INT404 ARTIFICIAL INTELLIGENCE

Constraint Satisfaction

Lecture 10

Formal Definition of CSP

- A constraint satisfaction problem (CSP) is a triple (V, D, C) where
  - V is a set of variables $X_1, \ldots, X_n$.
  - D is the union of a set of domain sets $D_1,\ldots,D_n$, where D_i is the domain of possible values for variable X_i.
  - C is a set of constraints on the values of the variables, which can be pairwise (simplest and most common) or k at a time.

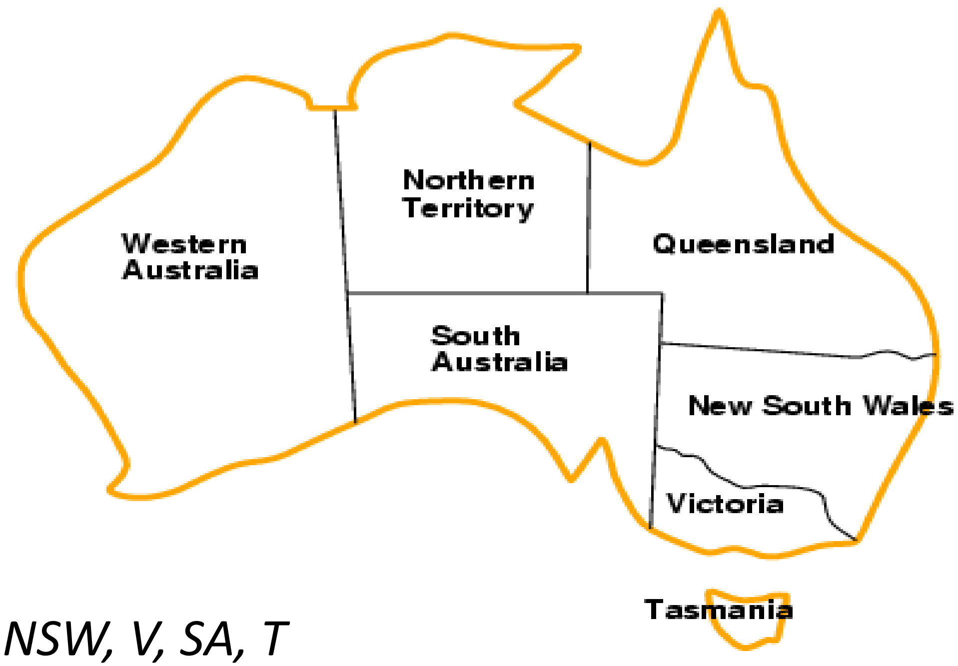
CSPs vs. Standard Search Problems

- Standard search problem:
  - state is a "black box“ – any data structure that supports successor function, heuristic function, and goal test
- CSP:
  - state is defined by variables X_i with values from domain D_i
  - goal test is a set of constraints specifying allowable combinations of values for subsets of variables

- Simple example of a formal representation language

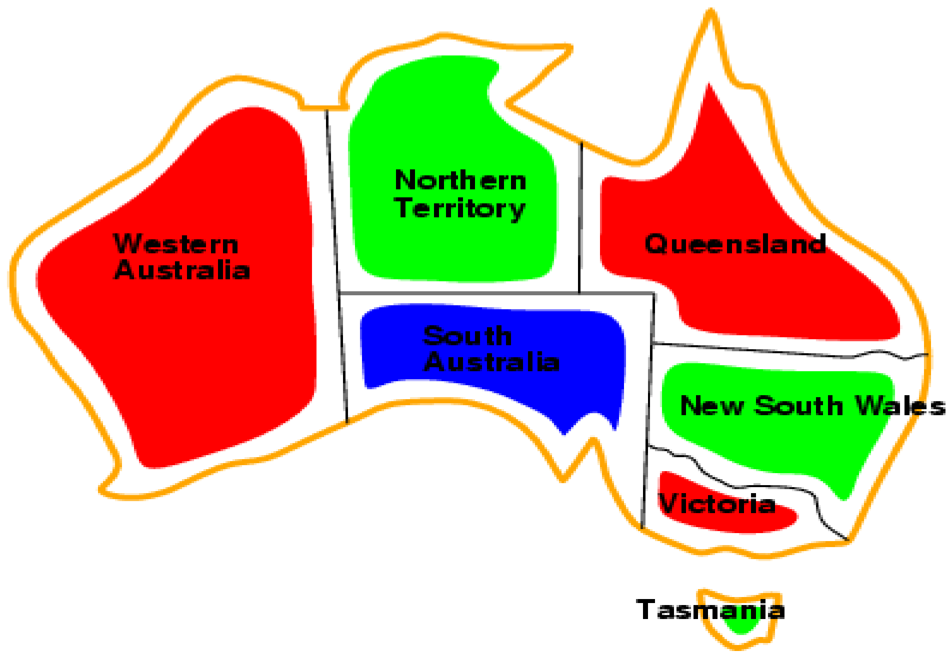
- **Allows useful general-purpose algorithms with more power than standard search algorithms**

