UNIT – 1

18CSC305J / ARTIFICIAL INTELLIGENCE

AI Techniques

AI Applications is spread across the Domains and across the complexities of the problem that includes the following:

- Various Day-to-Day Practical Problems
- Different Identification and Authentication Problems
- Various Classification problems resulting in decision making
- Interdependent and cross-domain problems

Challenges of AI Techniques

AI Techniques need to be build from problem-solving perspectives. the points needed for AI technique are:

- Need for analysis and large amount of data
- Analysis should be followed by the mapping of data with build in knowledge
- Dealing with constantly changing scenarios and situations
- Identification of relevant data and irrelevant data are challenges of AI Techniques.

The Top 4 AI Techniques are

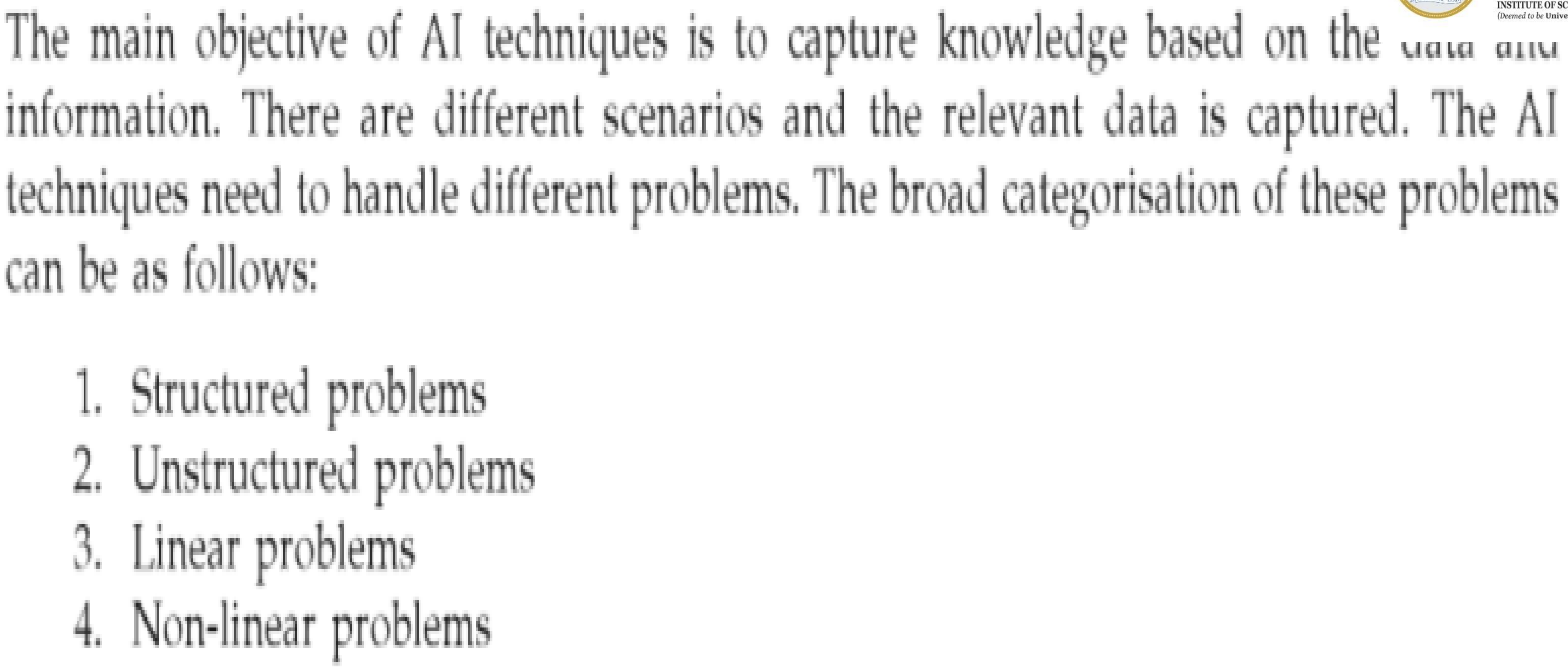
- Machine Learning
- NLP(Natural Language Problem)
- Automation and Robotics
- Machine Vision

Characteristics of AI Techniques

- Intelligence requires Knowledge.
- Knowledge possesses less desirable properties such as:
  - Voluminous
  - Hard to characterize accurately
  - Constantly changing

AI technique is a method that exploits knowledge that should be represented in such a way that:

- Knowledge captures generalization
- It can be understood by people who must provide it.
- It can be easily modified to correct errors.
- It can be used in variety of situations.



