Lecture 15

CS 6380 Artificial Intelligence

Constraint Satisfaction Problems

Department of Computer Science and Engineering

Indian Institute of Technology Madras (IITM), India

Material based on lecture slides for CSE 571: Artificial Intelligence by Siddharth Srivastava, ASU.

Sudoku

- Problem: fill in the blank squares such that
  - the cells in each row and column have distinct digits $\{1, 2, \dots, 9\}$
- Given
  - A variable s_i for each open square
- Compute a digit assignment σ such that
  - All assignments in each row are different
  - All assignments in each column are different
  - All assignments in each region are different

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Sudoku

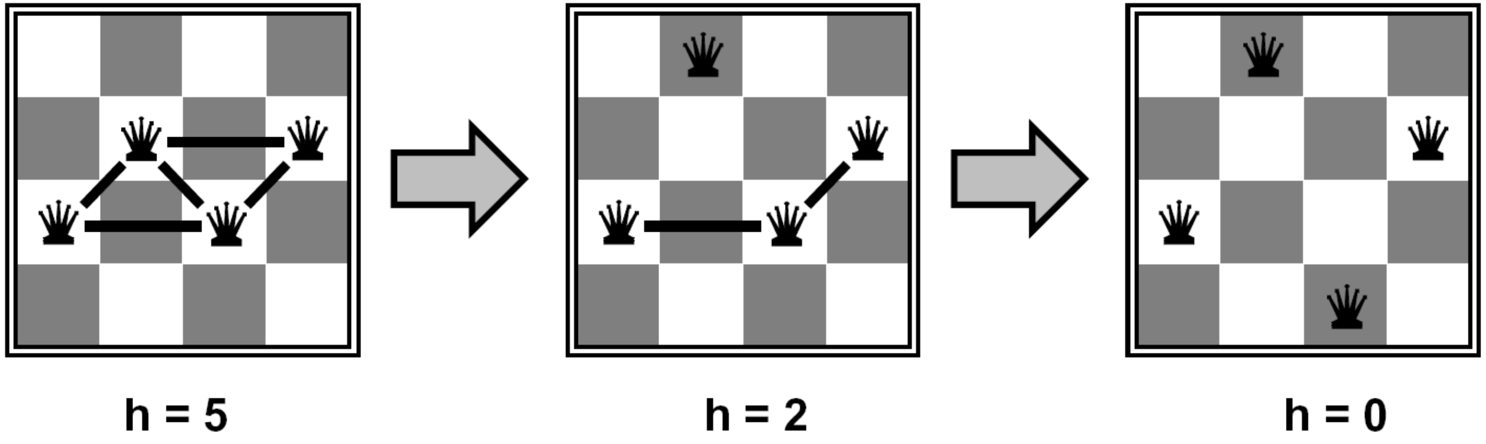
- Problem: fill in the blank squares such that
  - the cells in each row and column have distinct digits $\{1, 2, \ldots, 9\}$
- C
- C

Formulate as a search problem

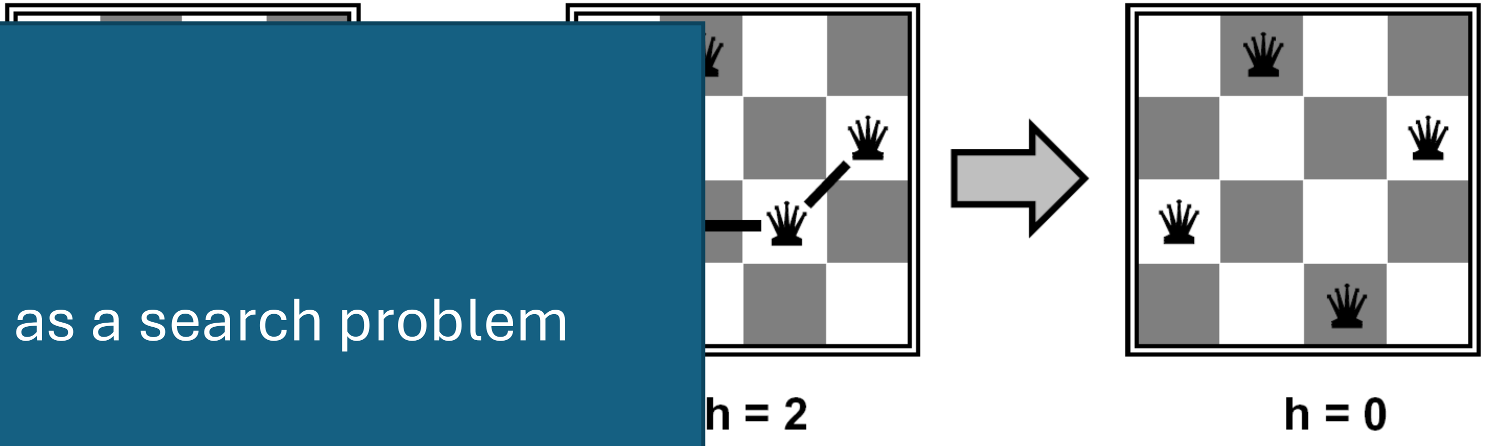
- States
- Actions
- Performance measure

