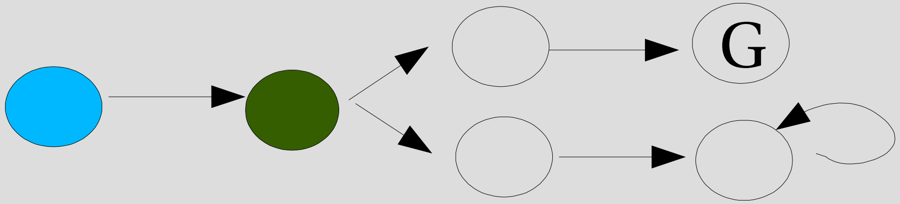
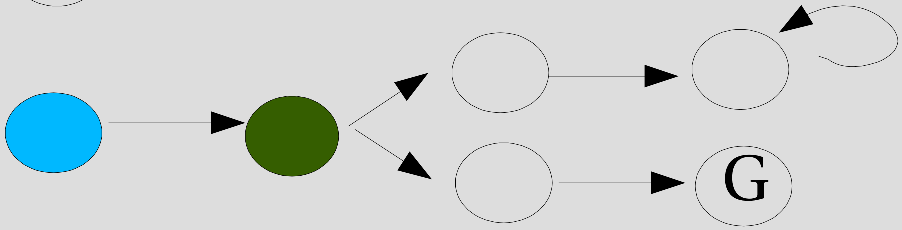
CSC421 Intro to Artificial Intelligence

UNIT 05:
Constraint Satisfaction Problems

Online search problems

- Interleave computation & execution
- Exploration problems (robot placed on new planet go from A to B)
- Competitive ratio (can be infinite)
  - Actual cost compared to the cost of the path the agent would follow *if it "knew" the search space in advance*

