

# Constraint Satisfaction

CS5491: Artificial Intelligence  
ZHICHAO LU

Content Credits: Prof. Wei's CS4486 Course  
and Prof. Boddeti's AI Course

# WHAT ARE CONSTRAINT SATISFACTION PROBLEMS?

---

Special kind of search problems.

- ›  $N$  variables
- › Values from domain  $D$
- › assignment satisfies constraints

**States:** partial assignment

**Goal Test:** satisfies all constraints

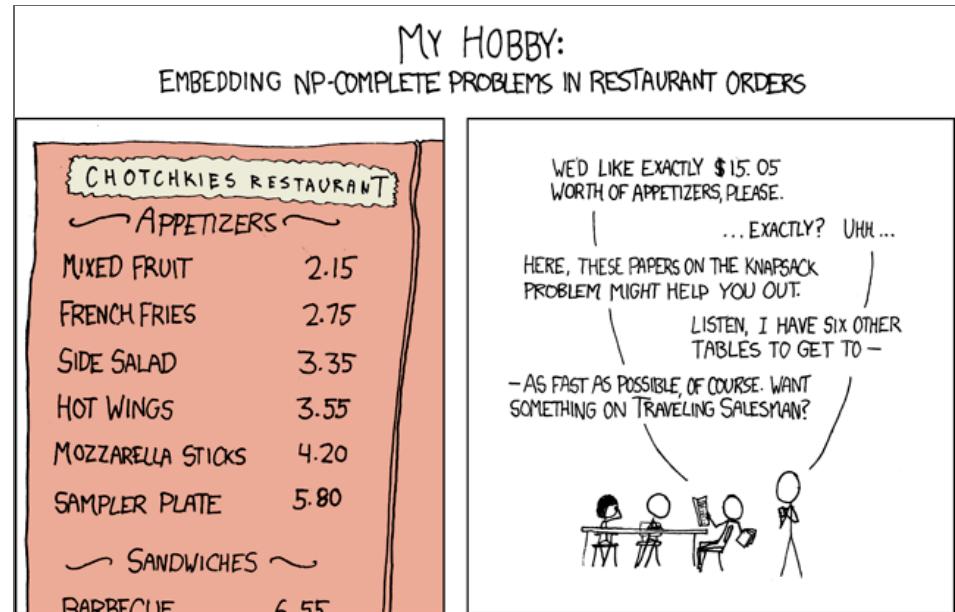
**Successor Function:** assign an unassigned variable

# TODAY

## Constraint Satisfaction

## Reading

- Today's Lecture: RN Chapter 6



XKCD

## WHAT IS SEARCH FOR?

---

Given 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 a specialized class of identification problems

# CONSTRAINT SATISFACTION PROBLEMS

---

# WHAT ARE CONSTRAINT SATISFACTION PROBLEMS?

Standard search problems:

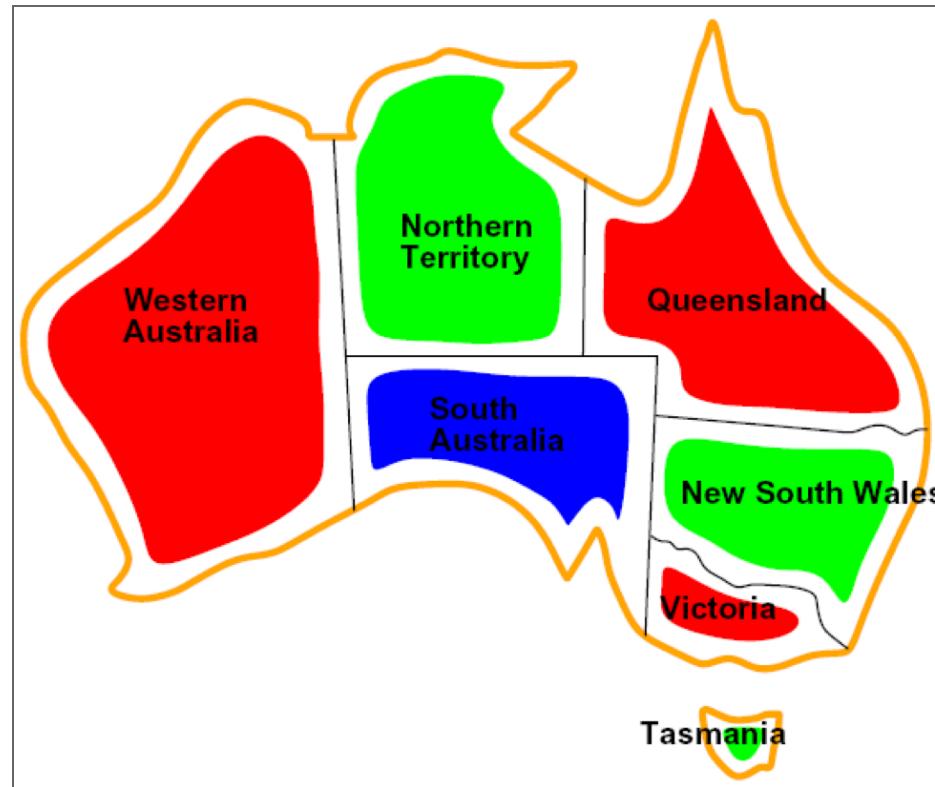
- › State is a “black box”: arbitrary data structure
- › Goal test can be any function over states
- › Successor function can also be anything

