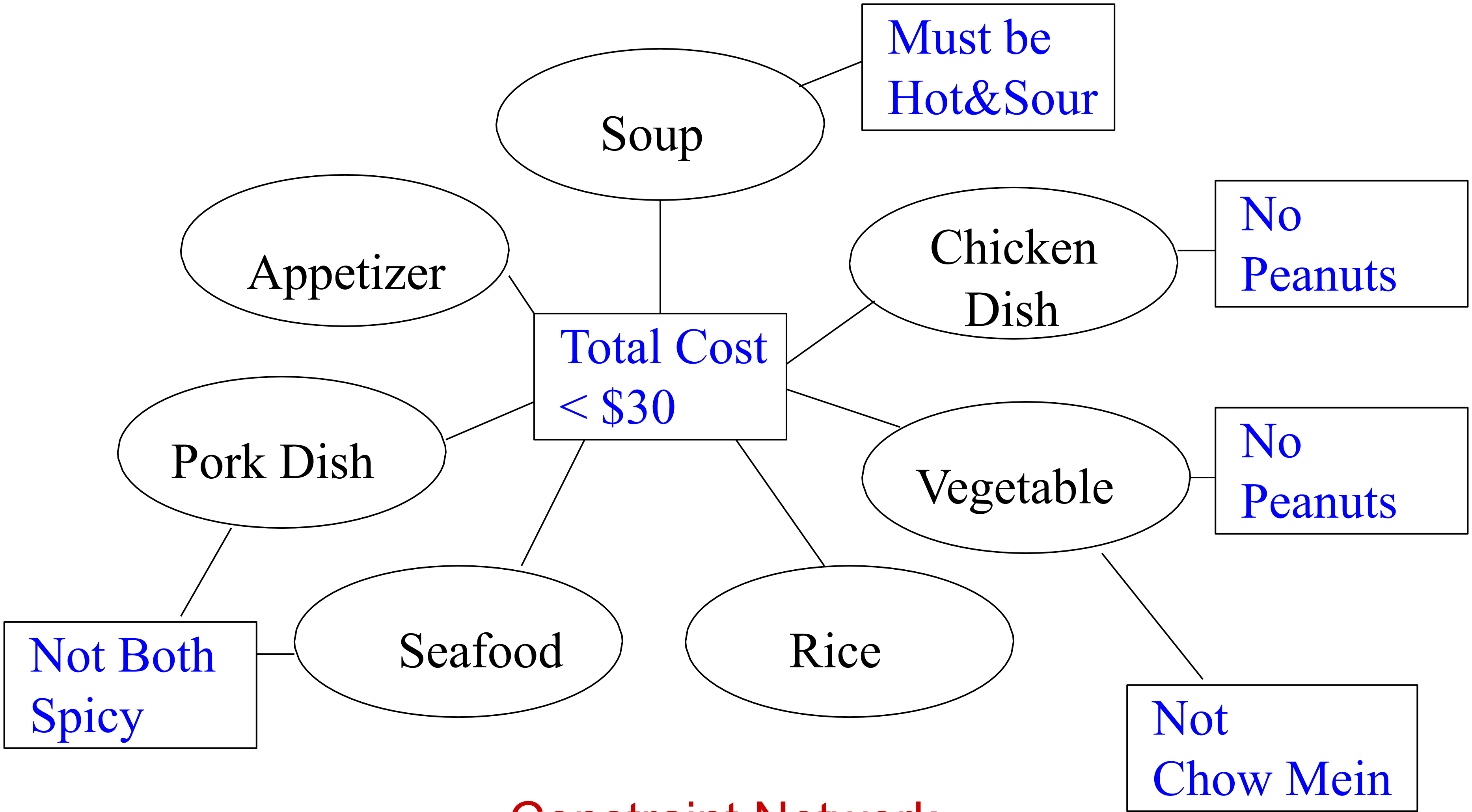
Constraint Satisfaction Problems



Constraint Network

Formal Definition of CSP

- A constraint satisfaction problem (CSP) is a triple (V, D, C) where
  - V is a set of variables $X_1, \ldots, X_n$.
  - D is the union of a set of domain sets $D_1,\ldots,D_n$, where D_i is the domain of possible values for variable X_i.
  - C is a set of constraints on the values of the variables, which can be pairwise (simplest and most common) or k at a time.

