Constraint Satisfaction Problems

Dr.SangeethaYalamanchili
Associate Professor
Department of IT
VRSEC

- Constraint satisfaction problems (CSPs) are mathematical problems defined as a set of objects whose state must satisfy a number of constraints or limitations.

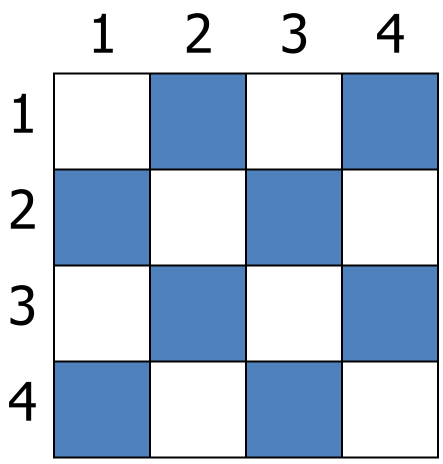
Constraint satisfaction problems

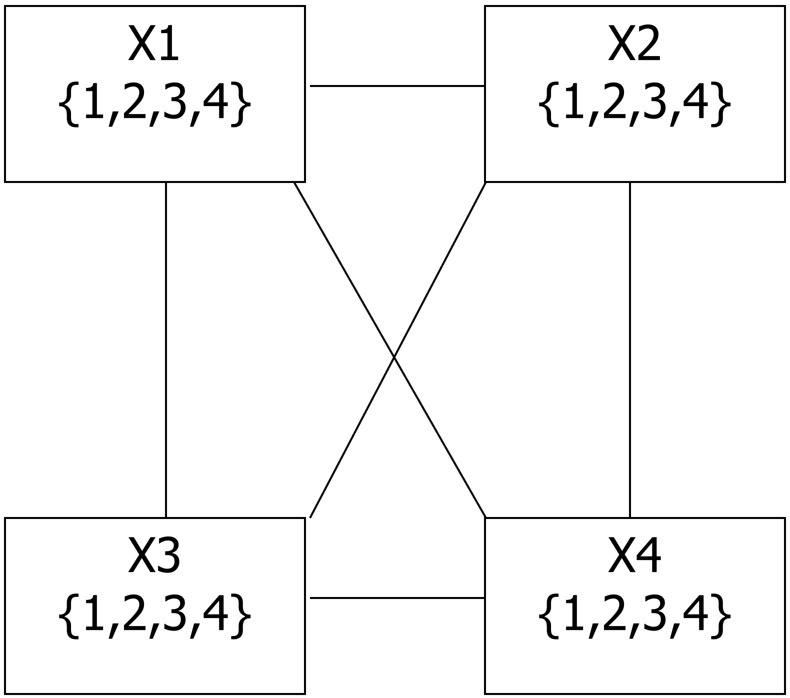
- What is a CSP?
  - Finite set of variables $X_1, X_2, \ldots, X_n$
    - Where each variable X_i is a Nonempty domain of possible values $D_1, D_2, \ldots D_d$
  - Finite set of constraints $C_1, C_2, \ldots, C_m$
    - Each constraint C_i limits the values that variables can take, e.g., $X_1 \neq X_2$
- A state is defined as an assignment of values to some or all variables.
- Consistent assignment: assignment does not violate the constraints.

Constraint satisfaction problems

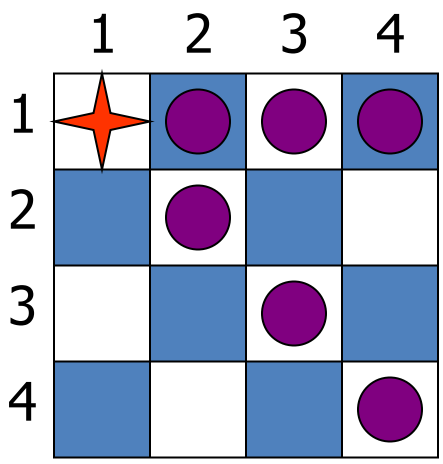
- An assignment is complete when every variable is assigned a value.
- A solution to a CSP is a complete assignment that satisfies all constraints.
- Applications:
    - Eight queens puzzle
    - Map coloring problem
    - Sudoku etc…..

