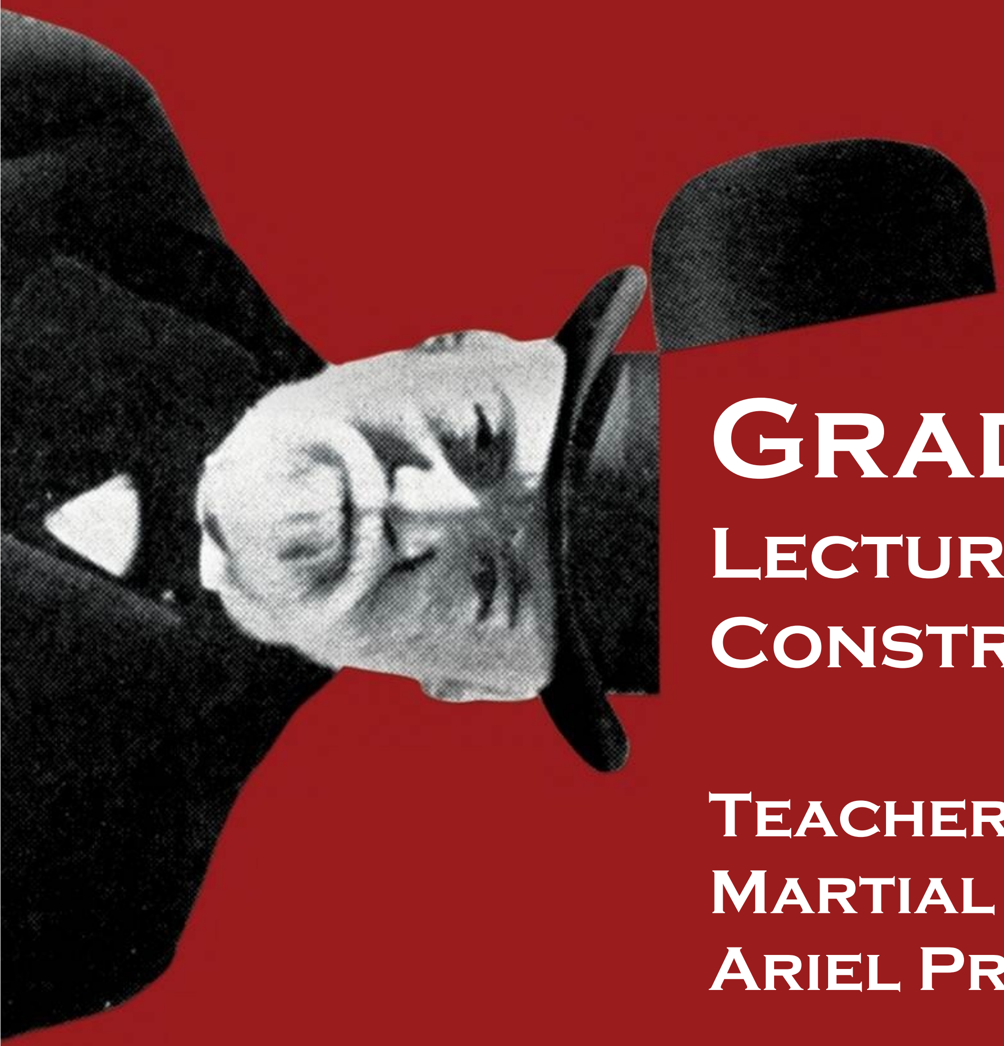


Graduate AI

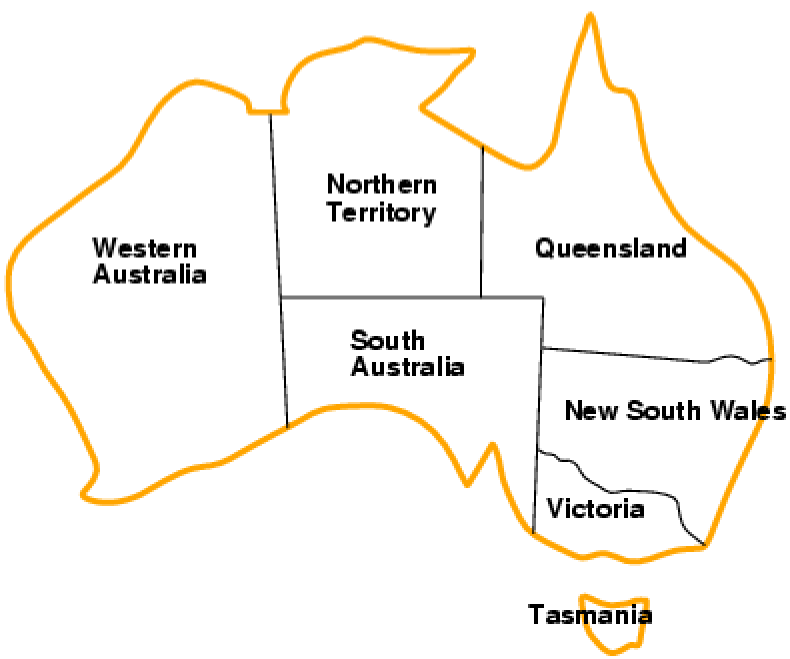
Lecture 13: Constraint Satisfaction 1

Teachers:
Martial Hebert
Ariel Procaccia (this time)

What are CSPs?

- A **constraint satisfaction problem** (CSP) consists of:
  - Variables $\{X_1,\ldots,X_n\}$
  - Domains $\{D_1,\ldots,D_n\}$
