Constraint Satisfaction Problems

- **N variables**
- **domain D**
- **constraints**

- **states**
- **goal test**
- **successor function**

Backtracking Search

- Backtracking search is the basic uninformed algorithm for solving CSPs
  - **Idea 1:** One variable at a time
    - Variable assignments are commutative, so fix ordering
    - i.e., [WA = red then NT = green] same as [NT = green then WA = red]
  - **Idea 2:** Check constraints as you go
    - i.e., consider only values which do not conflict previous assignments
    - Might have to do some computation to check the constraints
      - "Incremental goal test"
  - Depth-first search with these two improvements is called backtracking search (not the best name)
  - Can solve n-queens for n = 25

Backtracking Example

- Backtracking = DFS + variable-ordering + fail-on-violation

Filtering

- General-purpose ideas give huge gains in speed
- **Ordering:**
  - Which variable should be assigned next?
  - In what order should its values be tried?
- **Filtering:** Can we detect inevitable failure early?
- **Structure:** Can we exploit the problem structure?
Filtering: 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!
  - Why didn’t we detect this yet?
  - Constraint propagation: reason from constraint to constraint

Consistency of A Single Arc

- An arc $X \rightarrow Y$ is consistent if for every $x$ in the tail there is some $y$ in the head which could be assigned without violating a constraint:

Arc Consistency of an Entire CSP

- A simple form of propagation makes sure all arcs are consistent:
  - Important: 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
  - What’s the downside of enforcing arc consistency?

Enforcing Arc Consistency in a CSP

- After enforcing arc consistency:
  - Can have one solution left
  - Can have multiple solutions left
  - Can have no solutions left (and not know it)
  - Arc consistency still runs inside a backtracking search!

Limitations of Arc Consistency

- Runtime $O(n^3)$, can be reduced to $O(n^2)$
- but detecting all possible future problems is NP-hard - why?

Video of Demo Coloring – Backtracking with Forward Checking – Complex Graph

Video of Demo Coloring – Backtracking with Arc Consistency – Complex Graph

Video of Demo Coloring – Backtracking with Forward Checking

Video of Demo Coloring – Backtracking with Arc Consistency