The main objective of AI techniques is to capture knowledge based on the data and information. There are different scenarios and the relevant data is captured. The AI techniques need to handle different problems. The broad categorisation of these problems can be as follows:

1. Structured problems
2. Unstructured problems
3. Linear problems
4. Non-linear problems

Problem Solving with AI

- Define the problem.
- Analyze the problem.
- Isolate and represent the knowledge that is necessary for solution.
- Select the best problem solving technique.

Well-Structured Problems

- Well-Structured Problems-Commonly faced problems during day to day life.
- Well-structured problems are those in which the initial state, goal state, and constraints are clearly defined.
- eg:- Solving quadratic equation to find the value of X

Ill-structured problems

- Ill-structured problems are ones students face routinely in everyday life. They include important social, political, economic, and scientific problems

- Ill-Structured Problems-Problems which do not yield a particular answer

- .(eg:- Predicting how to dispose wet waste safely)

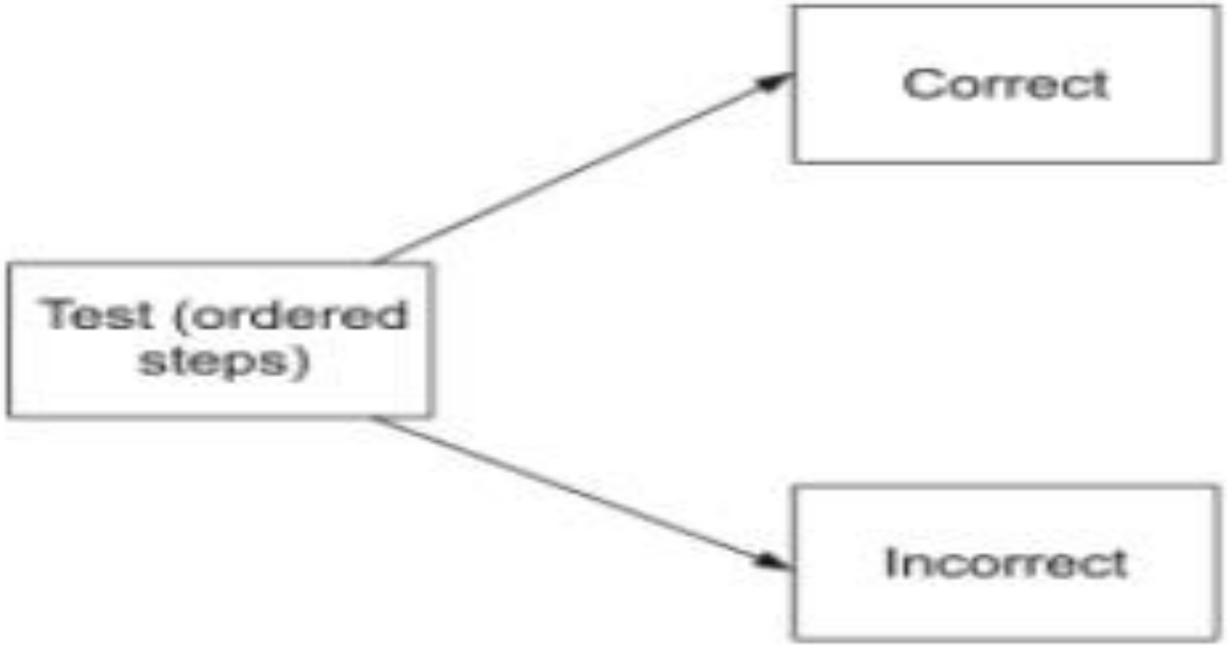


Figure 1.2 A typical well-structured problem analysis.

