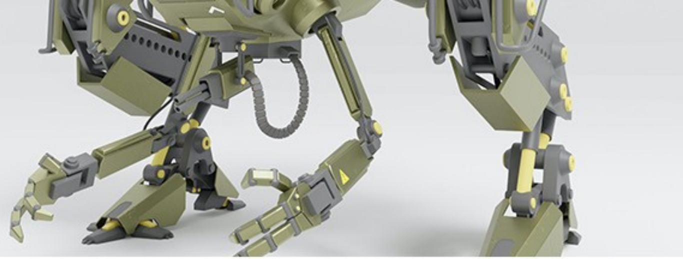
UNIT III

Adeversarial Search and Games

Contents

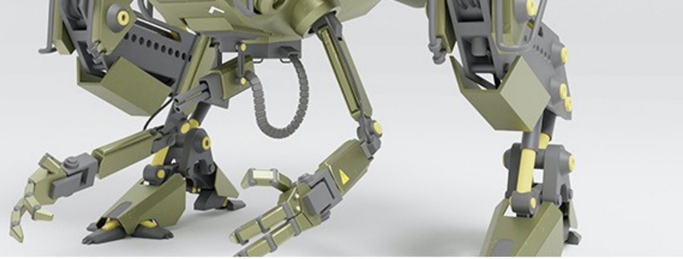
Unit III	Adversarial Search and Games	07 Hours
Game Theory, Optimal Decisions in Games, Heuristic Alpha–Beta Tree Search, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, Limitations of Game Search Algorithms, Constraint Satisfaction Problems (CSP), Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs.		
#Exemplar/Case Studies	Machine Learning At Google: The Amazing Use Case Of Becoming A Fully Sustainable Business	
*Mapping of Course Outcomes for Unit III	CO3, CO4	

Constraint Satisfaction Problems

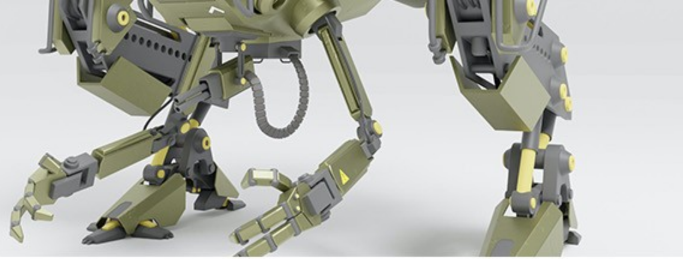


- Finding a solution that meets a set of constraints is the goal of constraint satisfaction problems (CSPs), a type of AI issue.
- Finding values for a group of variables that fulfill a set of restrictions or rules is the aim of constraint satisfaction problems.
- For tasks including resource allocation, planning, scheduling, and decision-making, CSPs are frequently employed in AI.
- There are mainly three basic components in the constraint satisfaction problem:
- \- Variables
- \- Domains
- \- Constraints

Constraint Satisfaction Problems

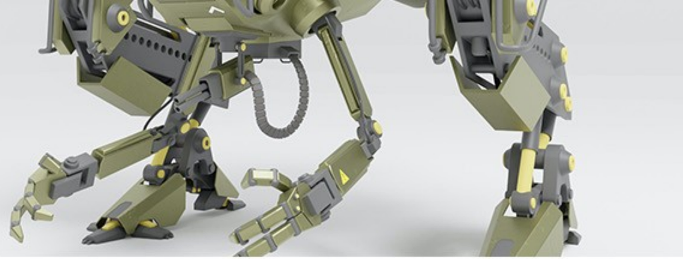


- **Constraint Satisfaction Problems (CSP) representation:**
- - The finite set of variables V1, V2, V3 ……………..Vn.
- - Non-empty domain for every single variable D1, D2, D3 …………..Dn.
- - The finite set of constraints C1, C2 ………., Cm.
  - - where each constraint Ci restricts the possible values for variables,
    - e.g., V1 ≠ V2
  - - Each constraint Ci is a pair <scope, relation>
    - Example: <(V1, V2), V1 not equal to V2>
  - - Scope = set of variables that participate in constraint.
  - - Relation = list of valid variable value combinations.
  - - There might be a clear list of permitted combinations. Perhaps a relation that is abstract and that allows for membership testing and listing.



Constraint Satisfaction Problems

- Constraint Satisfaction Problems (CSP) algorithms:
- **The backtracking algorithm** is a depth-first search algorithm that methodically investigates the search space of potential solutions up until a solution is discovered that satisfies all the restrictions.
- The method begins by choosing a variable and giving it a value before repeatedly attempting to give values to the other variables.
- The method returns to the prior variable and tries a different value if at any time a variable cannot be given a value that fulfills the requirements. Once all assignments have been tried or a solution that satisfies all constraints has been discovered, the algorithm ends.