Example: Map-Coloring



- Variables *WA, NT, Q, NSW, V, SA, T*
- Domains D_i = {red,green,blue}
- Constraints: adjacent regions must have different colors
- e.g., WA ≠ NT, or (WA,NT) in {(red,green),(red,blue),(green,red),(green,blue),(blue,red),(blue,green)}

Example: Map-Coloring



- Solutions are complete and consistent assignments, e.g., WA = red, NT = green,Q = red,NSW = green,V = red,SA = blue,T = green

Algorithm : Constraint Satisfaction

1. Propagate available constraints. To do this, first set *OPEN* to the set of all objects that must have values assigned to them in a complete solution. Then do until an inconsistency is detected or until *OPEN* is empty:

   (a) Select an object *OB* from *OPEN.* Strengthen as much as possible the set of constraints that apply to *OB.*

   (b) If this set is different from the set that was assigned the last time *OB* was examined or if this is the first time *OB* has been examined, then add to *OPEN* all objects that share any constraints with *OB.*

   (c) Remove *OB* from *OPEN.*

2. If the union of the constraints discovered above defines a solution, then quit and report the solution.

3. If the union of the constraints discovered above defines a contradiction, then return failure.

4. If neither of the above occurs, then it is necessary to make a guess at something in order to proceed. To do this, loop until a solution is found or all possible solutions have been eliminated:

   (a) Select an object whose value is not yet determined and select a way of strengthening the constraints on that object.

   (b) Recursively invoke constraint satisfaction with the current set of constraints augmented by the strengthening constraint just selected.

A Cryptarithmetic Problem

Problem:

```
  SEND
+ MORE
.................
 MONEY
```

Initial State:

No two letters have the same value.