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

N-Queens



N-Queens



Formulate as a search problem

- States
- Actions
- Performance measure

Map Coloring

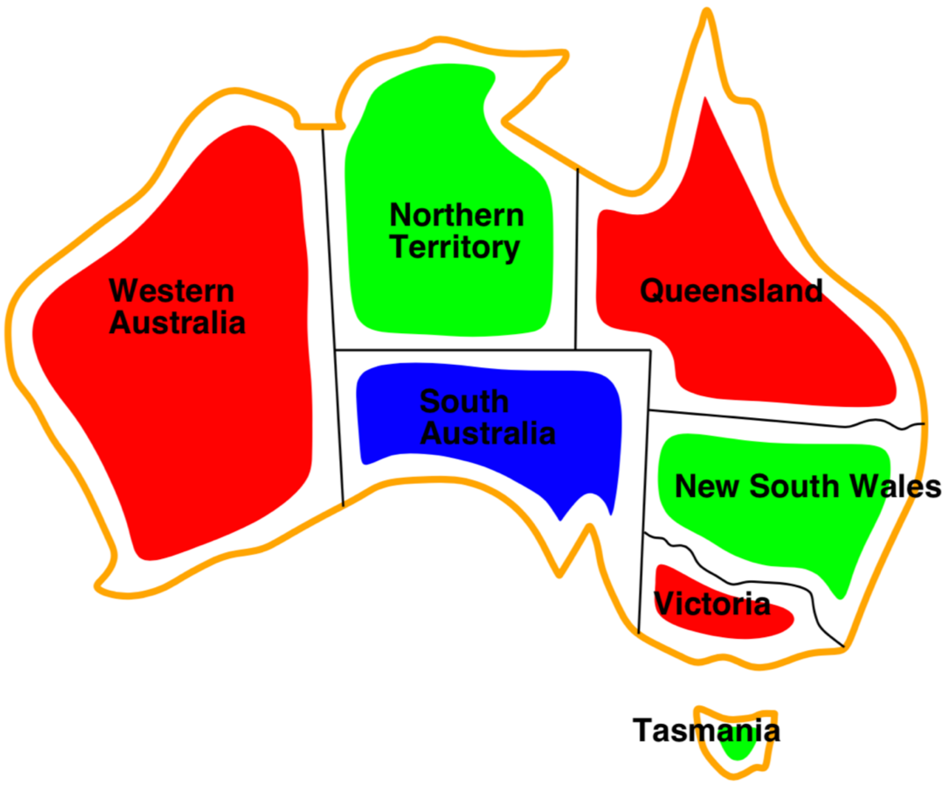
- Problem: color this map such that adjacent regions have different colors

- Colors: red, green, blue

Map Coloring

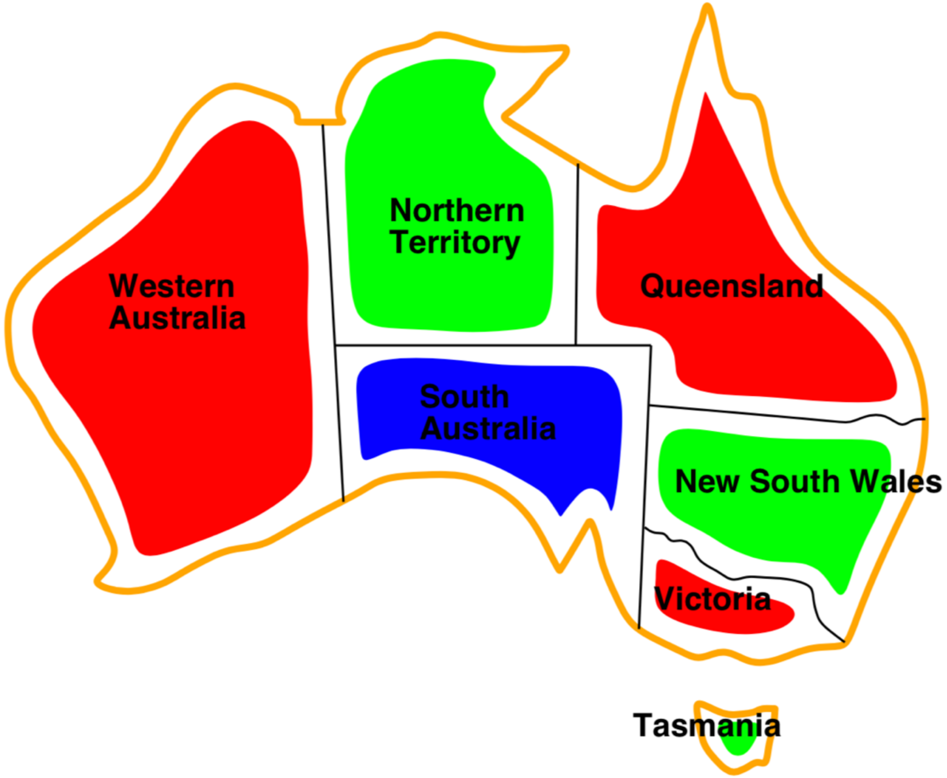
Formulate as a search problem

- States
- Actions
- Performance measure



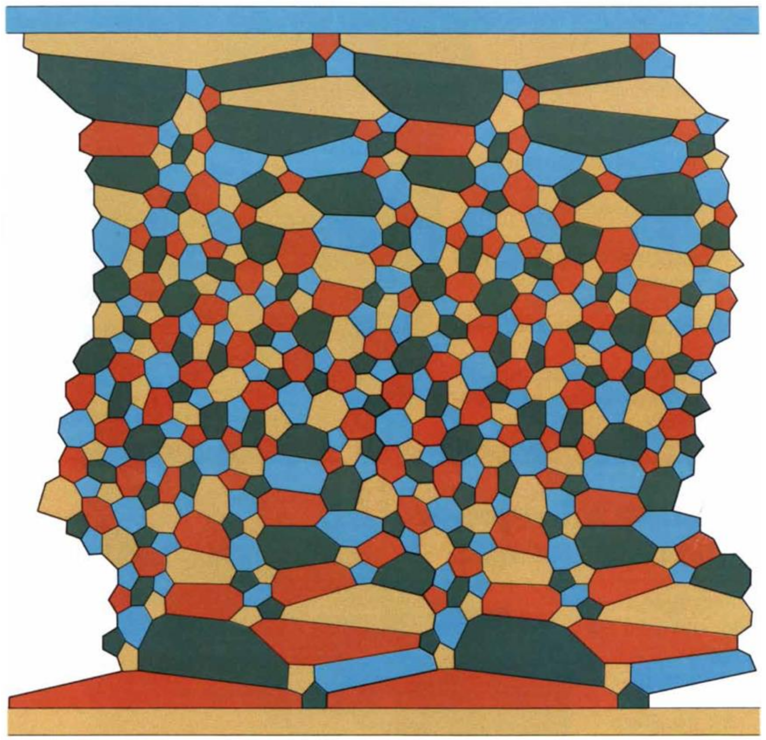
Map Coloring

- Problem: color this map such that
  - adjacent regions have different colors
- Colors: red, green, blue
- Can formulate as:
  - Regions WA, NT, Q, NSW, SA, T
  - Possible colors: $red, green, blue$
- Compute:
  - A color mapping $\sigma : Region \rightarrow Color$ such that
  - For all adjacent $r_1, r_2, \;\; \sigma(r_1) \neq \sigma(r_2)$



Did You Know?