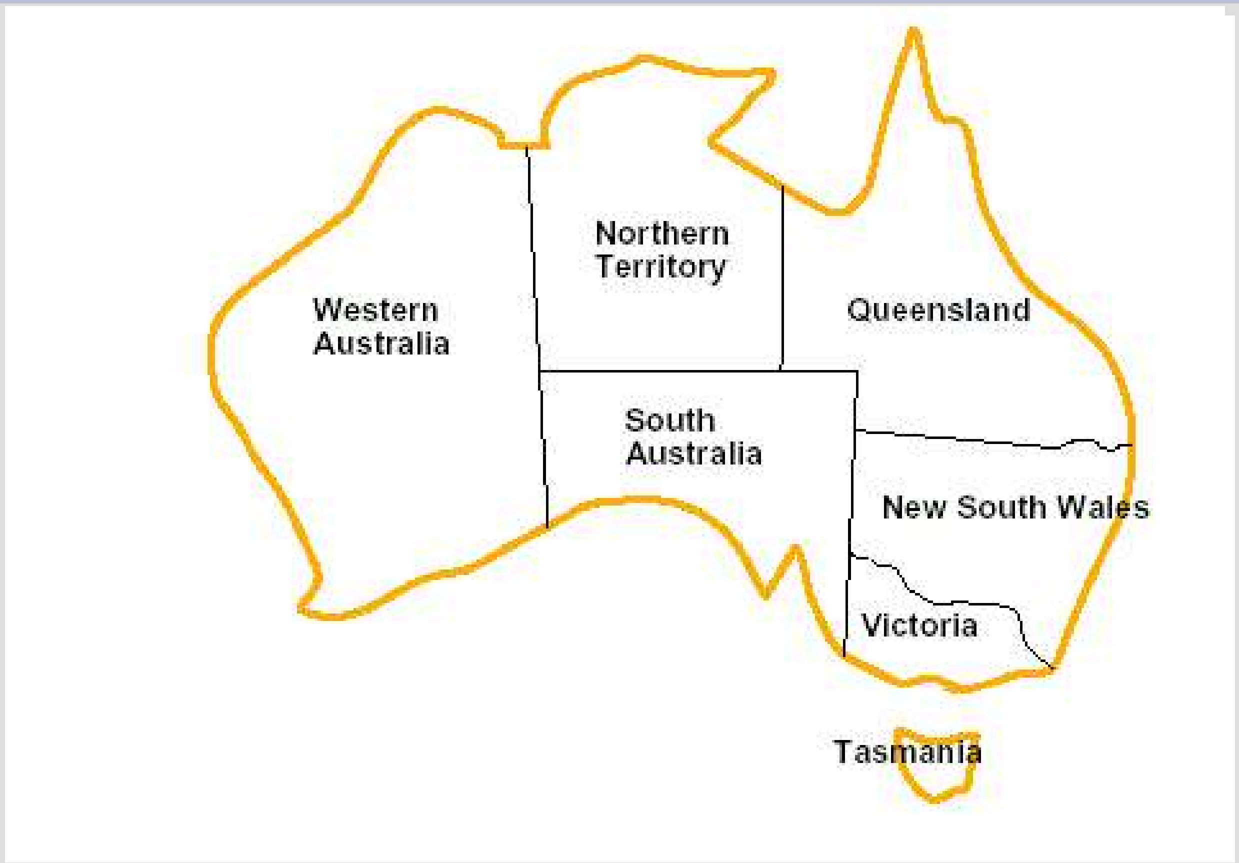




Constraint Satisfaction Problems (CSPs)

- Standard search problem
  - State is a “black box” - any data structure as long as it has goal test, eval, succesor
- CSP (more structured representation but still quite general)
  - State is defined by variables X_i with values from domain D_i
  - Goal test is is a set of constraints specifying allowable combinations of values for subsets of the variables
- Simple example of formal representation
- Allows useful general purpose algorithms with more power than standard search

Example: Map coloring



VARIABLES: WA, NT, Q, NSW, V, SA, T
Domains : {red, green, blue}

Constraints: adjacent regions must have different colors

WA $\neq$ NT (if the language allows it)
or
(WA, NT) $\in$ {(red, green), (red, blue), (blue, green), (blue, red), (green, red), (green, blue)}

Historical Sidenotes

- 4-colour conjecture (any planar graph can be colored with four or fewer colors)
- Probably first made by Francis Guthrie, student of De Morgan in 1852
- Despite efforts first proof in 1977 by Appel and Haken (with computer aid)
- Influential early example
  - SketchPad by Sutherland 1963
  - Forerunner of pointer/display interaction, CAD, etc

More map coloring

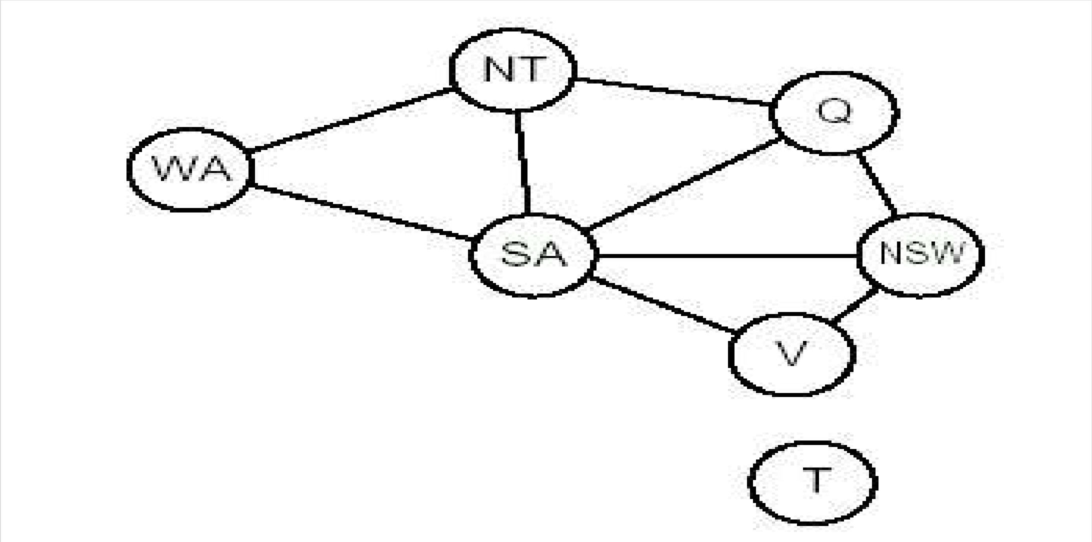
How would you formulate as a "standard" search problem ?

