CS 4100/5100: Foundations of AI

Constraint Satisfaction

Instructor: Rob Platt
rplatt@ccs.neu.edu

College of Computer and information Science
Northeastern University

Fall, 2014

CSP example

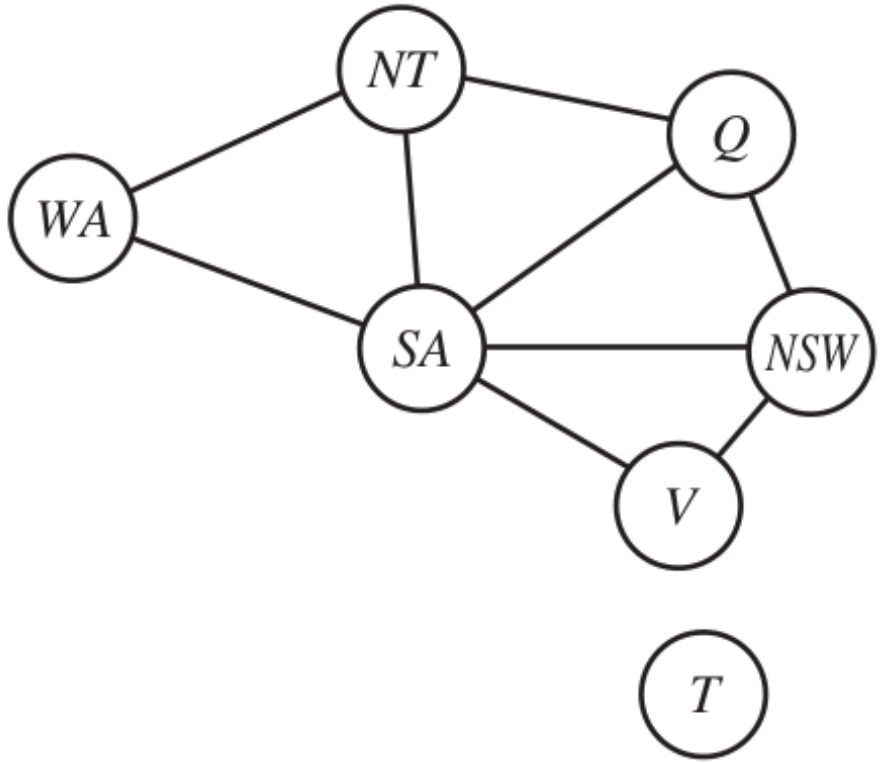


Image: Russell and Norvig

CSP example

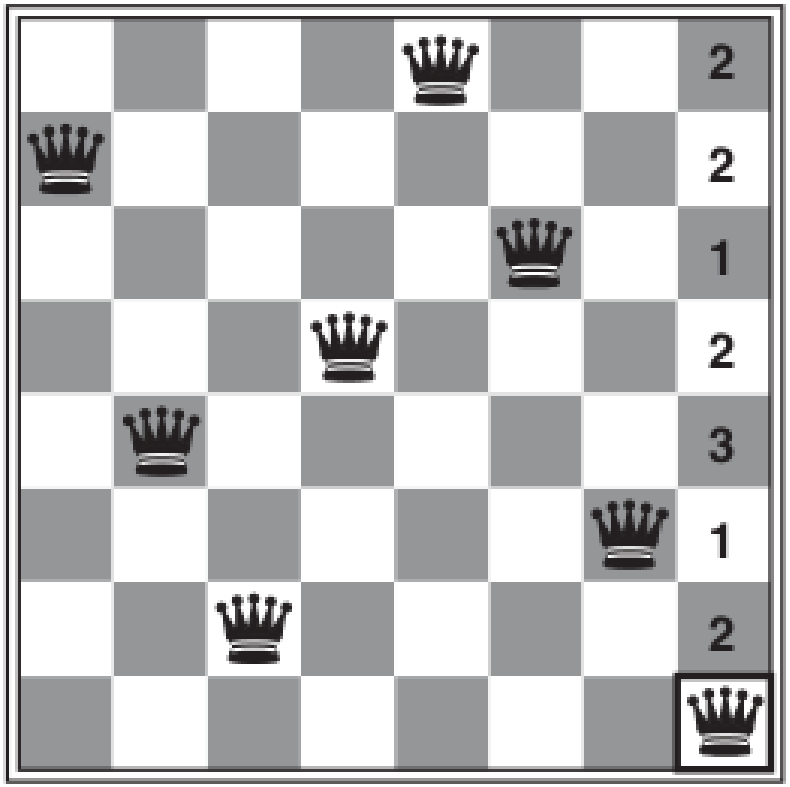


Image: Russell and Norvig

CSP example