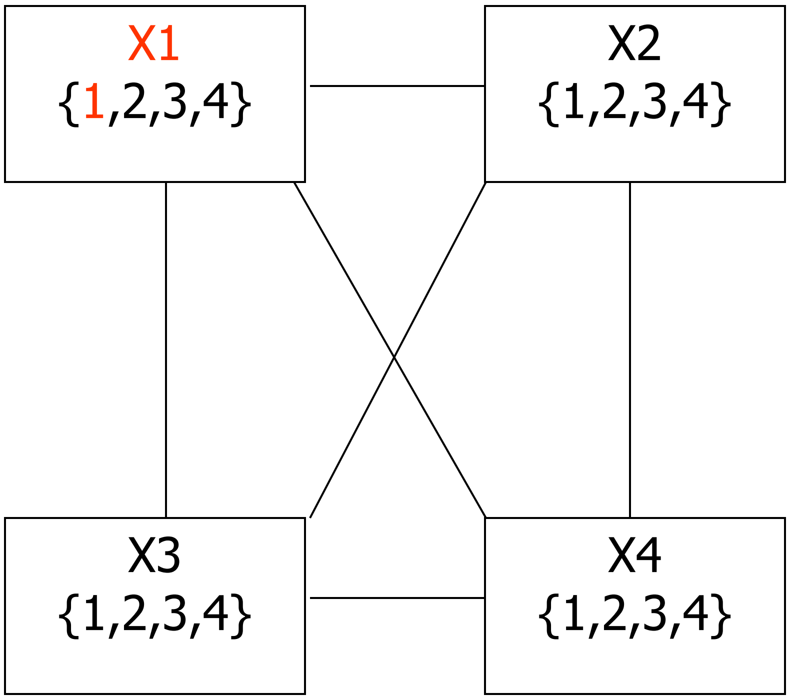
Example: 4-Queens Problem



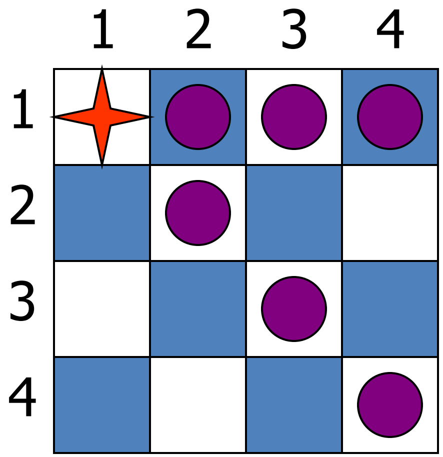


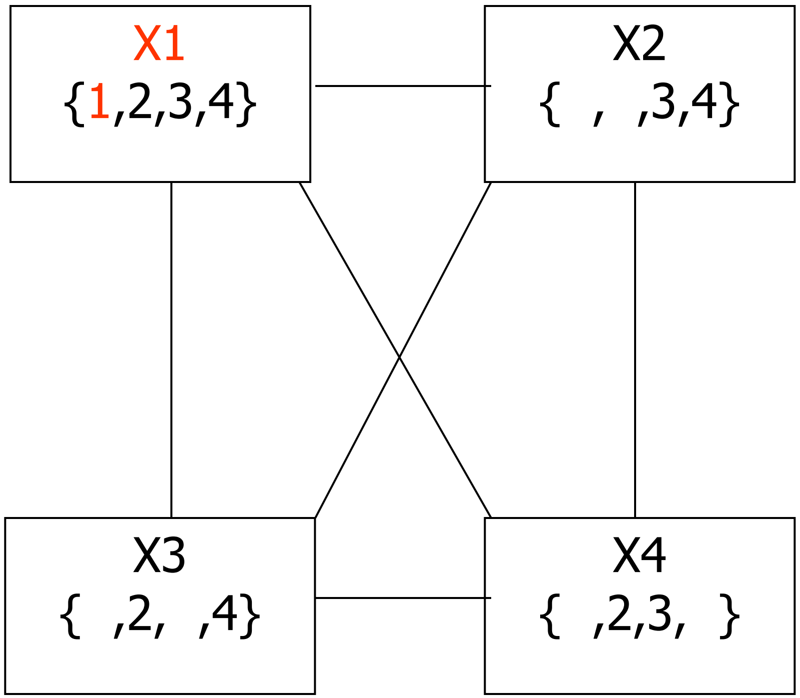
Example: 4-Queens Problem



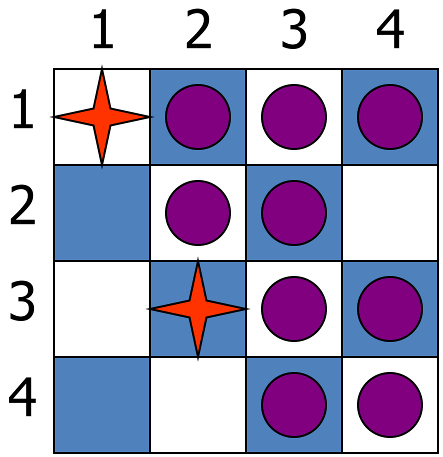


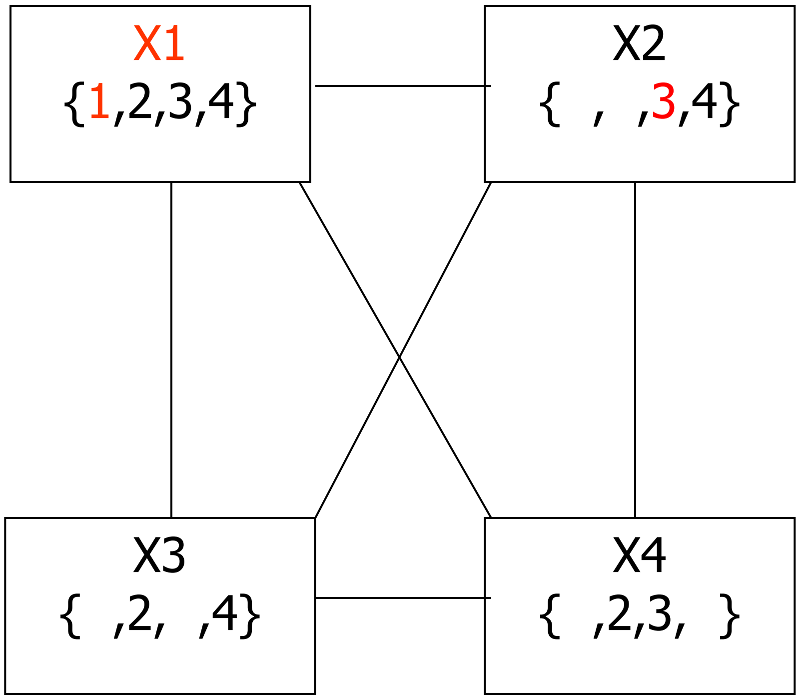
Example: 4-Queens Problem



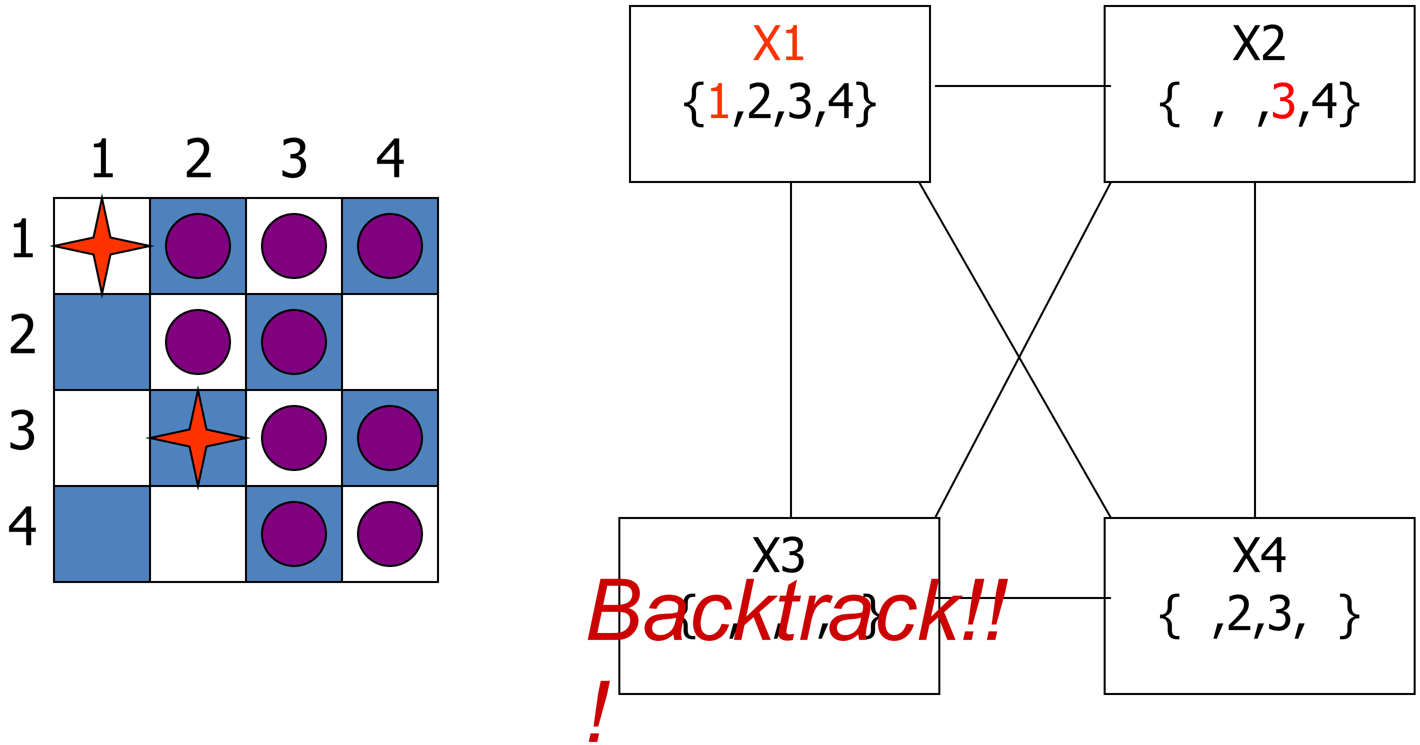


Example: 4-Queens Problem

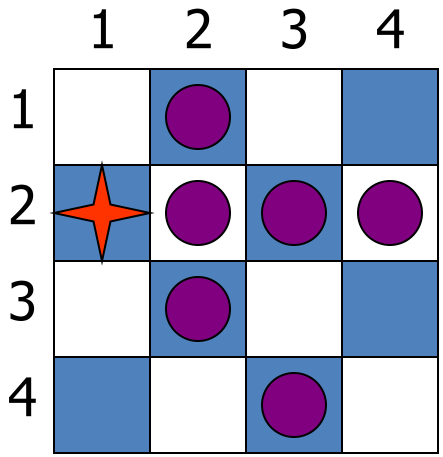


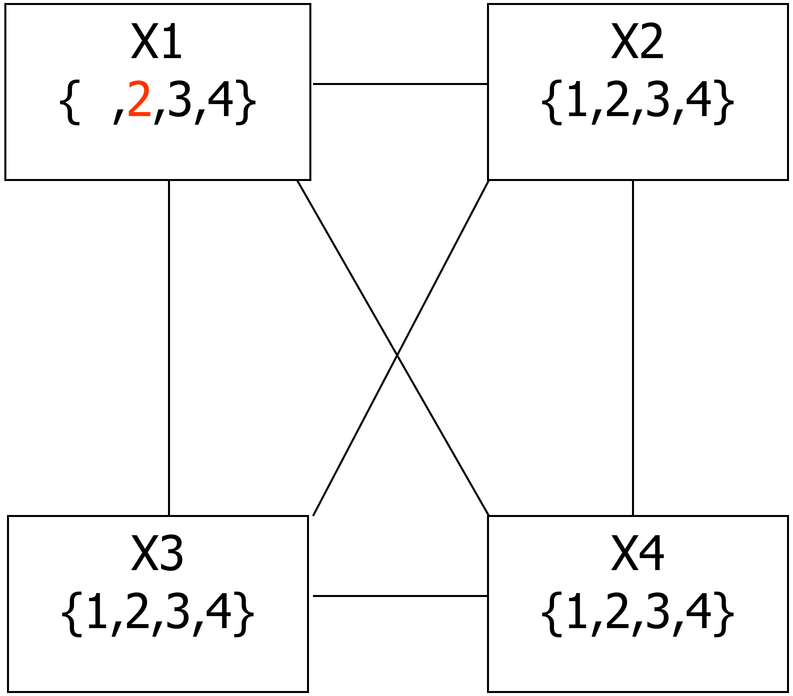


Example: 4-Queens Problem

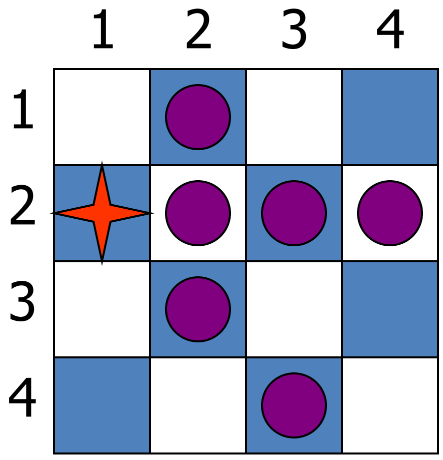


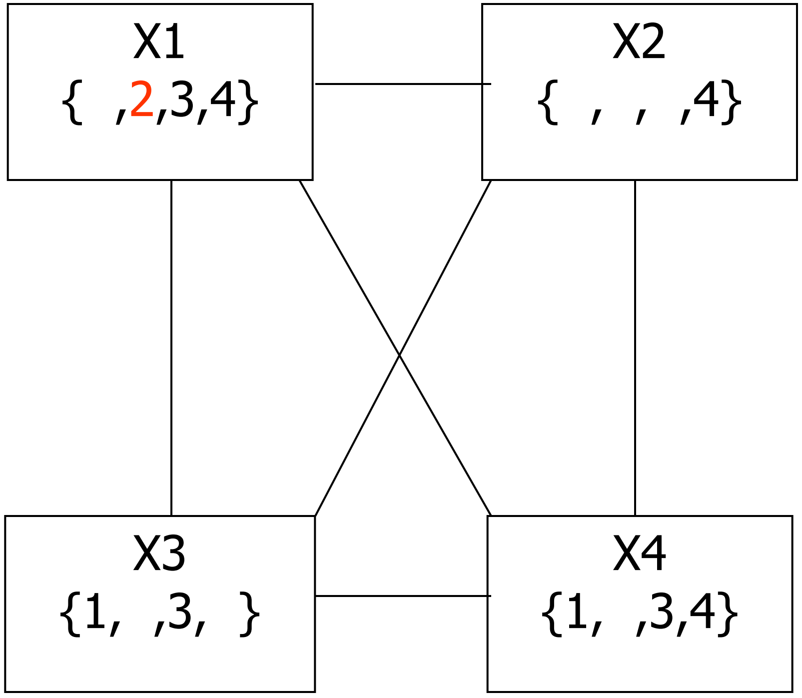
Example: 4-Queens Problem



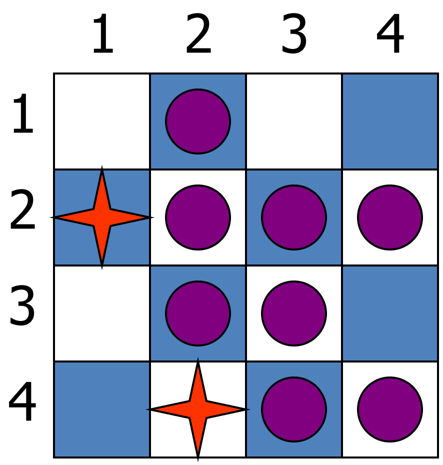


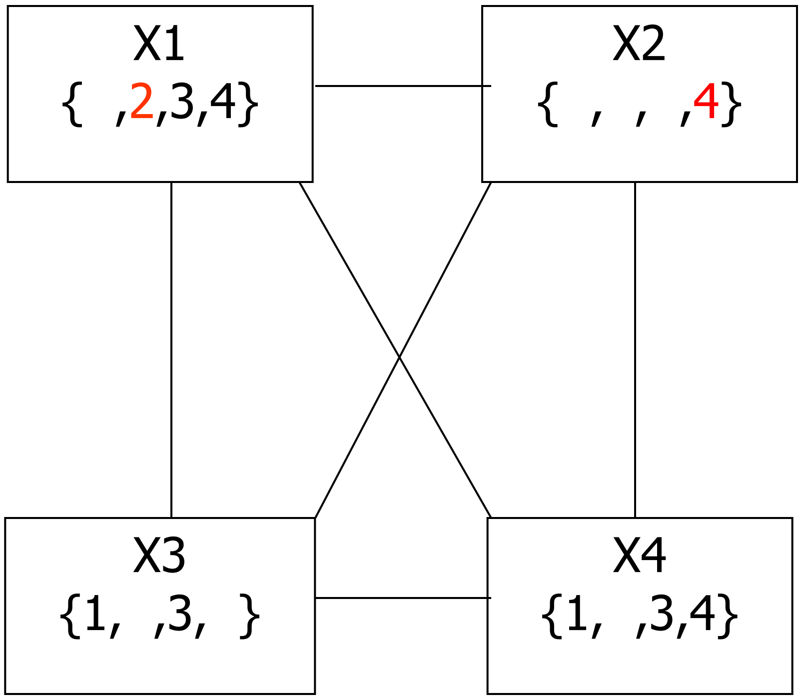
Example: 4-Queens Problem



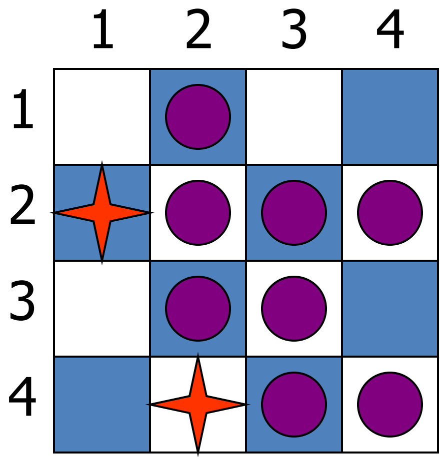


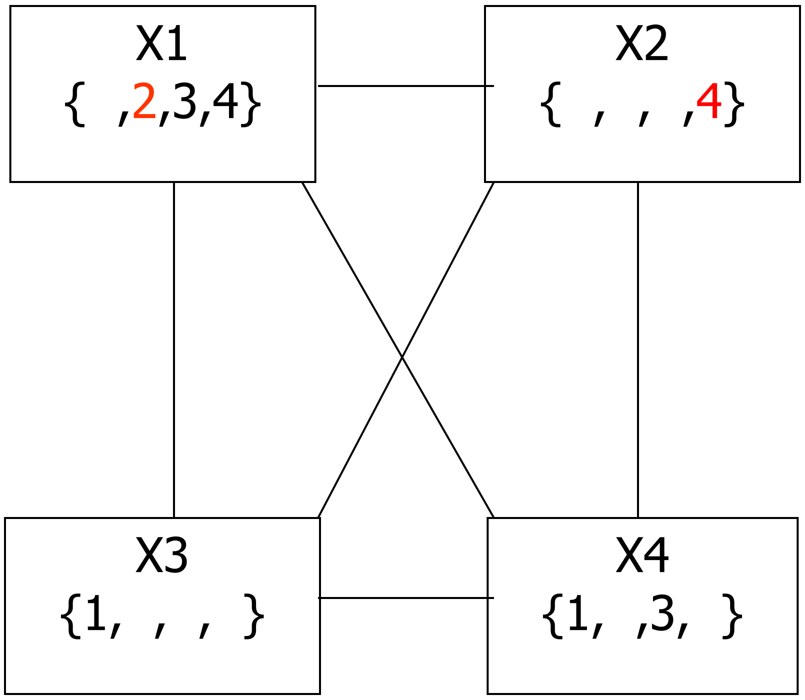
Example: 4-Queens Problem



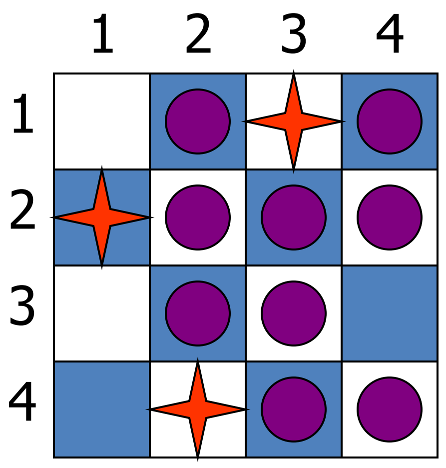


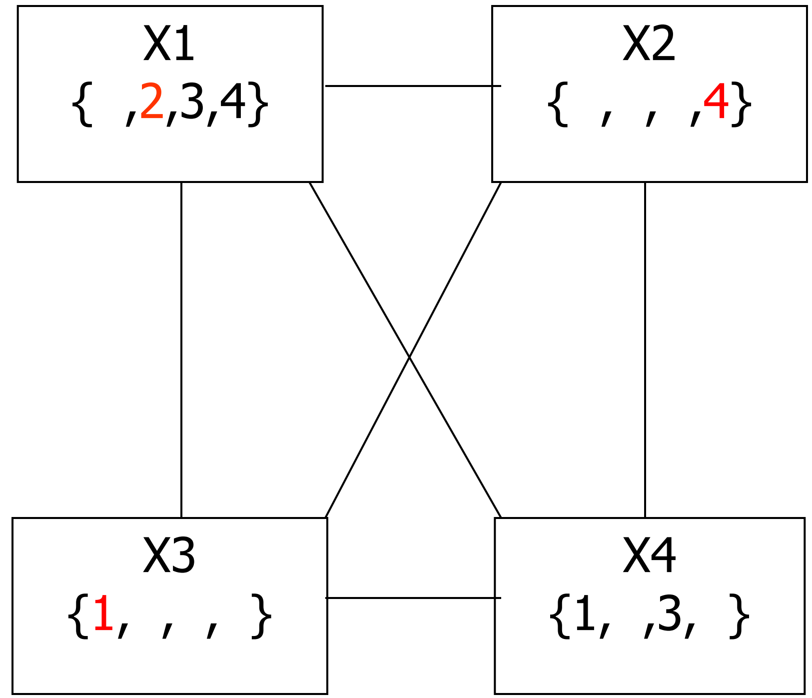
