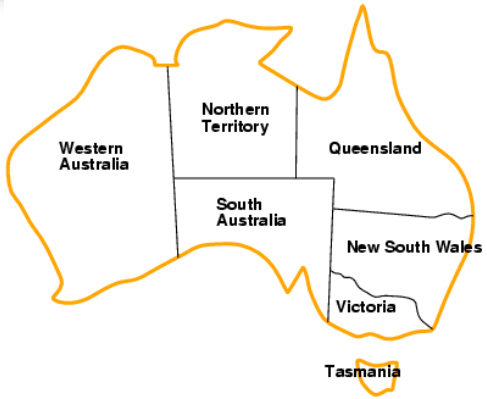
  choose the variable with the most constraint on



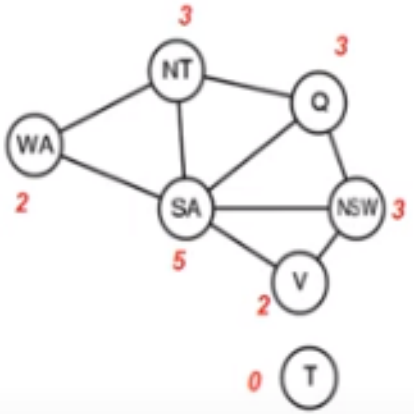


Which variable should be assigned next?

- Usually first applies (MRV) and break ties by MCV

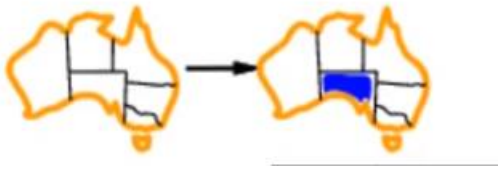


WA	3
NT	3
SA	3
Q	3
NSW	3
V	3
T	3

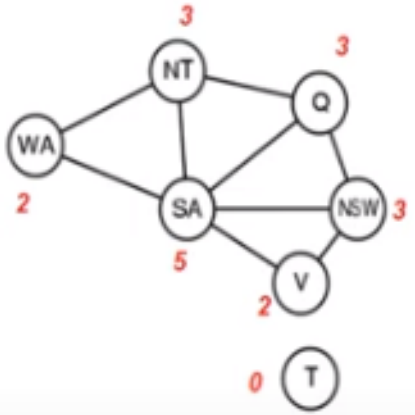


Which variable should be assigned next?

- Usually first applies (MRV) and break ties by MCV



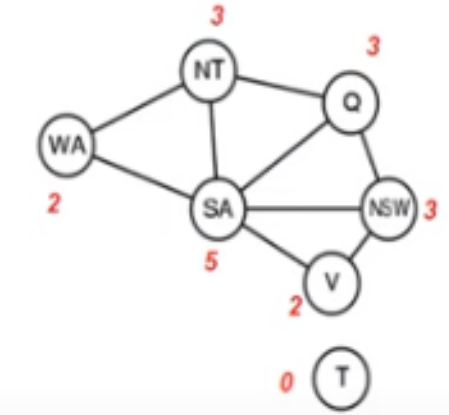
WA	2
NT	2
SA	Done..
Q	2
NSW	2
V	2
T	3



Which variable should be assigned next?

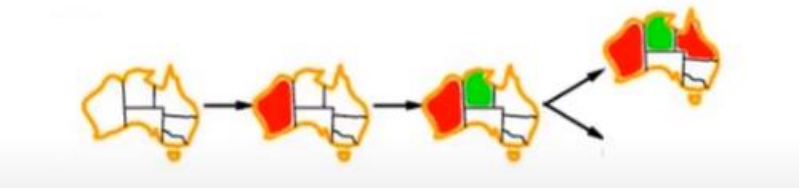
- Usually first applies (MRV) and break ties by MCV

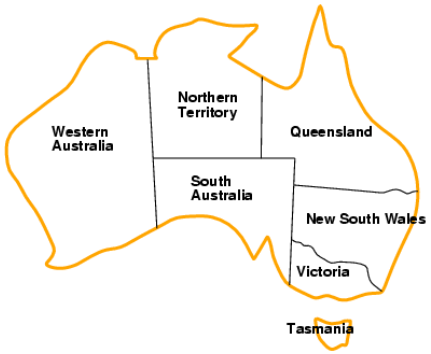
WA	1
NT	Done..
SA	Done..
Q	1
NSW	2
V	2
T	3



what order should its values be tried?