- 3 colors are not sufficient to color all possible maps
    - Counterexample…
- Would 4 colors suffice? (1852)
- 5 colors suffice to color any planar (flat) map [Heawood, 1890]



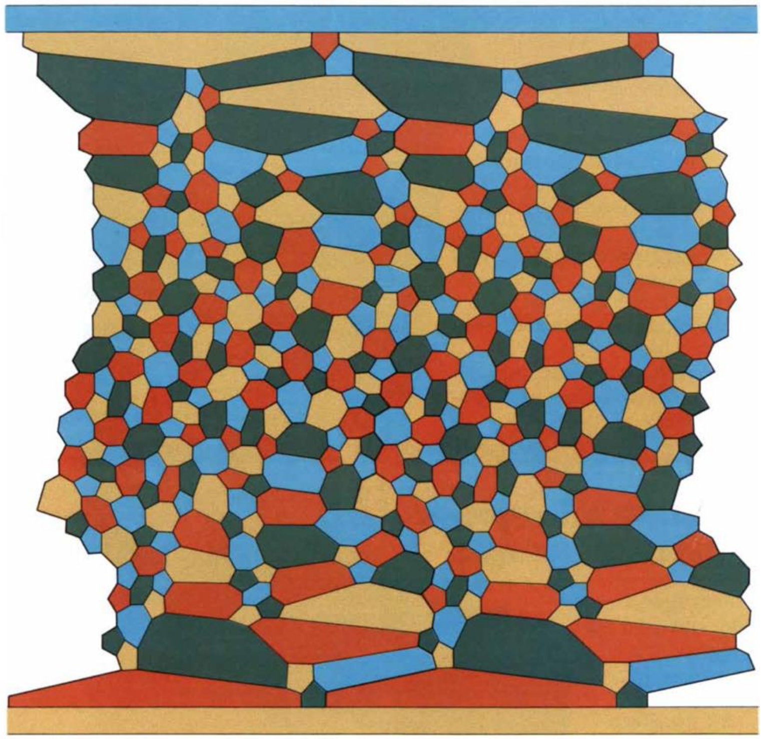
Did You Know?

- The 4 color question remained a mystery for more than 100 years, until…
  - Computer assisted proof [Appel, Haken 1976]

The Solution of the Four-Color-Map Problem

Four colors suffice to color any planar map so that no two adjacent countries are the same color. This famous conjecture has been proved true by a new kind of proof, one that relies on high-speed computers

by Kenneth Appel and Wolfgang Haken



An Early AI Success Story

- But doubts remained!
  - The proof was 400 pages long

- Philosophical issues were raised

The Solution
of the Four-Color-Map Problem

THE JOURNAL OF PHILOSOPHY

VOLUME LXXVI, NO. 2, FEBRUARY 1979

Thomas Tymoczko

THE FOUR-COLOR PROBLEM AND ITS PHILOSOPHICAL SIGNIFICANCE *

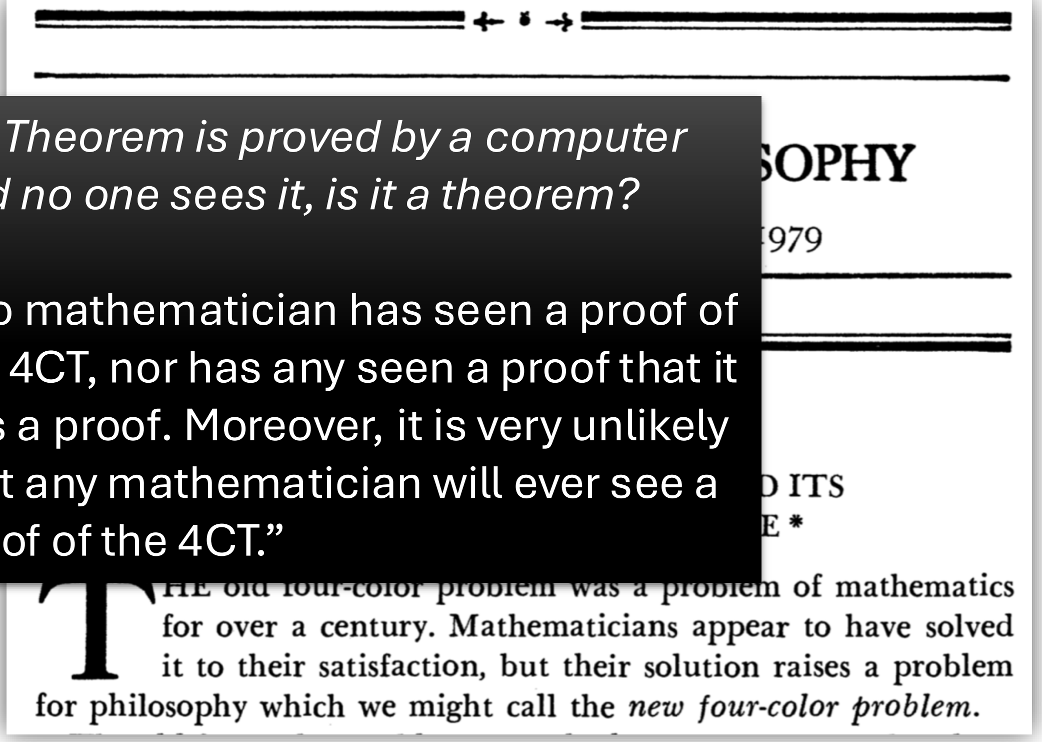
THE old four-color problem was a problem of mathematics for over a century. Mathematicians appear to have solved it to their satisfaction, but their solution raises a problem for philosophy which we might call the *new four-color problem*.

