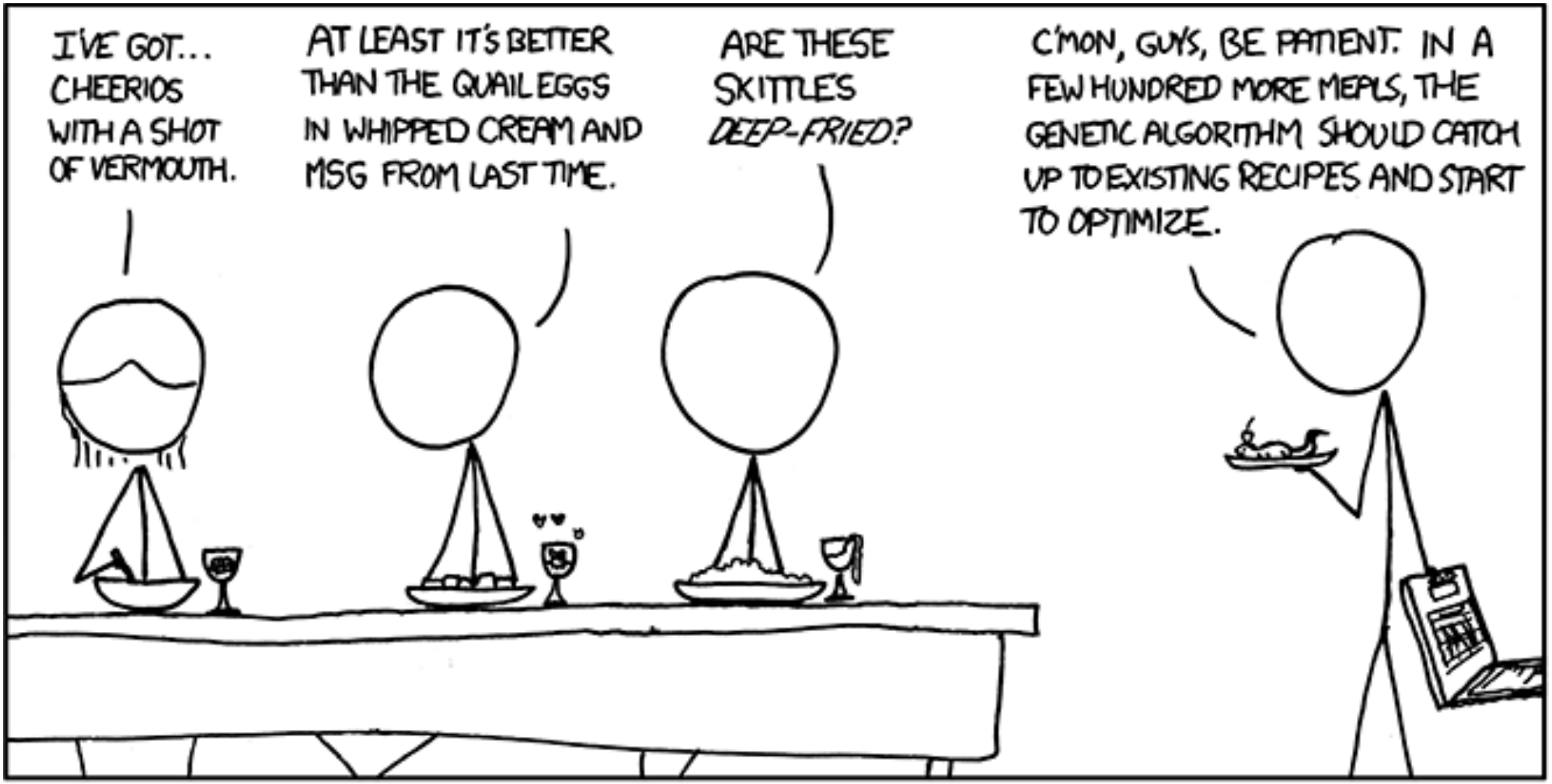
