

Advanced Topics in AI Constraint Satisfaction Problems



Instructor: Prof. Dr. techn. Wolfgang Nejdl

Leibniz University Hannover

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All materials are available at <http://ai.berkeley.edu>.]



Co-financed by the Connecting Europe
Facility of the European Union

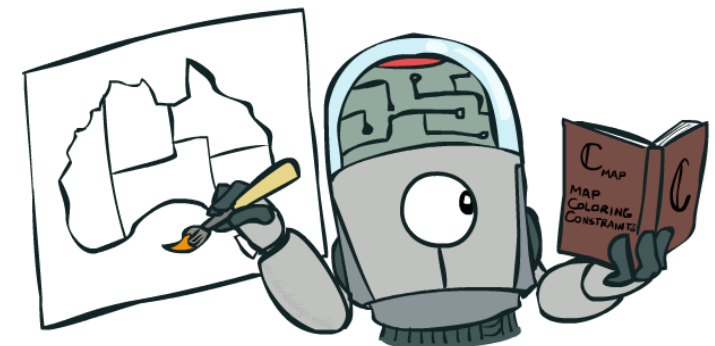
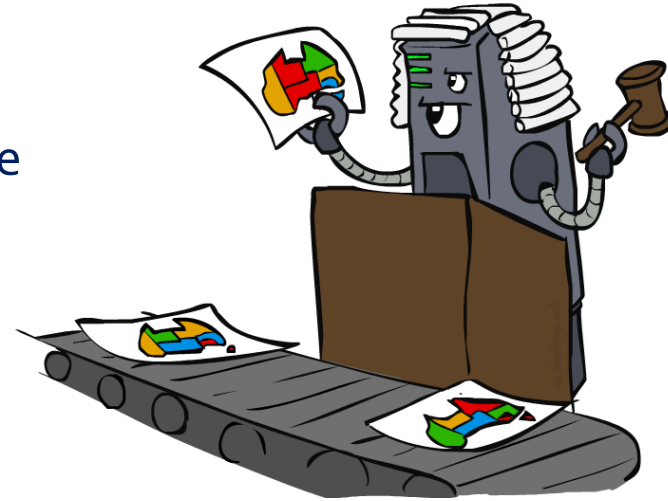
What is Search For?

- Assumptions about the world: a single agent, deterministic actions, fully observed state, discrete state space
- Planning: sequences of actions
 - The path to the goal is the important thing
 - Paths have various costs, depths
 - Heuristics give problem-specific guidance
- Identification: assignments to variables
 - The goal itself is important, not the path
 - All paths at the same depth (for some formulations)
 - CSPs are specialized for identification problems

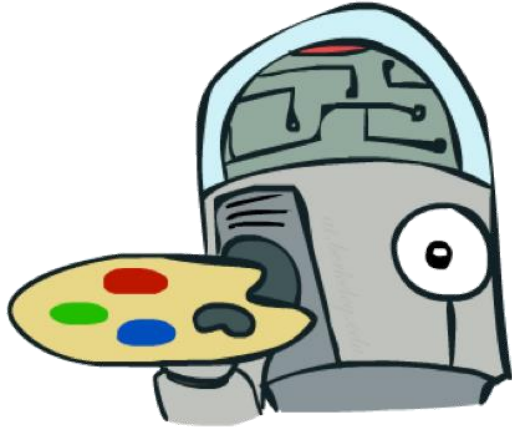


Constraint Satisfaction Problems

- Standard search problems:
 - State is a “black box”: arbitrary data structure
 - Goal test can be any function over states
 - Successor function can also be anything
- Constraint satisfaction problems (CSPs):
 - A special subset of search problems
 - State is defined by variables X_i with values from a domain D (sometimes D depends on i)
 - Goal test is a set of constraints specifying allowable combinations of values for subsets of variables
- Allows useful general-purpose algorithms with more power than standard search algorithms



Varieties of CSPs



- Discrete Variables
 - Finite domains
 - Size d means $O(d^n)$ complete assignments
 - E.g., Boolean CSPs, including Boolean satisfiability (NP-complete)
 - Infinite domains (integers, strings, etc.)
 - E.g., job scheduling, variables are start/end times for each job
 - Linear constraints solvable, nonlinear undecidable



- Continuous variables
 - E.g., start/end times for Hubble Telescope observations
 - Linear constraints solvable in polynomial time by LP methods

Varieties of Constraints

- Varieties of Constraints

- Unary constraints involve a single variable (equivalent to reducing domains), e.g.:

$$SA \neq green$$

- Binary constraints involve pairs of variables, e.g.:

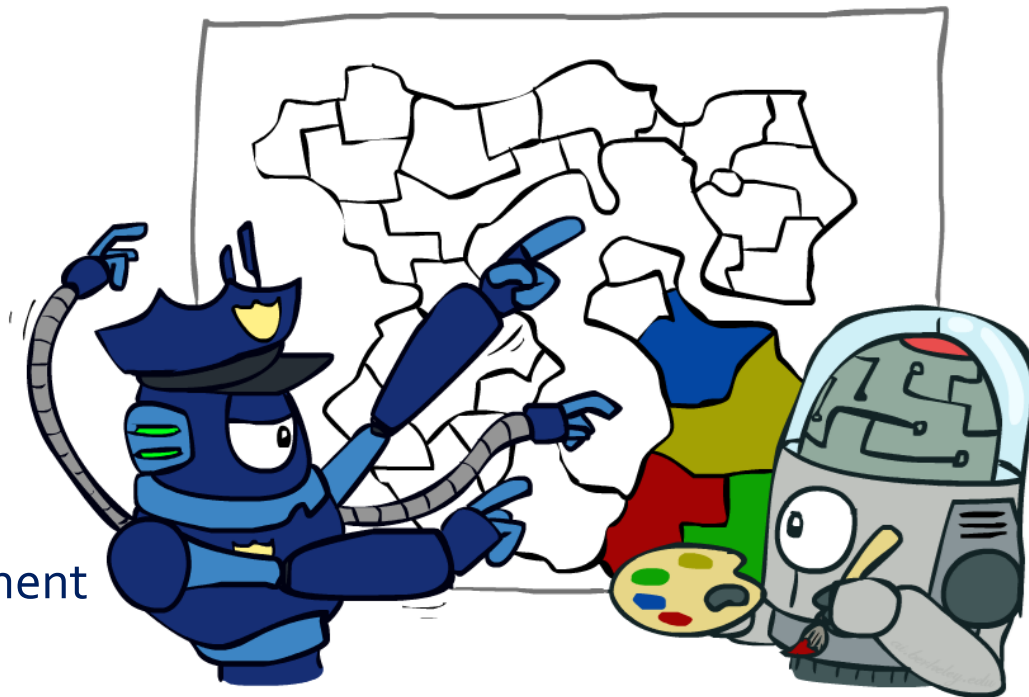
$$SA \neq WA$$

- Higher-order constraints involve 3 or more variables:

e.g., cryptarithmic column constraints

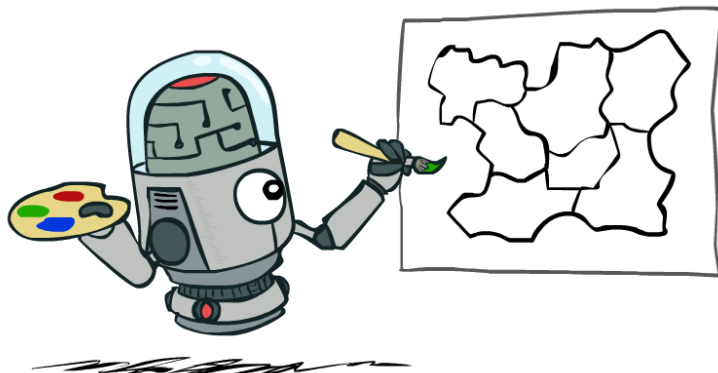
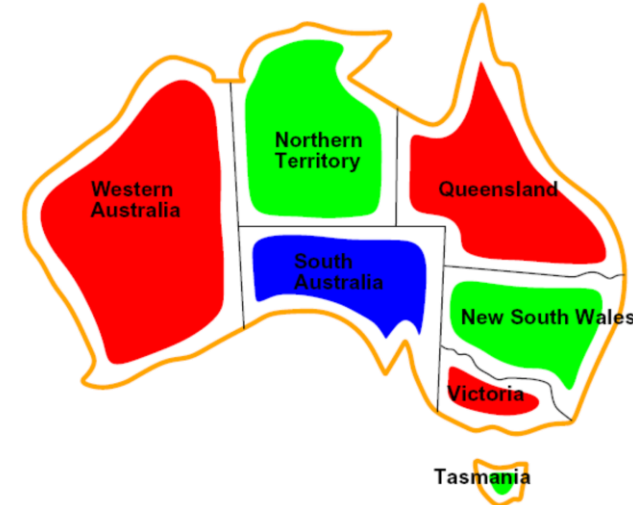
- Preferences (soft constraints):

- E.g., red is better than green
- Often representable by a cost for each variable assignment
- Gives constrained optimization problems
- (We'll ignore these)