Constraint Satisfaction Problems

- **Constraint Satisfaction Problems (CSP) algorithms:**
- **The forward-checking** algorithm is a variation of the backtracking algorithm that condenses the search space using a type of local consistency. For each unassigned variable, the method keeps a list of remaining values and applies local constraints to eliminate inconsistent values from these sets. The algorithm examines a variable's neighbors after it is given a value to see whether any of its remaining values become inconsistent and removes them from the sets if they do. The algorithm goes backward if, after forward checking, a variable has no more values.
- Algorithms for propagating constraints are a class that uses local consistency and inference to condense the search space. These algorithms operate by propagating restrictions between variables and removing inconsistent values from the variable domains using the information obtained.

Constraint Satisfaction Problems

- **Real-World Examples of CSP in AI:**
- **Resource Allocation:** In resource allocation problems, variables represent tasks or jobs, domains represent resource assignments, and constraints ensure that resource limits are not exceeded. Soft constraints may be used to optimize resource usage while considering costs.
- **Job Scheduling:** Job scheduling problems involve assigning tasks to available time slots. Constraints include task dependencies and resource constraints. Optimization can aim to minimize makespan or maximize resource utilization.
- **Game Playing:** In game playing, CSPs can represent game states, and constraints define the rules of the game. Global constraints ensure that game moves are legal, and optimization may aim to find the best move based on a scoring function.

Constraint Satisfaction Problems

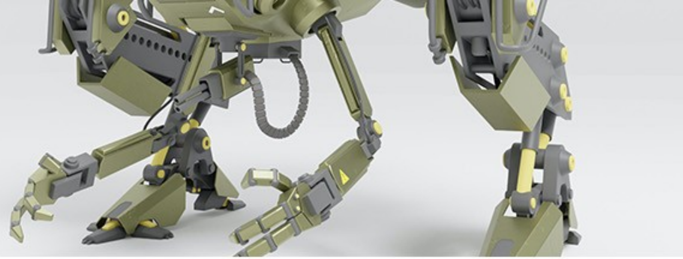
Types of Constraints in CSP-

Unary constraints involve a single variable,
e.g., SA ≠ green

Binary constraints involve pairs of variables,
e.g., SA ≠ WA

Higher-order constraints involve 3 or more variables,
e.g., cryptarithmetic column constraints

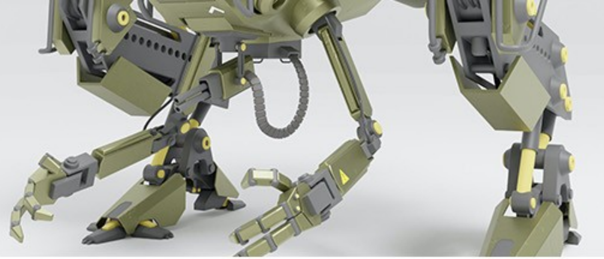
Constraint Satisfaction Problems



Domain Categories in CSP

- In Constraint Satisfaction Problems (CSPs), domain categories refer to the set of possible values that can be assigned to each variable in the problem. The specific categories or domains can vary depending on the nature of the CSP, but here are some common domain categories:
- **Finite Domain:** Variables in many CSPs have finite domains that are made up of discrete values. Examples comprise:
- **Binary Domains:** Domains that only have two values (for binary CSPs, this would be 0 and 1).
- **Integer Domains:** Domains made up of a limited number of integer values, such as 1, 2, 3, and 4, are known as integer domains.
- **Enumeration Domains:** Domains containing a limited number of distinct values, such as “red, green, and blue” in an issue involving color assignment.

Constraint Satisfaction Problems



Domain Categories in CSP

- **Continuous Domains:** Some CSPs contain variables whose domains are continuous, i.e., they can accept any real number falling within a given range. Examples comprise:
- **Real-valued Domains:** Variables may accept any real number that falls within a given range (for example, X [0, 1]).
- **Interval Domains:** Variables are limited to a specific range of real values in the interval domain. e.g., $X \in [-\pi, \pi]$)