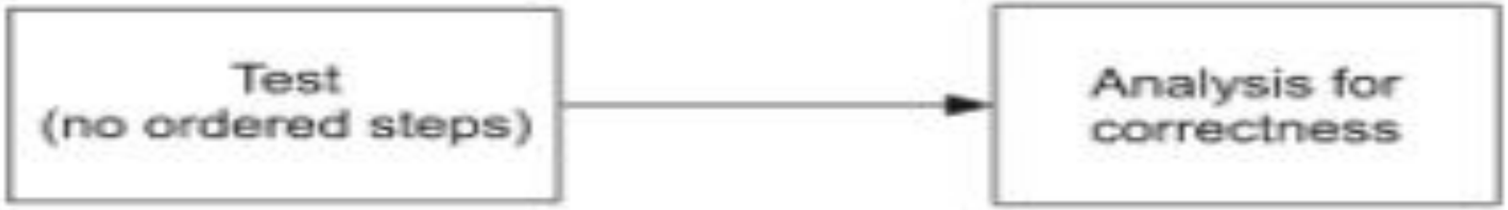
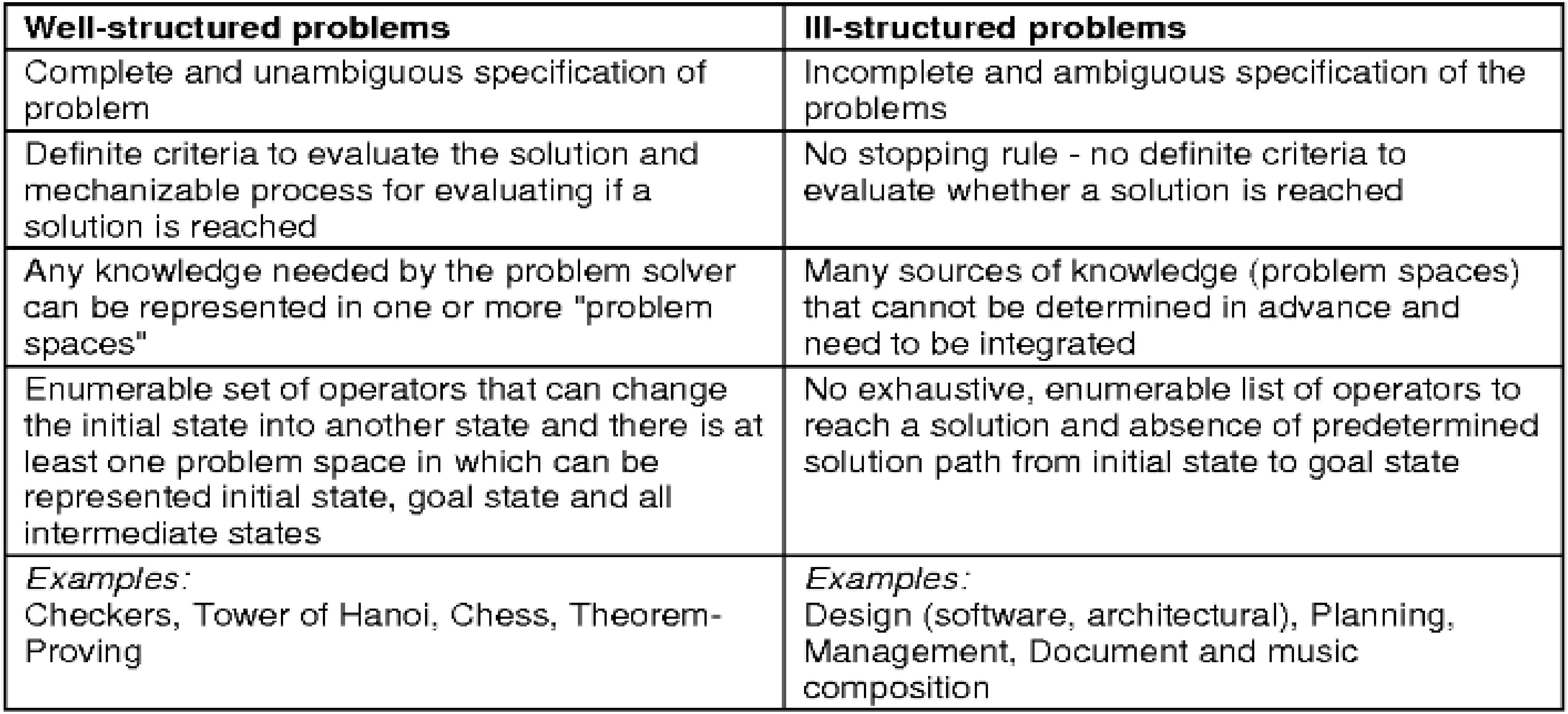


Figure 1.3 A typical ill-structured problem analysis.