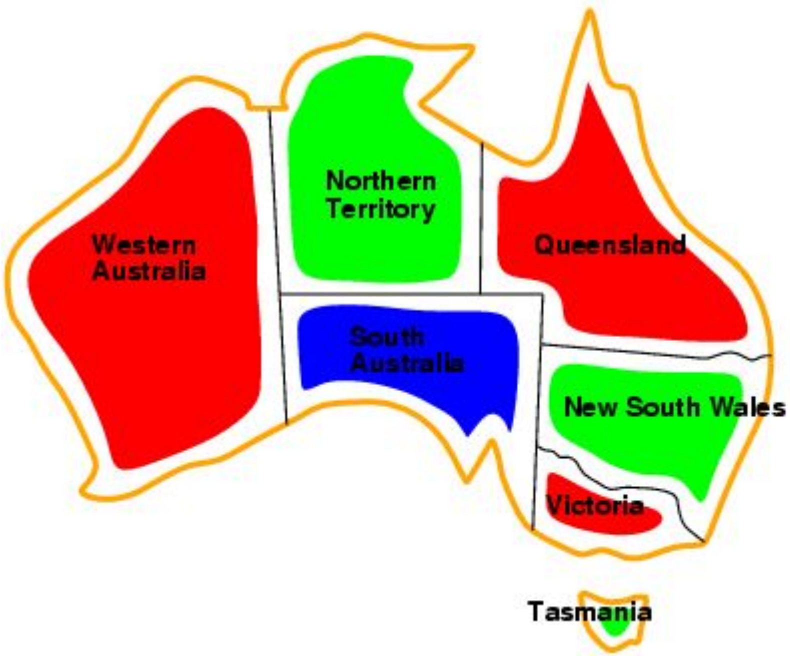
CSPs vs. Standard Search Problems

- Standard search problem:
  - state is a "black box“ - any data structure that supports successor function, heuristic function, and goal test
- 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
- Simple example of a formal representation language
- 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, or (WA,NT) in {(red,green),(red,blue),(green,red),(green,blue),(blue,red),(blue,green)}

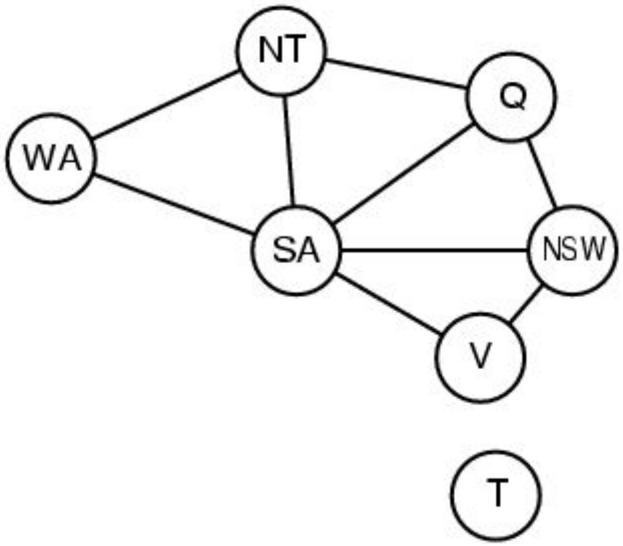
Example: Map-Coloring



- 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., $SA \neq green$

- Binary constraints involve pairs of variables,
  - e.g., $value(SA) \neq value(WA)$

- Higher-order constraints involve 3 or more variables,
  - e.g., cryptarithmetic column constraints

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

What Kinds of Algorithms are used for CSP?