Example: 4-Queens Problem



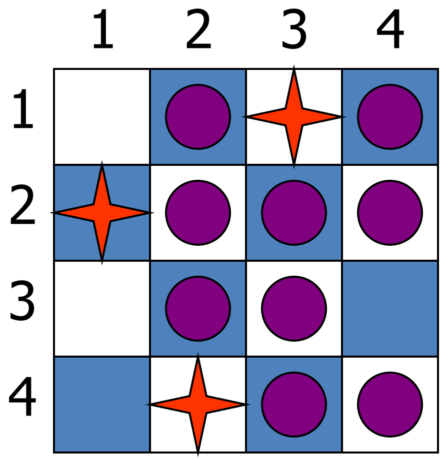


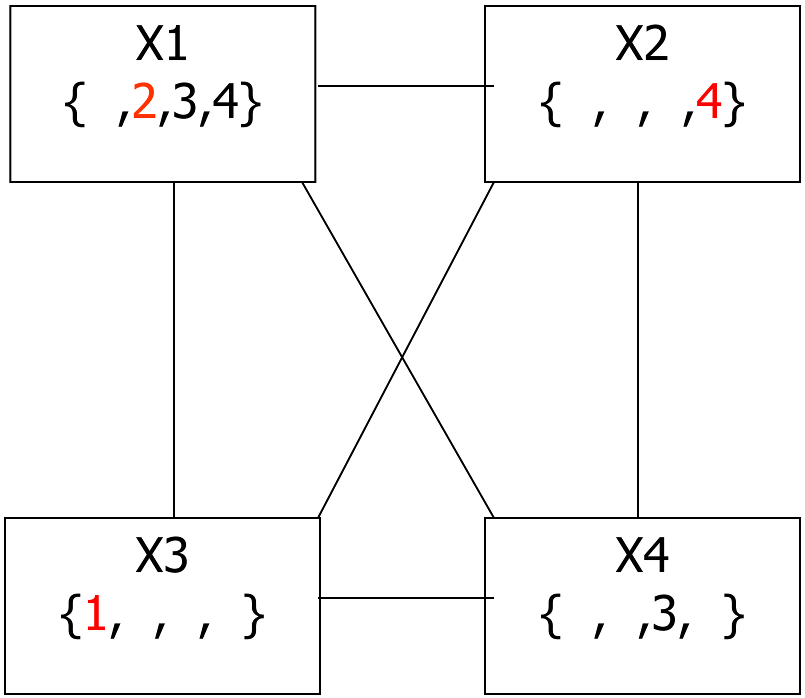
Example: 4-Queens Problem



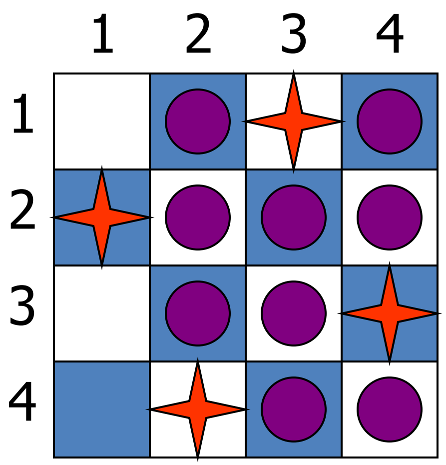


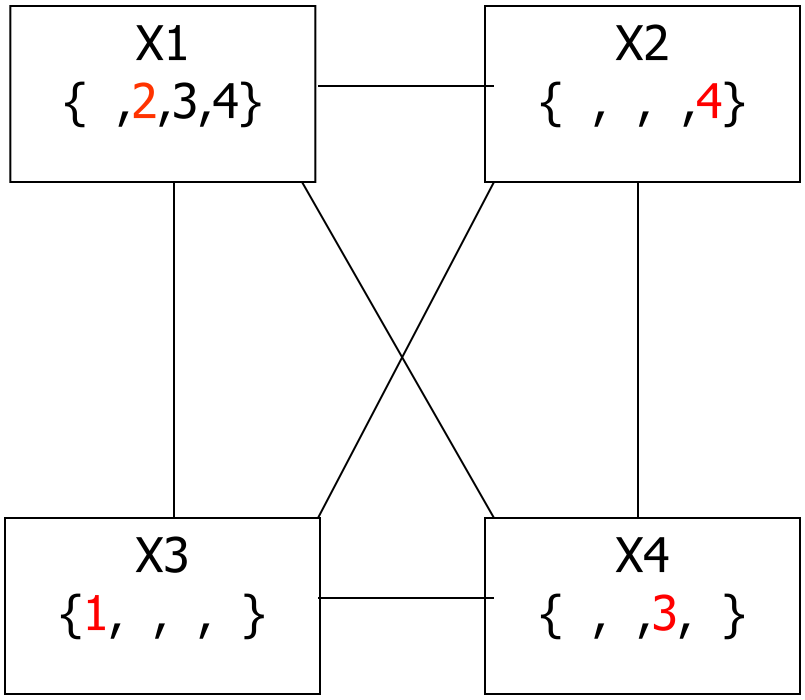
Example: 4-Queens Problem





Example: 4-Queens Problem





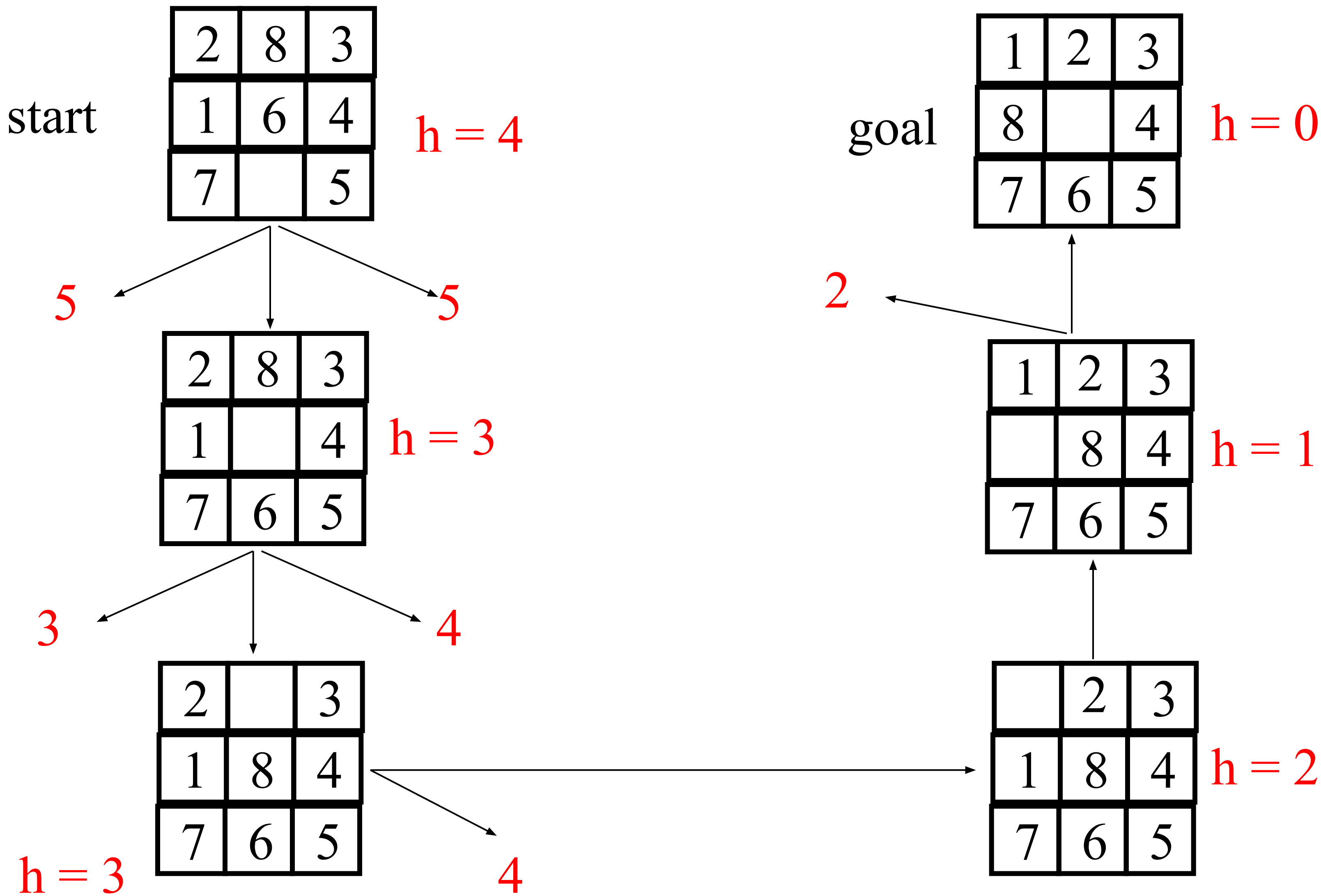
The Silver Bullet?

Is there an “intelligence algorithm”?

1957 GPS (General Problem Solver)

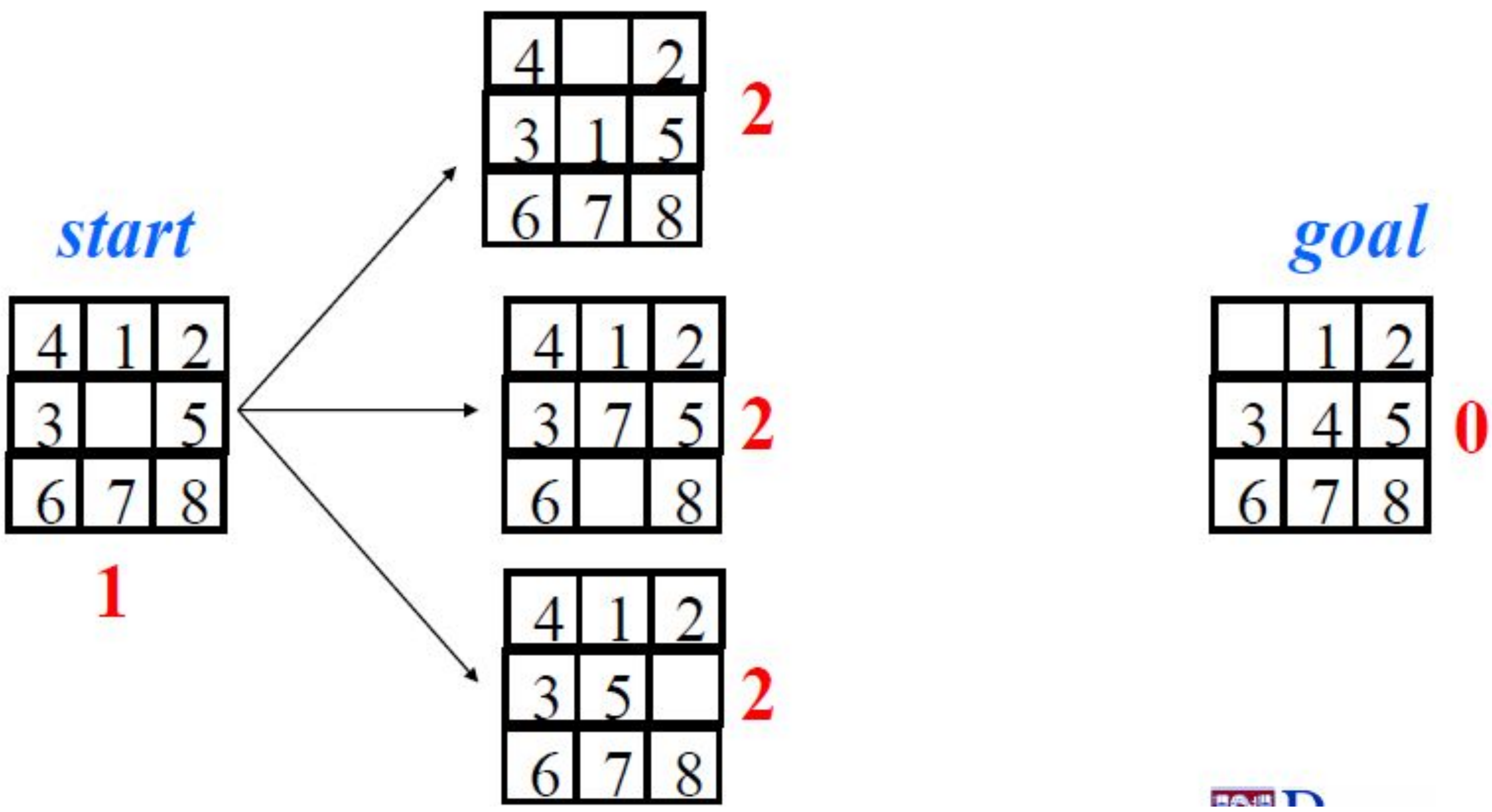
Start Goal

8-Puzzle example



f(n) = (number of tiles out of place)

Toy Example of a local "maximum"



Varieties of CSPs

Discrete variables

finite domains:

n variables, domain size d □ $O(d^n)$ complete assignments

e.g., Boolean CSPs, includes Boolean satisfiability (NP-complete) ***Line Drawing Interpretation***

infinite domains:

integers, strings, etc.

e.g., job scheduling, variables are start/end days for each job need a constraint language, e.g., $StartJob_1 + 5 \leq StartJob_3$