  - A set of constraints: defined on subsets of variables, give allowable tuples of values
- Consider (possibly partial) assignments of values to variables
- **Solution** = **complete**+**consistent** assignment

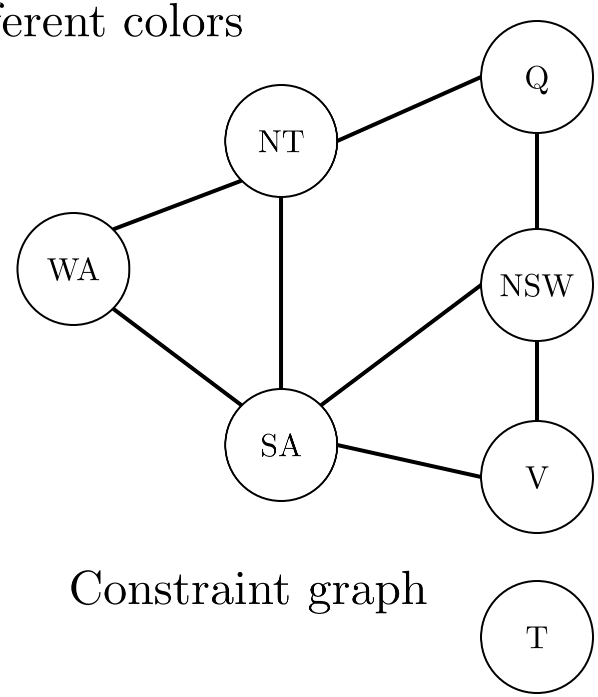
Example: map coloring



Variables = {WA,NT,SA,Q,NSW,V,T}

D_i = {■ ■ ■}

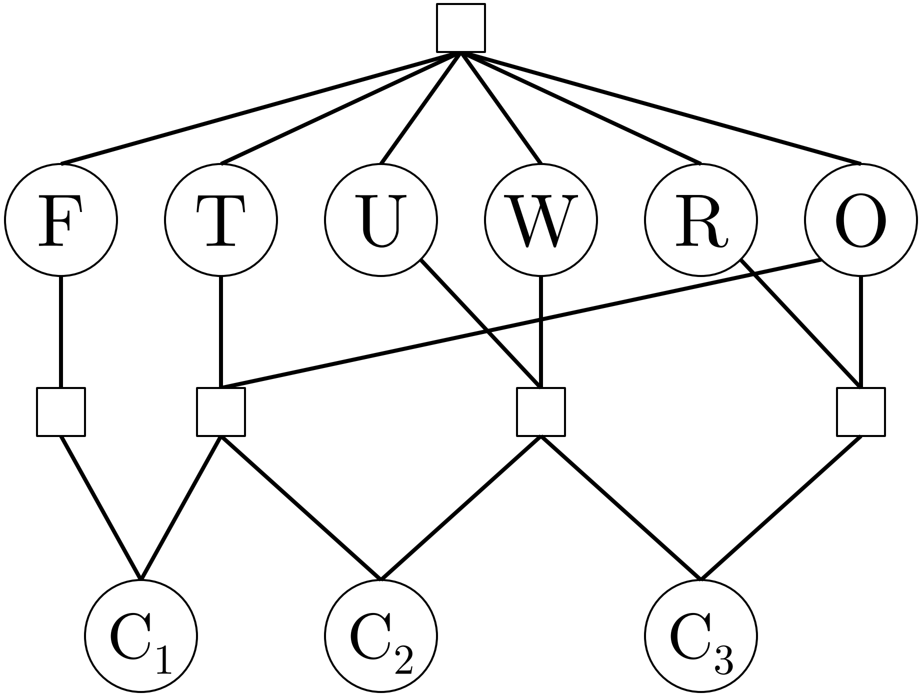
Constraints: adjacent regions have different colors