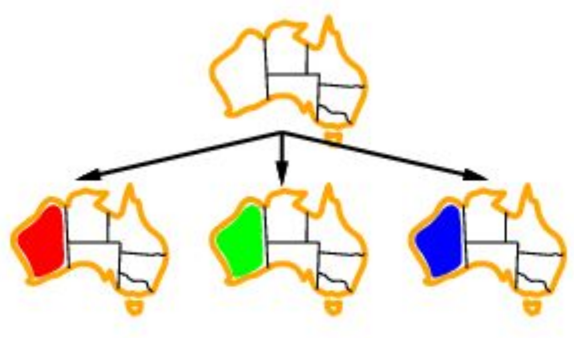
- Backtracking Tree Search
- Tree Search with Forward Checking
- Tree Search with Discrete Relaxation (arc consistency, k-consistency)
- Many other variants
-

Backtracking Tree Search

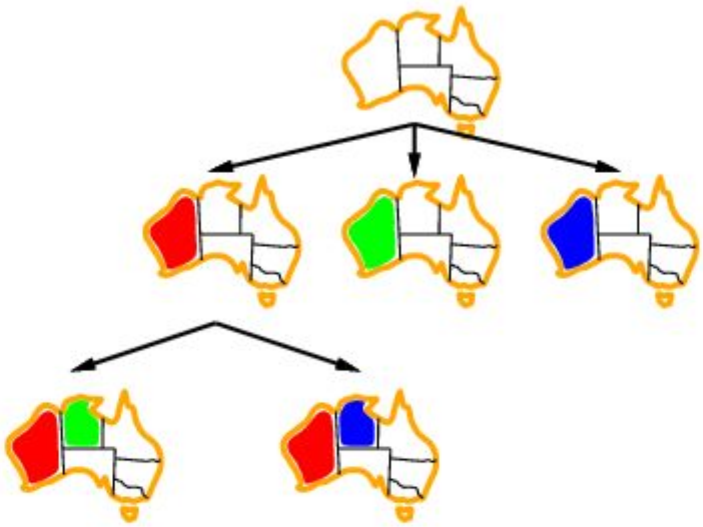
- Variable assignments are commutative}, i.e.,
  [WA = red then NT = green] same as [NT = green then WA = red]
- Only need to consider assignments to a single variable at each node.
- Depth-first search for CSPs with single-variable assignments is called backtracking search.
- Backtracking search is the basic uninformed algorithm for CSPs.
- Can solve *n*-queens for $n \approx 25$.

