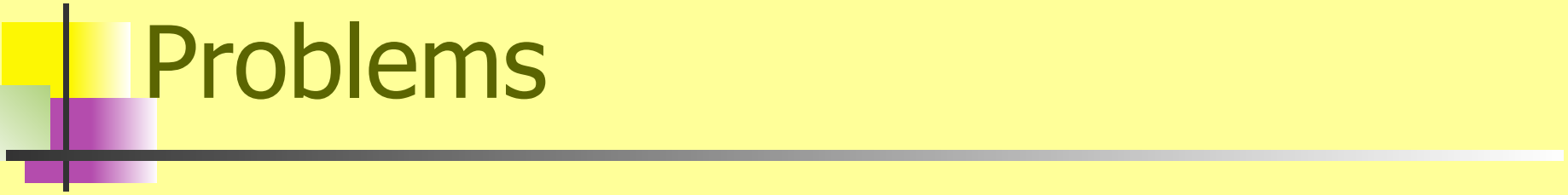
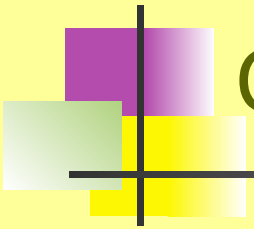


# Constraint Satisfaction Problems



# Constraint satisfaction problems (CSPs)

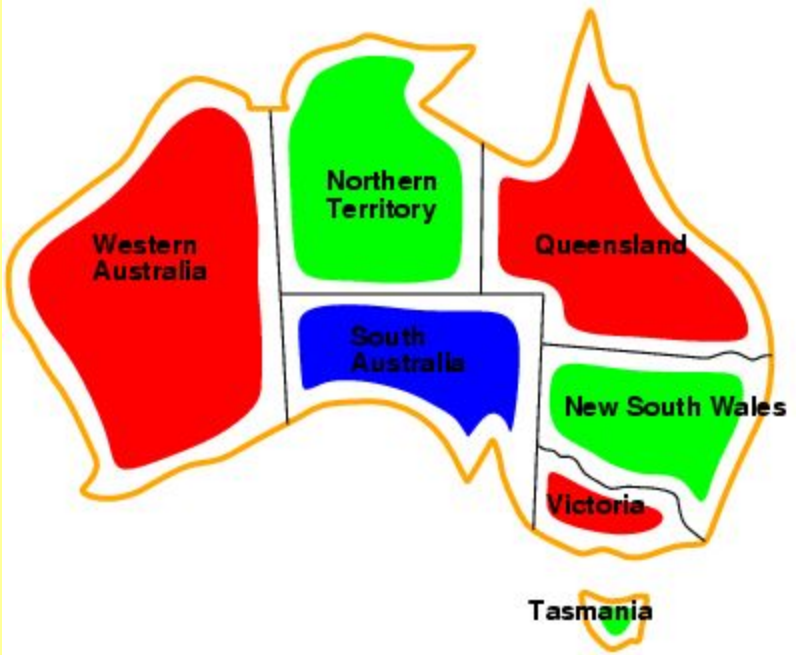
- CSP:
  - state is defined by variables $X_i$ with values from domain $D_i$
  - goal test is a set of constraints specifying allowable combinations of values for subsets of variables

- Allows useful general-purpose algorithms with more power than standard search algorithms