$$\begin{array}{r} T\ W\ O \\ +\ T\ W\ O \\ \hline F\ O\ U\ R \end{array}$$

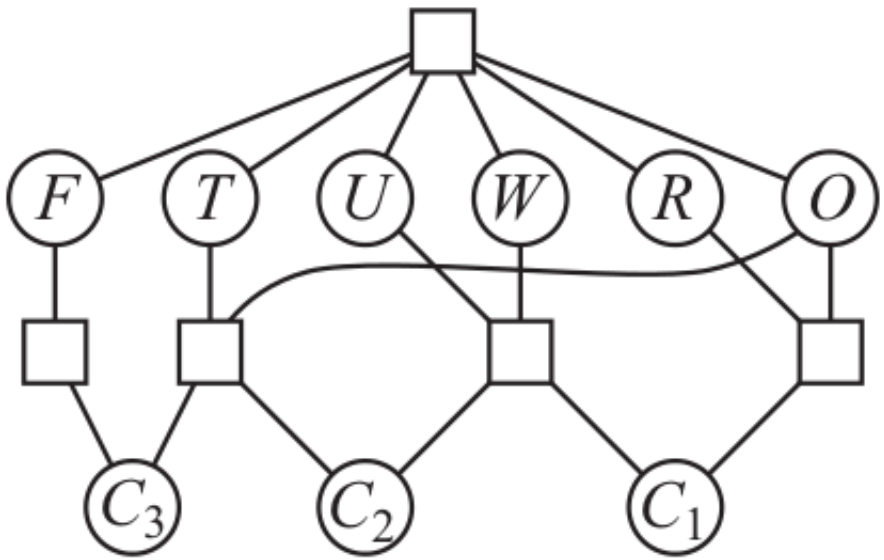


Image: Russell and Norvig

Example of backtracking search

Image: Hwee Tou Ng

Example of backtracking search

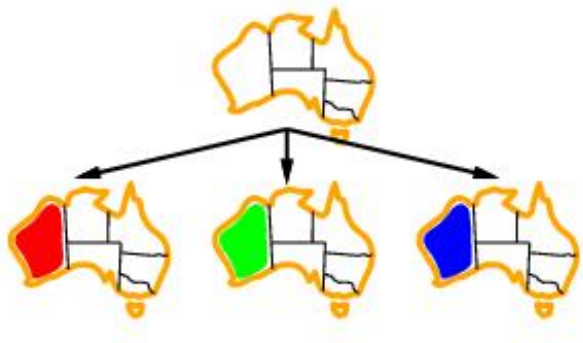


Image: Hwee Tou Ng

Example of backtracking search

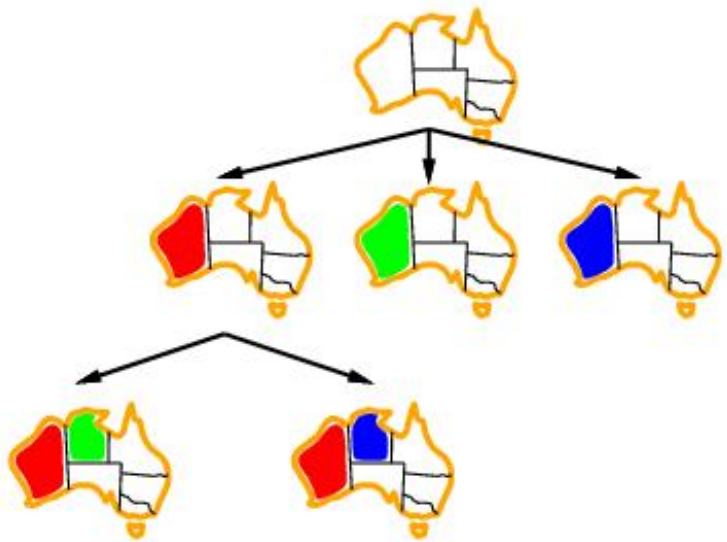


Image: Hwee Tou Ng