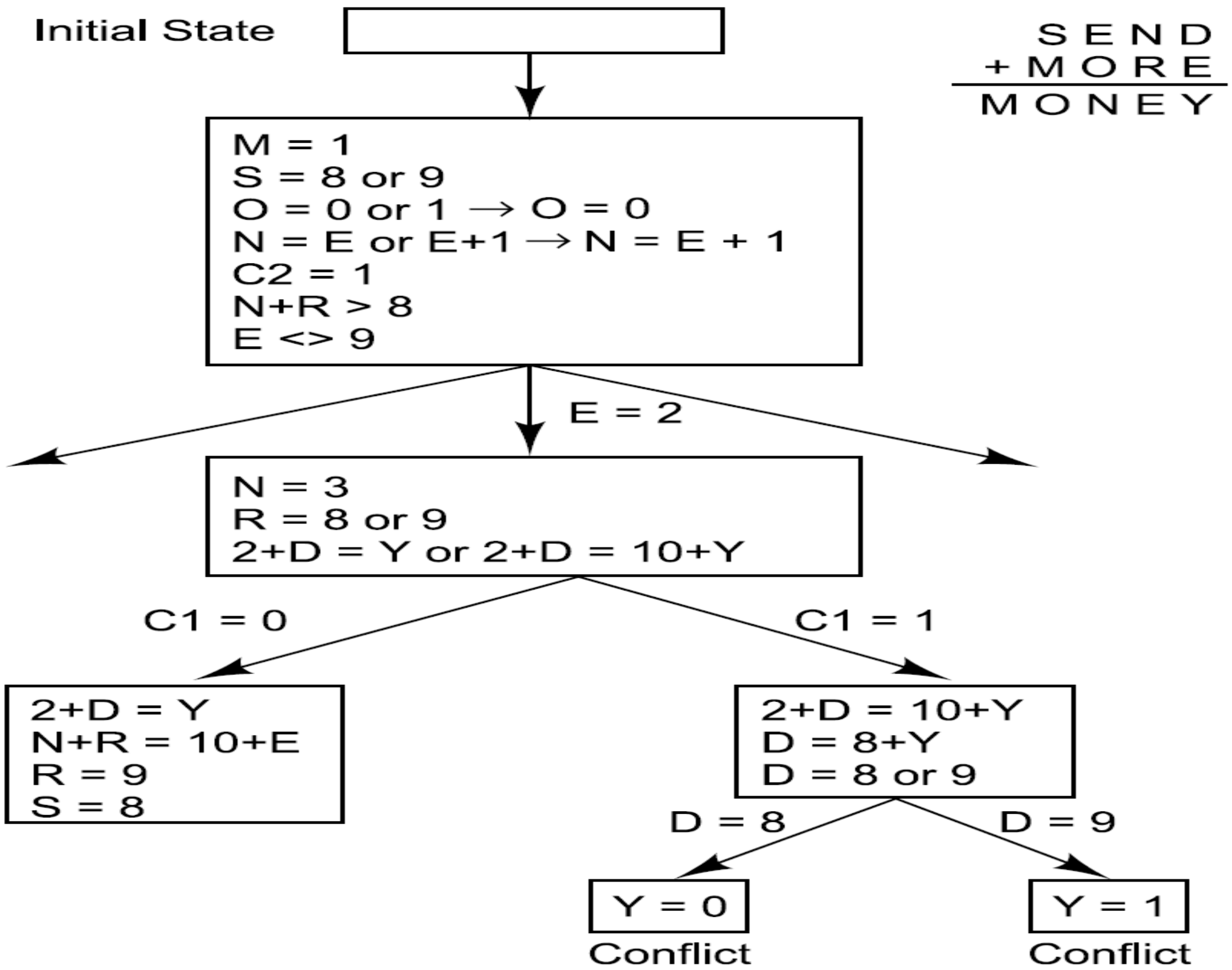
The sums of the digits must be as shown in the problem.

S E N D

\+ M O R E

M O N E Y

Solving a Cryptarithmetic Problem



	S	E	N	D
+	M	O	R	E
M	O	N	E	Y

From column 5, **M = 1** since it is the only carry-over possible from the sum of two single digit numbers in column 4.
Since there is a carry in column 5, and M = 1, then **O = 0**
Since O = 0 there cannot be a carry in column 4 (otherwise N would also be 0 in column 3) so **S = 9**.
If there were no carry in column 3 then E = N, which is impossible. Therefore there is a carry and N = E + 1.
If there were no carry in column 2, then (N + R) mod 10 = E, and N = E + 1, so (E + 1 + R) mod 10 = E which means (1 + R) mod 10 = 0, so R = 9. But S = 9, so there must be a carry in column 2 so **R = 8**.
To produce a carry in column 2, we must have D + E = 10 + Y.
Y is at least 2 so D + E is at least 12.
The only two pairs of available numbers that sum to at least 12 are (5,7) and (6,7) so either E = 7 or D = 7.
Since N = E + 1, E can't be 7 because then N = 8 = R so **D = 7**.
E can't be 6 because then N = 7 = D so **E = 5** and **N = 6**.
D + E = 12 so **Y = 2**.

	N	O	O	N
	M	O	O	N
+	S	O	O	N
	J	U	N	E