Difference between Well-Structured and Ill-Structured Problems



Well-structured problems	Ill-structured problems
Complete and unambiguous specification of problem	Incomplete and ambiguous specification of the problems
Definite criteria to evaluate the solution and mechanizable process for evaluating if a solution is reached	No stopping rule - no definite criteria to evaluate whether a solution is reached
Any knowledge needed by the problem solver can be represented in one or more "problem spaces"	Many sources of knowledge (problem spaces) that cannot be determined in advance and need to be integrated
Enumerable set of operators that can change the initial state into another state and there is at least one problem space in which can be represented initial state, goal state and all intermediate states	No exhaustive, enumerable list of operators to reach a solution and absence of predetermined solution path from initial state to goal state
Examples: Checkers, Tower of Hanoi, Chess, Theorem-Proving	*Examples:* Design (software, architectural), Planning, Management, Document and music composition

- **Linear Problem :**
  - It is the problem which can be solved or where the decision can be obtained by linear solution.

- **Non-Linear Problem:**
  - It is the problem which can be solved or seperated by linear equation.
  - The relationship between input and output is not linear.

AI Models

- To understand the different types of AI learning models, we can use two of the main elements of human learning processes: **knowledge and feedback.**
- From the knowledge perspective, learning models can be classified based on the representation of input and output data points.
- In terms of the feedback, AI learning models can be classified based on the interactions with the outside environment, users and other external factors.

Knowledge-Based Classification

- Factoring its representation of knowledge, AI learning models can be classified in two main types: **inductive and deductive.**
- **Inductive Learning:**
  - This type of AI learning model is based on inferring a general rule from datasets of input-output pairs.
  - Algorithms such as knowledge based inductive learning(KBIL) are a great example of this type of AI learning technique.
  - KBIL focused on finding inductive hypotheses on a dataset with the help of background information.

- **Deductive Learning:**
  - This type of AI learning technique starts with the series of rules and infers new rules that are more efficient in the context of a specific AI algorithm.
  - Explanation-Based Learning(EBL) and Relevance-0Based Learning(RBL) are examples examples of deductive techniques.
  - EBL extracts general rules from examples by “generalizing” the explanation.
  - RBL focuses on identifying attributes and deductive generalizations from simple example.

Feedback-Based Classification

- Based on the feedback characteristics, AI learning models can be classified as **supervised, unsupervised, semi-supervised or reinforced.**
  - **Unsupervised Learning:** Unsupervised models focus on learning a pattern in the input data without any external feedback. Clustering is a classic example of unsupervised learning models.
  - **Supervised Learning:** Supervised learning models use external feedback to learning functions that map inputs to output observations. In those models the external environment acts as a “teacher” of the AI algorithms.

- **Semi-supervised Learning:** Semi-Supervised learning uses a set of curated, labeled data and tries to infer new labels/attributes on new data data sets. Semi-Supervised learning models are a solid middle ground between supervised and unsupervised models.
- **Reinforcement Learning:** Reinforcement learning models use opposite dynamics such as rewards and punishment to “reinforce” different types of knowledge. This type of learning technique is becoming really popular in modern AI solutions.

What is data acquisition in artificial intellige

- Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer.
- Signal conditioning circuitry, to convert sensor signals into a form that can be converted to digital values.

Data acquisition and learning aspects in AI

- Knowledge Discovery
- Computational Learning Theory(COLT)
- Neural and Evolutionary computation
- Intelligent Agent and Multi Agent System
- Multi prespective Integrated Intelligence

Knowledge Discovery

- It is about finding understandable knowledge.
- Machine learning is a field concerned with the study of algorithms that will improve its performance with experience.
- machine learning is more focused on improving performance of an agent.
- Here the mining plays an implicit part.

Computational Learning Theory(COLT)

- In COLT, formal mathematical problems are defined.
- These models helps in analyzing the efficiency and complexity in terms of computation, prediction and feasibility of the algorithm.
- computational learning finds its importance in the field of machine learning,pattern recognition and many more.
- the goal is to inductively learn the target function.
- there are two frameworks for analysing pattern probably approximate correct and mistake bound.