An Early AI Success Story

- But doubts remained!
  - The proof was 400 pages long

- Philosophical issues were raised

The Solution
of the Four-Color-Map Problem


THE old four-color problem was a problem of mathematics for over a century. Mathematicians appear to have solved it to their satisfaction, but their solution raises a problem for philosophy which we might call the *new four-color problem*.

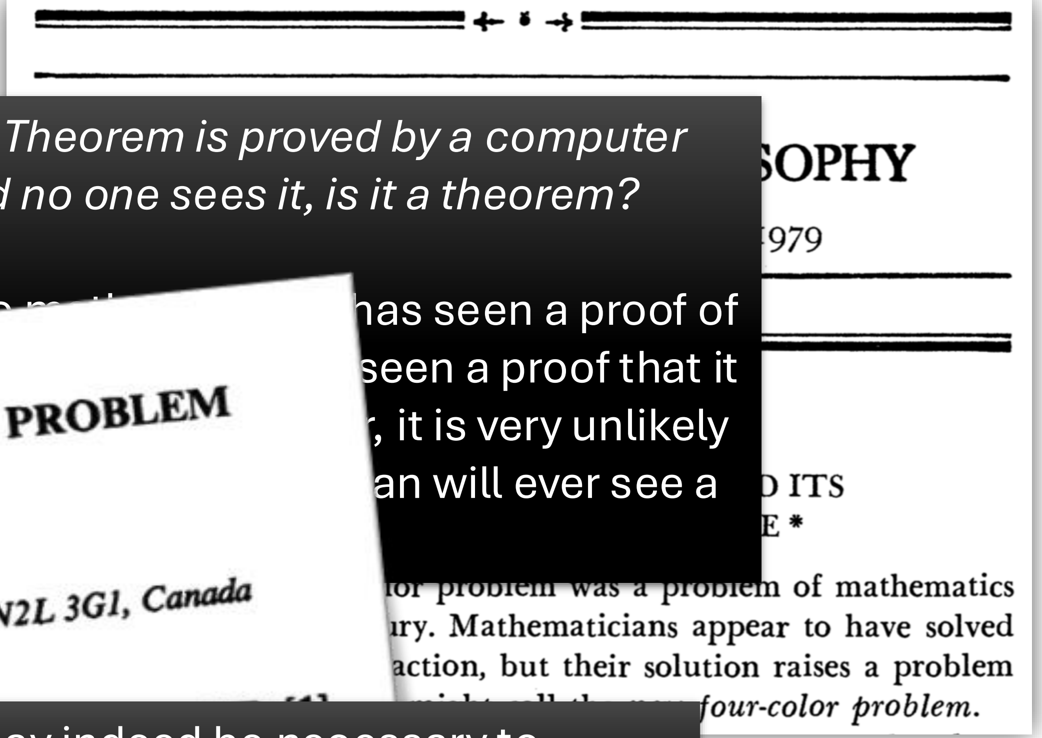
If a Theorem is proved by a computer and no one sees it, is it a theorem?

"No mathematician has seen a proof of the 4CT, nor has any seen a proof that it has a proof. Moreover, it is very unlikely that any mathematician will ever see a proof of the 4CT."

An Early AI Success Story

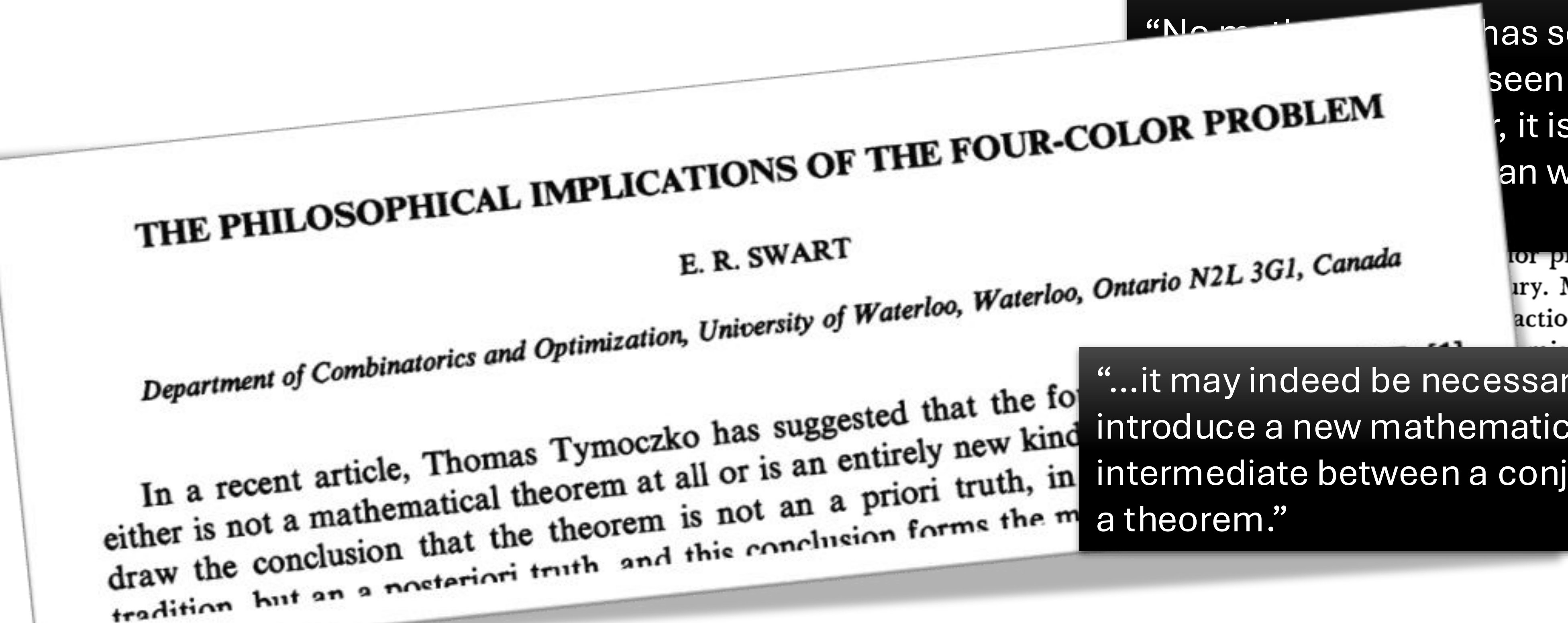
- But doubts remained!
  - The proof was 400 pages long
- Philosophical issues were raised

The Solution
of the Four-Color-Map Problem



If a Theorem is proved by a computer and no one sees it, is it a theorem?

