Constraint Satisfaction Problems (CSP)

- What is a CSP?
  - Finite set of variables $V_1, V_2, \ldots, V_n$
    - Nonempty domain of possible values for each variable $D_{V1}, D_{V2}, \ldots D_{Vn}$
  - Finite set of constraints $C_1, C_2, \ldots, C_m$
    - Each constraint C_i limits the values that variables can take,
      - e.g., $V_3 > 0$ or $V_1 \neq V_2$,
- A *state* is defined as an *assignment* of values to some or all variables.
- *Consistent assignment*
  - assignment does not violate the constraints
- *Complete assignment*
  - when every variable is mentioned in the assignment

Constraint Satisfaction Problems (CSP)

- A *solution* to a CSP is a complete assignment that satisfies all constraints.
- If a solution to a CSP exists then it can be found with a *backtracking search* over the states (assignments).
- Scheduling a meeting of X number of people with constraints on their available time is the premier example of a CSP.

Backtracking Search

```
function BACKTRACKING-SEARCH(csp)
    return RECURSIVE-BACKTRACKING({} , csp)

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

Note: *csp* is the representation of our constraint satisfaction problem, VARIABLES(csp) and CONSTRAINTS(csp) are accessor functions that access the respective parts of the respective parts of the representation.

Numeric CSPs

- Our CSP:
  - Two variables: X,Y
  - Domains: [1,9] (for both vars)
  - Constraint: X+Y=X*Y
  - Problem: Find a value for X and Y that satisfies the constraint.
- It is easy to see that the complete assignment {X=2,Y=2} is a solution.
- Compute the solution with recursive backtracking.

Scheduling

- Schedule a meeting:
  - Variables: Peter, Paul, Mary
  - Domains: available times
    - Peter: {10-11,11-12,2-3}
    - Paul: {11-12,1-2}
    - Mary: {10-11,11-12,3-4}
  - Problem: Find a common meeting time.

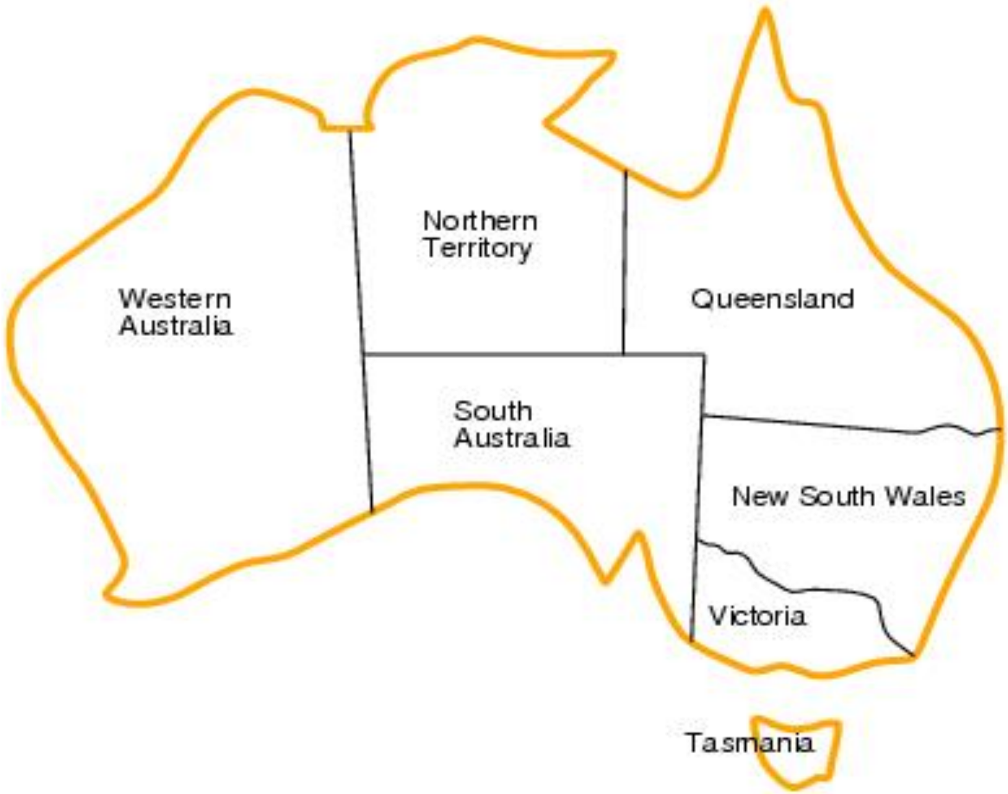
Scheduling (over-constrained problems)

- Schedule a meeting:
  - Variables: Peter, Paul, Mary
  - Domains: available times
    - Peter: {10-11,11-12,2-3}
    - Paul: {11-12,1-2}
    - Mary: {10-11,2-3,3-4}
  - Problem: Find a common meeting time.