Constraint satisfaction problems (CSPs):

- › A special subset of search problems
- › State is defined by variables  $\mathbf{X}_i$  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**

# CSP EXAMPLE



## EXAMPLE: MAP COLORING

Variables:  $WA, NT, Q, NSW, V, SA, T$

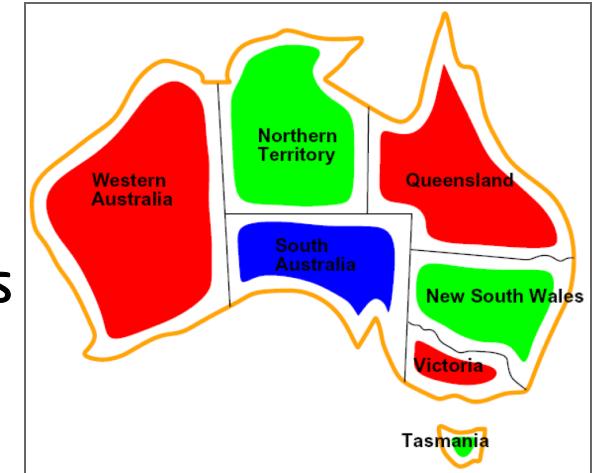
Domains:  $D = \{red, green, blue\}$

Constraints: adjacent regions must have different colors

- › Implicit:  $WA \neq NT$
- › Explicit:  $(WA, NT) \in (red, green), (red, blue), \dots$

Solutions: Assignments that satisfy all constraints, e.g.:

- ›  $\{WA=\text{red}, NT=\text{green}, Q=\text{red}, NSW=\text{green}, V=\text{red}, SA=\text{blue}, T=\text{green}\}$

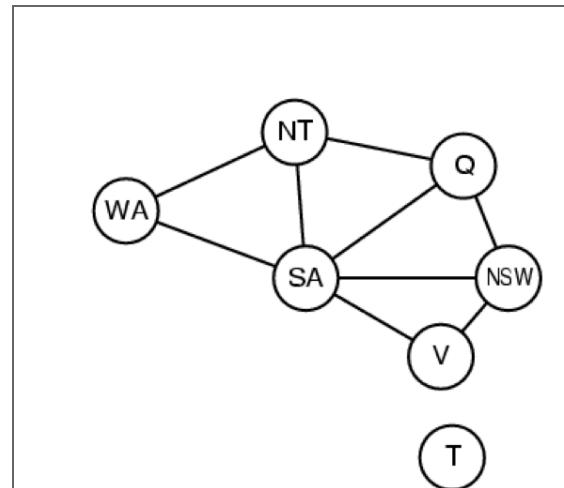


# CONSTRAINT GRAPHS

**Binary CSP:** each constraint relates two variables

**Constraint Graph:** nodes are variables, arcs are constraints

**General-purpose CSP algorithms use the graph structure to speed up search.** E.g., Tasmania is an independent subproblem.



# EXAMPLE: N QUEENS

## Formulation 1:

- Variables:  $X_{ij}$
- Domains:  $\{0, 1\}$
- Constraints:

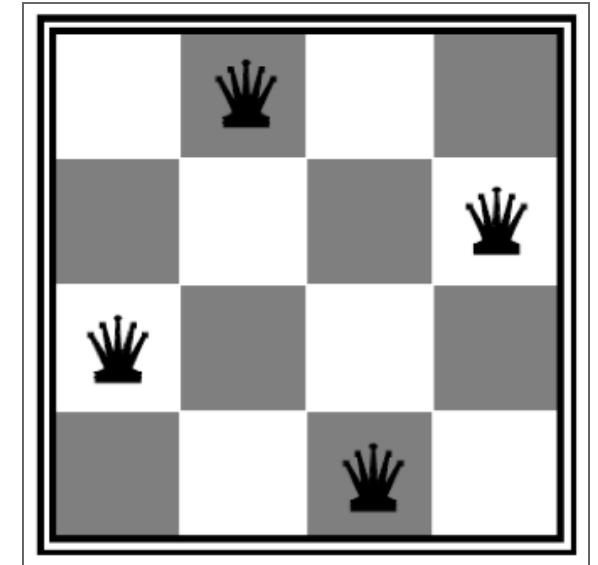
$$\forall i, j, k \quad (X_{ij}, X_{ik}) \in (0, 0), (0, 1), (1, 0)$$