"No m… has seen a proof of … seen a proof that it …, it is very unlikely … an will ever see a



"…it may indeed be necessary to introduce a new mathematical entity intermediate between a conjecture and a theorem."

An Early AI Success Story

- But doubts remained!
  - The proof was 400 pages long
- Philosophical issues were raised
- 1996: $O(n^2)$ algorithm for 4-coloring maps!
  - Proof is computer assisted (but less exhaustive checking: 633 vs 1,476 graphs)
- 2005: Complete proof using general purpose theorem prover (Gonthier)
  - Yay, AI theorem proving!

Formal Proof—The Four-Color Theorem

Georges Gonthier

The Tale of a Brainteaser

Francis Guthrie certainly did it, when he coined his innocent little coloring puzzle in 1852. He managed to embarrass successively his mathematician brother, his brother's professor, Augustus de Morgan, and all of de Morgan's visitors, who couldn't solve it; the Royal Society, who only realized ten years later that Alfred Kempe's 1879 solution was wrong; and the three following generations of mathematicians who couldn't fix it [19].

even with a language rich enough to express all mathematics.

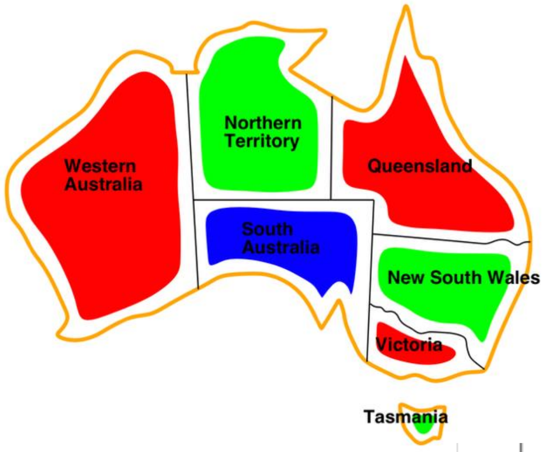
In 2000 we tried to produce such a proof for part of code from [13], just to evaluate how the field had progressed. We succeeded, but now a new question emerged: was the statement of the correctness proof (the *specification*) itself correct? The only solution to that conundrum was to formalize the *entire* proof of the Four-Color Theorem, not just its code. This we finally achieved in 2005.

"…we were pleasantly surprised to uncover new and rather elegant nuggets of mathematics in the process. "

Can such problems be solved automatically?

What's Common?

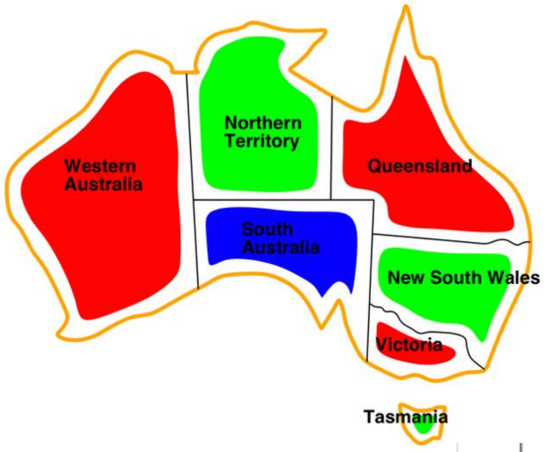
- Problem states are defined using
  - variables X_i with values from domains D_i
- Goal test is a set of constraints on variable values
- Constraint satisfaction problem:
  - Find an assignment to variable satisfying the constraints
- This captures a range of applications, including
  - Task allocation to heterogeneous robots,
  - Taxi services
  - Hubble telescope scheduling, Time tables…
- It seems to be a special case of a search problem!



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

CSPs as a Special Case of Search

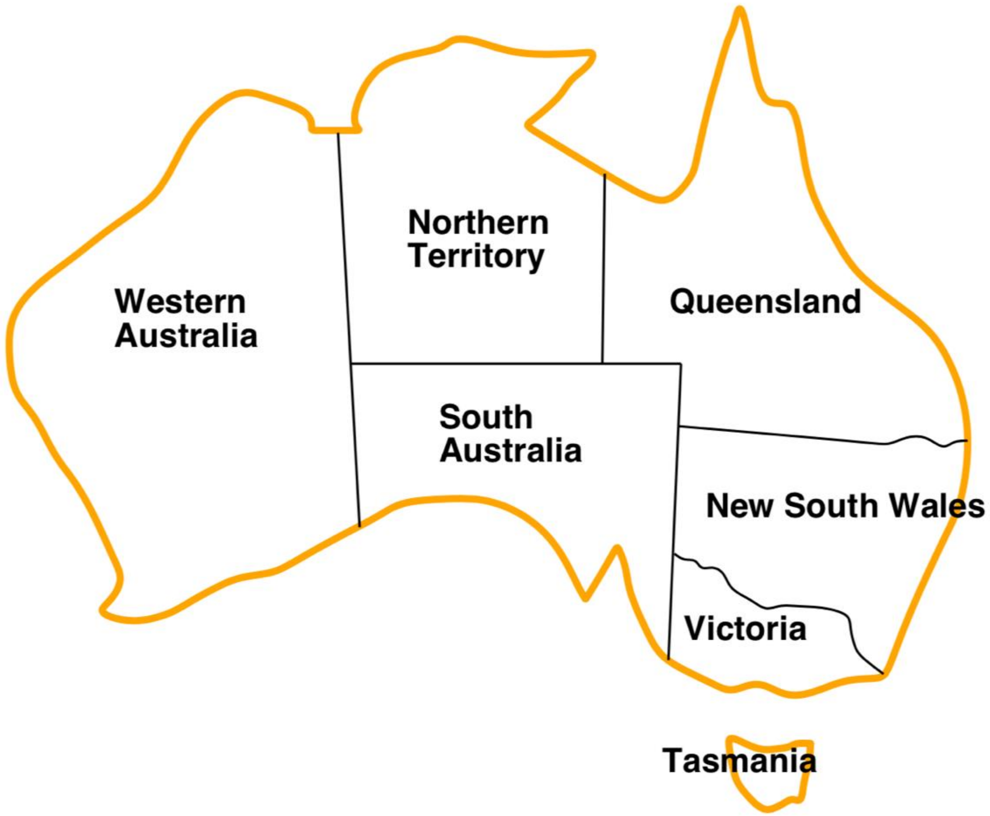
- What's different?
  - There's a lot of "structure"
  - States are rich entities (not black-box data structures): variables, domains
  - Constraints may involve subsets of variables
    - Tasmania
- Can we exploit this structure?