Continuous variables

e.g., start/end times for Hubble Space Telescope observations linear constraints solvable in polynomial time by linear programming

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., crypt-arithmetic column constraints

Preference (soft constraints) e.g. *red* is better than *green* can be represented by a cost for each variable assignment
Constrained optimization problems.

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

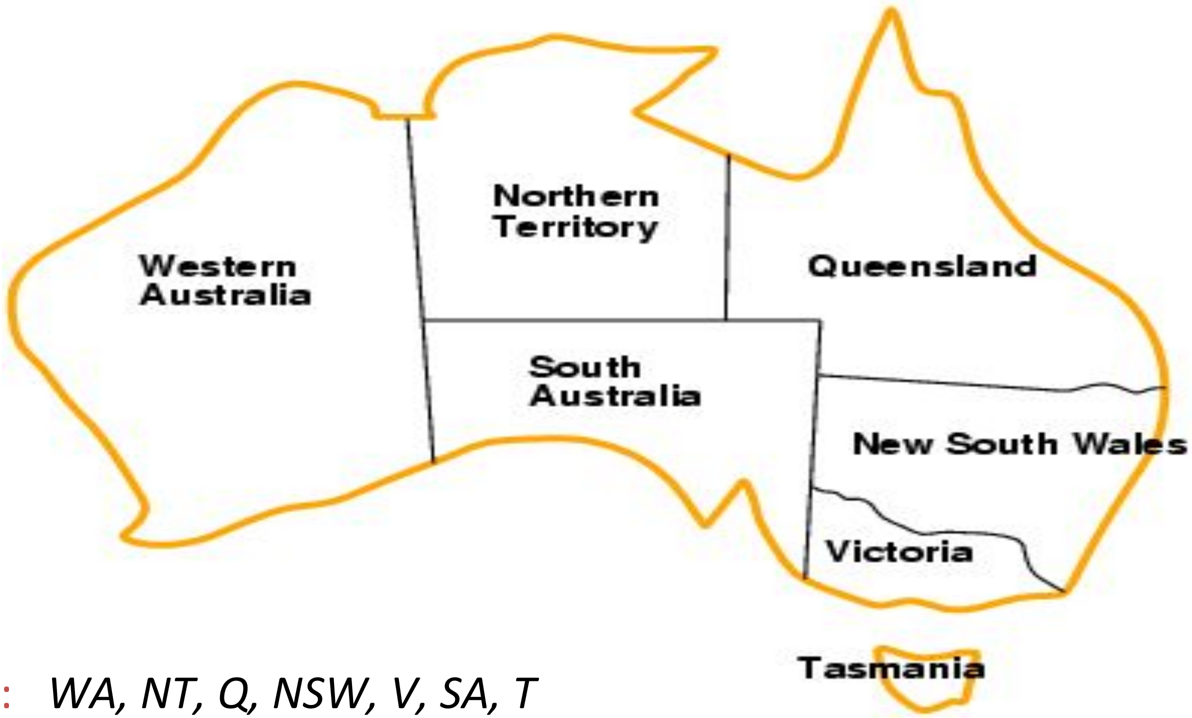
- Notice that many real-world problems involve real-valued variables

Comparison of Methods

- **Backtracking** tree search is a blind search.
- **Forward checking** checks constraints between the current variable and all future ones.
- **Arc consistency** then checks constraints between all pairs of future (unassigned) variables.