# 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

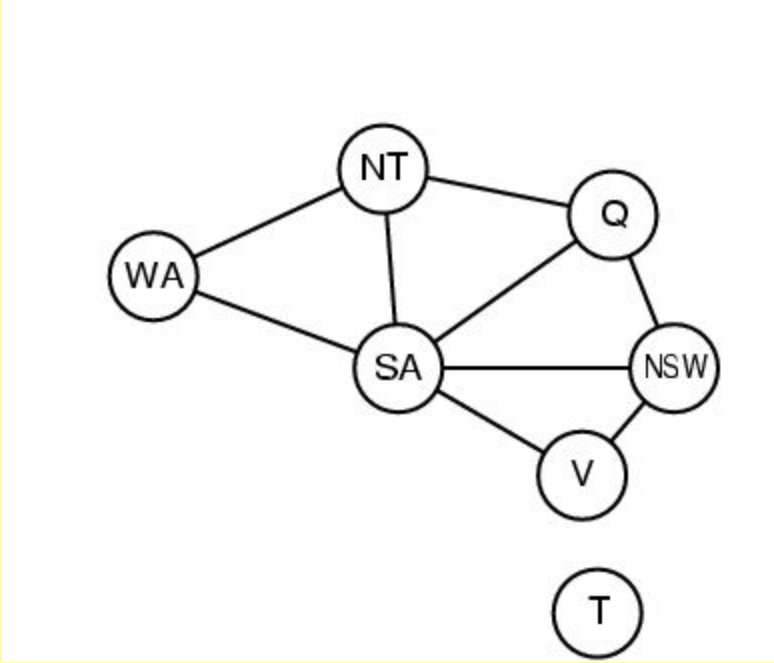
# Example: Map-Coloring




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

# Constraint graph

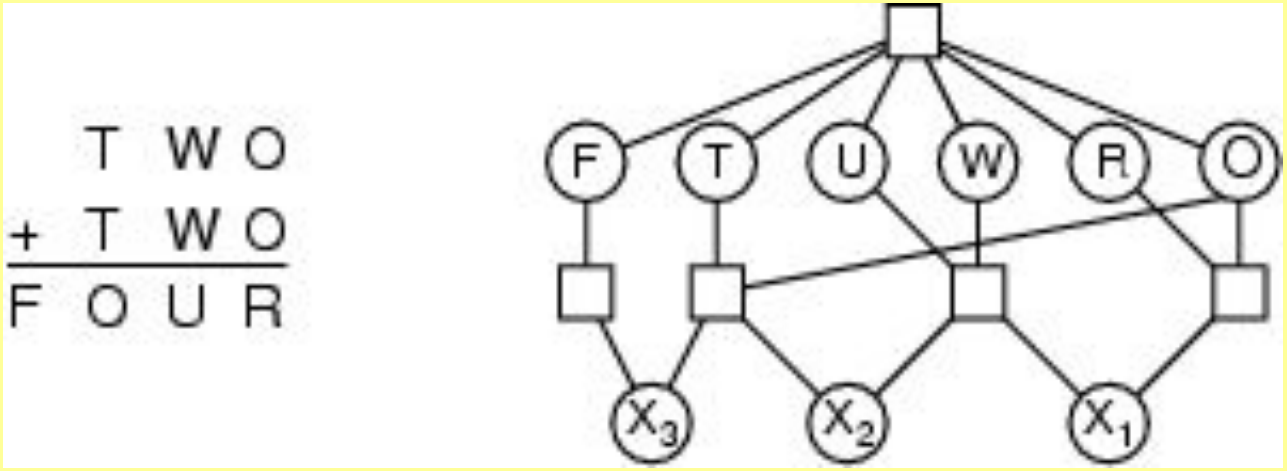
- Binary CSP: each constraint relates two variables
- Constraint graph: nodes are variables, arcs are constraints

# 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., SA ≠ WA ≠ NT

# Example: Cryptarithmetic



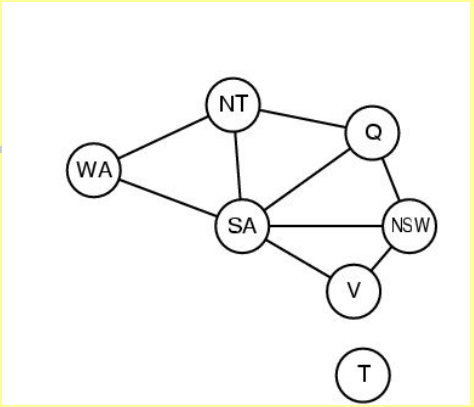

- Variables: *F T U W R O* $X_1\ X_2\ X_3$
- Domains: {*0,1,2,3,4,5,6,7,8,9*} {0,1}
- Constraints: *Alldiff (F,T,U,W,R,O)*
  - $O + O = R + 10 \cdot X_1$
  - $X_1 + W + W = U + 10 \cdot X_2$
  - $X_2 + T + T = O + 10 \cdot X_3$
  - $X_3 = F,\ T \neq 0,\ F \neq 0$

# Real-world CSPs

- Assignment problems
  - e.g., who teaches what class
- Timetabling problems
  - e.g., which class is offered when and where?
- Transportation scheduling
- Factory scheduling