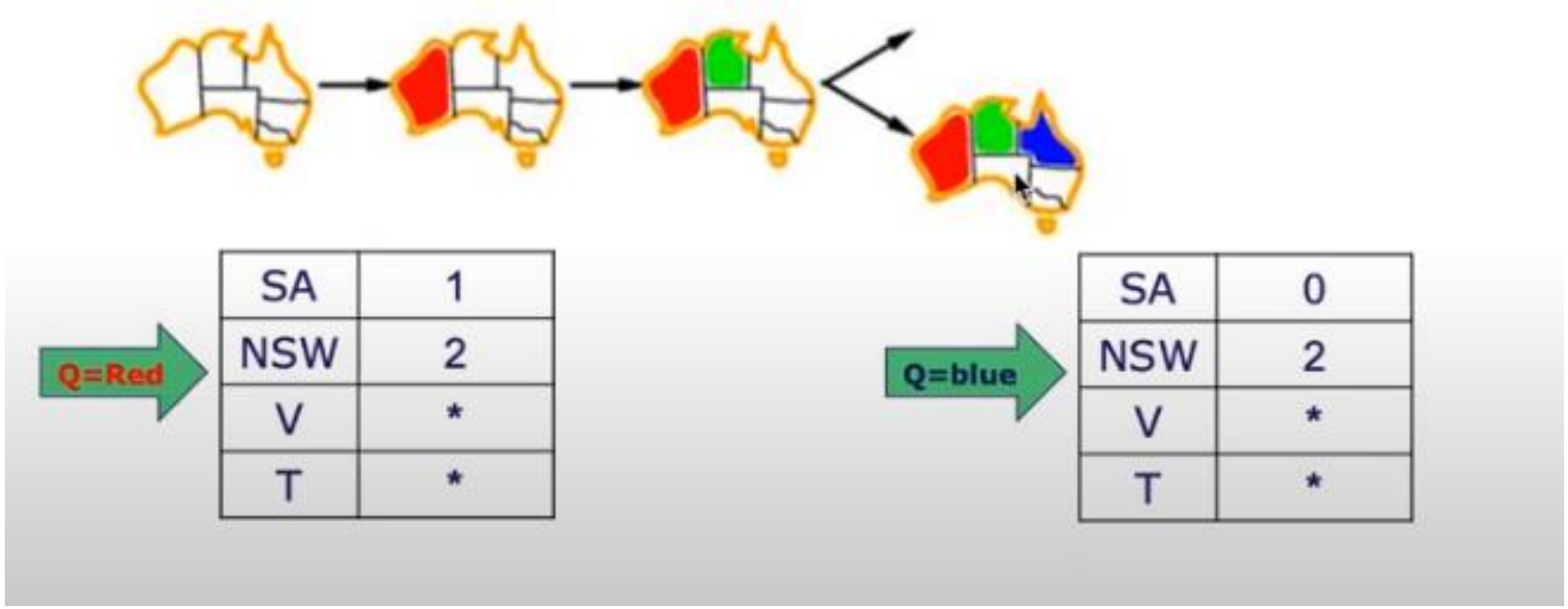
- Least constraint value(LCV)
  - Given available, choose the lest constraint value





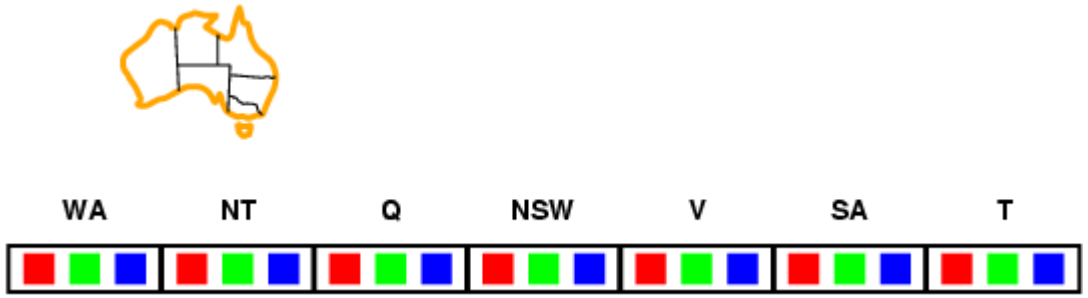
what order should its values be tried?

- Least constraint value(LCV)
  - Given available, choose the lest constraint value



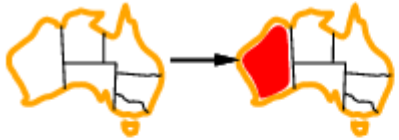
Can we detect inevitable failure early? Forward checking

- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values.