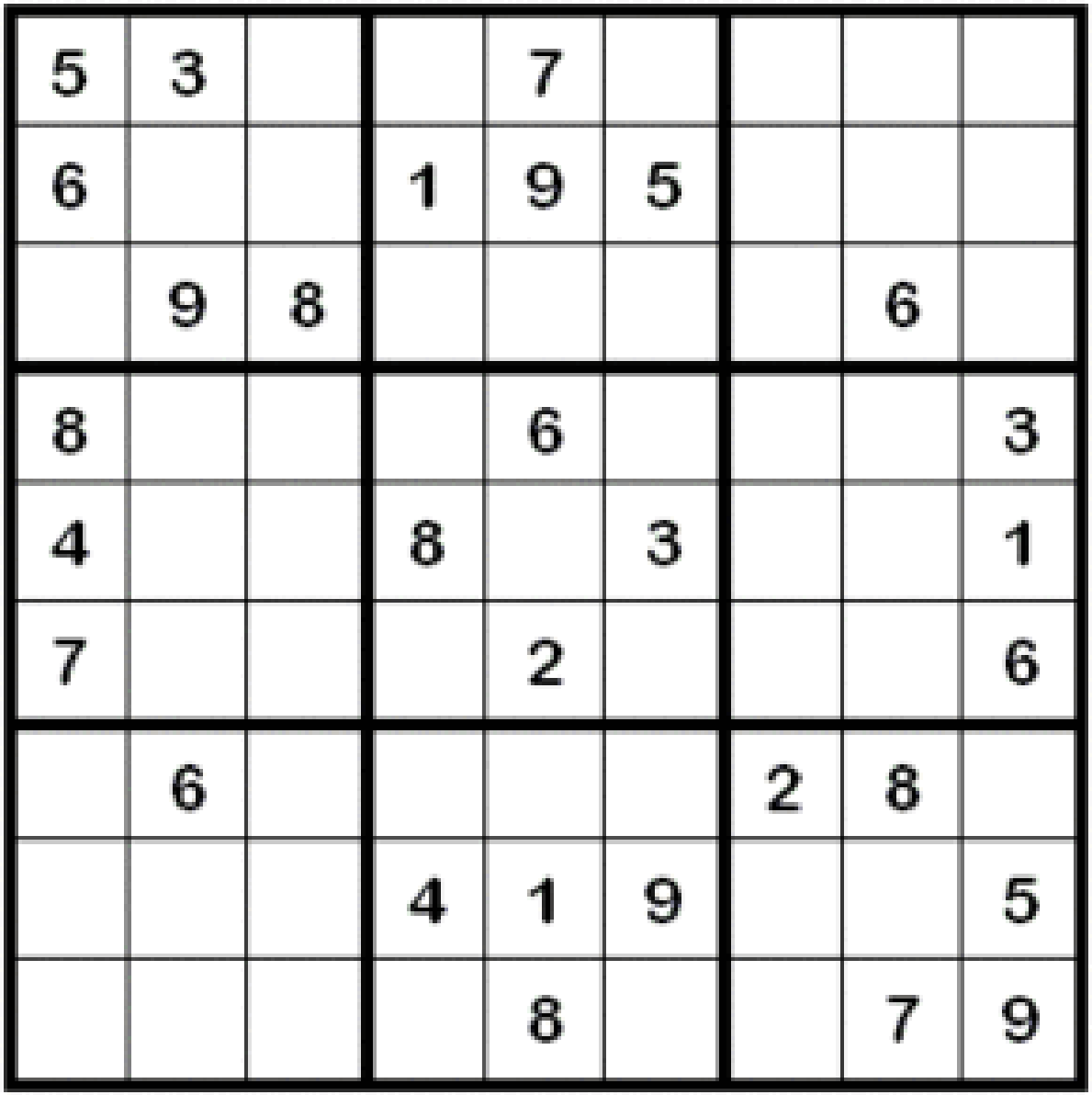