- Over-constrained problems do not have solutions!

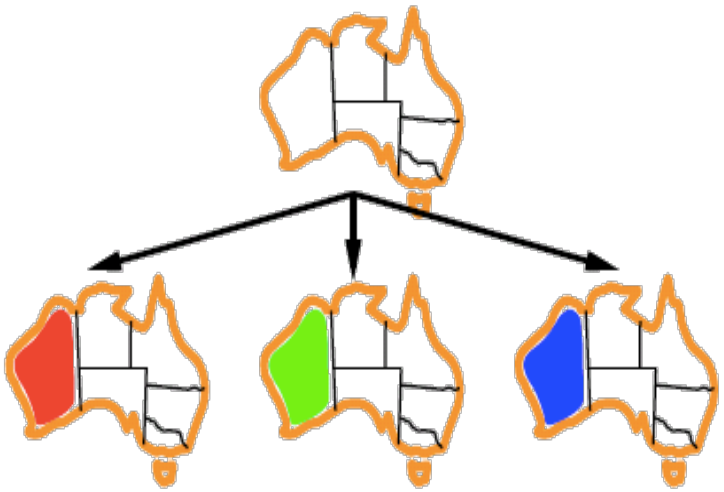
Map Coloring

- Perhaps the most famous CSP: Color a map such that no two adjacent areas have the same color.
- CSP:
  - Variables: areas on the map
  - Domain: a set of colors (turns out for maps we only need four different colors - the *four color theorem*)
  - Constraint: no two adjacent areas can have the same color.