Backtracking Example

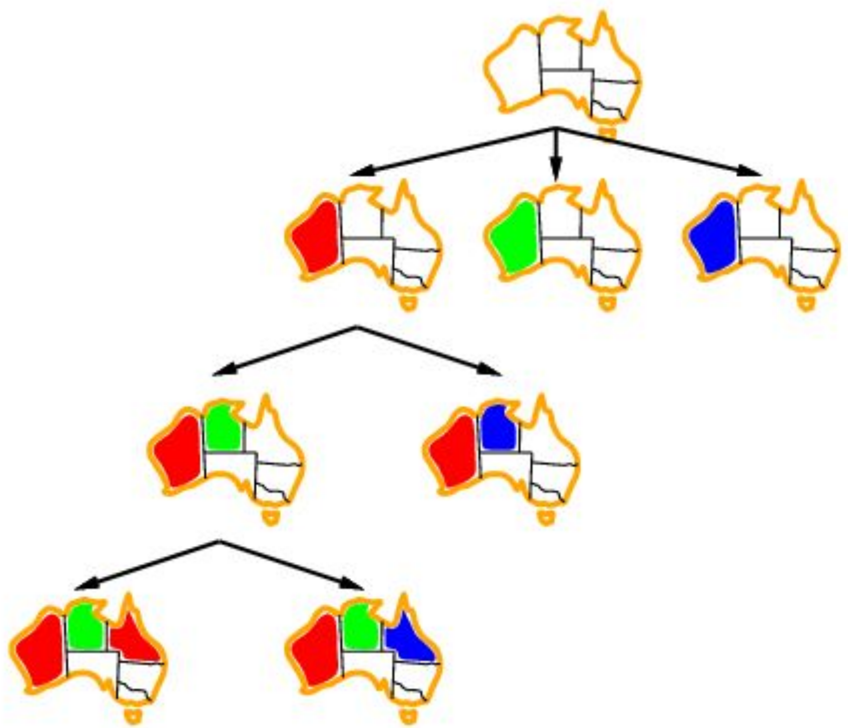
Backtracking Example



Backtracking Example

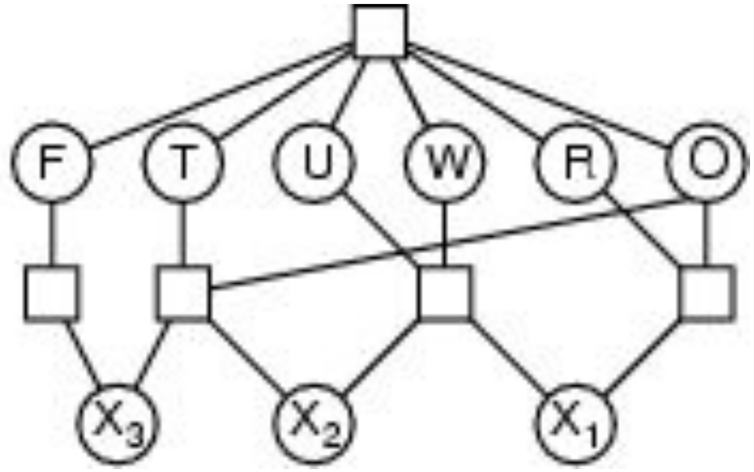


Backtracking Example



Example: Cryptarithmetic

```
  T W O
+ T W O
-------
F O U R
```



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

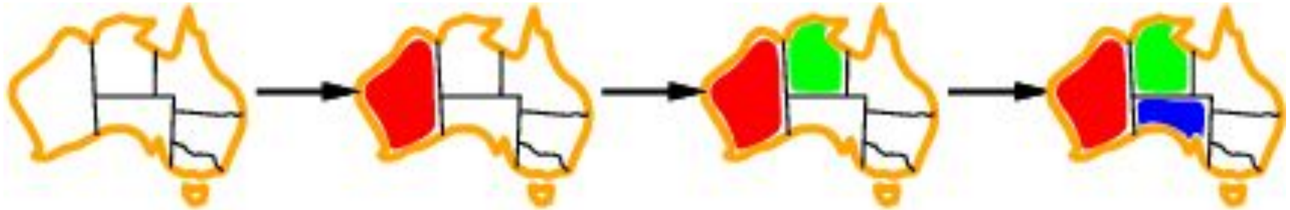
Improving Backtracking Efficiency

- General-purpose methods can give huge gains in speed:
  - Which variable should be assigned next?
  - In what order should its values be tried?
  - Can we detect inevitable failure early?

Most Constrained Variable

- Most constrained variable:

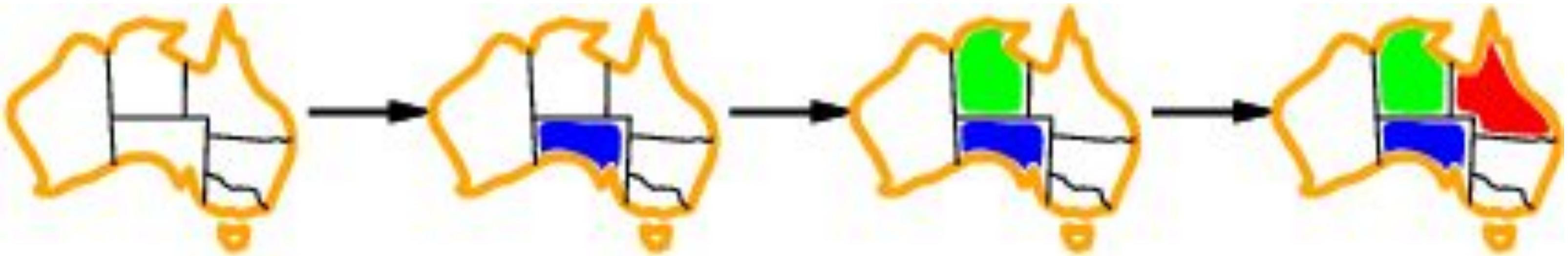
  choose the variable with the fewest legal values