WE'VE DECIDED TO DROP THE CS DEPARTMENT FROM OUR WEEKLY DINNER PARTY HOSTING ROTATION.

Constraint Satisfaction Problems

Toolbox so far

- Uninformed search
  - BFS, DFS, Iterative deepening DFS, Uniform cost search
- Heuristic search
  - A*
- Local search
  - Hill climbing, simulated annealing, genetic algorithms


5 3 7
6 1 9 5
9 8 6
8 6 3
4 8 3 1
7 2 6
6 2 8
4 1 9 5
8 7 9

Constraint Satisfaction Problems

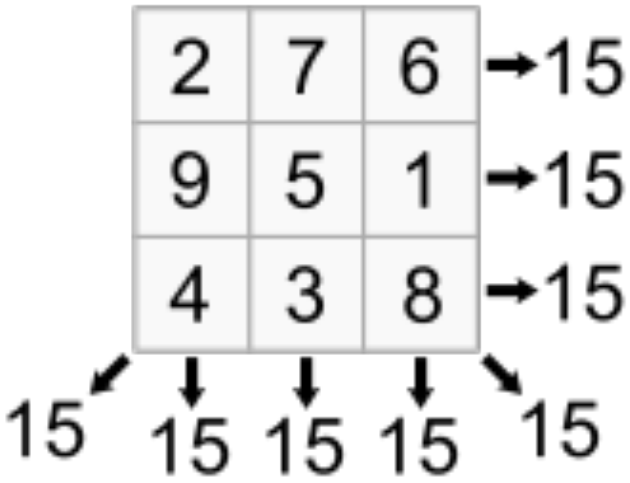
- Another variation on search.
- Requires a specific kind of problem (less general than heuristic or local search).
- Algorithms for solving CSPs can be very fast because they eliminate large areas of the search space at once.

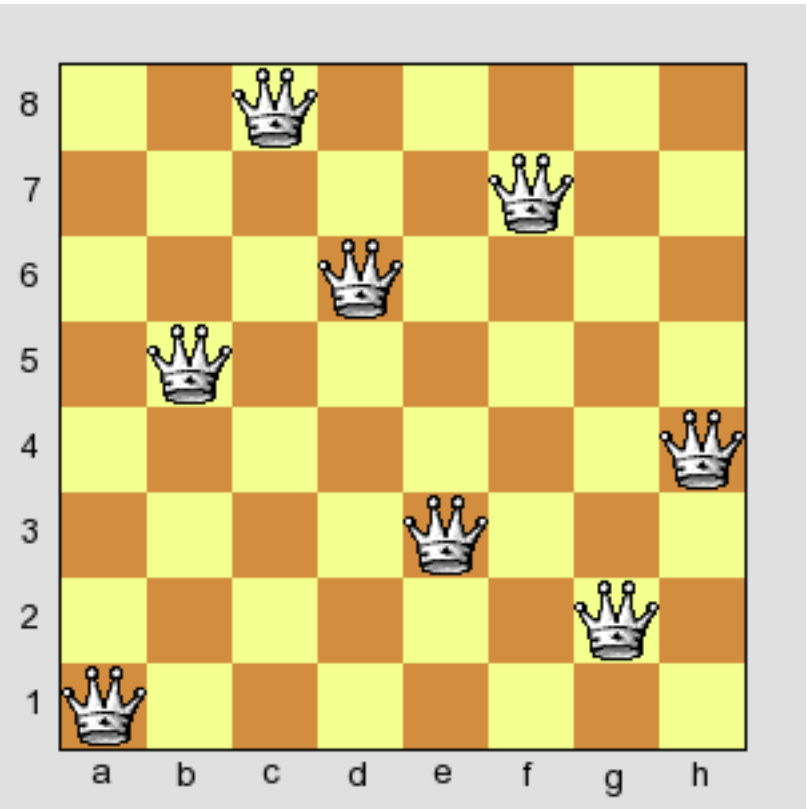
CSP Needs:

- A set of variables
- Each variable has a domain (finite and discrete)
  - Some CSPs use infinite domains.
- Set of constraints that specify what combinations of variables are legal.
  - Each constraint is a mathematical relation that must be satisfied.
  - OK, you don't like relations…think of constraints as boolean functions on the variables.