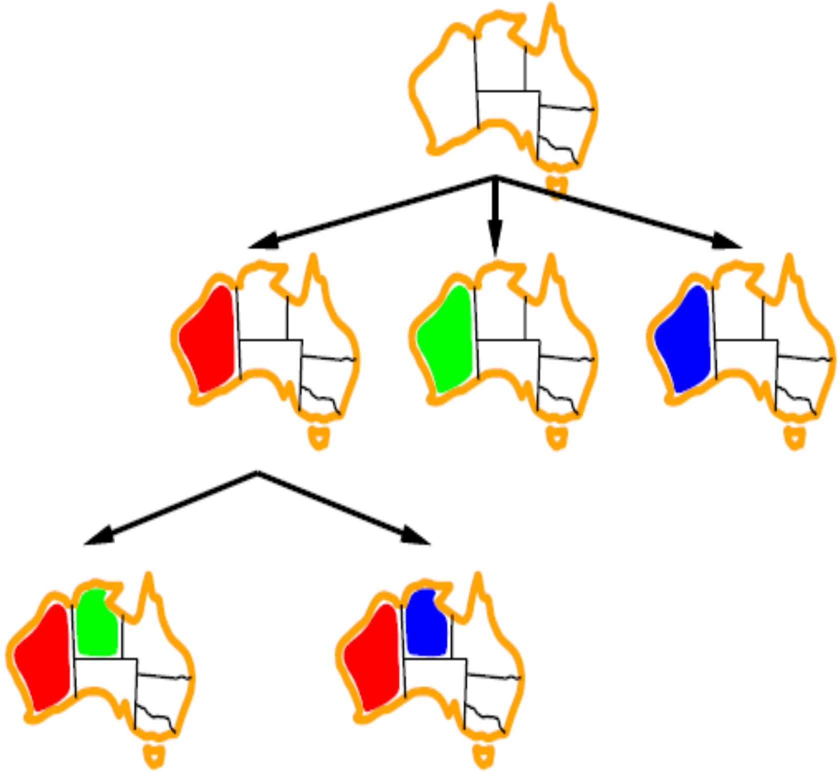
Map Coloring



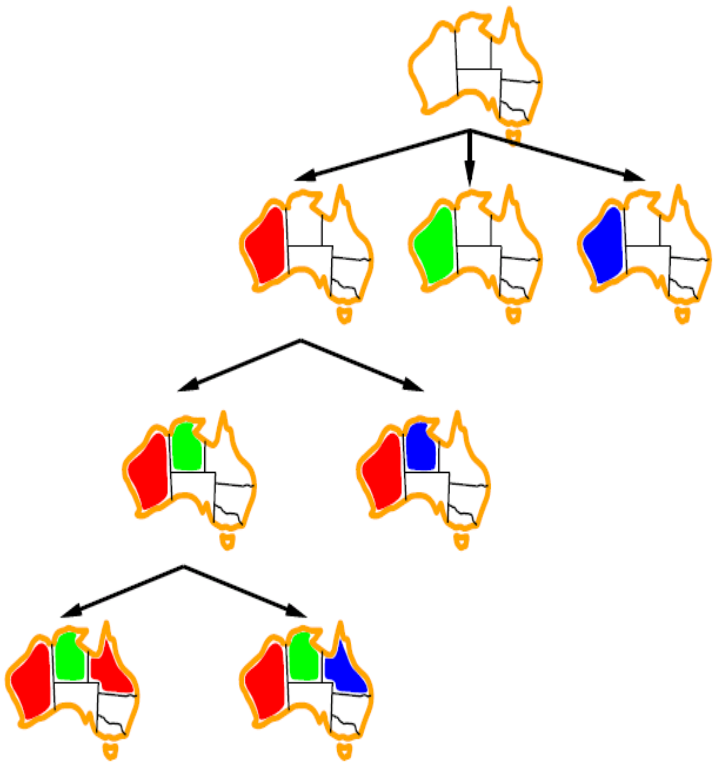
Map Coloring



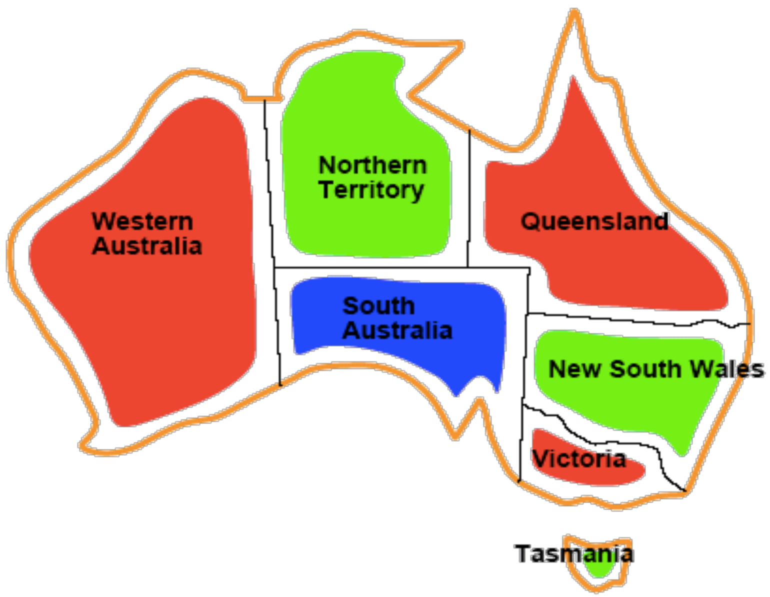
Map Coloring



Map Coloring



Map Coloring



- Solutions are assignments satisfying all constraints, e.g.

  {WA=red,NT=green,Q=red,NSW=green,V=red,SA=blue,T=green}

Reading

- Chapter 4 (up to including 4.4)