- a.k.a. minimum remaining values (MRV) heuristic

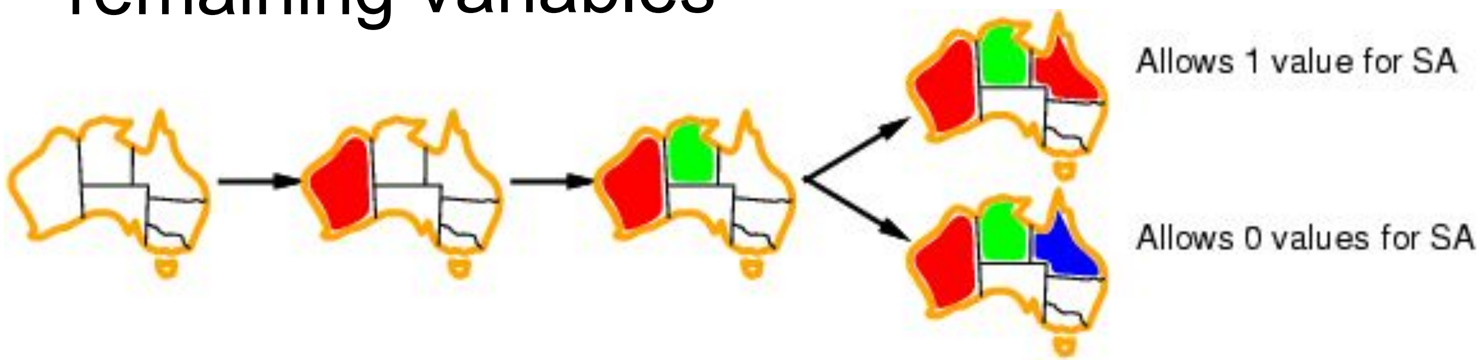
Most Constraining Variable

- Tie-breaker among most constrained variables
- Most constraining variable:
  - choose the variable with the most constraints on remaining variables



Least Constraining Value

- Given a variable, choose the least constraining value:
  - the one that rules out the fewest values in the remaining variables

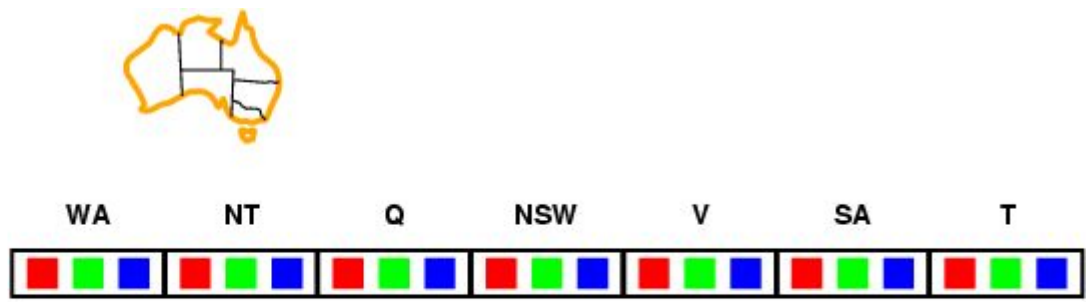


- Combining these heuristics makes 1000 queens feasible

How do we incorporate forward checking into a backtracking depth-first search?