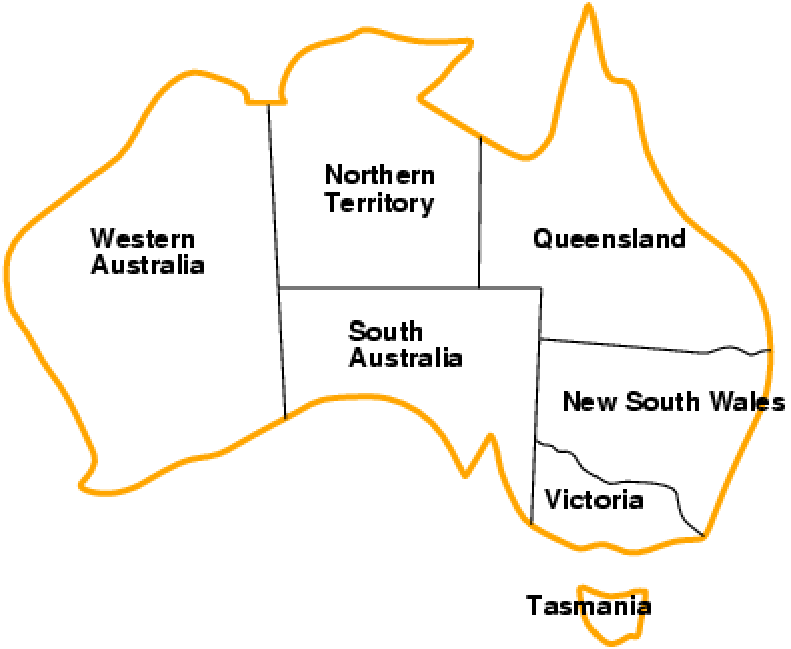
Constraint Satisfaction Problems

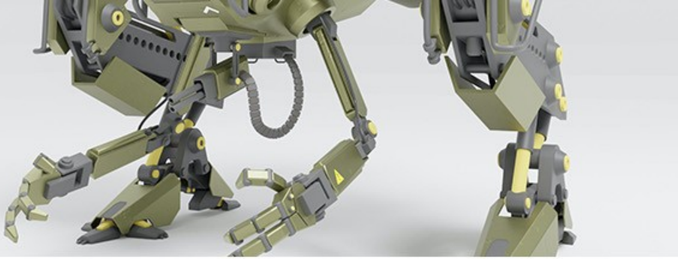
Variables ***WA, NT, Q, NSW, V, SA, T***