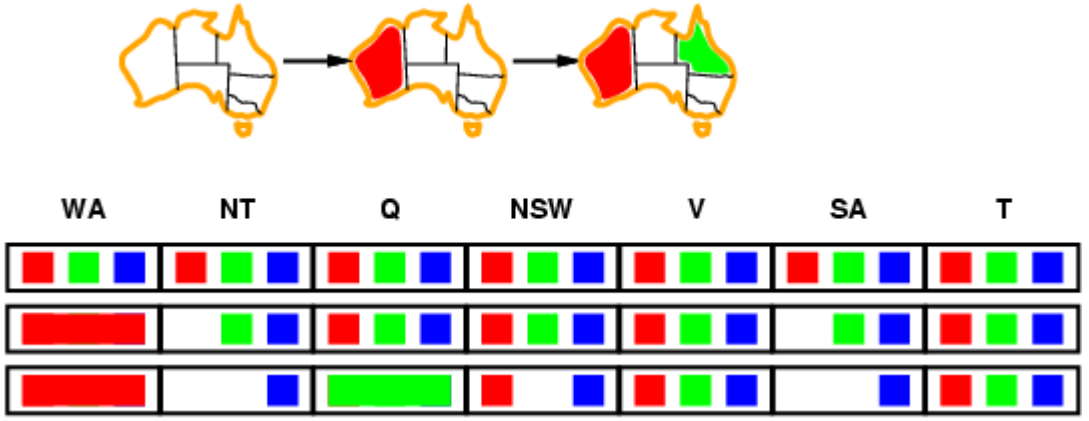
Forward checking

- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values.



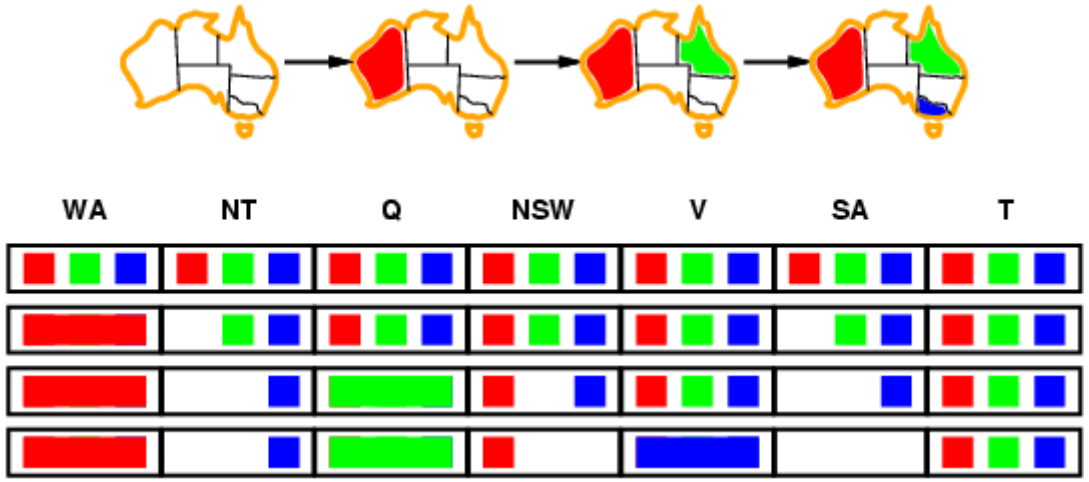
Forward checking

- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values



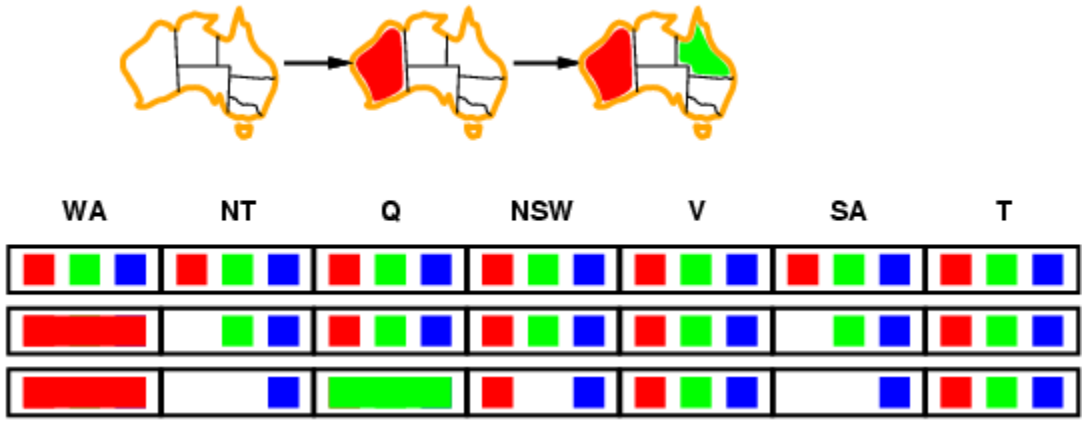
Forward checking

- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values



Constraint propagation

- Forward checking propagates information from assigned to unassigned variables, but doesn't provide early detection for all failures:



- NT and SA cannot both be blue!
- Constraint propagation repeatedly enforces constraints locally.


AI
THANKYOU