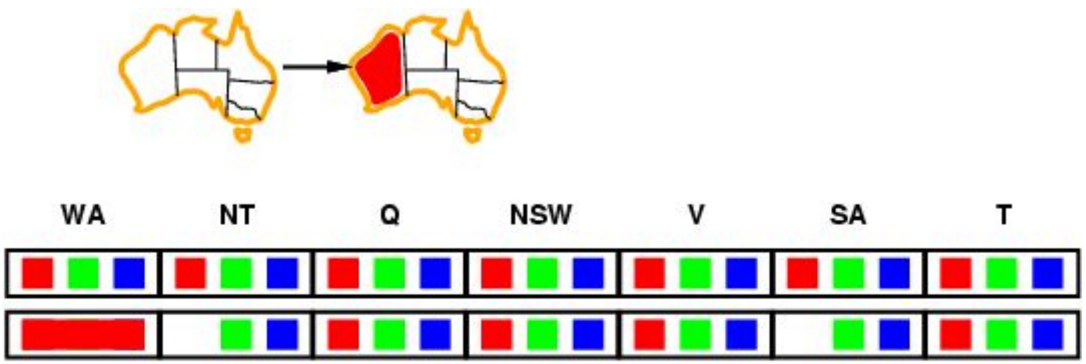
Book’s Forward Checking Example

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



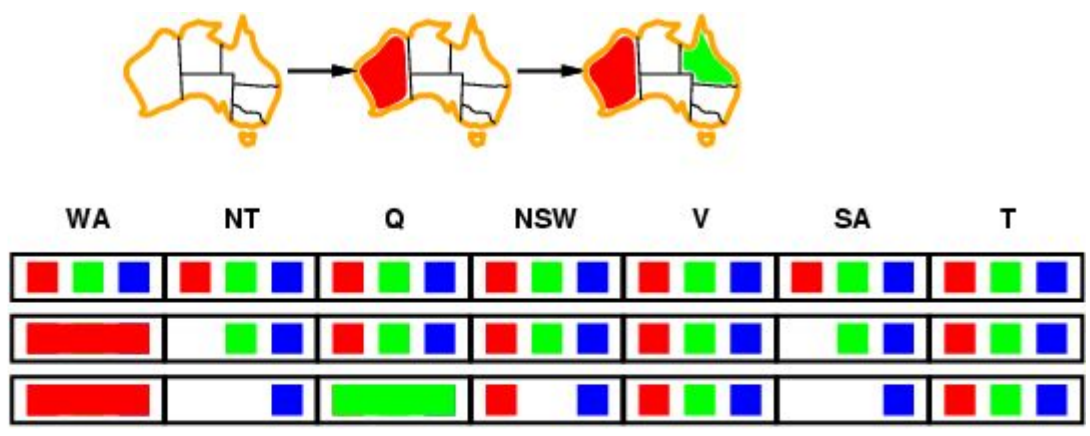
Forward Checking

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



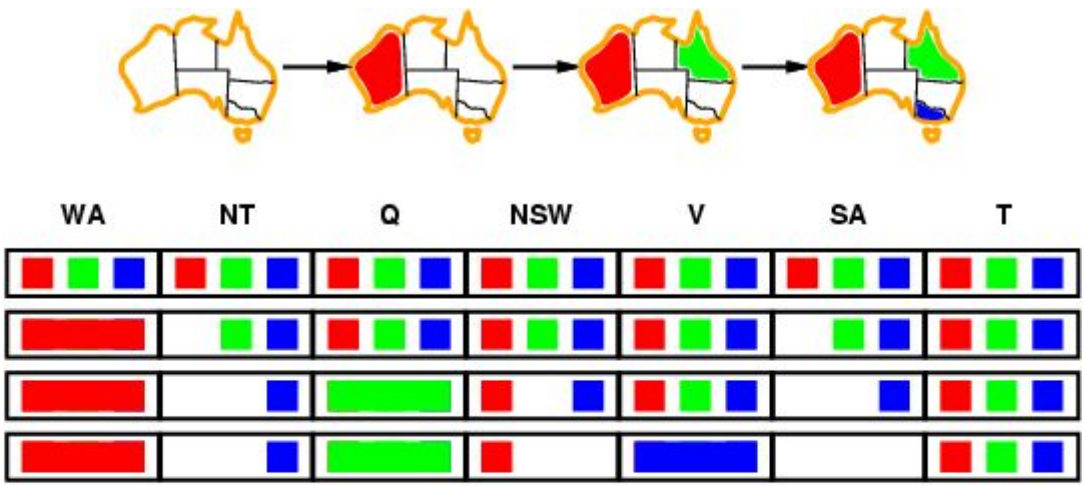
Forward Checking

