Artificial Intelligence (CS 3011)

CHAPTER 5: **Constraint satisfaction problem**

Sourav Kumar Giri
Asst. Professor

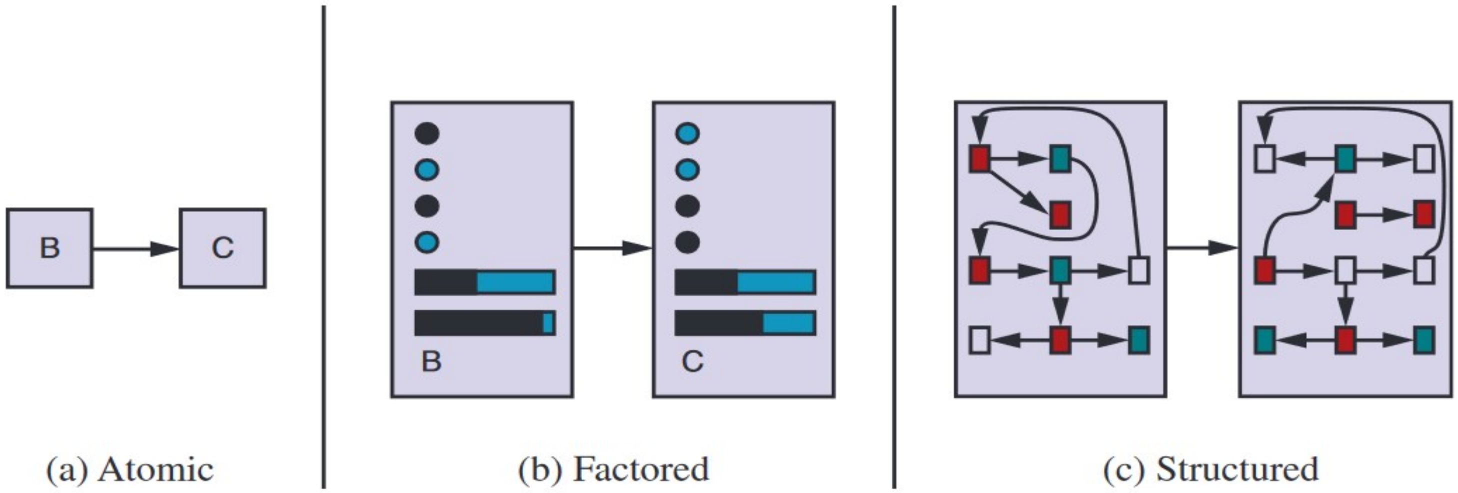
School of Computer Engineering
KIIT Deemed to be University

Chapter Outline

- Constraint Satisfaction Problems
- Examples-Australian color mapping, Job shop scheduling, Sudoku game, Cryptarithmetic
- Types of variables
- Types of constraints
- Types of consistencies & constraint propagation,
- Backtracking search for CSPs and local search for CSPs etc.

Constraint Satisfaction Problems

- CSP search algorithms take advantage of the **structure of states** and use general-purpose rather than problem-specific heuristics to enable the solution of complex problems.
- The main idea is to **eliminate large portions of the search space** all at once by identifying variable/value combinations that violate the constraints.



Constraint Satisfaction Problems

- A constraint satisfaction problem consists of three components, ***X***, ***D***, and ***C***:
  - ***X*** is a set of variables, $\{X_1, \ldots, X_n\}$.
  - ***D*** is a set of domains, $\{D_1, \ldots, D_n\}$, one for each variable.
  - ***C*** is a set of constraints that specify allowable combinations of values.
- Each domain D_i consists of a set of allowable values, $\{v_1, \ldots, v_k\}$ for variable X_i.
- Each constraint C_i consists of a pair (***scope, rel***), where
  - ***scope*** is a tuple of variables that participate in the constraint and
  - ***rel*** is a relation that defines the values that those variables can take on.
- A relation can be represented as an explicit list of all tuples of values that satisfy the constraint, or as an abstract relation that supports two operations: ***testing if a tuple is a member of the relation*** and ***enumerating the members of the relation***.