- Notice that many real-world problems involve real-valued variables

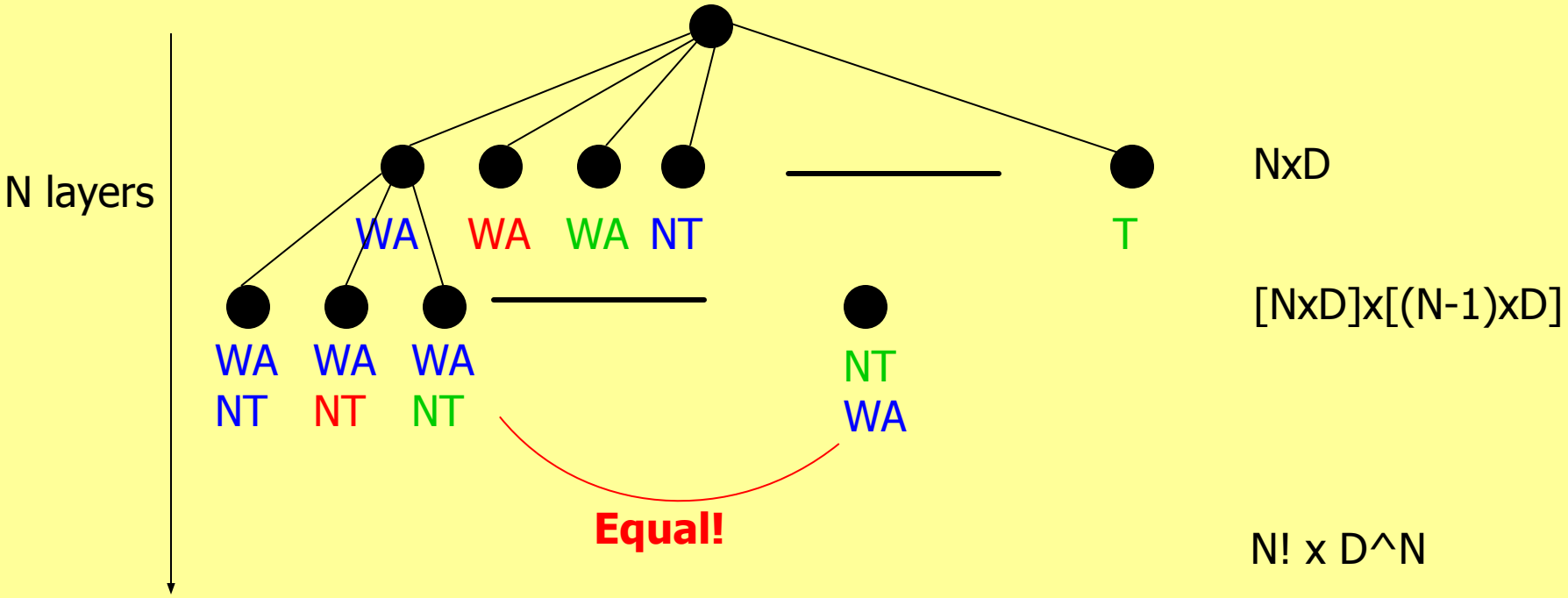
# Standard search formulation




Let's try the standard search formulation.

We need:

- Initial state: none of the variables has a value (color)
- Successor state: one of the variables without a value will get some value.
- Goal: all variables have a value and none of the constraints is violated.



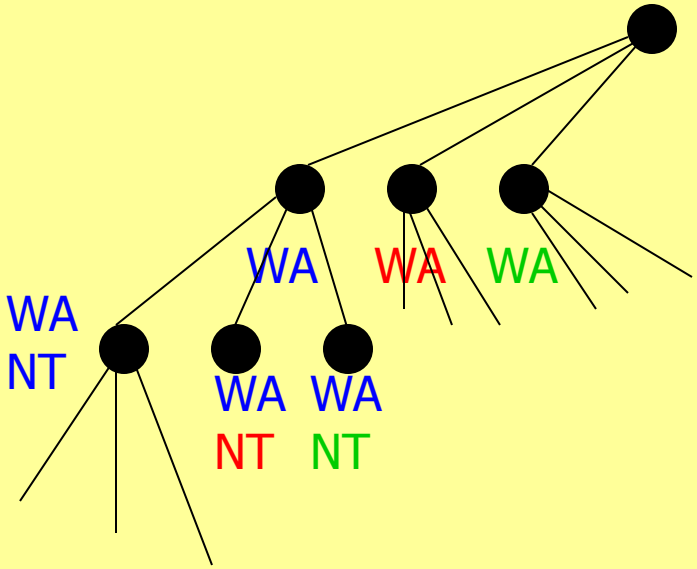

There are N! x D^N nodes in the tree but only D^N distinct states??

# Backtracking (Depth-First) search

- Special property of CSPs: They are commutative:
  This means: the order in which we assign variables does not matter.

NT WA = WA NT

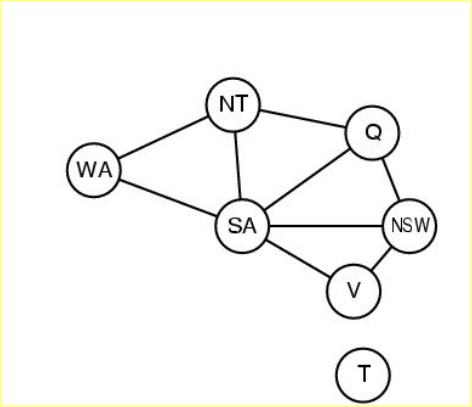
- Better search tree: First order variables, then assign them values one-by-one.