Constraints

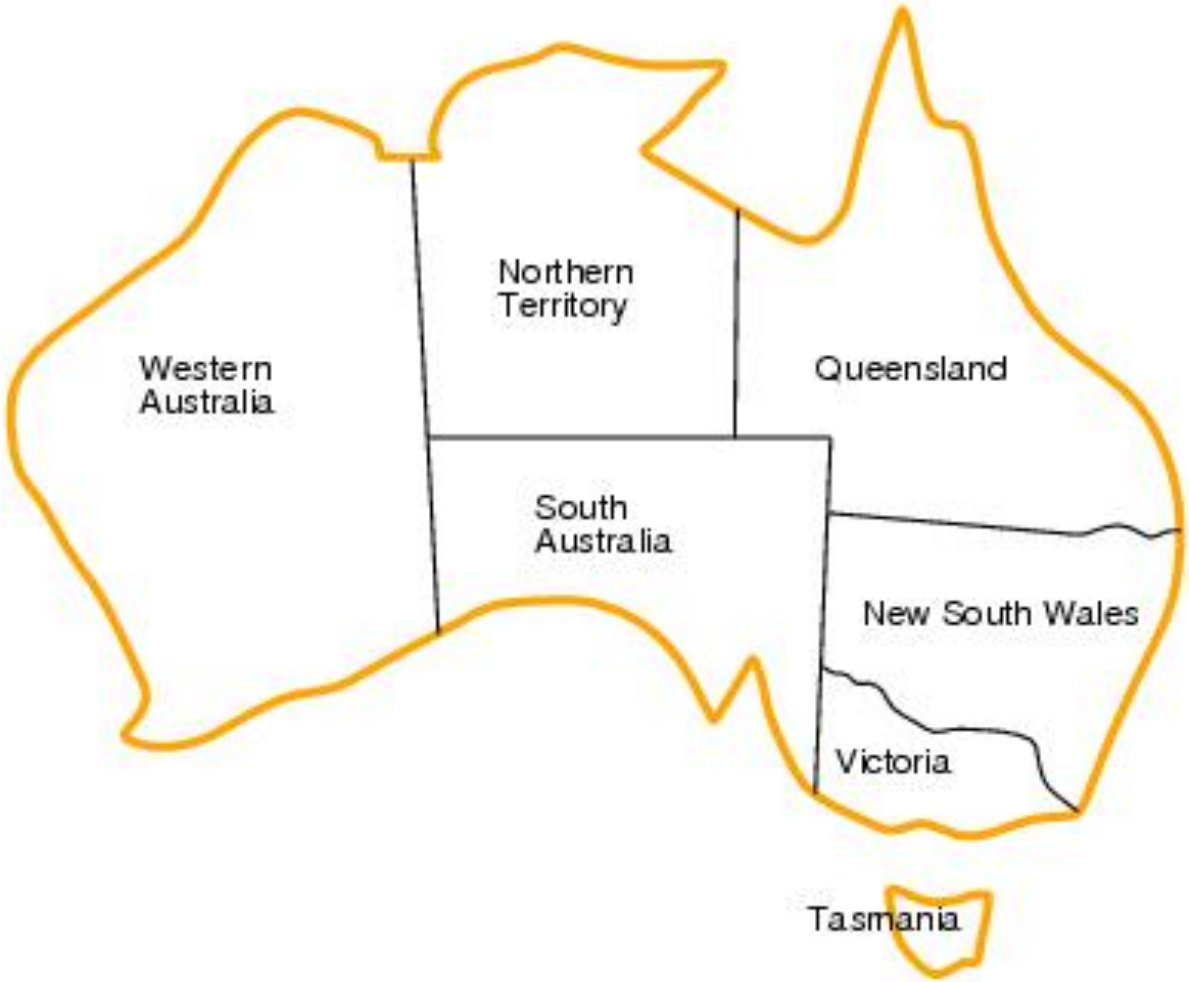
- Unary
- Binary
- Larger (or global)
- Preference constraints (not discussed)