Example of backtracking search

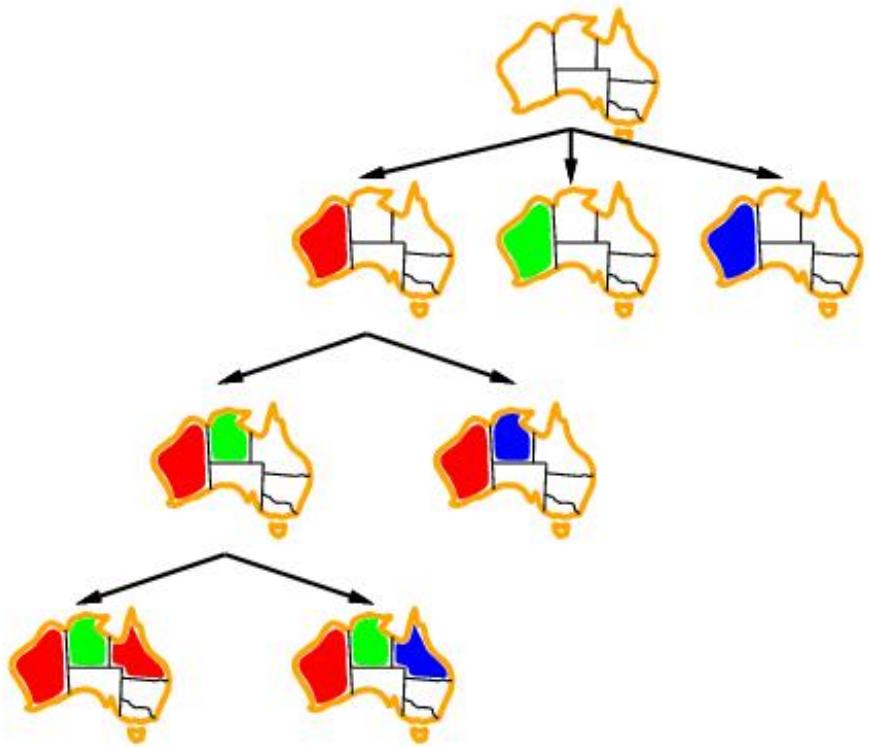


Image: Hwee Tou Ng

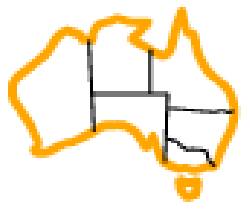
Backtracking algorithm

```
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
```

Image: Russell and Norvig

Forward checking example



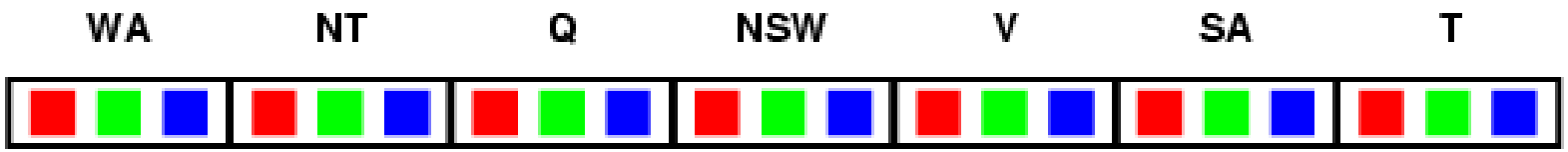
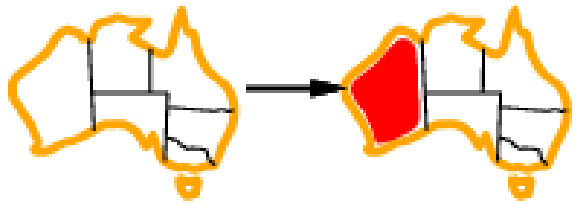