Constraint graph

Example: cryptarithmetic

```
  TWO
+ TWO
-----
 FOUR
```



Other examples

- Sudoku
- Assignment problems, e.g., who teaches which class
- Scheduling problems, e.g, meetings, transportation, manufacturing
- SAT (on Monday)

Binary CSPs

- Binary CSP = constraints involve exactly two variables
- Map coloring is binary, cryptarithmetic is not
- Any CSP can be transformed into a binary CSP; how?
- Hint: add new variable for each non-unary constraint

Complexity of CSPs

- CSPs are NP-complete
- Clearly in NP: given an assignment, check that it is legal
- Graph coloring is a special case

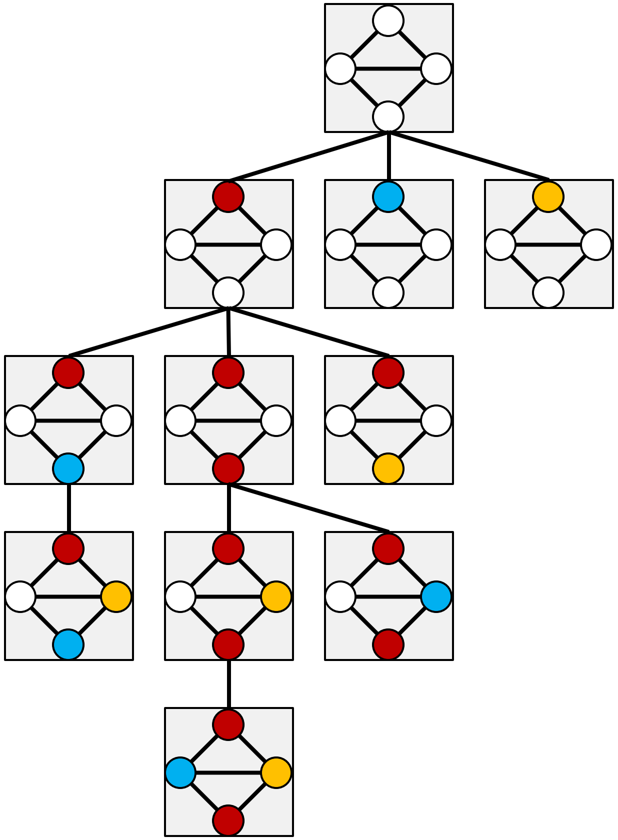
CSPs vs. Search

- Informed search:
  - State is a black box
  - Heuristics are problem-specific
- In contrast, CSPs:
  - States are assignments; have structure
  - General-purpose algorithms that do not require domain-specific knowledge

Search formulation

- Initial state: empty assignment
- Successor: consistent assignment to unassigned variable
- Goal test: assignment is complete
- Variable assignments are commutative $\Rightarrow$ At each node, only consider legal assignments to one of the variables

Example: backtracking



Improving efficiency

- Which variable to assign next?
- Most constrained variable: least # legal values
- Which value to assign next?
- Least constraining: largest # legal values for remaining variables
- Weird?

K-CONSISTENCY

- A CSP is **k-consistent** if for every $Y_1,\ldots,Y_k$, any legal assignment for $Y_1,\ldots,Y_{k-1}$ can be extended to a legal assignment for $Y_1,\ldots,Y_k$
- **Strong k-consistency** = k' consistency for every $k' \leq k$
- **Global consistency** = strong n-consistency
- Global consistency $\Rightarrow$ backtrack-free search
- Practitioners usually enforce 2-consistency

Solvability via global properties

- We prove the following statements on the board / in the addendum
- Assume D_i=D for all i, denote |D|=d, and assume the CSP has arity r (each constraint has at most r variables). Then strong (d(r–1)+1)-consistency ⇒ global consistency
- Let there be a CSP with arity r. Let t be an upper bound on the number of constraints each variable appears in. Let q be a lower bound on the probability of choosing a satisfying assignment for a constraint. If $q \geq 1-1/e(r(t-1)+1)$ then there is a satisfying assignment