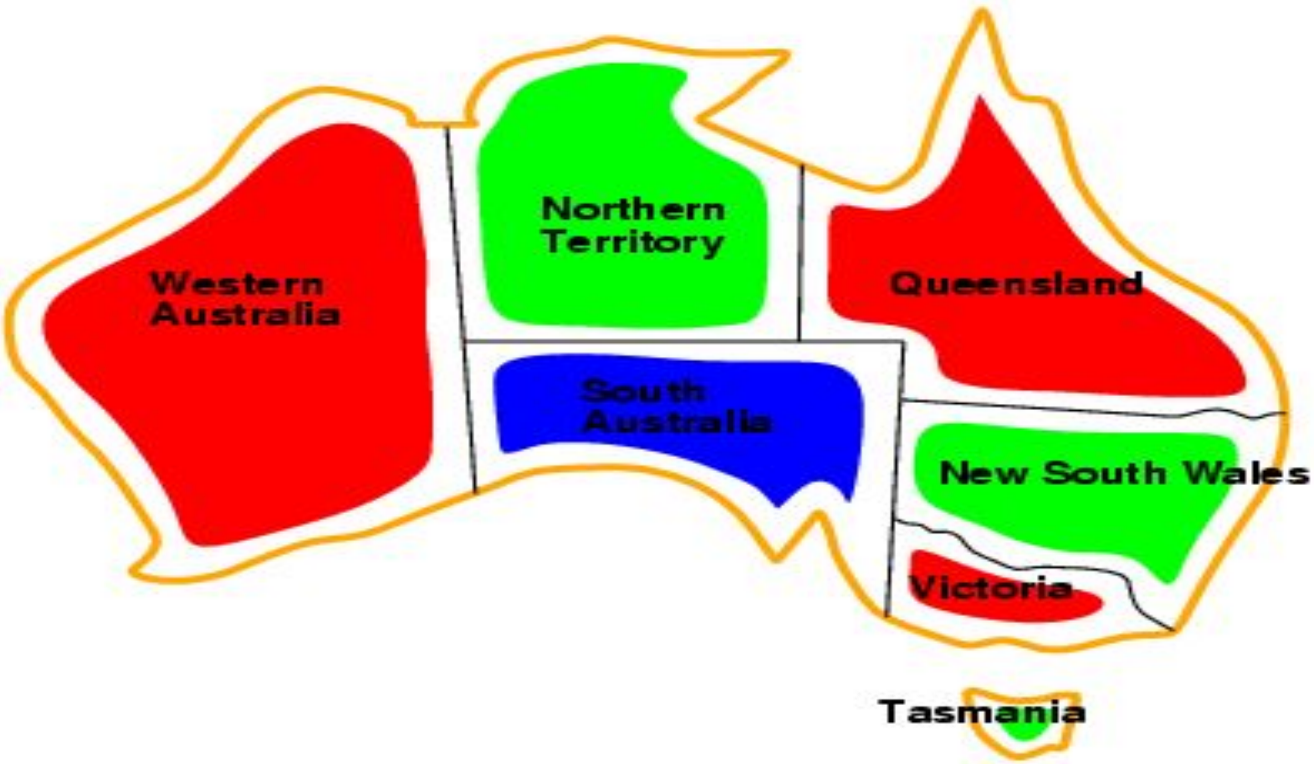
- What is the complexity of a backtracking tree search?
- How do forward checking and arc consistency affect that?

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 So (WA,NT) must be in {(red,green),(red,blue),(green,red), …}

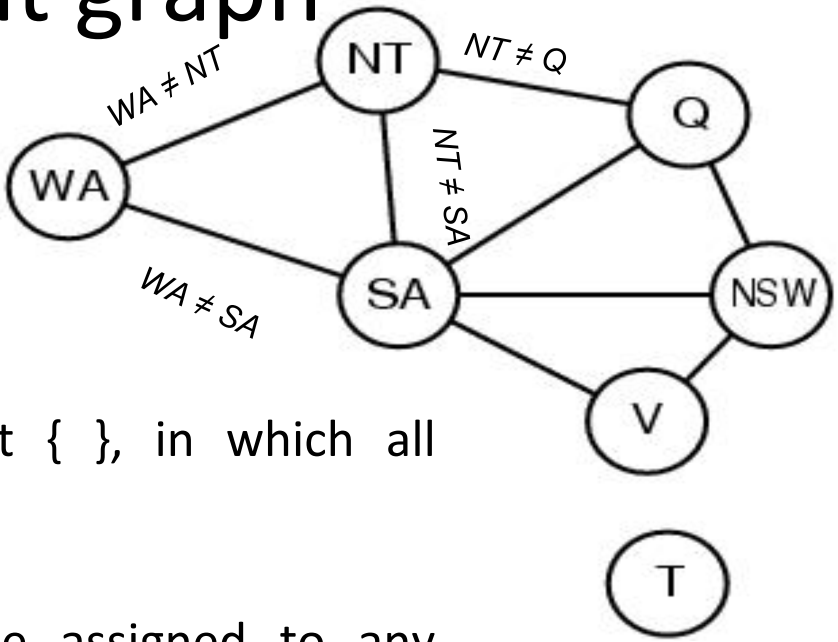
Example: Map-Coloring



Solutions are complete and consistent assignments,

– e.g., WA = red, NT = green,Q = red,NSW = green,
V = red,SA = blue,T = green

Constraint graph



- Constraint graph:
  - nodes are variables
  - arcs are constraints
- **Initial state:** the empty assignment { }, in which all variables are unassigned.
- **Successor function:** a value can be assigned to any unassigned variable, provided that it does not conflict with previously assigned variables.
- **Goal test:** the current assignment is complete.
- **Path cost:** a constant cost (e.g., 1) for every step.

Backtracking search for CSPs

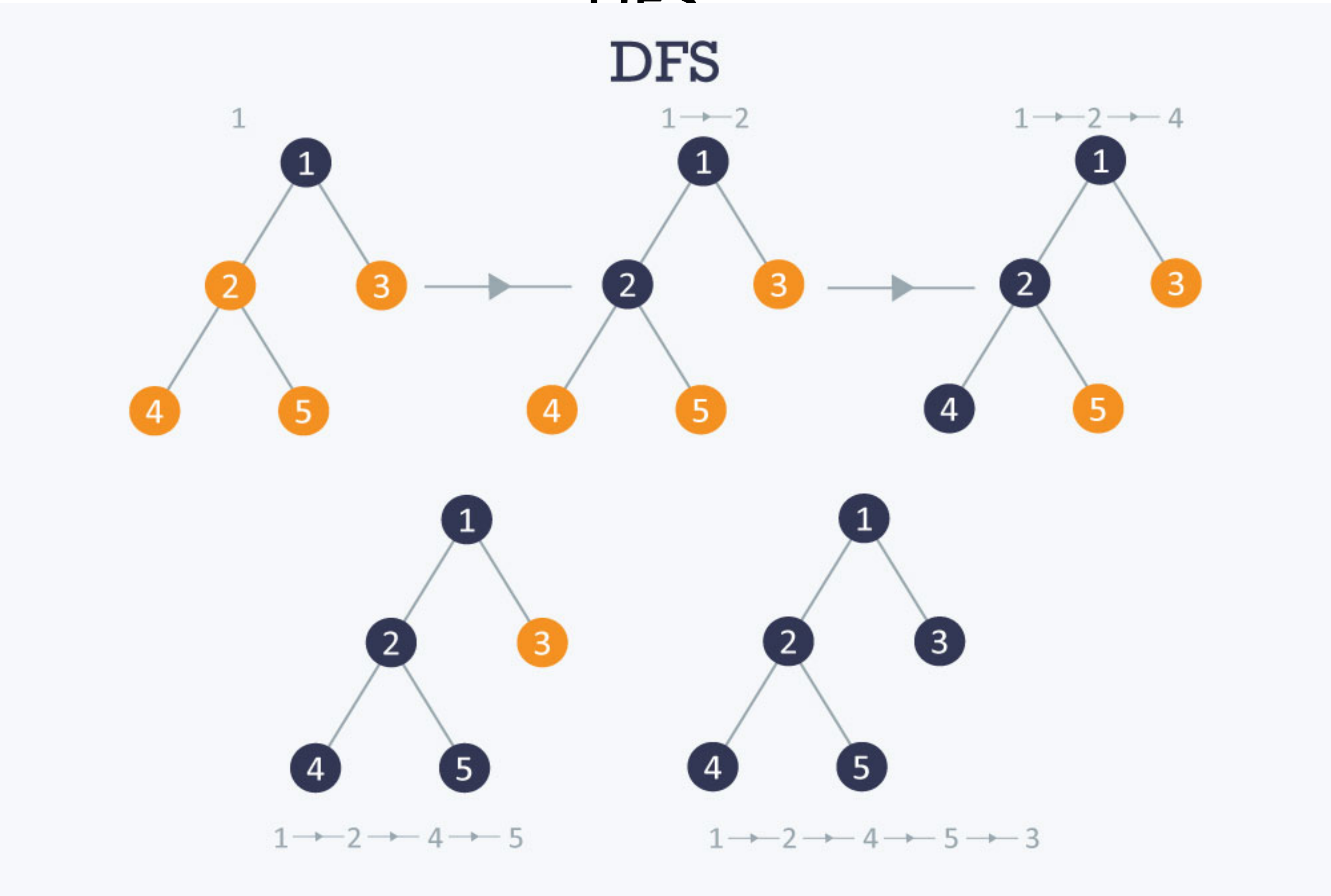
- Crucial property to all CSPs is Commutatively .(eg: AB=BA)
- The term backtracking search is used for a depth-first search that chooses values for BACKTRACKING SEARCH **one variable at a time** and backtracks when a variable has no legal values left to assign.
- The DFS algorithm is a recursive algorithm that uses the idea of backtracking. It involves exhaustive searches of all the nodes by going ahead, if possible, else by backtracking.

Backtracking search algorithm

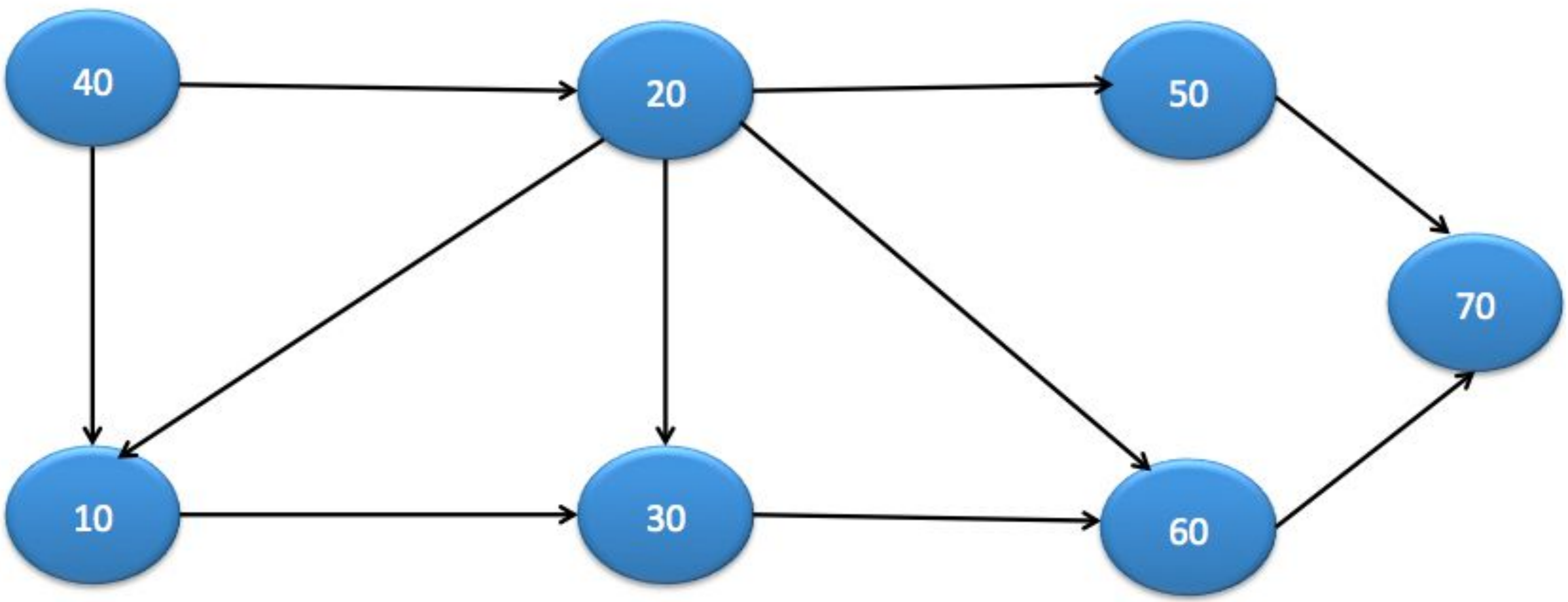
```
function BACKTRACKING-SEARCH(csp) returns a solution, or failure
   return RECURSIVE-BACKTRACKING({ }, csp)

function RECURSIVE-BACKTRACKING(assignment, csp) returns a solution, or failure
   if assignment is complete then return assignment
   var ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[csp], assignment, csp)
   for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
      if value is consistent with assignment according to CONSTRAINTS[csp] then
         add {var = value} to assignment
         result ← RECURSIVE-BACKTRACKING(assignment, csp)
         if result ≠ failure then return result
         remove {var = value} from assignment
   return failure
```