D

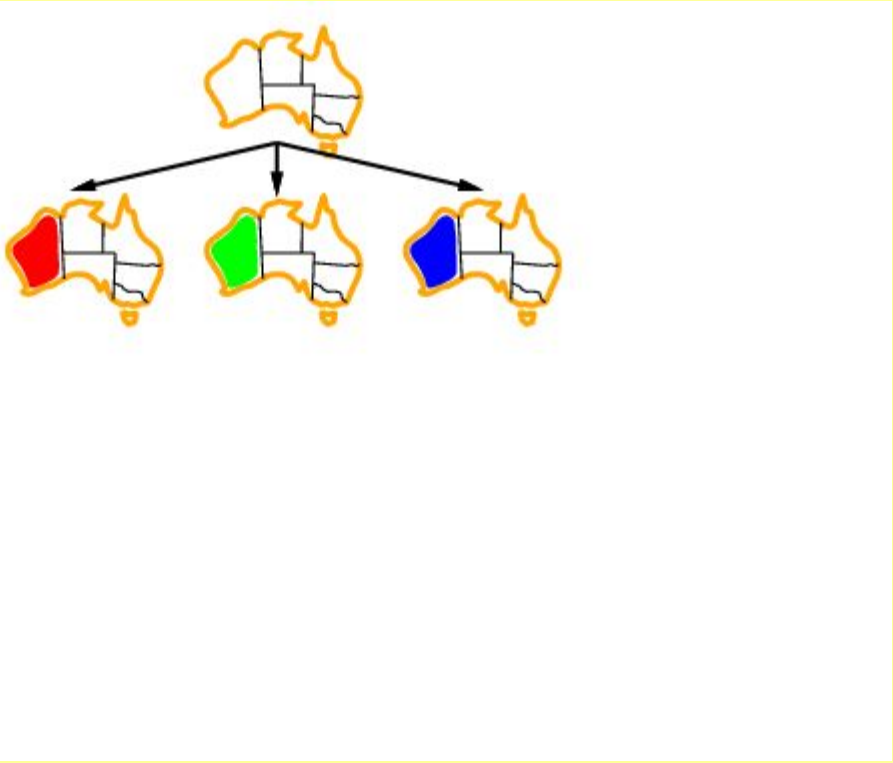
D^2

D^N

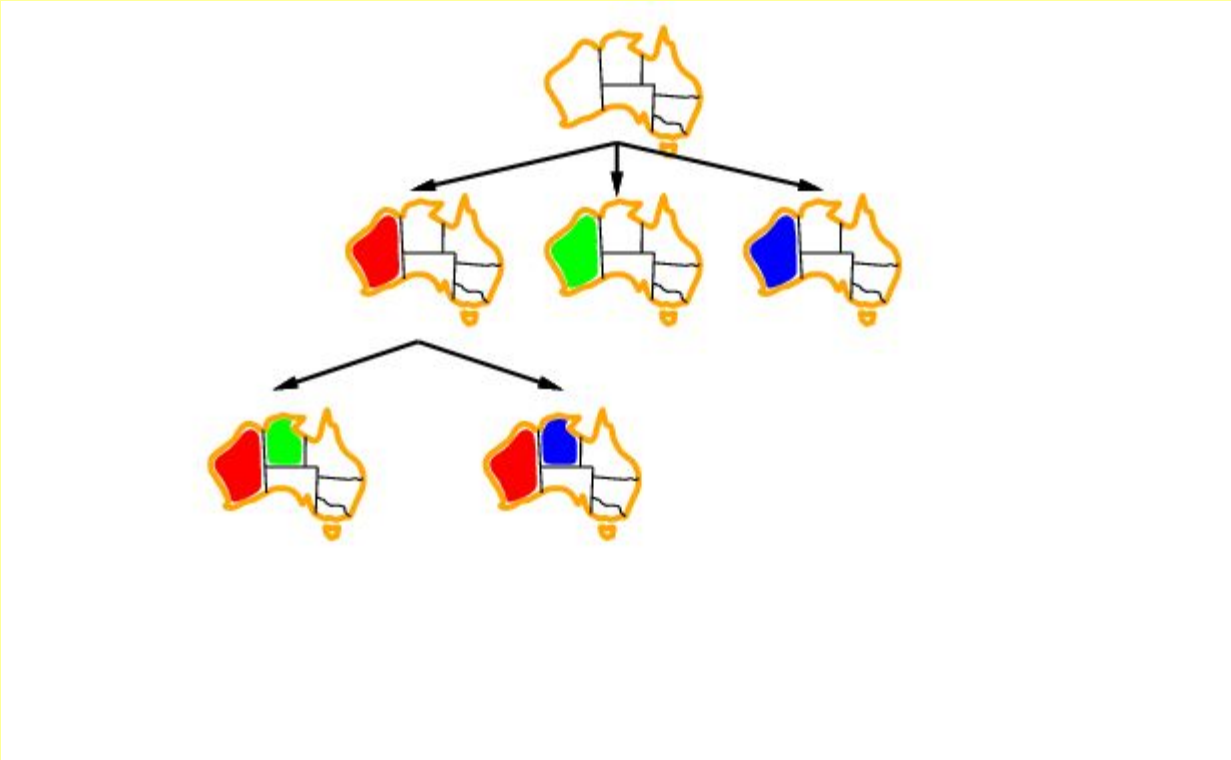


# Backtracking example

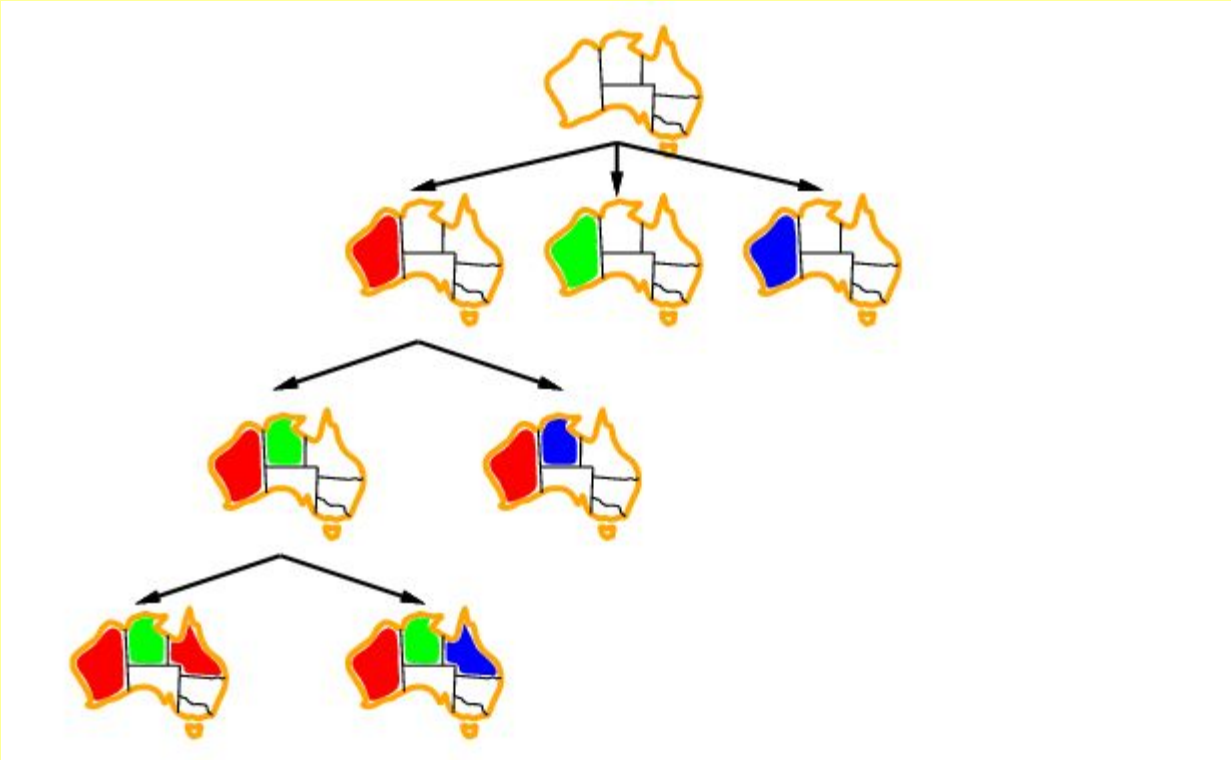
# Backtracking example



# Backtracking example