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)}

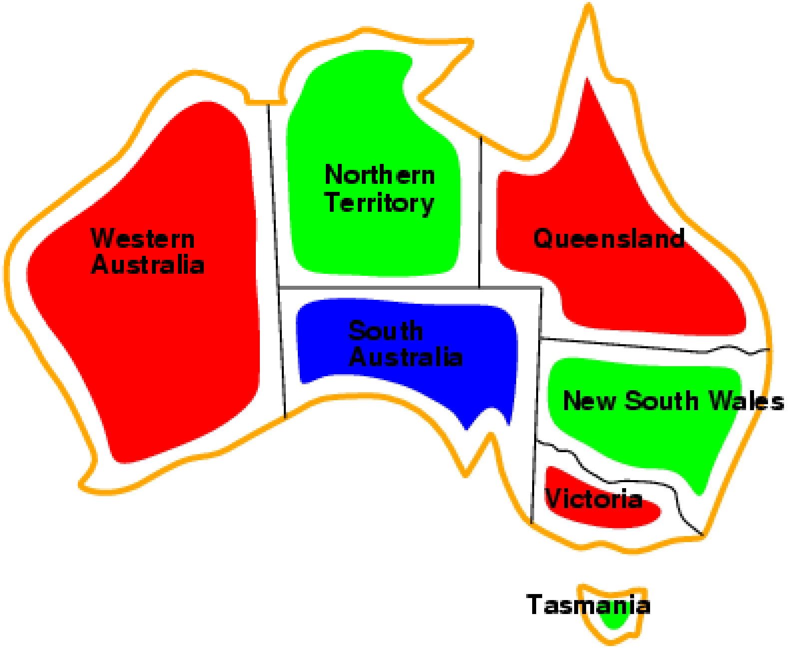




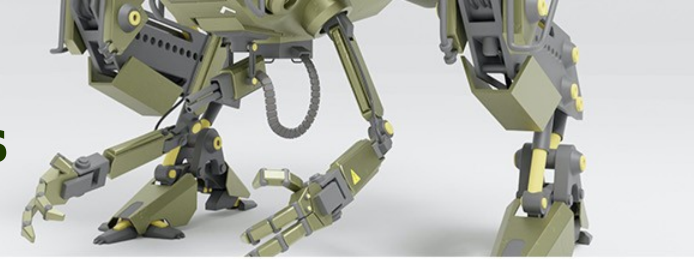
Constraint Satisfaction Problems

Solutions are complete and consistent assignments

e.g., WA = red, NT = green, Q = red, NSW = green,V = red,SA = blue,T = green



Constraint Satisfaction Problems



Example: Cryptarithmetic

Variables: ***F T U W***

R O $\boldsymbol{X_1\ X_2\ X_3}$

Domains: {*0,1,2,3,4,5,6,7,8,9*}

Constraints: *Alldiff (F,T,U,W,R,O)*

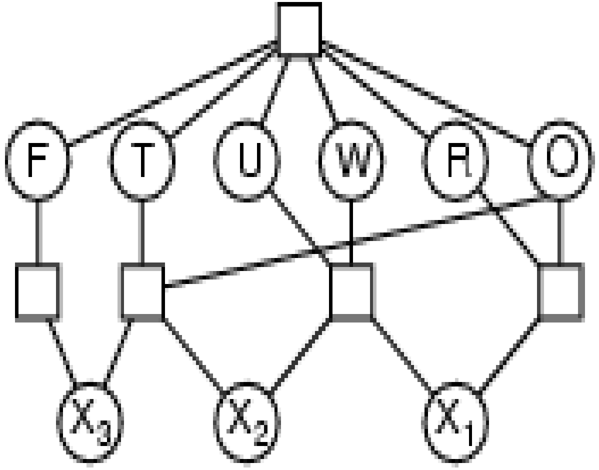
$O + O = R + 10 \cdot X_1$

$X_1 + W + W = U + 10 \cdot X_2$

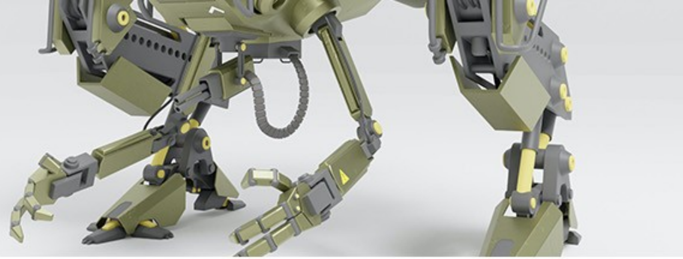
$X_2 + T + T = O + 10 \cdot X_3$

$X_3 = F$, $T \neq 0$, $F \neq 0$

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



Constraint Satisfaction Problems



Real-world CSPs
Assignment problems
e.g., who teaches what class
Timetabling problems
e.g., which class is offered when and where?
Transportation scheduling
Factory scheduling
Notice that many real-world problems involve real-valued variables

Constraint propagation

Constraint propagation is the process of communicating the domain reduction of a decision variable to all of the constraints that are stated over this variable.

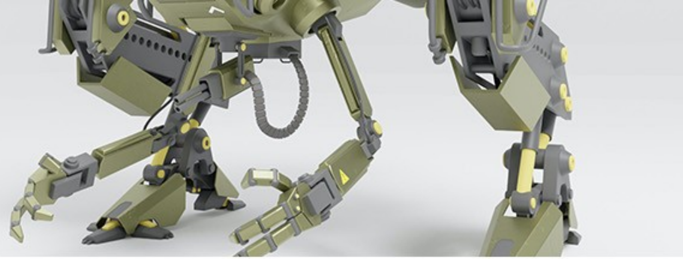
This process can result in more domain reductions. These domain reductions, in turn, are communicated to the appropriate constraints.

Constraint propagation

What are the advantages of constraint propagation?

- One of the main advantages of constraint propagation is that it can reduce the search space and prune branches that lead to dead ends. This can make the problem easier to solve and improve the performance of your algorithm.
- For example, if you use constraint propagation to assign colors to a map, you might find that some regions have only one possible color left, and you can assign it without further exploration.
- Another advantage of constraint propagation is that it can reveal hidden structures and symmetries in the problem, and help you find more elegant and general solutions. For example, if you use constraint propagation to solve a Sudoku puzzle, you might discover that some cells belong to a subset that can be solved independently of the rest.