DFS



DFS

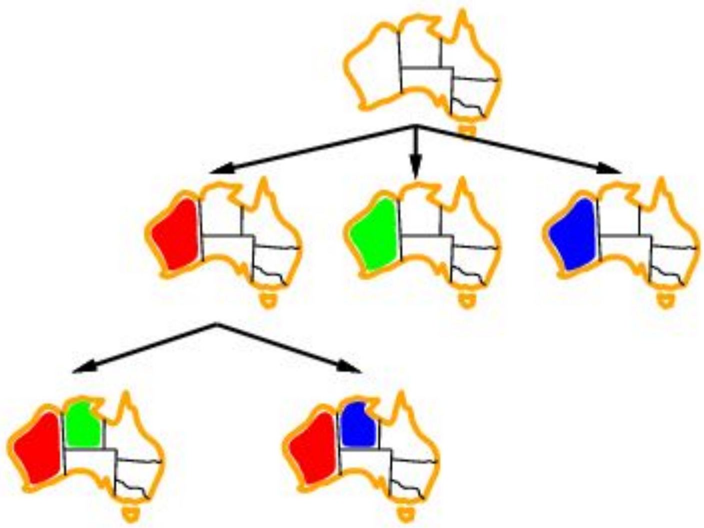


Depth first traversal of above graph can be :40,20,50,70,60,30,10

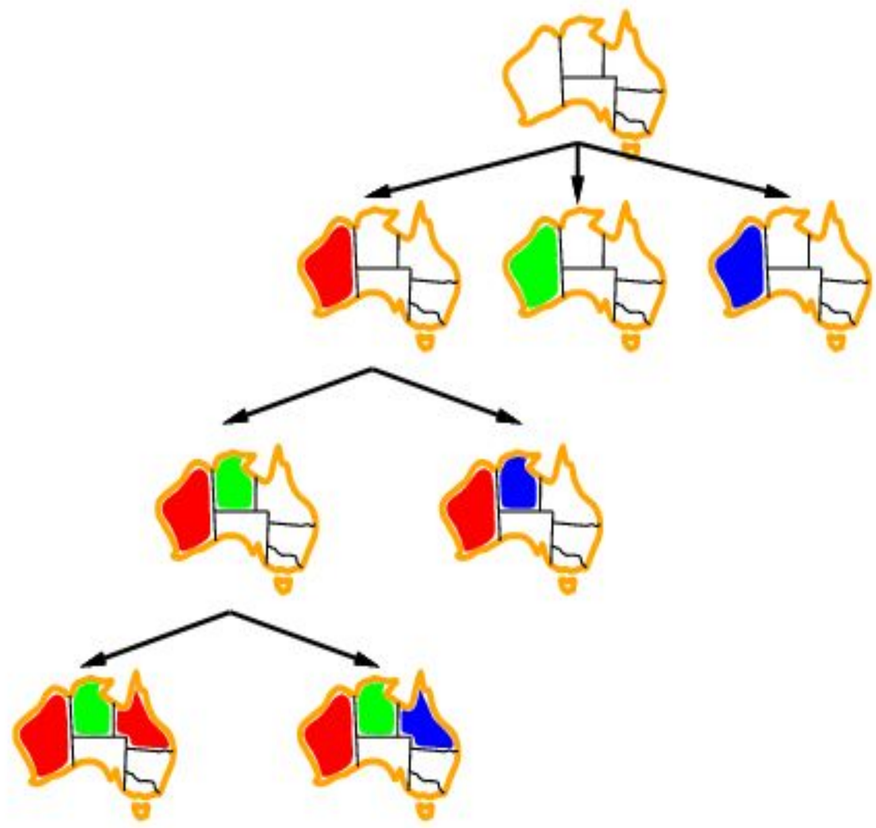
Backtracking example

Backtracking example

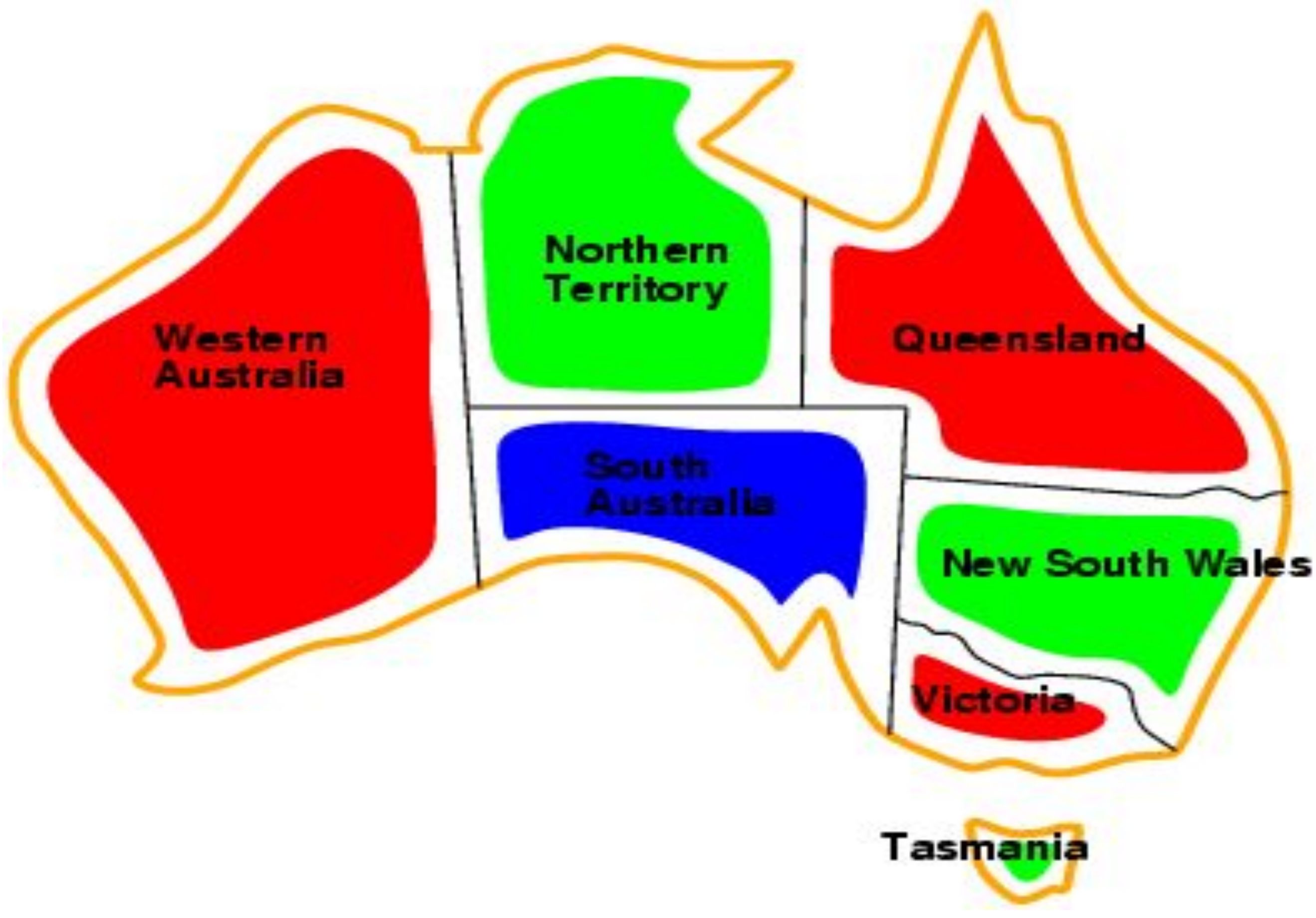
Backtracking example



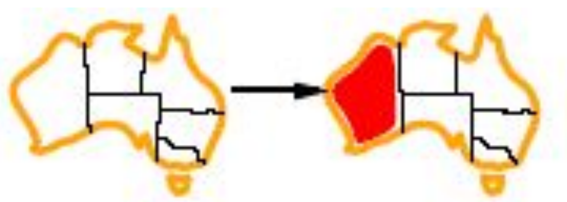
Backtracking example

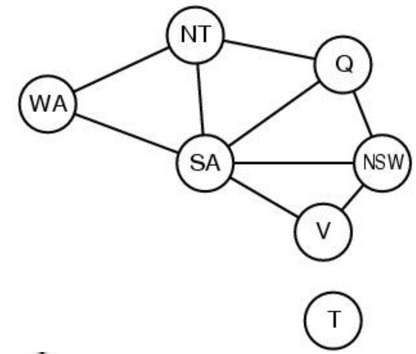


Final (one of the solution)