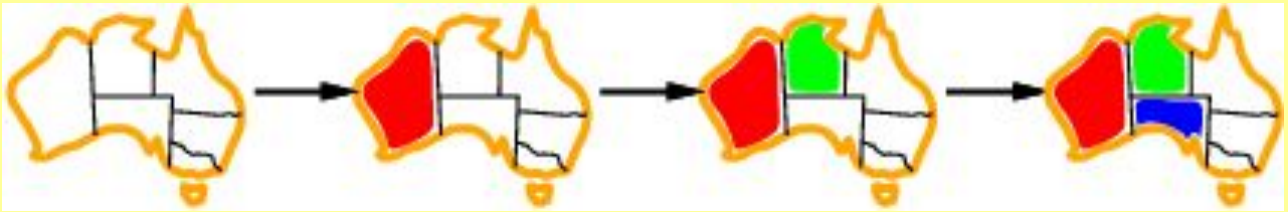
# Backtracking example



# Most constrained variable

- Most constrained variable:

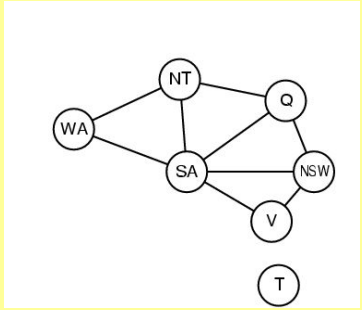
  choose the variable with the fewest legal values