2 7 6 →15
9 5 1 →15
4 3 8 →15
15 15 15 15 15

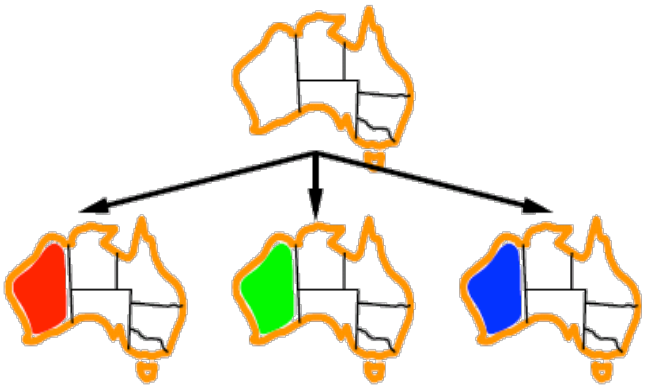

8
7
6
5
4
3
2
1
a b c d e f g h

Solving a CSP is still search!

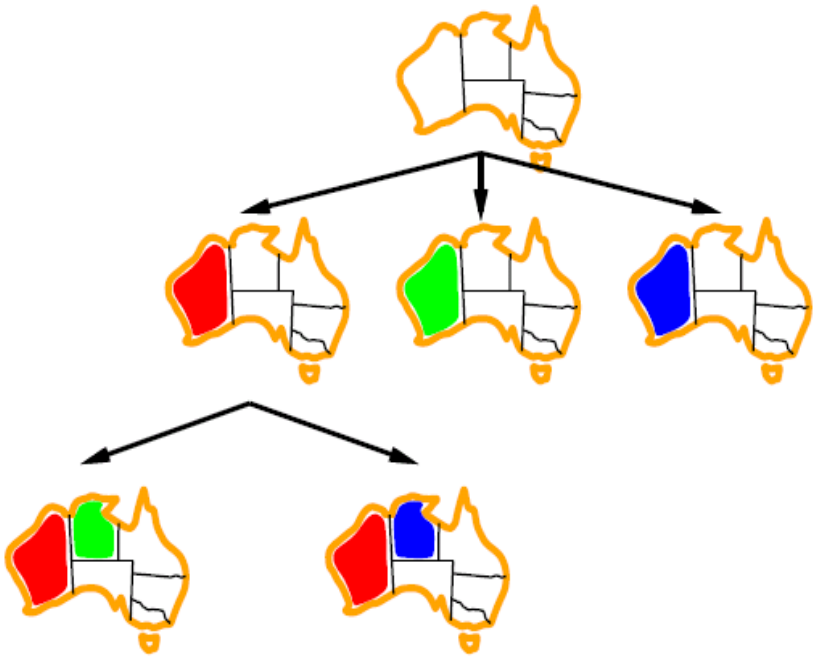
- Each state is a (possibly partial) set of variable assignments.
- Goal state is any complete set of variable assignments that satisfies all the constraints.