Forward checking



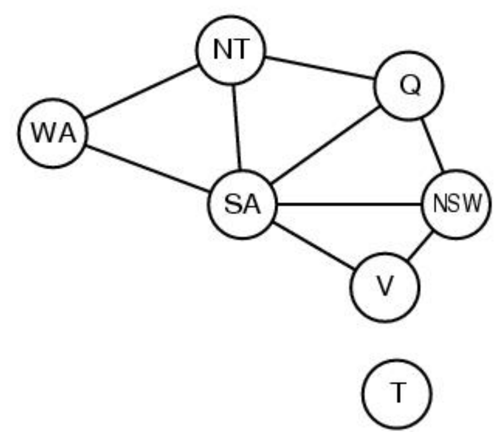


- **Assign *{WA=red}***

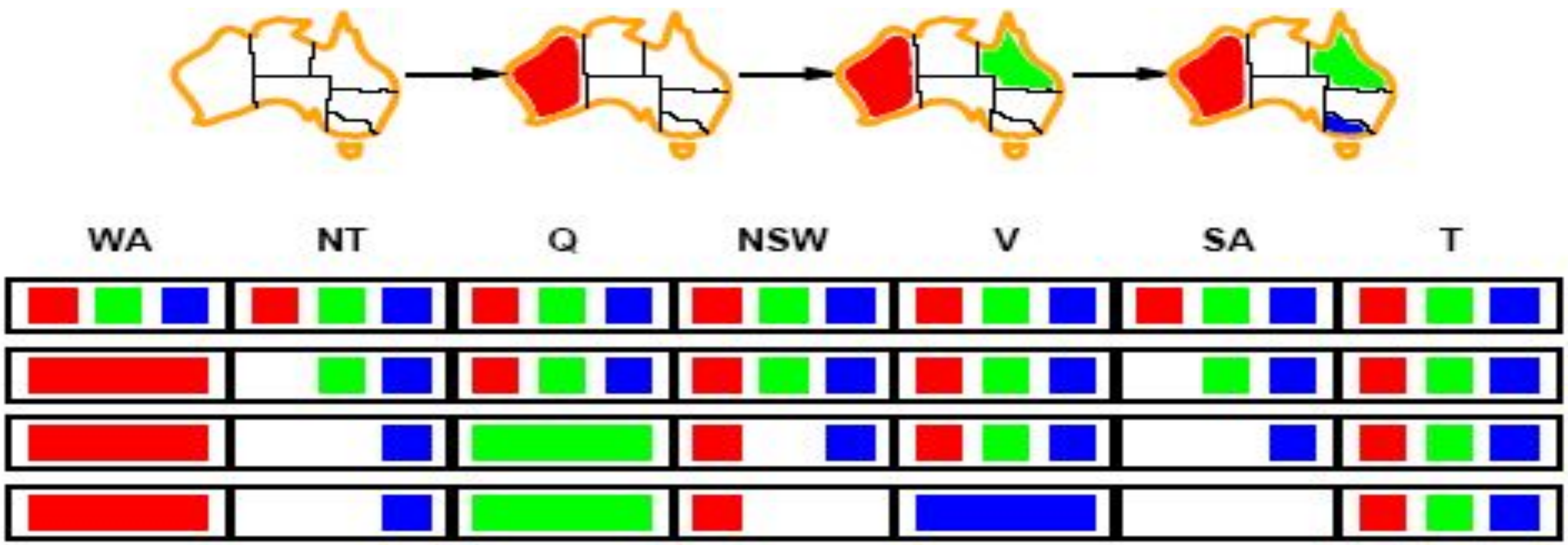
- **Effects on other variables connected by constraints to WA**
  - ***NT can no longer be red***
  - ***SA can no longer be red***

Forward checking

- **Assign *{Q=green}***
- **Effects on other variables connected by constraints with**
  - ***NT can no longer be green***
  - ***NSW can no longer be green***
  - ***SA can no longer be green***
- ***MRV heuristic* would automatically select NT or SA next**



Forward checking



- If *V* is assigned *blue*

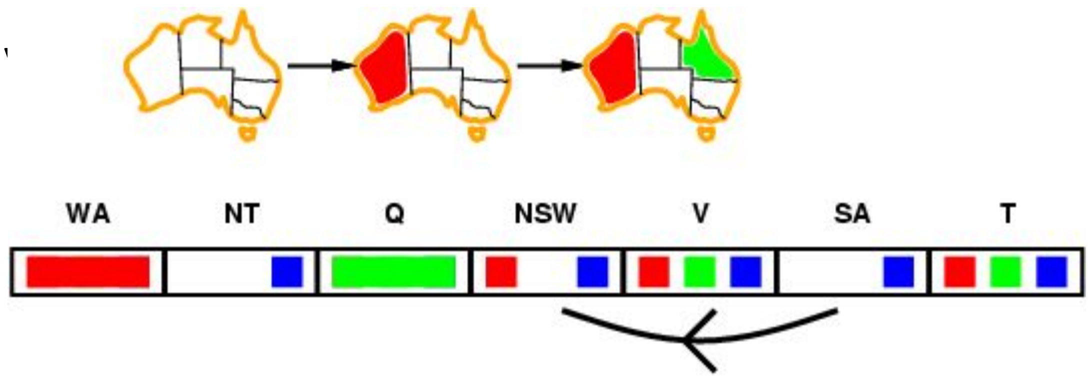
- Effects on other variables connected by constraints with WA
  - *NSW can no longer be blue*
  - *SA is empty*

- FC has detected that partial assignment is *inconsistent* with the constraints and backtracking can occur.
- No more value for SA: backtrack

Arc Consistency

- Simplest form of propagation makes each arc consistent
- $X \rightarrow Y$ is consistent iff

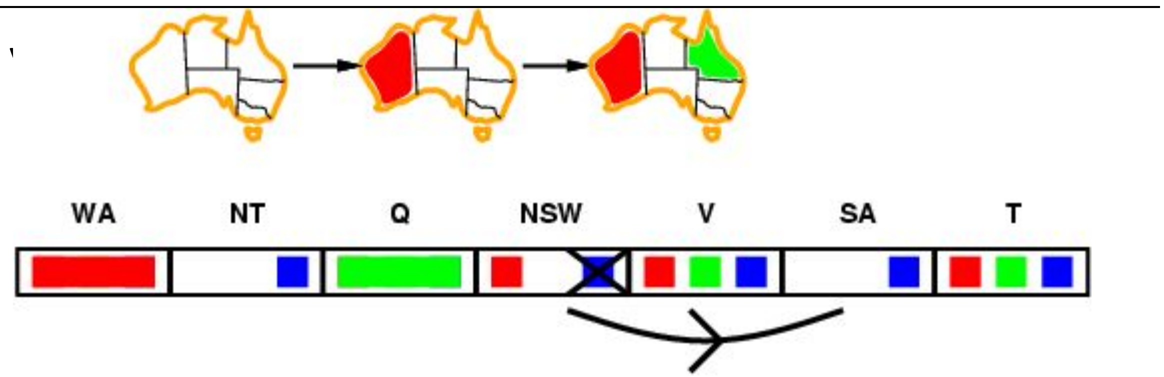
for every



Arc Consistency

- Simplest form of propagation makes each arc consistent
- *X* → *Y* is consistent iff

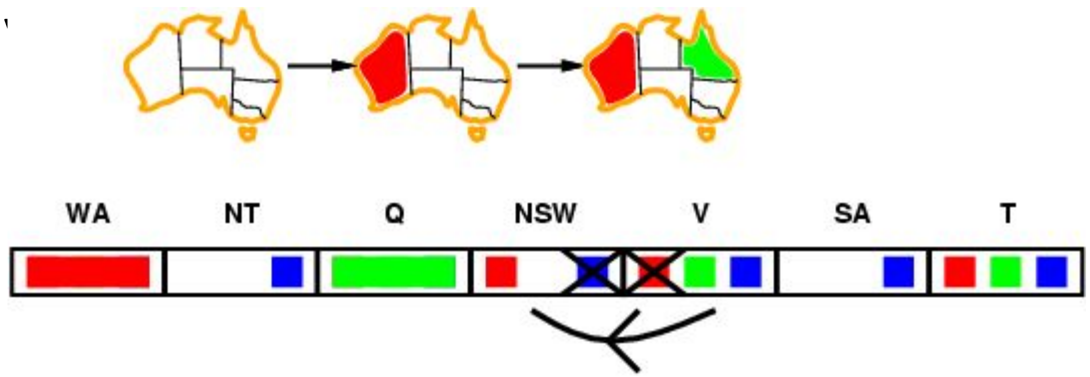
for every



Arc Consistency

- Simplest form of propagation makes each arc consistent
- *X* ⍰ *Y* is consistent iff

for every
of Y



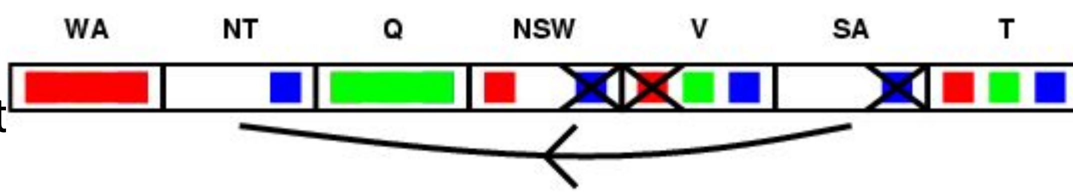
- If *X* loses a value, neighbors of *X* need to be rechecked

Arc consistency

- Simplest form of propagation makes each arc consistent
- *X* ⍰*Y* is consistent iff

  for every value *x* of *X* there is some allowed value *y* of Y

- If *X* loses a
- Arc consist



g

- Can be run as a preprocessor or after each assignment

Arc Consistency Algorithm AC-3
Sometimes called Discrete Relaxation

function AC-3(*csp*) **returns** the CSP, possibly with reduced domains
  inputs: *csp*, a binary CSP with variables $\{X_1, X_2, \ldots, X_n\}$
  local variables: *queue*, a queue of arcs, initially all the arcs in *csp*

  while *queue* is not empty **do**
    $(X_i, X_j) \leftarrow$ REMOVE-FIRST(*queue*)
    if RM-INCONSISTENT-VALUES(X_i, X_j) **then**
      for each X_k **in** NEIGHBORS[X_i] **do**
        add (X_k, X_i) to *queue*

---

function RM-INCONSISTENT-VALUES(X_i, X_j) **returns** true iff remove a value
  removed $\leftarrow$ *false*
  for each x **in** DOMAIN[X_i] **do**
    if no value y in DOMAIN[X_j] allows (x,y) to satisfy constraint(X_i, X_j)
      then delete x from DOMAIN[X_i]; *removed* $\leftarrow$ *true*
  return *removed*

- Time complexity: $O(n^2d^3)$

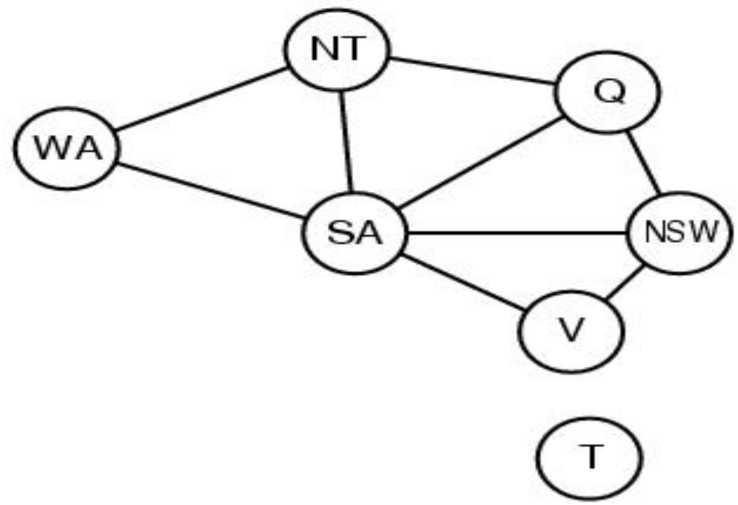
Variable and value ordering

- For Ex:
  - After assigning WA=red and NT=green there is only one possible value for SA i.e blue.
  - After SA is assigned, Values for Q,NSW, V are all forced
  - Choosing the variable with fewest legal moves is called MRV heuristic or MCV or fail-first heuristic.