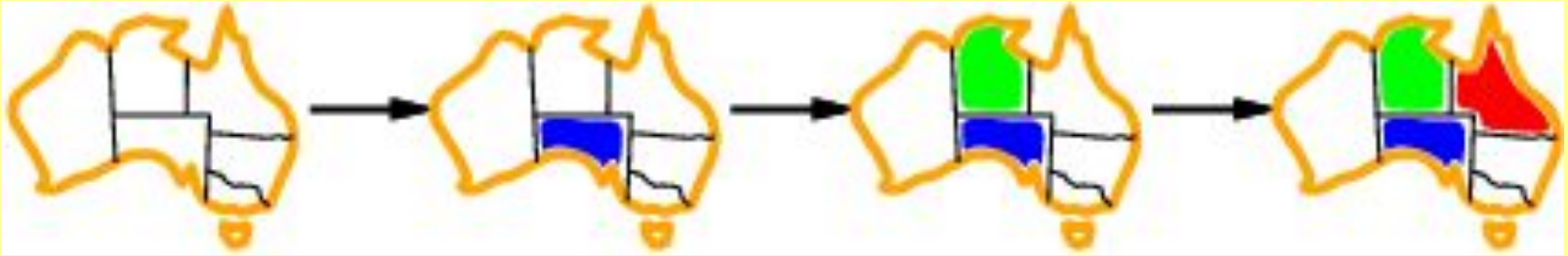
- a.k.a. minimum remaining values (MRV) heuristic
- Picks a variable which will cause failure as soon as possible, allowing the tree to be pruned.

# Most constraining variable

- Tie-breaker among most constrained variables

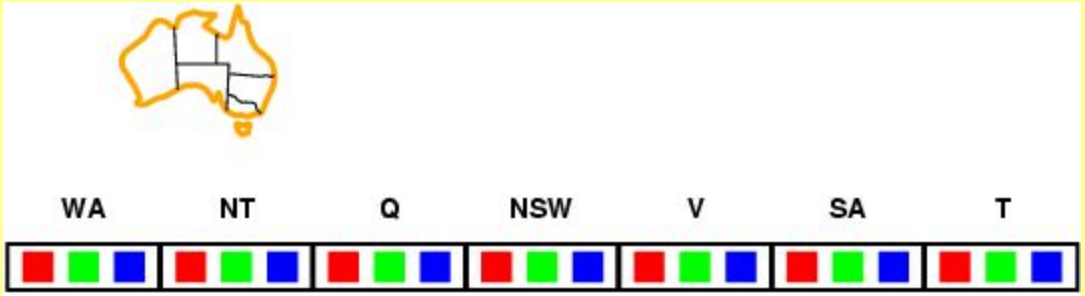


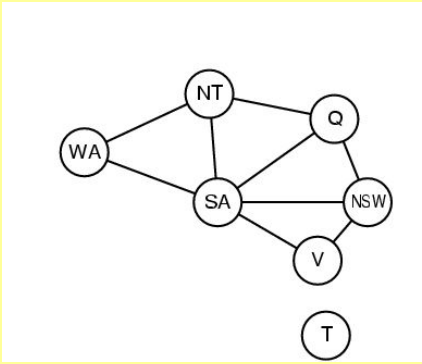

- Most constraining variable:
  - choose the variable with the most constraints on remaining variables (most edges in graph)



# Forward checking

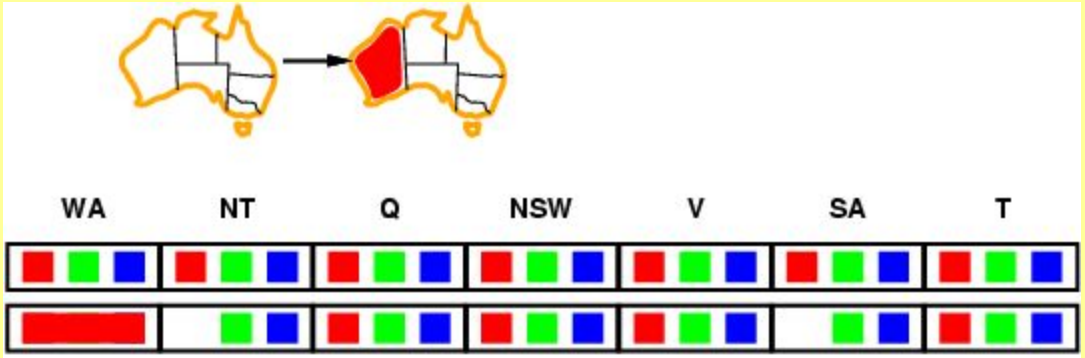
- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values