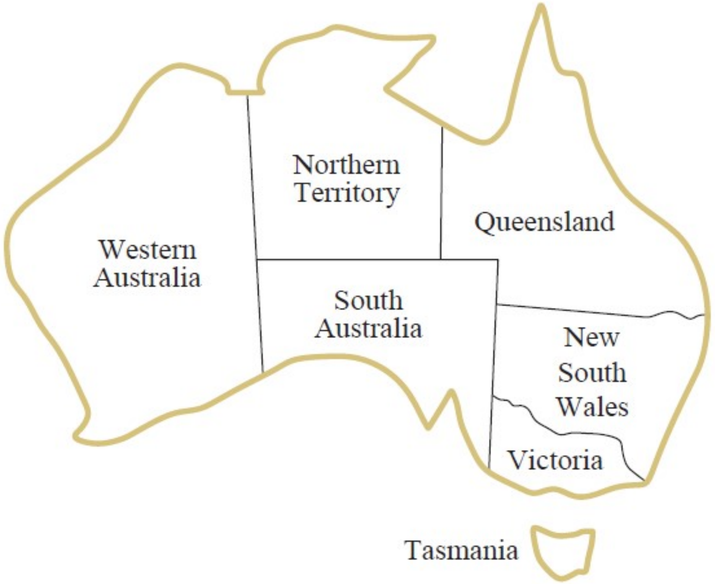
Constraint Satisfaction Problems

- For example,
  1) If X_1 and X_2 both have the domain $\{A,B\}$, then the constraint saying the two variables must have different values can be written as:

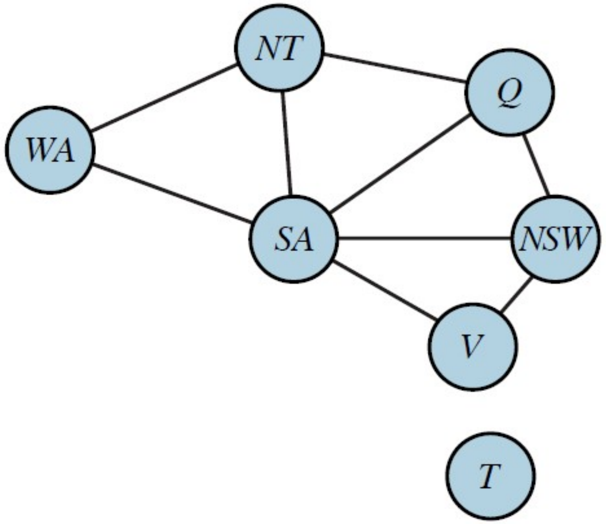
     $<(X_1,X_2), [(A,B),(B,A)]>$ or as $<(X_1,X_2), X_1 \neq X_2>$
  2) If X_1 and X_2 both have the domain $\{1, 2, 3\}$, then the constraint value of X_1 is greater than the value of X_2 can be written as:

     $<(X_1,X_2), [(2,1), (3,1), (3,2)]>$ or as $<(X_1,X_2), (X_1>X_2)>$

Example problem: Map coloring



(a)



(b)

- Figure. **(a)** The principal states and territories of Australia. Coloring this map can be viewed as a constraint satisfaction problem (CSP). The goal is to assign colors to each region so that no neighboring regions have the same color. **(b)** The map-coloring problemrepresented as a constraint graph.