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

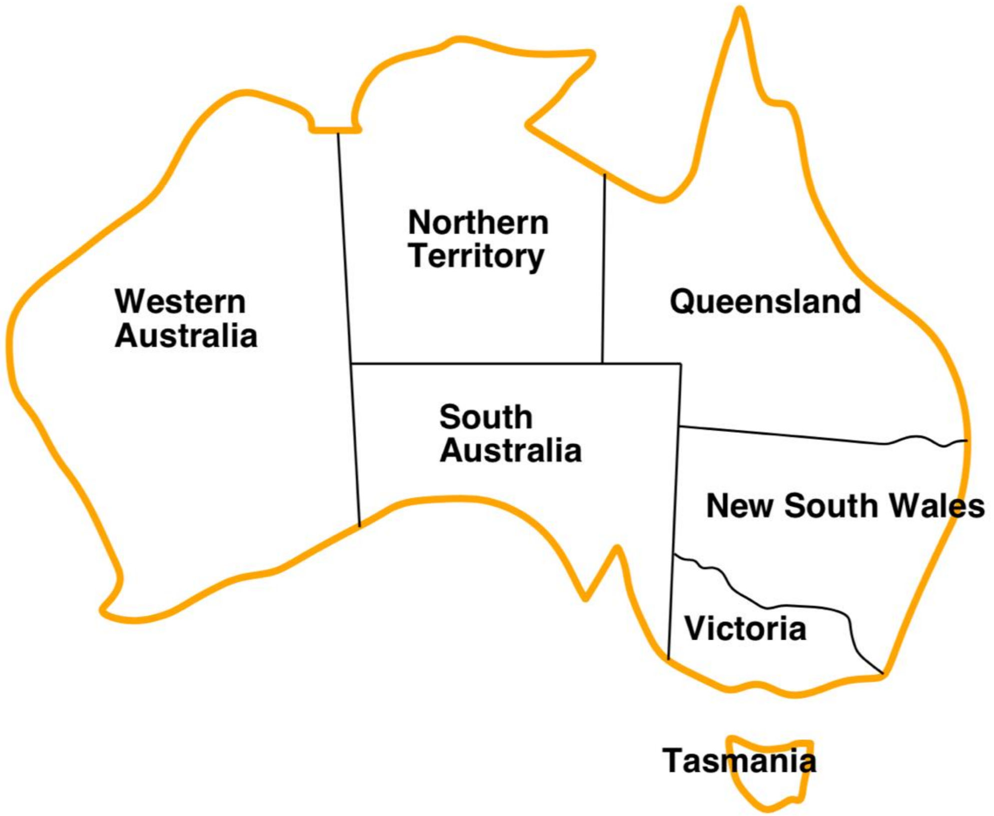
Constraint Graph

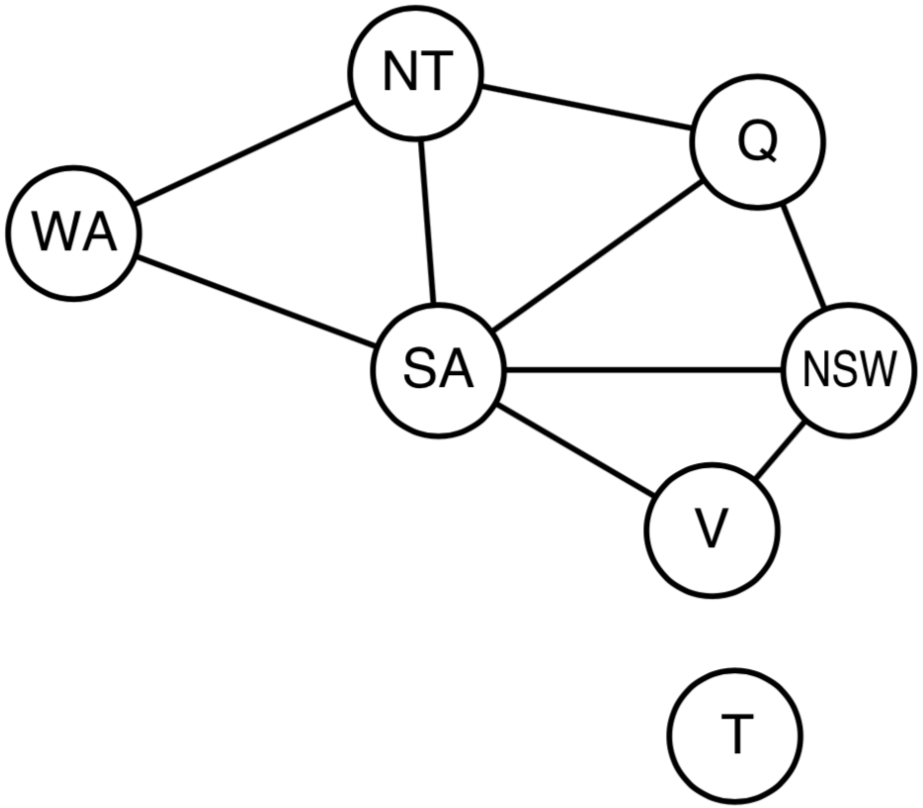
- Vertices: variables
- Edges between any pair of variables that are involved in a constraint