Image: Hwee Tou Ng

Forward checking example



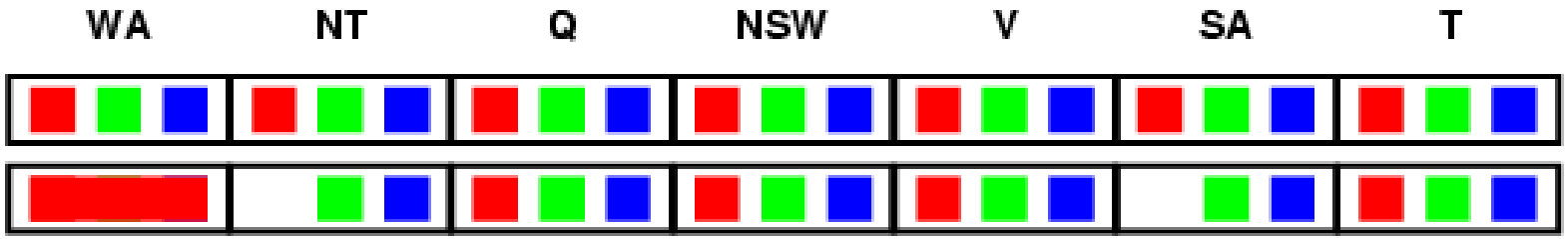


Image: Hwee Tou Ng

Forward checking example

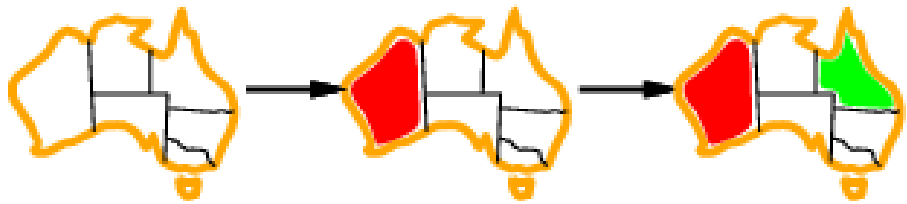
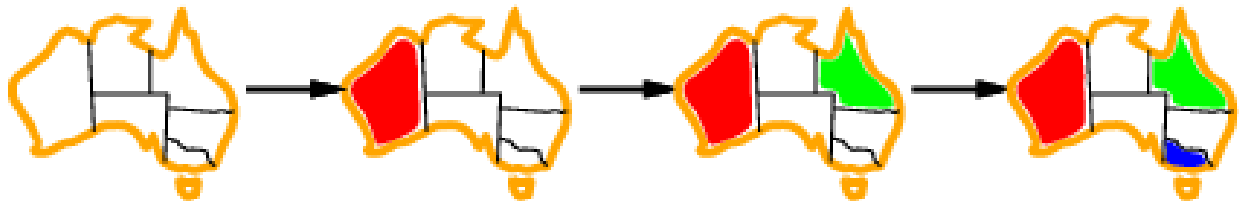