Example problem: Map coloring

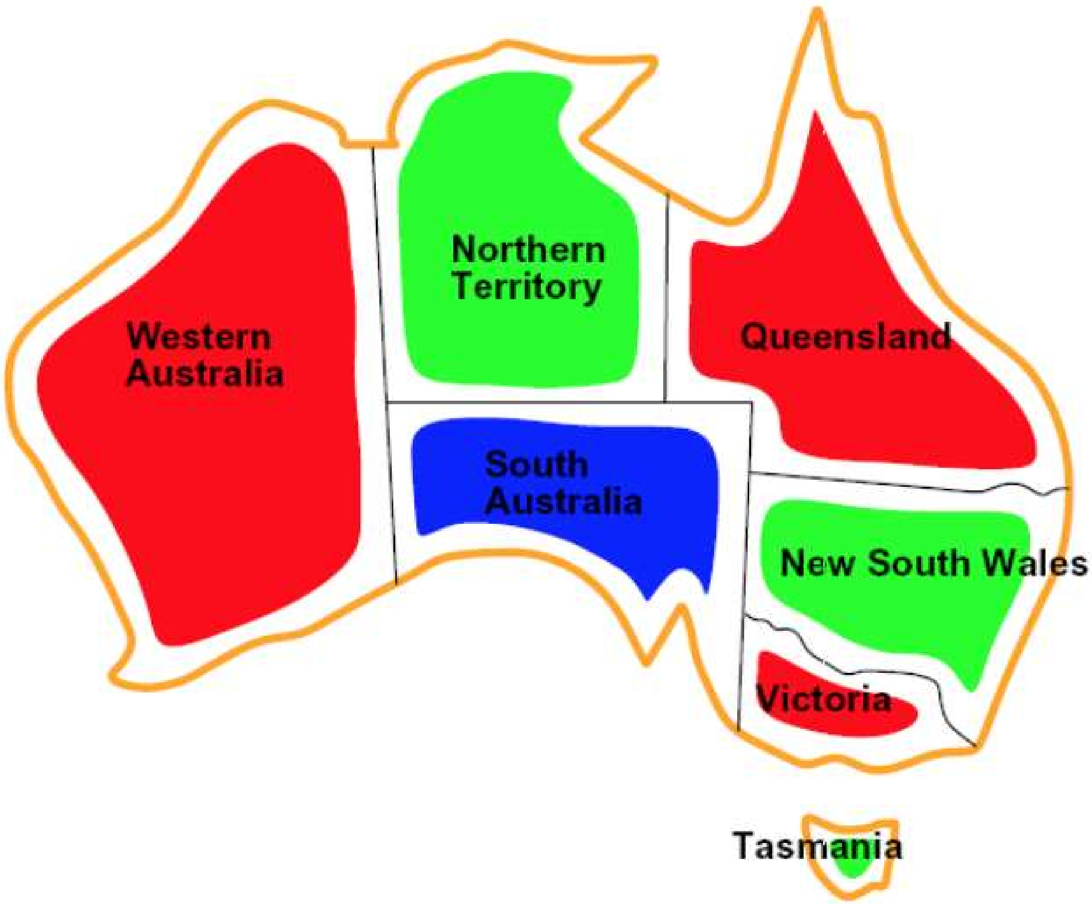
- Given a map of Australia showing each of its states and territories.
- **Goal-** To color each region as either red, green, or blue in such a way that no neighboring regions have the same color.
- CSP Formulation-
  - We define the variables to be the regions:

    ***X* = {WA, NT, Q, NSW, V, SA, T}**
  - The domain of each variable is the set D_i

    D_i = {red , green, blue}
  - The constraints require neighboring regions to have distinct colors. Since there are nine places where regions border, there are nine constraints:

    ***C* = {SA≠WA, SA≠NT, SA≠Q, SA≠NSW, SA≠V, WA≠NT, NT≠Q, Q≠ NSW, NSW≠V}**

Example problem: Map coloring



Example problem: Map coloring

❑ Why formulate a problem as a CSP?

- CSPs yield a natural representation for a wide variety of problems
- CSP solvers can be faster than state-space searchers because the CSP solver can quickly eliminate large swatches of the search space.
- For example, once we have chosen {SA=blue} in the Australia problem, we can conclude that none of the five neighboring variables can take on the value blue. Without taking advantage of constraint propagation, a search procedure would have to consider 3^5 =243 assignments for the five neighboring variables; with constraint propagation we never have to consider blue as a value, so we have only 2^5 =32 assignments to look at, a reduction of 87%.

Example problem: Job-shop scheduling

❑ Why formulate a problem as a CSP?

- CSPs yield a natural representation for a wide variety of problems
- CSP solvers can be faster than state-space searchers because the CSP solver can quickly eliminate large swatches of the search space.
- For example, once we have chosen {SA=blue} in the Australia problem, we can conclude that none of the five neighboring variables can take on the value blue. Without taking advantage of constraint propagation, a search procedure would have to consider $3^5 = 243$ assignments for the five neighboring variables; with constraint propagation we never have to consider blue as a value, so we have only $2^5 = 32$ assignments to look at, a reduction of 87%.

Example problem: cryptarithmetic problem

- Cryptarithmetic problems are mathematical puzzles in which the **digits are replaced by letters of the alphabet**.

- **Rules for Solving Cryptarithmetic Problems**
  - **Each Letter, Symbol** represents only **one digit** throughout the problem.
  - Numbers **must not begin with zero** i.e. 0567 (wrong), 567 (correct).
  - The aim is to find the value of each letter in the Cryptarithmetic problems
  - There must be **only one solution** to the Cryptarithmetic problems
  - The **numerical base**, unless specifically stated, is **10.**
  - After replacing letters with their digits, the resulting arithmetic operations must be correct.
  - **Carryover can only be 1** in Cryptarithmetic problems involving 2 numbers.

Example problem: cryptarithmetic problem

To be continued