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



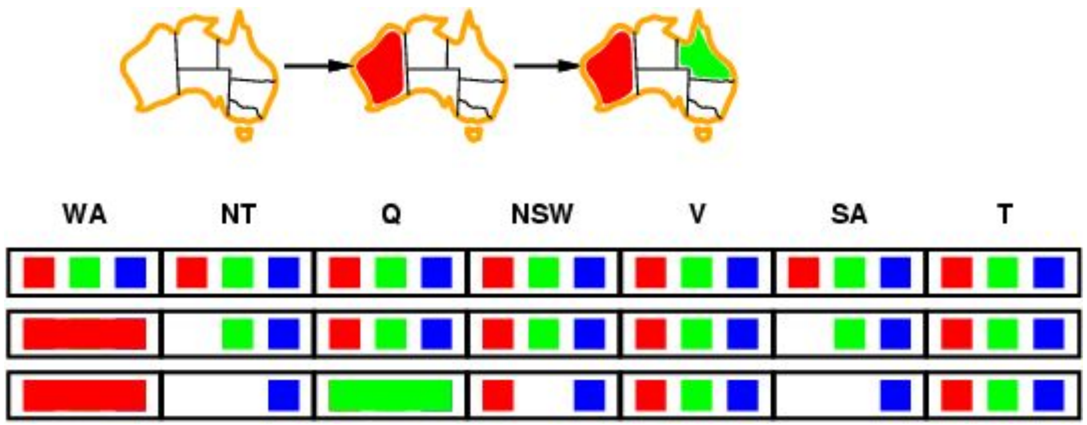
Forward Checking

- 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:

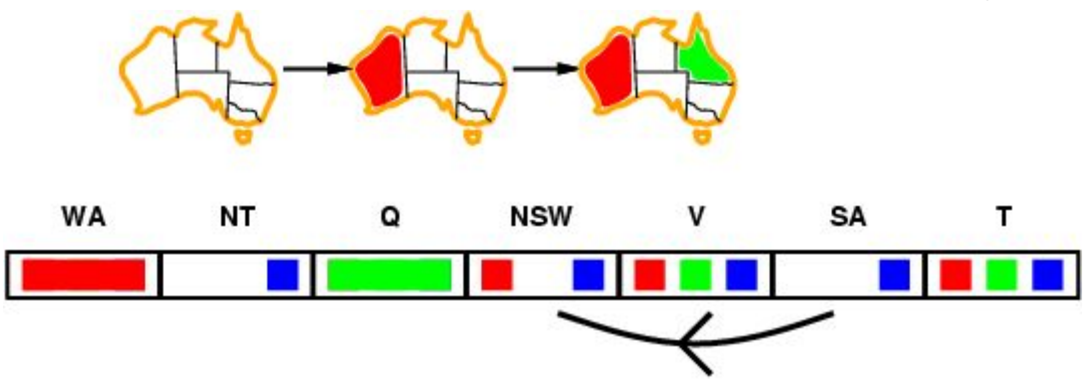


- NT and SA cannot both be blue!
- Constraint propagation repeatedly enforces constraints locally