Image: Hwee Tou Ng

Forward checking example



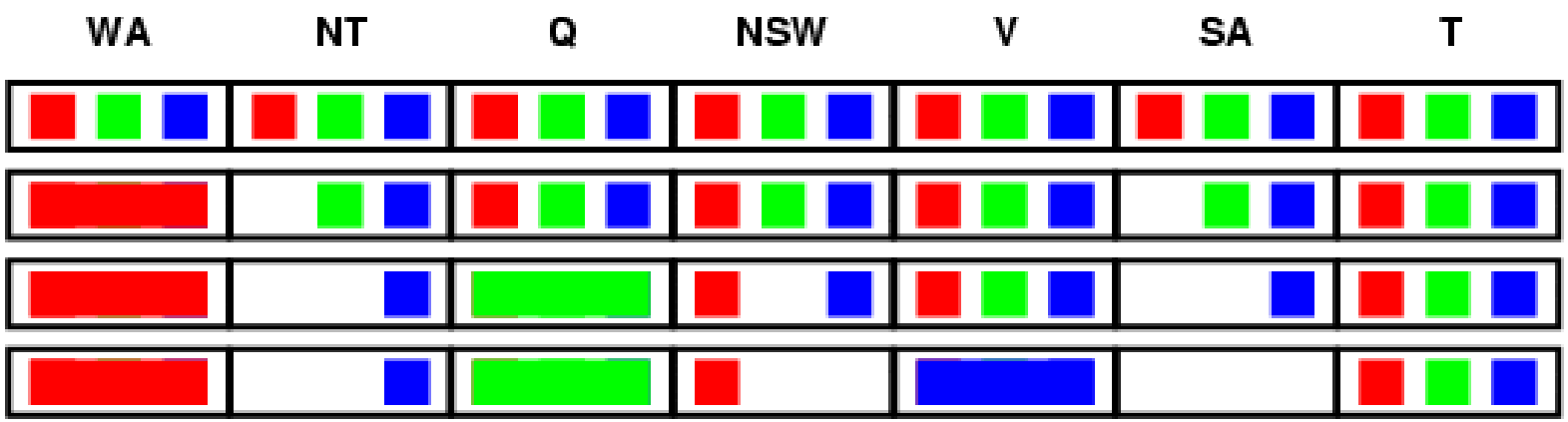


Image: Hwee Tou Ng

AC3

function AC-3(csp) **returns** false if an inconsistency is found and true otherwise
  inputs: csp, a binary CSP with components $(X,\ D,\ C)$
  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 REVISE(csp, X_i, X_j) **then**
      if size of $D_i\ =\ 0$ **then return** $false$
      for each X_k **in** X_i.NEIGHBORS - $\{X_j\}$ **do**
        add $(X_k,\ X_i)$ to $queue$
  return $true$

---

function REVISE(csp, X_i, X_j) **returns** true iff we revise the domain of X_i
  $revised \leftarrow false$
  for each x **in** D_i **do**
    if no value y in D_j allows (x,y) to satisfy the constraint between X_i and X_j **then**
      delete x from D_i
      $revised \leftarrow true$
  return $revised$