Constraint Graph

- Vertices: variables
- Edges between any pair of variables that are involved in a constraint

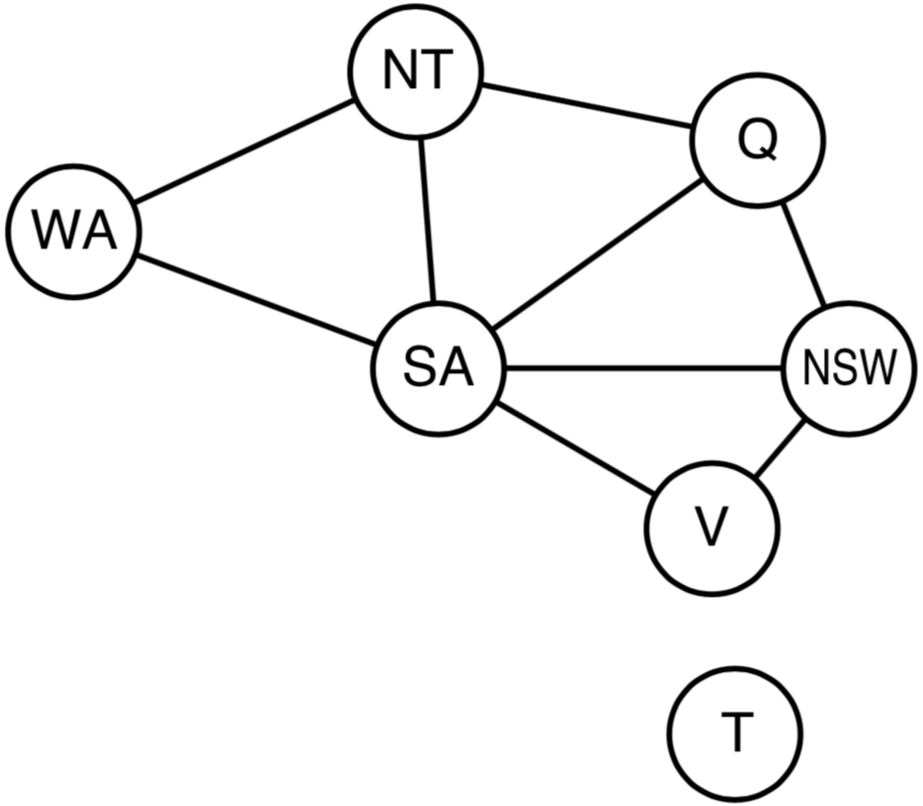




Constraint Graph

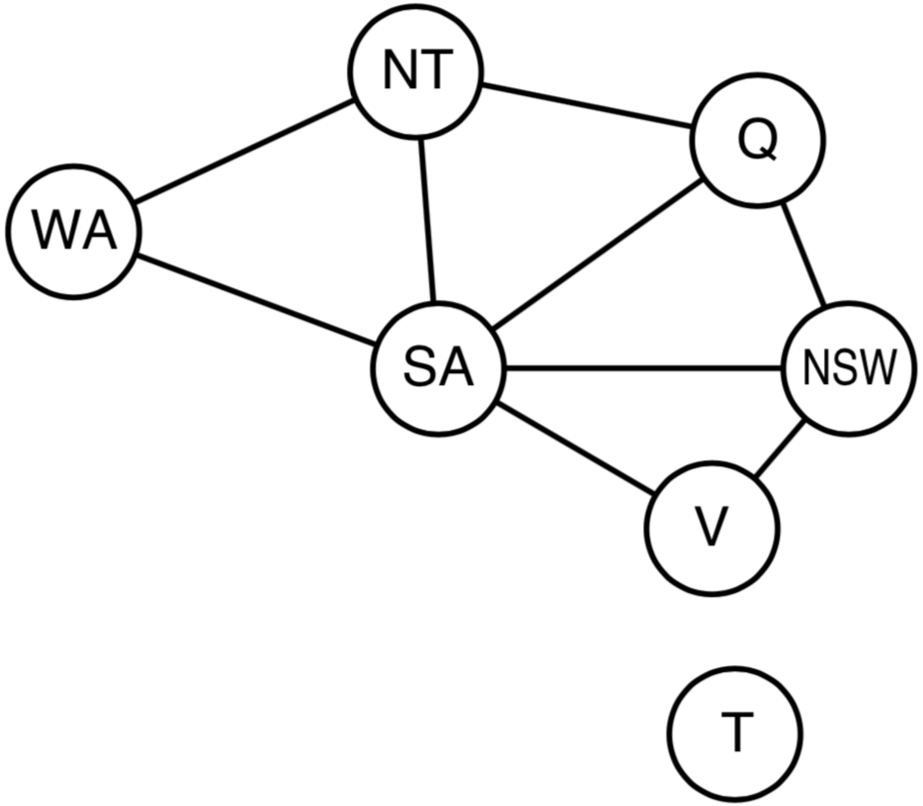
- Binary CSP: each constraint relates at most two variables