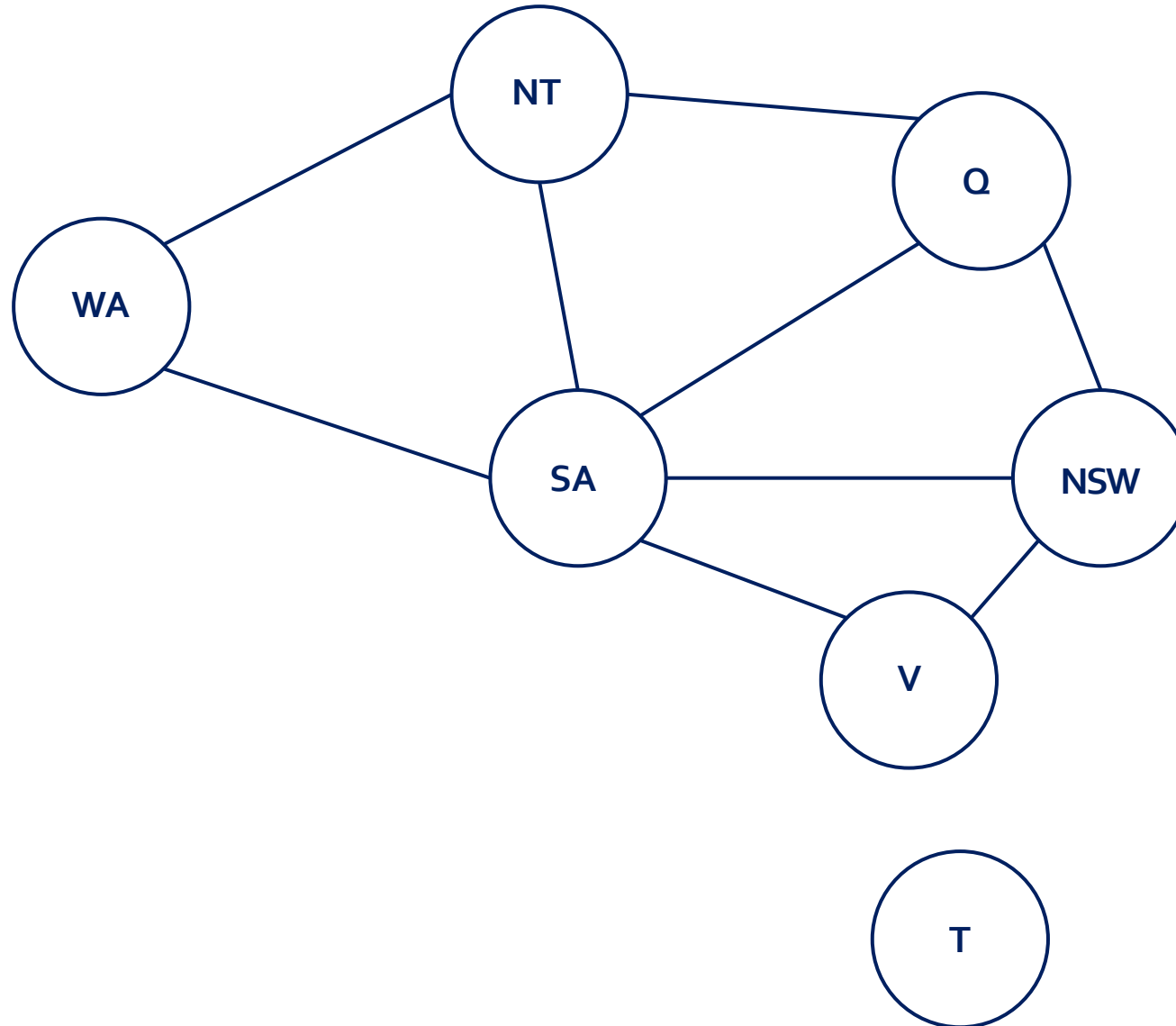
Example: Map Coloring

- Variables:
 - WA, NT, Q, NSW, V, SA, T
- Domains:
 - $D = \{red, green, blue\}$
- Constraints:
 - adjacent regions must have different colors
 - Implicit: $WA \neq NT$
 - Explicit: $(WA, NT) \in \{(red, green), (red, blue), \dots\}$



