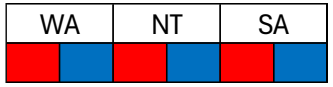
CONSTRAINT PROPAGATION: INFERENCE IN CSPS



Arc Consistent: YES
Path Consistent: NO

PATH CONSISTENCY

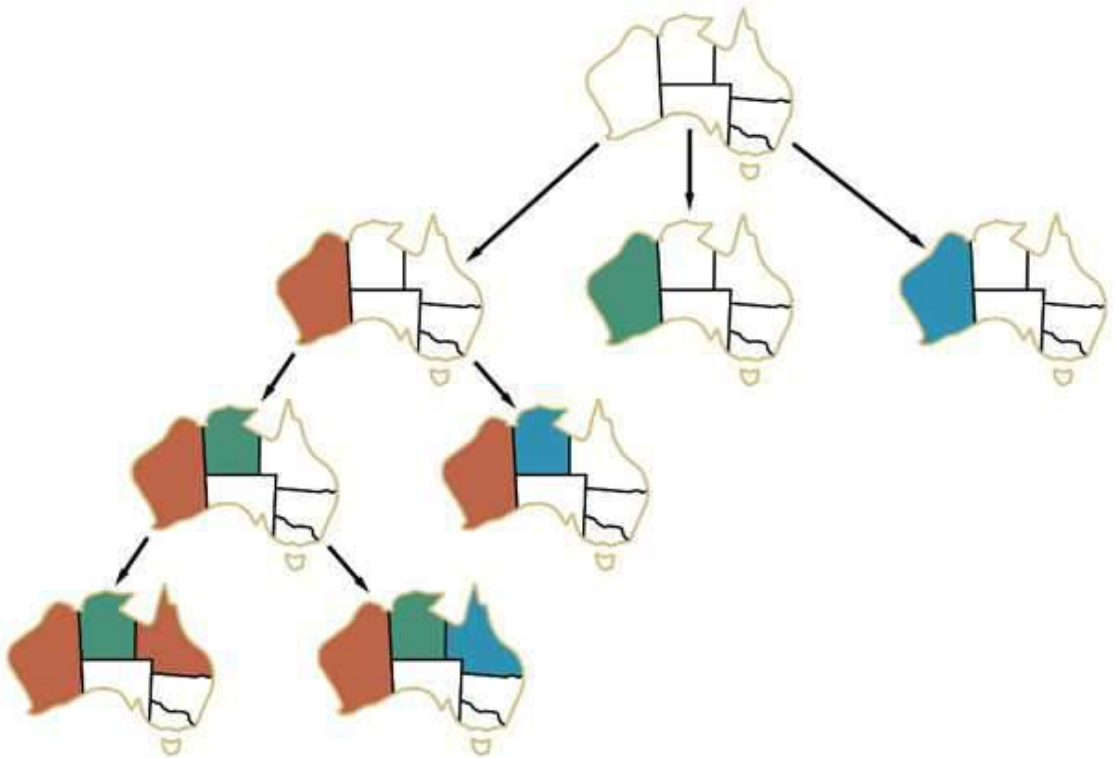
- Path consistency tightens the binary constraints by using implicit constraints that are inferred by looking at triples of variables.

- A two-variable set $\{X_i, X_j\}$ is path-consistent with respect to a third variable X_m if, for every assignment $\{X_i = a, X_j = b\}$ consistent with the constraints (if any) on $\{X_i, X_j\}$, there is an assignment to X_m that satisfies the constraints on $\{X_i, X_m\}$ and $\{X_m, X_j\}$.
- The name refers to the overall consistency of the path from X_i to X_j with X_m in the middle.

K-CONSISTENCY

- A CSP is k-consistent if, for any set of k-1 variables and for any consistent assignment to those variables, a consistent value can always be assigned to any kth variable.

BACKTRACKING SEARCH FOR CSP



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

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

BACKTRACKING SEARCH FOR CSP

- Which variable should be assigned next (SELECT-UNASSIGNED-VARIABLE) and in which order should its values be tried (ORDER-DOMAIN-VALUES)?
- What inferences should be performed at each step in the search (INFERENCE)?
- Can we BACKTRACK more than one step when appropriate?
- Can we save and reuse partial results from the search?

VARIABLE AND VALUE ORDERING

- Defined order!
- Random order!

- **Minimum-remaining-values (MRV) heuristic:** Picks a variable that is most likely to cause a failure soon.
- **Degree heuristic:** Picks a variable that is involved in the largest number of constraints on other unassigned variables.
  - Useful as a tie-breaker.
- **Least-constraining-value heuristic:** It prefers the value that rules out the fewest choices for the neighboring variables in the constraint graph.
- Variable selection is fail-first, but value selection is fail-last!