$$\forall i, j, k \quad (X_{ij}, X_{kj}) \in (0, 0), (0, 1), (1, 0)$$

$$\forall i, j, k \quad (X_{ij}, X_{i+k, j+k}) \in (0, 0), (0, 1), (1, 0)$$

$$\forall i, j, k \quad (X_{ij}, X_{i+k, j-k}) \in (0, 0), (0, 1), (1, 0)$$

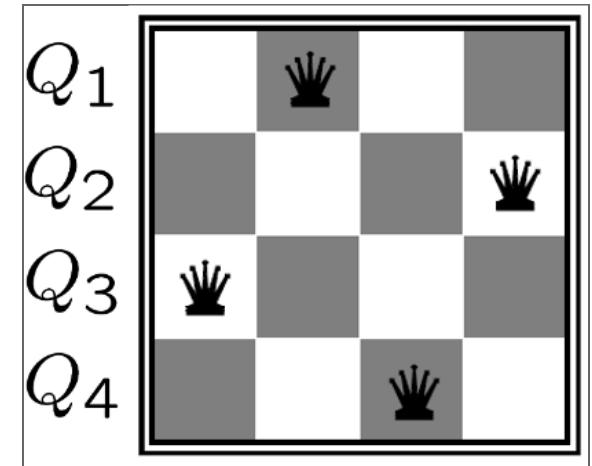
$$\sum_{ij} X_{ij} = N$$



## EXAMPLE: N QUEENS

### Formulation 2:

- Variables:  $Q_k$
- Domains:  $1, 2, 3, \dots, N$
- Constraints:
  - Implicit:  $\forall (i, j)$  non – threatening ( $Q_i, Q_j$ )
  - Explicit:  $(Q_1, Q_2) \in \{(1, 3), (1, 4), \dots\}$





# VARIETY 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., cryptarithmetic column constraints

## Preferences (soft constraints):

- › E.g., red is better than green
- › Often representable by a cost for each variable assignment
- › Leads to constrained optimization problems

## REAL-WORLD CSPS

---

Scheduling problems: e.g., when can we all meet?

Timetabling problems: e.g., which class is offered when and where?

Assignment problems: e.g., who teaches what class

Hardware configuration

Transportation scheduling

Factory scheduling

Circuit layout

Fault diagnosis

...

# SOLVING CSPS

---

## EXAMPLE: MAP COLORING

Variables:  $WA, NT, Q, NSW, V, SA, T$

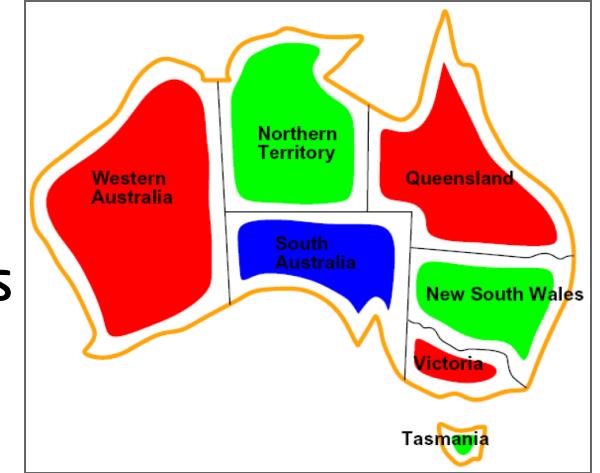
Domains:  $D = \{red, green, blue\}$

Constraints: adjacent regions must have different colors

- › Implicit:  $WA \neq NT$
- › Explicit:  $(WA, NT) \in (red, green), (red, blue), \dots$

Solutions: Assignments that satisfy all constraints, e.g.:

- ›  $\{WA=\text{red}, NT=\text{green}, Q=\text{red}, NSW=\text{green}, V=\text{red}, SA=\text{blue}, T=\text{green}\}$



# STANDARD SEARCH FORMULATION

---

Standard search formulation of CSPs

States defined by the values assigned so far (partial assignments)

- › Initial state: the empty assignment, {}
- › Successor function: assign a value to an unassigned variable
- › Goal test: the current assignment is complete and satisfies all constraints