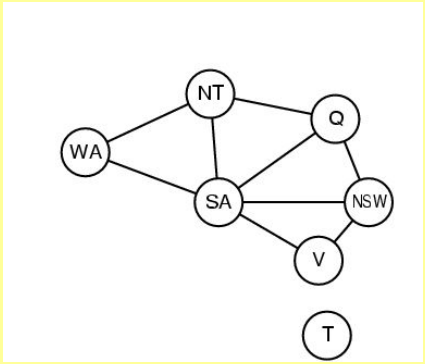


# Forward checking

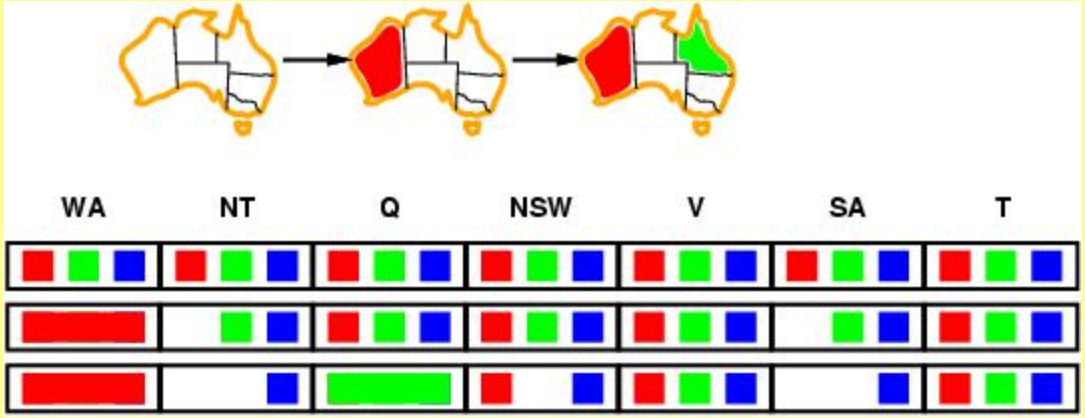
- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values

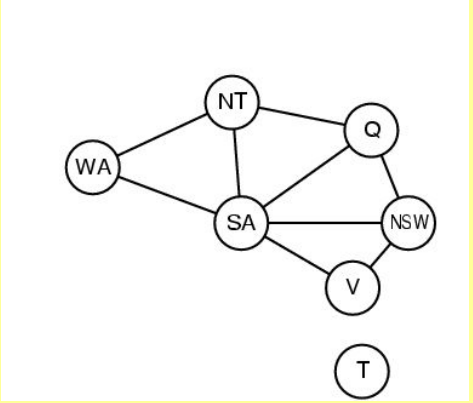


# Forward checking

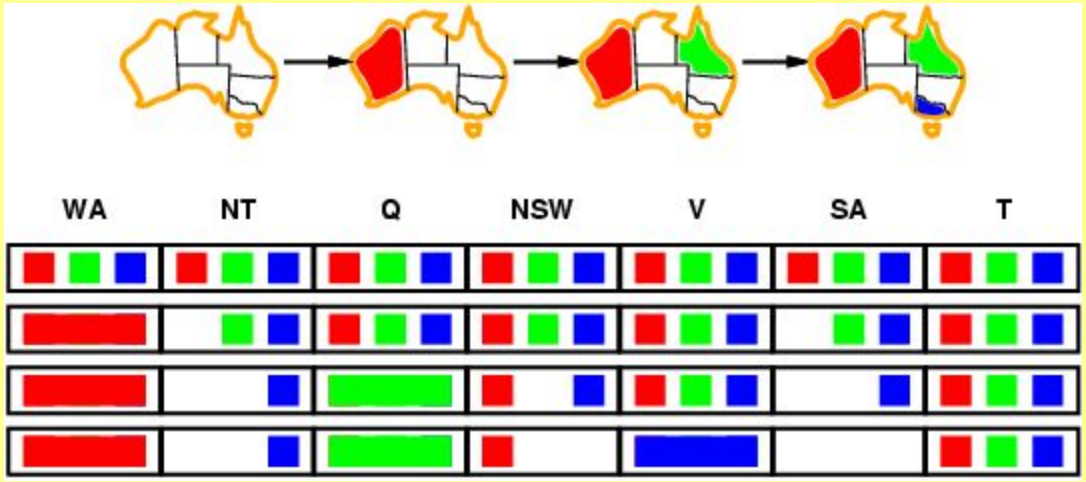
- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values