- CSP algorithms use the structure of this graph to optimize search



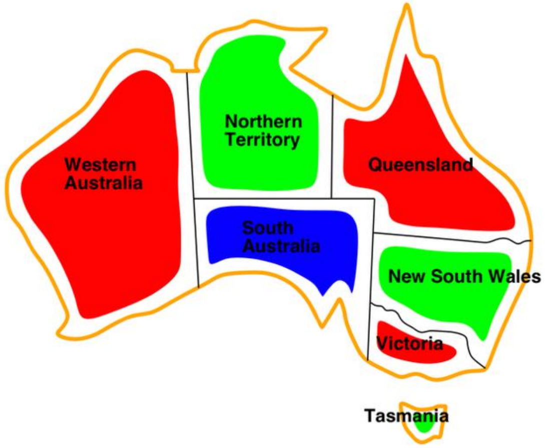
Varieties of CSPs

- Variables can be
  - Discrete
    - Finite domains (Boolean CSP: NP-complete)
      - d^n complete assignments
    - Infinite domains (integers, strings)
      - Project/Mission scheduling (start/end days)
  - Continuous
    - Start/end times for Hubble telescope
- Constraints
  - Linear, discrete variables: solvable
  - Nonlinear, discrete, infinite domains: undecidable



Types of Constraints

- Binary (recall)
- Unary constraints: involve a single variable
  - $SA \neq green$
- Higher-order constraints:
  - Involve 3 or more variables
  - E.g., Sudoku
- Preferences: soft constraints
  - E.g., Avoid making SA green
  - Can be associated with costs per variable assignment
- CSPs that include soft constraints are called constraint optimization problems



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

A More Complex Example: Cryptarithmetic

Assign unique values to letters to make this true:

$$\begin{array}{r} T\ W\ O \\ +\ T\ W\ O \\ \hline F\ O\ U\ R \end{array}$$

A More Complex Example: Cryptarithmetic

Assign unique values to letters to make this true:

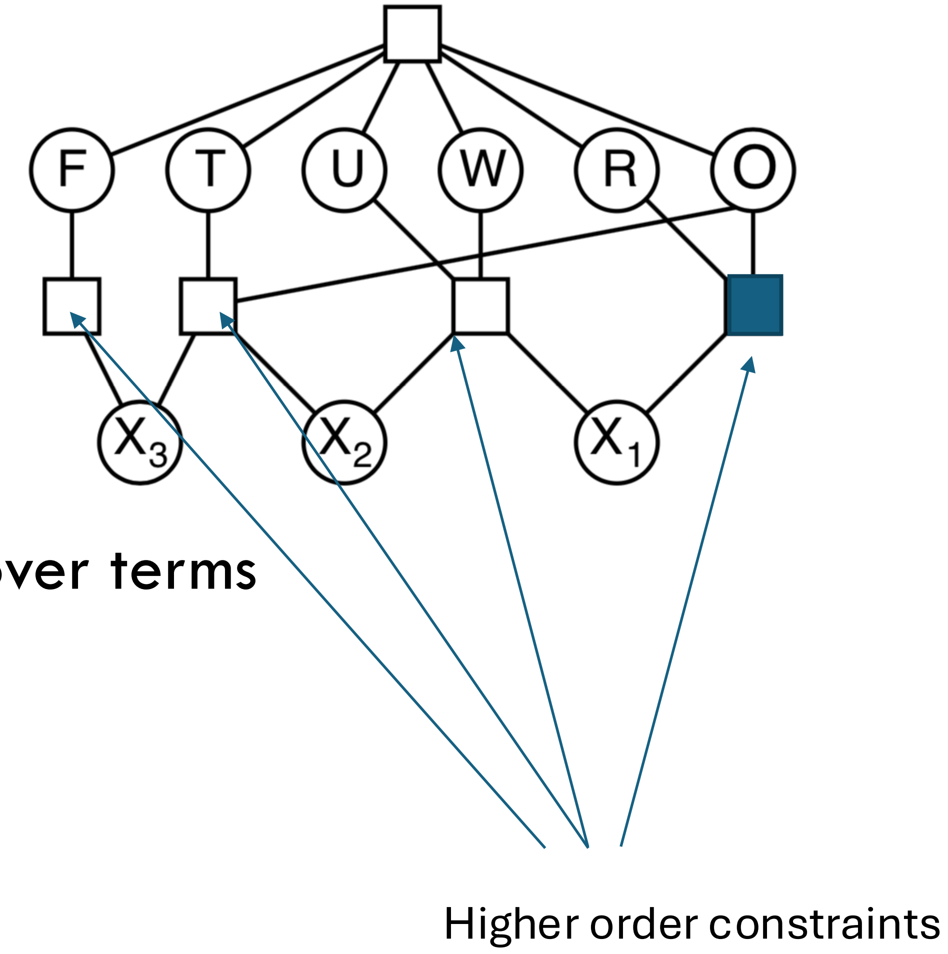
$$
\begin{array}{cccc}
 & T & W & O \\
+ & T & W & O \\
\hline
F & O & U & R
\end{array}
$$

- Need to add variables denoting carry-over terms
- Variables: $F, T, O, W, U, R, X_1, X_2, X_3$
- Domains: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Constraints:
  - alldiff(F, T, U, W, R, O)
  - $O + O = R + 10 \cdot X_1$
  - …

A More Complex Example: Cryptarithmetic

Assign unique values to letters to make this true:

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



- Need to add variables denoting carry-over terms
- Variables: $F, T, O, W, U, R, X_1, X_2, X_3$
- Domains: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Constraints:
  - alldiff(F, T, U, W, R, O)
  - $O + O = R + 10 \cdot X_1$
  - …

Standard Search Formulation

- Initial state: empty assignment $\sigma = \{\}$
- Successor function:
  - Select an unassigned variable
  - Assign a value from its domain that doesn't break any constraints
  - Fail if no such assignment possible
- Goal test: complete, constraint satisfying assignment

Standard Search Formulation

- Initial state: empty assignment $\sigma = \{\}$
- Successor function:
  - Select an unassigned variable
  - Assign a value from its domain that doesn't break any constraints
  - Fail if no such assignment possible
- Goal test: complete, constraint satisfying assignment

- Works for all CSPs!
  - All solutions appear at depth n for problems with n variables
  - Use DFS?
  - Number of leaves
    - At depth m, $b = (n - m)D$

    $\rightarrow n!\,D^n$ leaves