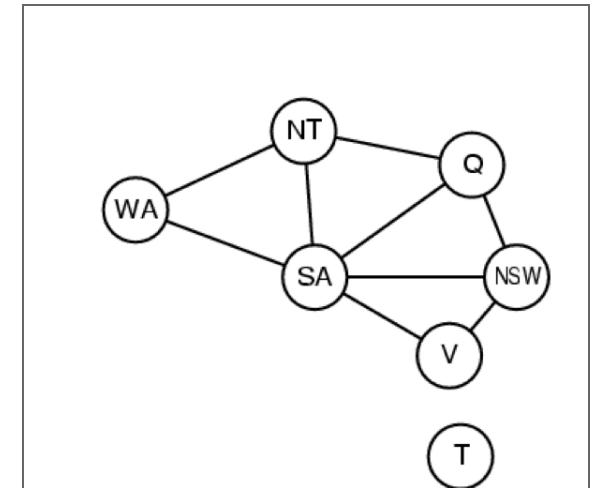
Start with straightforward search and then improve.

# SEARCH METHODS

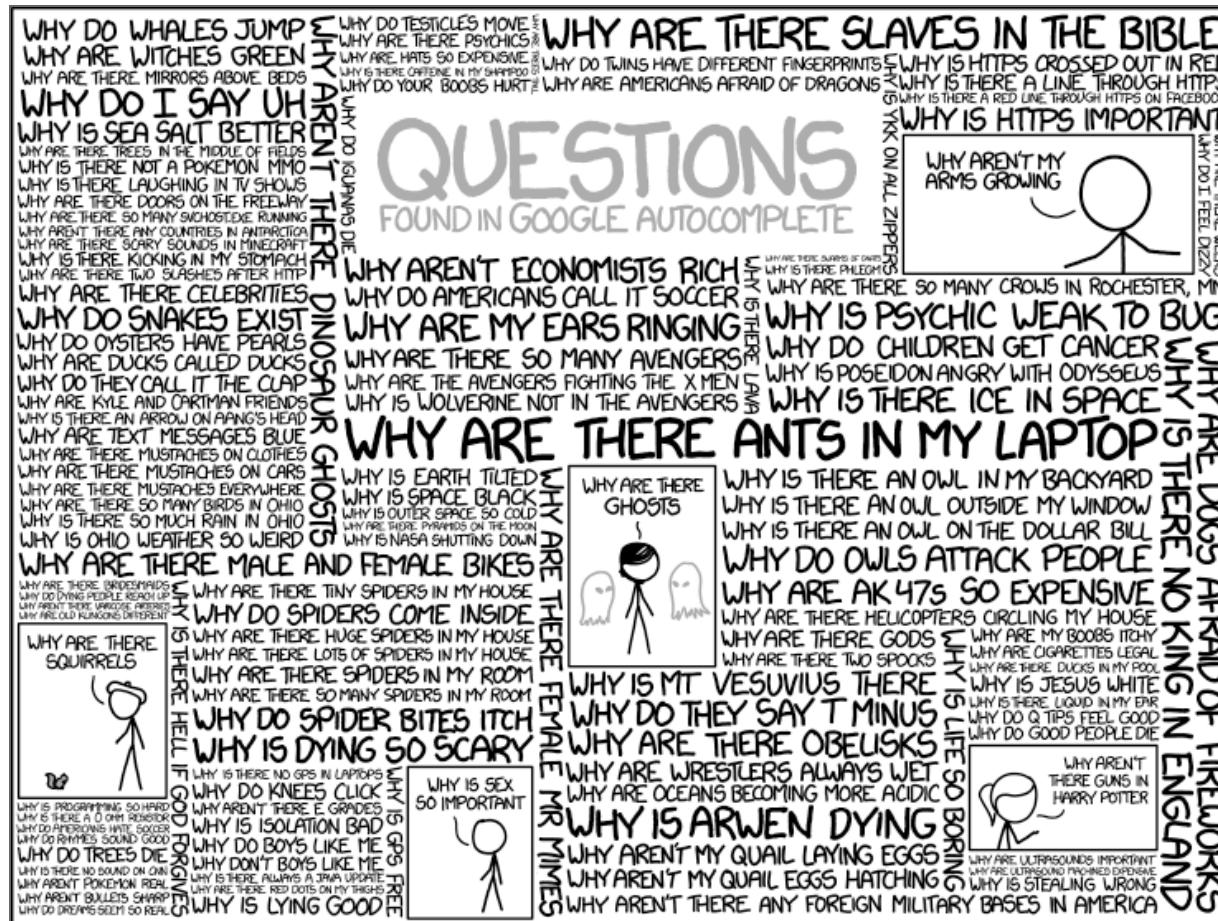
What would BFS do?

What would DFS do?

What problems does naïve search have?



# Q & A



XKCD







## Speaker notes

≡  
Q & A