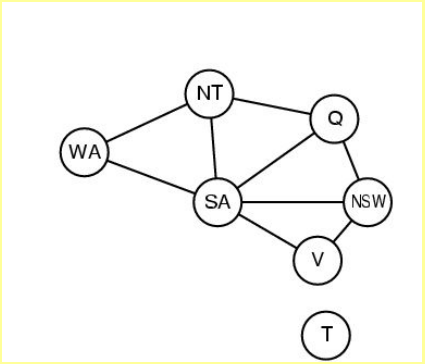


# Forward checking

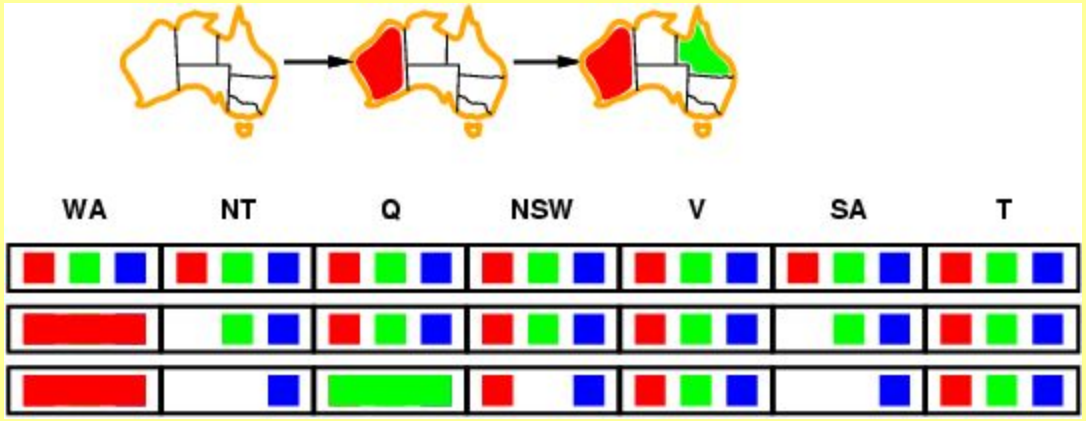
- Idea:
  - Keep track of remaining legal values for unassigned variables
  - Terminate search when any variable has no legal values

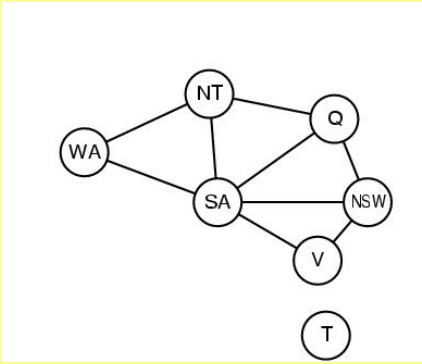


# Constraint propagation

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

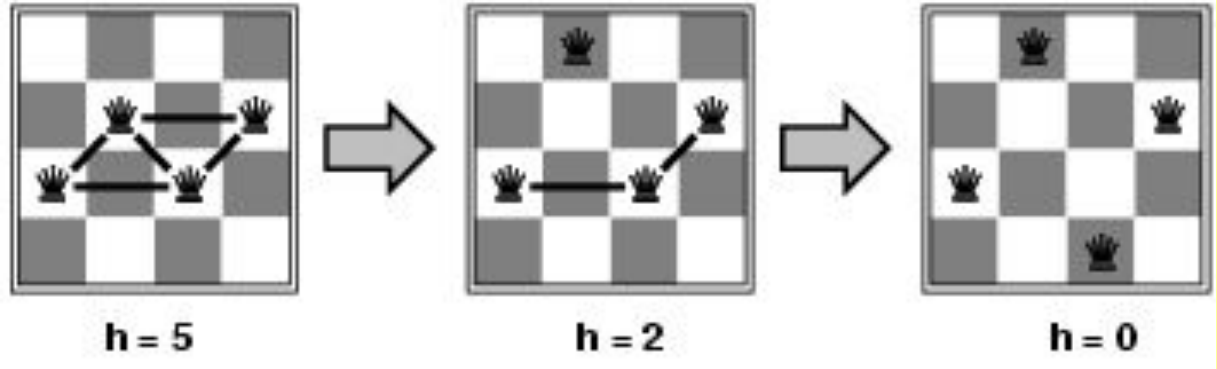


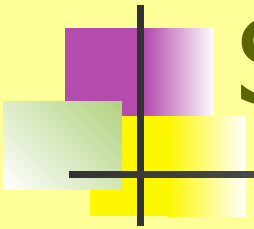




- NT and SA cannot both be blue!
- Constraint propagation repeatedly enforces constraints locally

# Example: 4-Queens

- States: 4 queens in 4 columns ($4^4$ = 256 states)
- Actions: move queen in column
- Goal test: no attacks
- Evaluation: *h(n)* = number of attacks





# Summary

- CSPs are a special kind of problem:
  - states defined by values of a fixed set of variables
  - goal test defined by constraints on variable values
- Backtracking = depth-first search with one variable assigned per node
- Variable ordering and value selection heuristics help significantly
- Forward checking prevents assignments that guarantee later failure