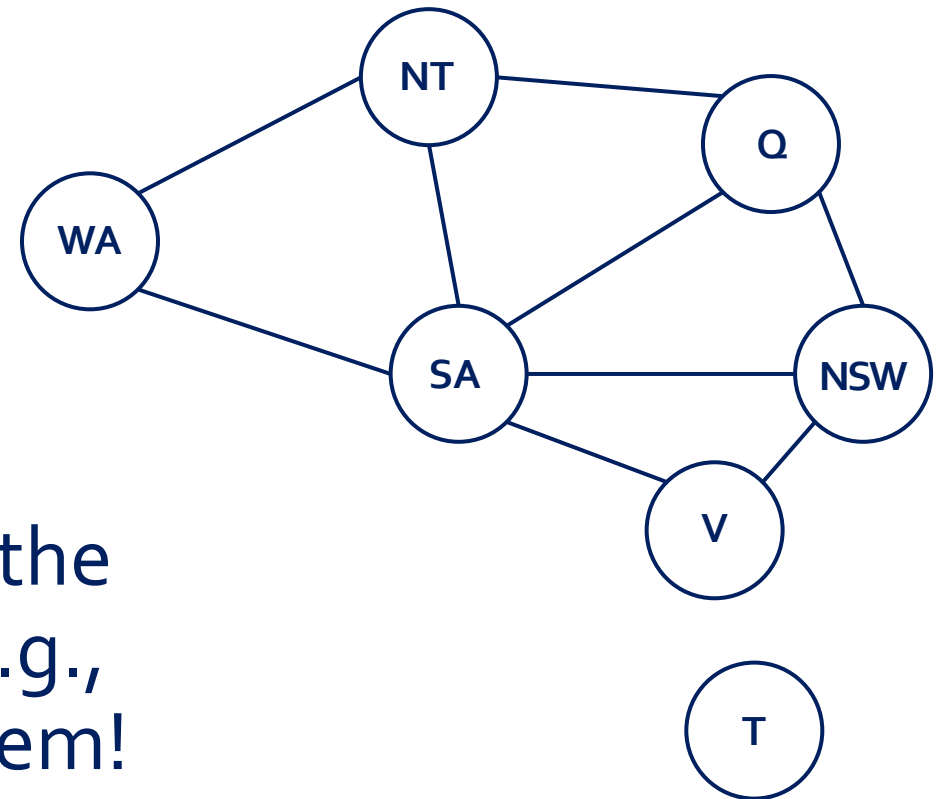
- Solutions are assignments satisfying all constraints, e.g.:
 - $\{WA = red, NT = green, Q = red, NSW = green, V = red, SA = Blue, T = green\}$

Constraint Graphs



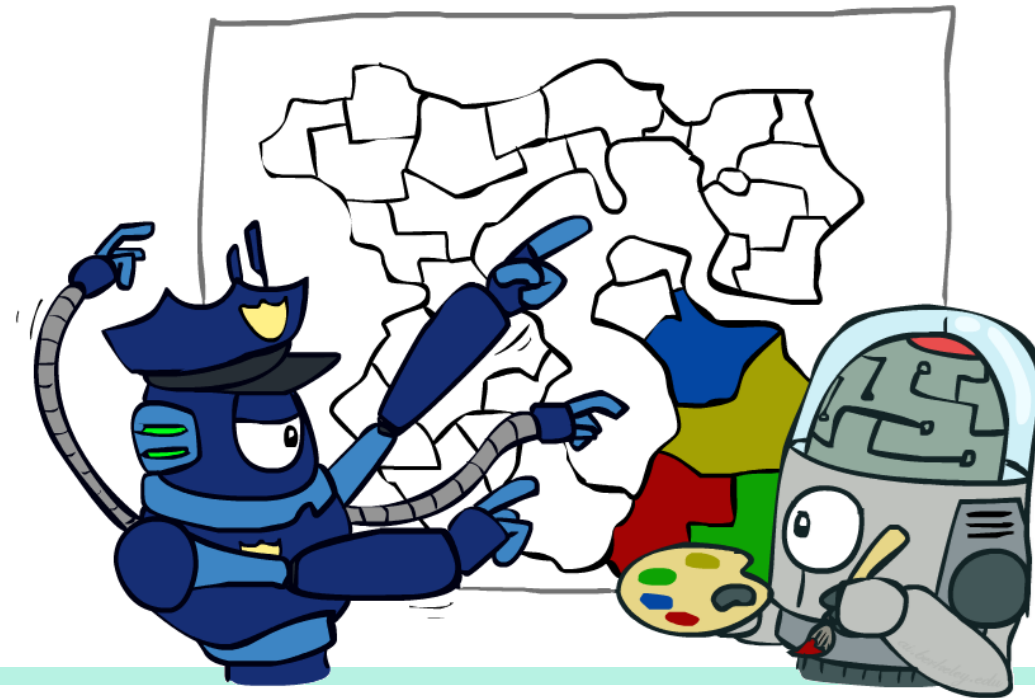
Constraint Graphs

- Binary CSP: each constraint relates (at most) two variables
- Binary constraint graph: nodes are variables, arcs show constraints
- General-purpose CSP algorithms use the graph structure to speed up search. E.g., Tasmania is an independent subproblem!



Advanced Topics in AI

Next: Examples of CSPs



Instructor: Prof. Dr. techn. Wolfgang Nejdl

Leibniz University Hannover



[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All materials are available at <http://ai.berkeley.edu>.]



Co-financed by the Connecting Europe
Facility of the European Union