Other examples of CSPs ?

Solutions are assignments that satisfy all the constraints
{WA = red, NT = green, SA = blue, Q = red, NS = green,
V = red, T = green}

Constraint Graph

- Binary CSP: each constraint relates at most two variables
- Constraint graph: nodes are variables, arcs show constrains
- General purpose CSP use the graph structure to speed up search
- For example ?



Varieties of CSP

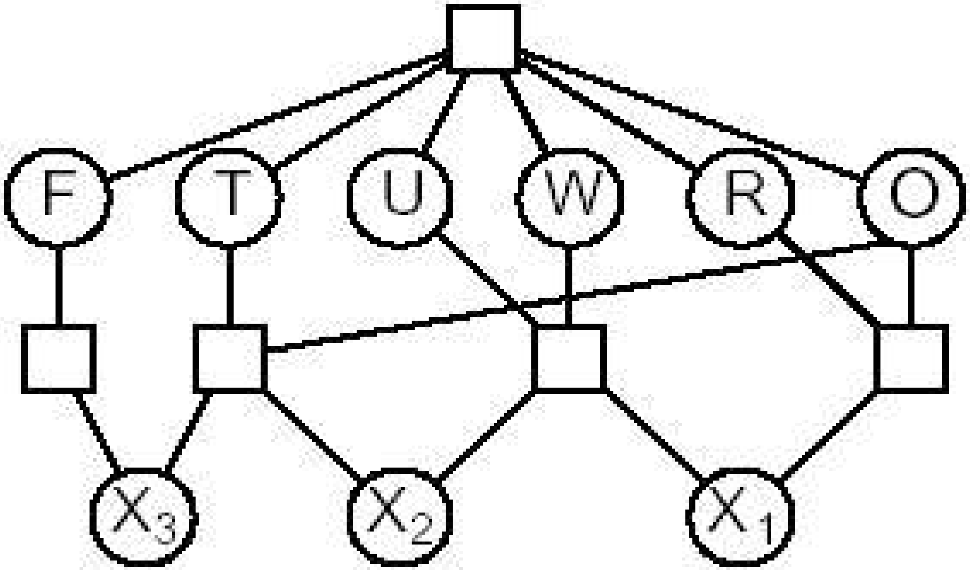
- Discrete variables
  - Finite domains; size d => $O(d^n)$ complete assignments (Boolean CSP, incl. Boolean satisfiability (NP-complete))
  - Infinite domains (integers, strings)
    - e.g job scheduling (start, end days for each job)
    - Need a constraint language e.g., startJob1 + 5 <= startJob2
    - Linear constraints solvable, non-linear undecidable
- Continuous variables
  - start/end time for Hubble Telescope observations
  - Linear constraints solvable in polynomial time by Linear Programming methods

Varieties of constraints

- Unary constraints involve a single variable
  - e.g, SA $\neq$ green
- Binary constraints involve pairs of variables
  - e.g SA $\neq$ WA
- Higher-order constraints involve 3 or more variables e.g. Cryptarithmetic column constraints
- Preferences (soft constraints) e.g. Red is better than green often represented by a cost for each variable assignment
  - Constraint optimization problems

Cryptarithmetic

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



Variables: F T U W R O X_1 X_2 X_3

Domains: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Constraints:

alldiff(F, T, U, W, R, O)

O + O = R + X1 etc

Real world CSP

- Assignment problem
  - e.g. Who teaches what class
- Timetabling problem
  - e.g. Which class is offered when and where
- Hardware configuration
- Spreadsheets
- Transportation scheduling
- Factory scheduling
- Floor planning
- Most real-world problems involve real-valued variables

Exercise 5.11

- Prove that every higher-order, finit-domain constrain can be reduced to a set of binary constraints if enough auxiliary variables are introduced