Arc Consistency

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

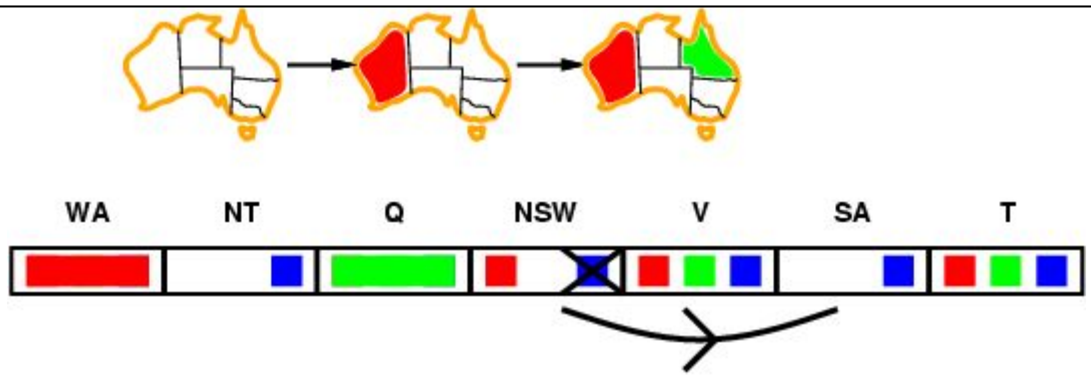
  for every value *x* of *X* there is some allowed value y of *Y*



Arc Consistency

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

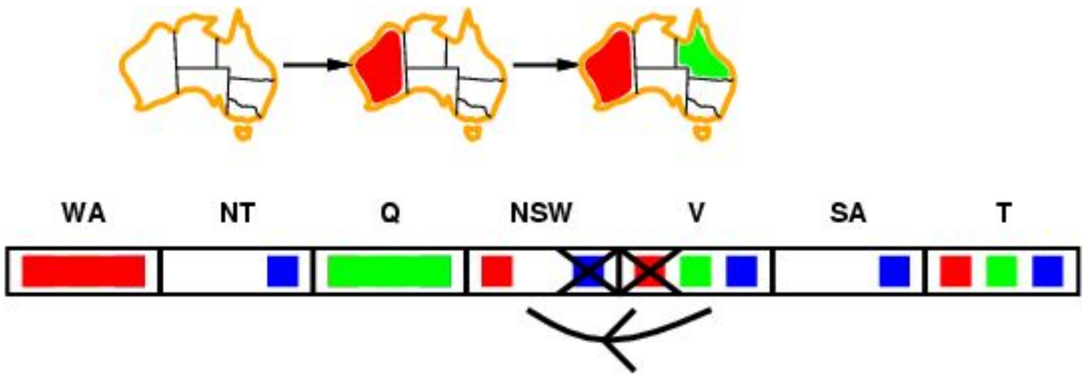
  for every value *x* of *X* there is some allowed value *y of Y*



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

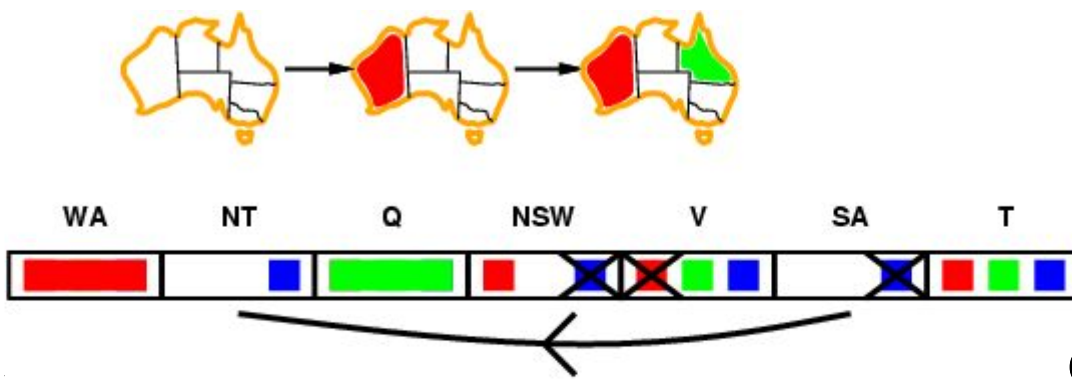


- If *X* loses a value, neighbors of *X* need to be rechecked

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 value, neighbors of *X* need to be rechecked
- Arc consistency detects failure earlier than forward checking
- Can be run as a preprocessor or after each assignment

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?