Northern Territory
Western Australia
Queensland
South Australia
New South Wales
Victoria
Tasmania

Variable and value ordering

- Degree heuristic: Degree heuristic can be useful as a tie breaker after MRV
- MRV heuristic is a more powerful guide, but the degree heuristic can be useful as a tie breaker



- LCV heuristic: The value that rules out the fewest values in the remaining variables(i.e what value will leave the most other values for other variable)

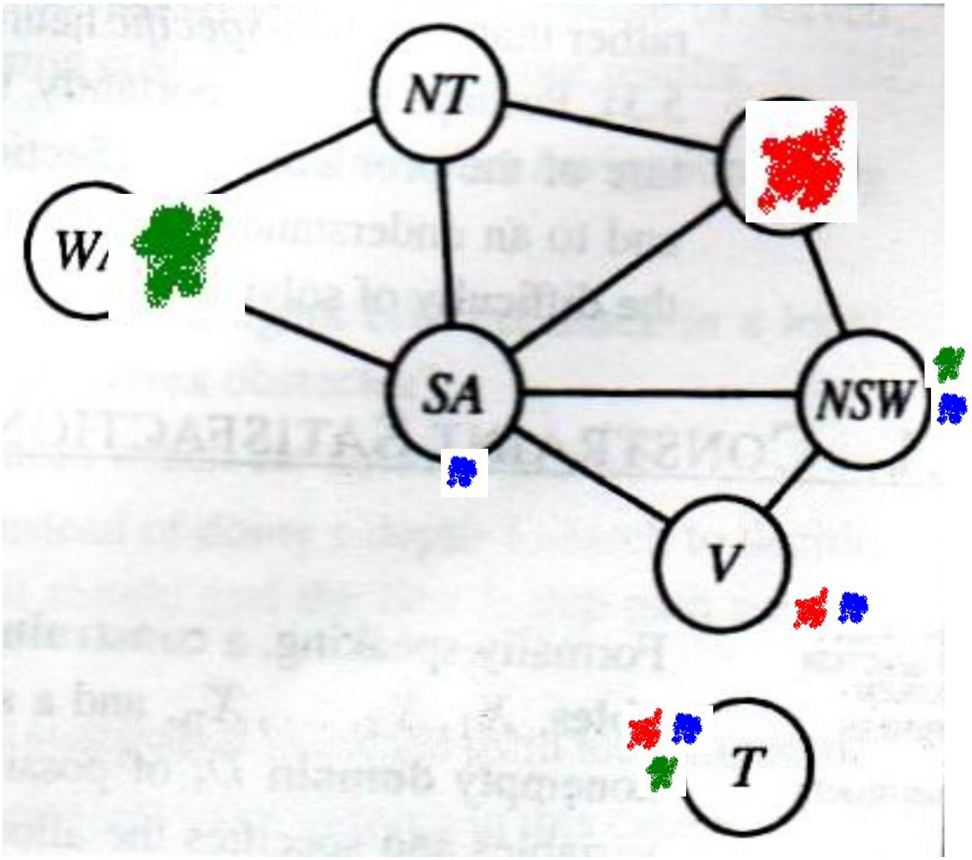
LCV: Least constraining value

Constraint propagation

- There are some inconsistencies with forward checking
- Constraint propagation is stronger than forward checking by considering arc consistency
- the arc is consistent if, for *every* value x of SA, there is *some* value y of NSW that is consistent with x.
- For Ex: SA - {blue} and NSW = {red, blue}

Arc Consistency

- A→B is consistent if for each remaining value in domain of A, there may be a consistent value in domain of B.
  - Consistent:
    - SA→NSW, NSW→V,…
  - Not Consistent:
    - NSW→SA, NT→SA,…



Arc consistency

- Simplest form of propagation makes each arc consistent
- $X \rightarrow Y$ is consistent iff

  for every value x of X there is some allowed y

Special Constraints…

- All Different
  - Sort remaining variables based on their number of choices…
- Resource Constraints
  - Checking the sums, …
    - Flight1 $\in$ [0,165] , Flight2 $\in$ [0,385]
    - Flight1+Flight2 = 420

     Flight1 $\in$ [35,165] , Flight2 $\in$ [255,385]

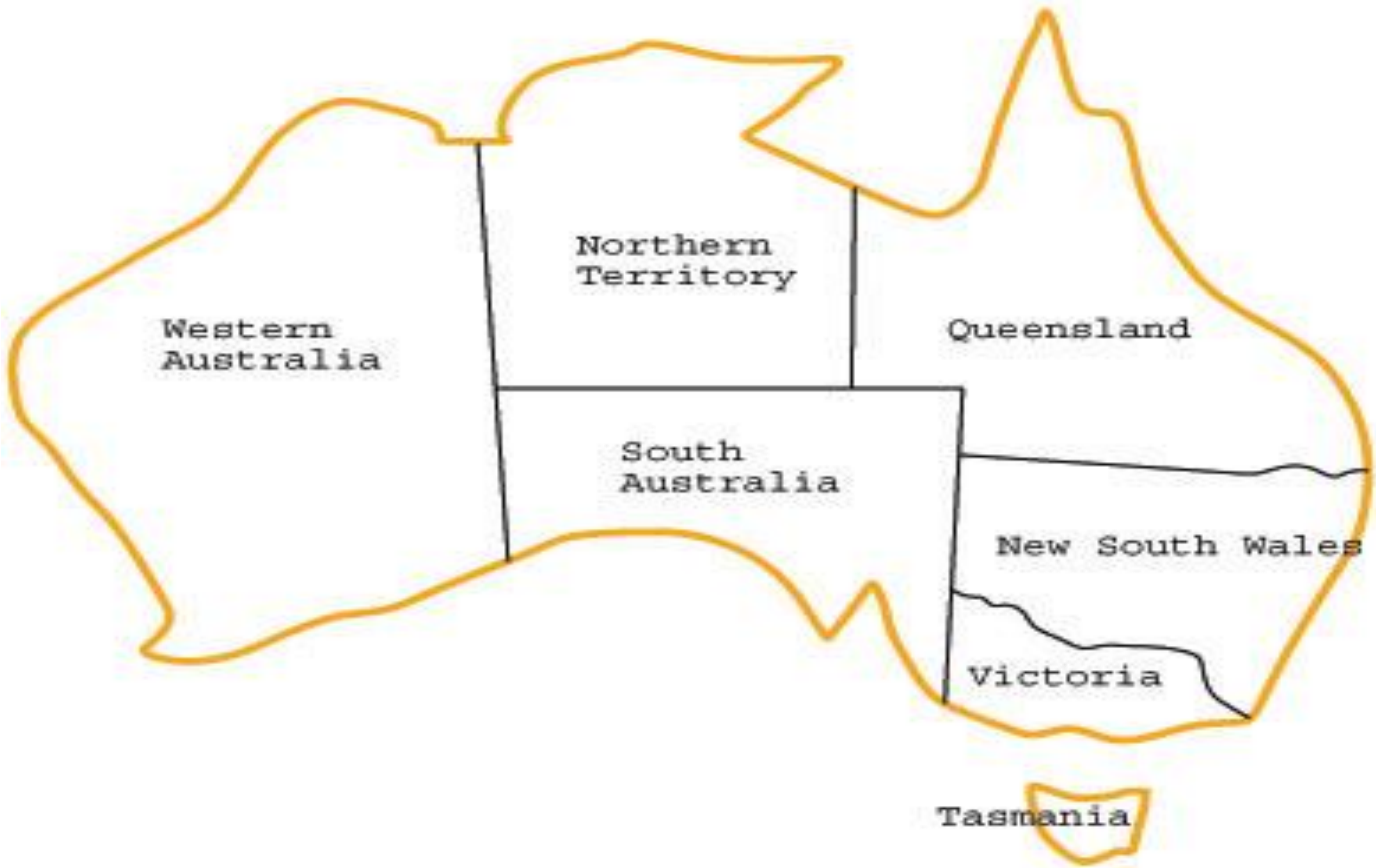
    - <u>Bounds Propagation</u>

Intelligent Backtracking

- The BACKTRACKING-SEARCH algorithm has a very simple policy for what to do when a branch of the search fails: back up to the preceding variable and try a different value for it.
- This is called chronological backtracking, because the *most recent decision* point is revisited.
- There are much better ways
- One way to get rid of the problem is using **intelligent backtracking** algorithms
- **Backjumping** (**BJ**) is different from BT in the following:

BJ vs. BT

We want to color each area in the map with a different color



BJ vs. BT

- Let's consider what BT does in the map coloring problem
  - Assume that variables are assigned in the order *Q, NSW, V, T, SA, WA, NT*
  - Assume that we have reached the partial assignment

    Q = red, NSW = green, V = blue, T = red

  - When we try to give a value to the next variable *SA*, we find out that all possible values violate constraints
    - Dead end!
  - BT will backtrack to try a new value for variable *T*!
    - Not a good idea!

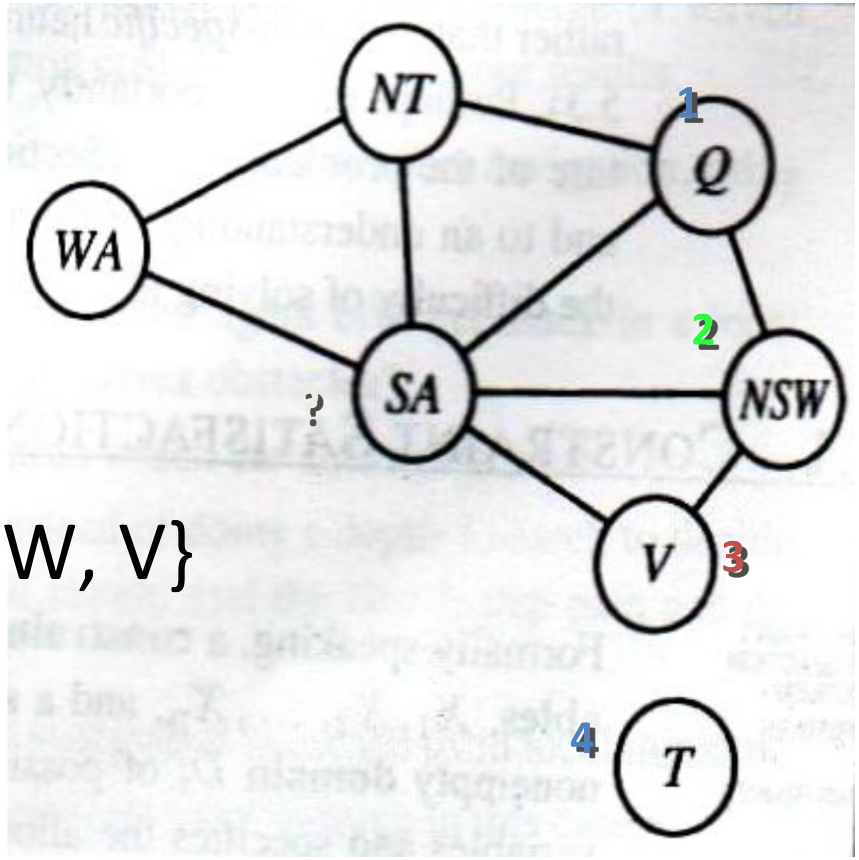
BJ vs. BT

- BJ has a smarter approach to backtracking
  - A more intelligent approach to backtracking is to go all the way back to one of the set of variables that *caused the **failure***
  - The set of these variables is called a **conflict set**
  - The conflict set for SA is {Q, NSW, V}

Back Jumping

1. Q Red
2. NSW Green
3. V Blue
4. T Red
5. SA ?

- Conflict Set: {Q,NSW, V}



The end of CSP