Constraint propagation

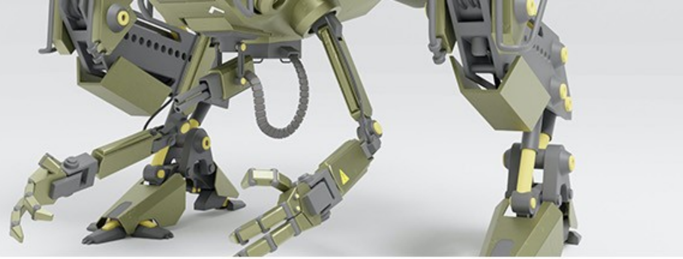


A number of inference techniques use the constraints to infer which variable/value pairs are consistent and which are not. These include node, arc, path, and k-consistent.

constraint propagation: Using the constraints to reduce the number of legal values for a variable, which in turn can reduce the legal values for another variable, and so on.

local consistency: If we treat each variable as a node in a graph and each binary constraint as an arc, then the process of enforcing local consistency in each part of the graph causes inconsistent values to be eliminated throughout the graph.

Constraint propagation



Node consistency
A single variable (a node in the CSP network) is node-consistent if all the values in the variable’s domain satisfy the variable’s unary constraint.

Arc consistency
A variable in a CSP is arc-consistent if every value in its domain satisfies the variable’s binary constraints.

Path consistency
K-consistency: A CSP is k-consistent if, for any set of k-1 variables and for any consistent assignment to those variables, a consistent value can always be assigned to any kth variable.

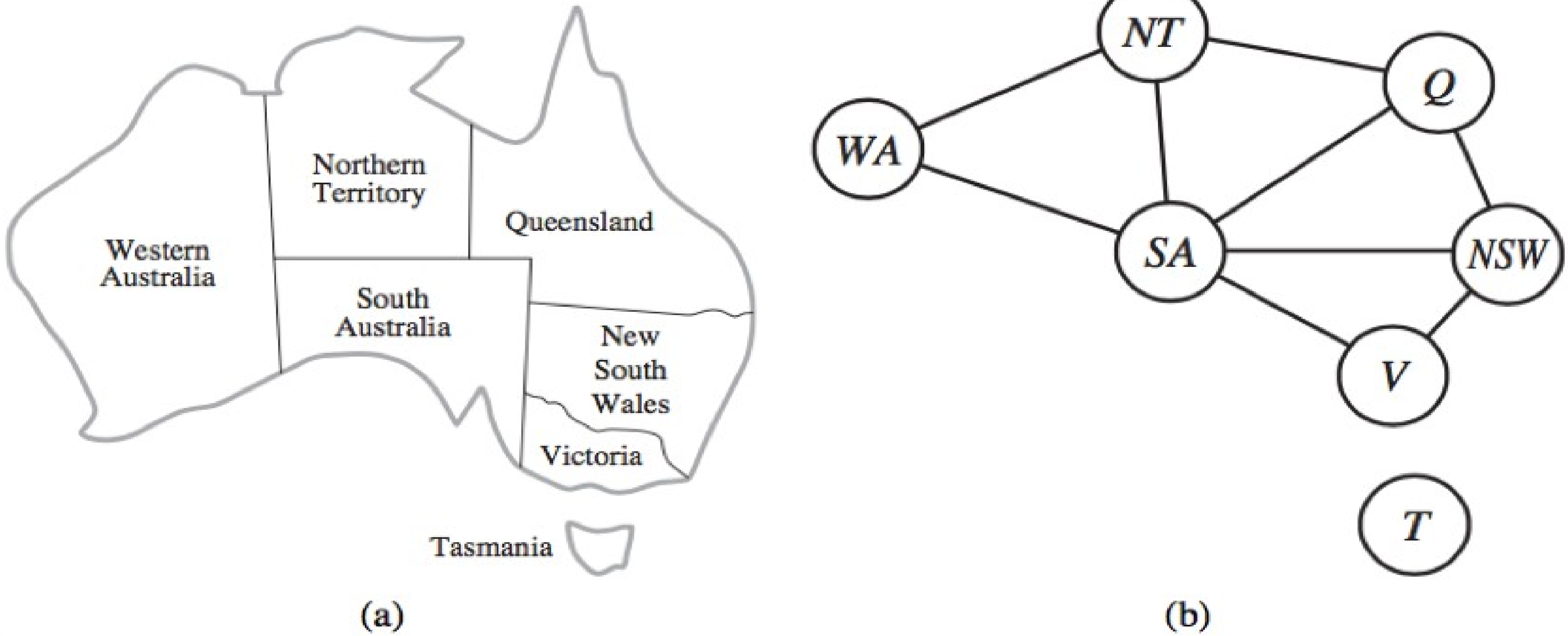


Figure 6.1 (a) The principal states and territories of Australia. Coloring this map can be viewed as a constraint satisfaction problem (CSP). The goal is to assign colors to each region so that no neighboring regions have the same color. (b) The map-coloring problem represented as a constraint graph.