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

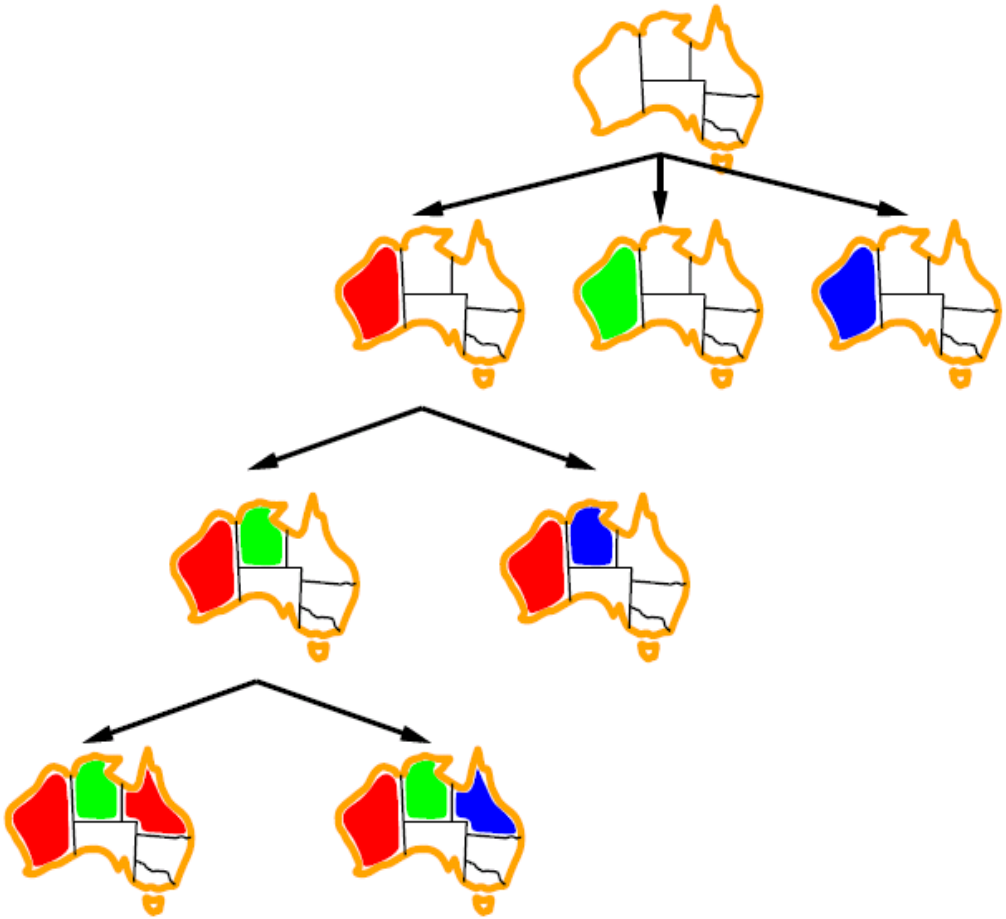
Search example



Search example



Search example



…but a different kind of search

- Doesn't use a "global" heuristic function.
- Algorithms are often fast because once a constraint is violated, any further search from that state is pointless.
- When a constraint is violated, we know exactly which variables have bad values and should be changed.

CSP search

- What is the start state?
- What is the goal?
- What are the actions? (i.e., how do we generate successor states?)
- What kind of search algorithm can we use?

Reducing the search space size

- Constraint propagation: Reducing the number of possible values for a variable.
  - Can happen before or during search.
- First we need a constraint graph.
  - Binary constraints only!

- Arc consistency
  - A variable Xi is arc consistent wrt Xj if for every value in Di there is some value in Dj that satisfies the constraint on arc (Xi, Xj).
- AC-3 Algorithm:
  - Add all arcs in graph to queue.
  - For arc (Xi, Xj), remove any impossible values in Di.
  - If Di changed, add all arcs (Xk, Xi) to queue. ($k \neq j$)

```
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) ← 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 ← 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 ← true
  return revised
```

Figure 6.3 The arc-consistency algorithm AC-3. After applying AC-3, either every arc is arc-consistent, or some variable has an empty domain, indicating that the CSP cannot be solved. The name "AC-3" was used by the algorithm's inventor (?) because it's the third version developed in the paper.

Using AC-3

- Sometimes AC-3 solves a CSP all on its own, without any search at all.
- If it doesn't, we can use backtracking search to find a solution.
  - Variation on DFS that backtracks whenever a variable has no legal values left to assign.

Backtracking search

- How do we pick a variable to assign to?
  - Minimum remaining values heuristic
  - Degree heuristic
- How do we pick a value to assign to the variable?
  - Least constraining value heuristic