Image: Russell and Norvig

MRV heuristic

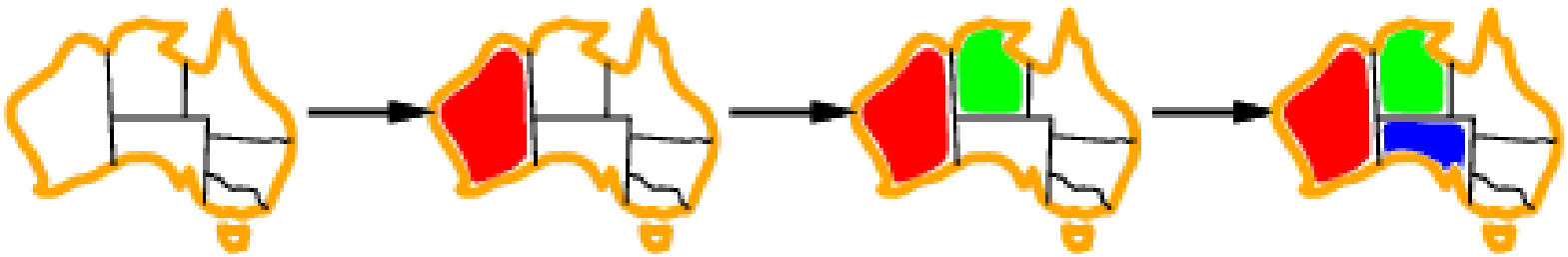


Image: Hwee Tou Ng

LCV heuristic

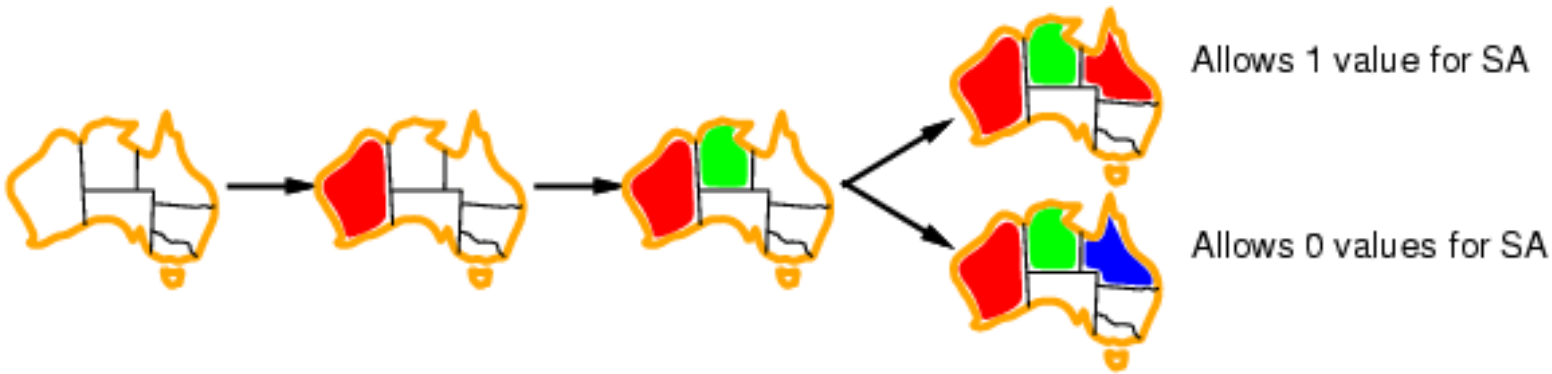


Image: Hwee Tou Ng

Local search: min conflicts

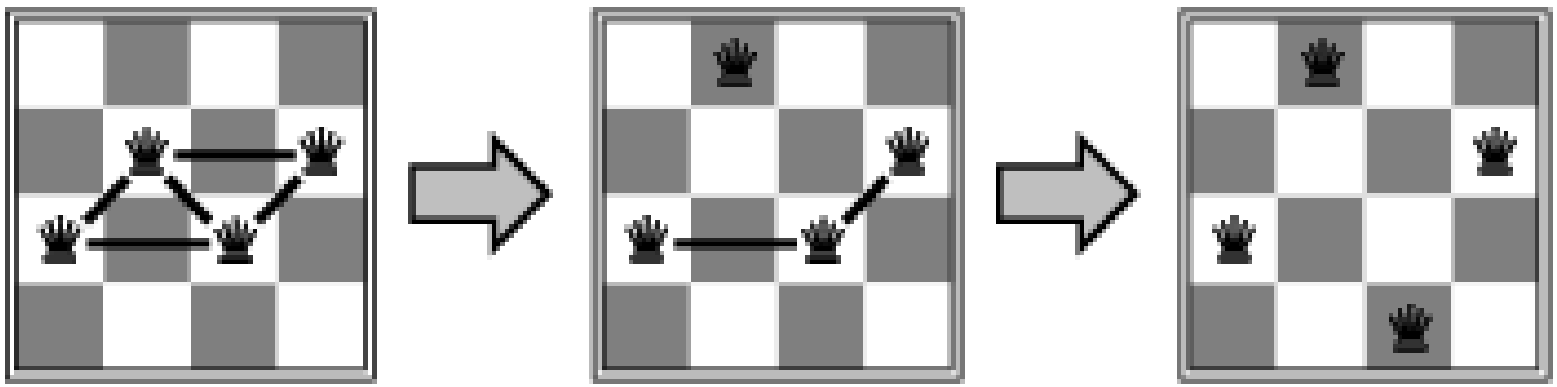


Image: Hwee Tou Ng