Neural and Evolutionary computation

- The new technique is enabled to speed up the mining of data.
- computational technique that are based on biological properties fall under the category of evolutionary computing.
- In the case of neural computing, the neural behaviour of human being is simulated to enable machine to learn.
- An ANN is formed for specific application like pattern recognition or classification.

Intelligent Agent and Multi Agent System

- It is the core part of intelligent systems, which allows timely decision making in complex scenarios.
- Intelligent agent is the one which is flexible in terms of its action to get the desired outcome.
- It is goal directed and reacts with the environment and acts accordingly.
- The percept of individual agent i always limited.
- For complex tasks and decision making more agents are required to solve the problem that is called MAS.

Multi perspective Integrated Intelligence

- Utilising and exploiting the knowledge from different perspective to build up an intelligent system giving accurate results builds the multi- perspective intelligence.

These approaches work in association with respect to the application they would be suited for. the good understanding of requirements and domain will result in an accurate predictions and decision making for solving a problem.

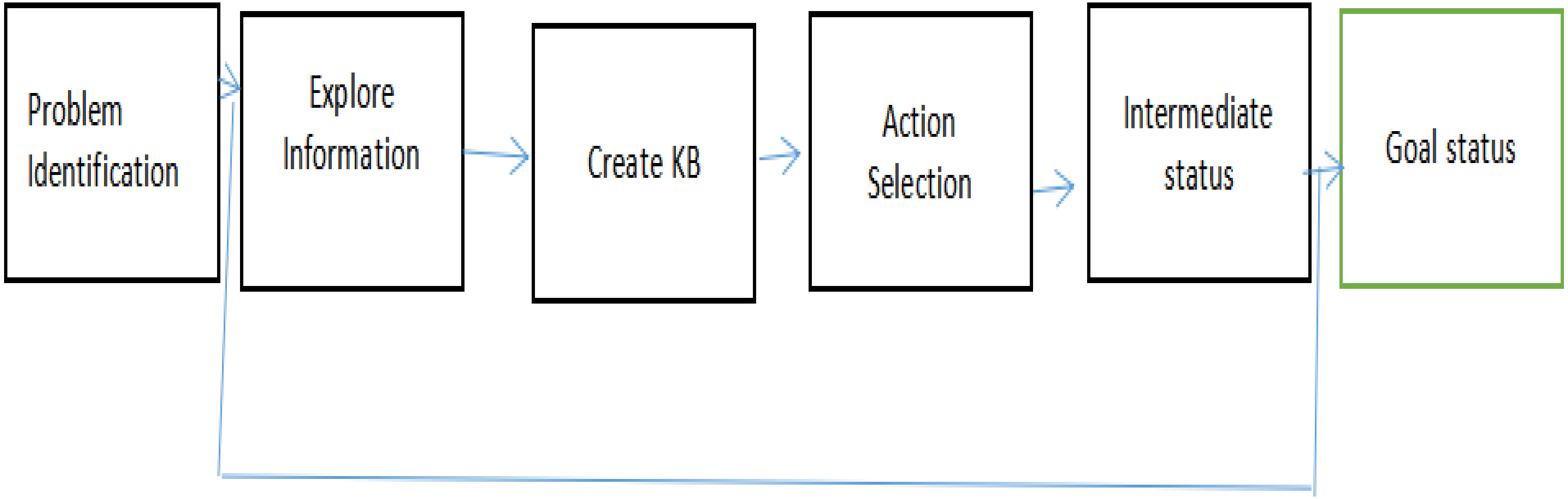
Problem Solving

- It is an area to deal with finding answers for some unknown situations.
- It involves understanding, representation,formulation and solving.
- Simple problem : deterministic procedure.
- Complex / Real world problem - search method
- Problem solving - planning and decision making.

Problem Solving Process

- Problem desired objective is not obvious/computational task.
- problem solving - generating solutions for a given situation.
- AI problem is formulated and solved by searching for a solution in space of possible solutions.
- It may include sophisticated information,storage retrival,information extraction,decision making and so on.

Problem Solving Process



- Problem can be defined with following condition:

1. Every problem is defined in a context. n this context it has certain assumptions under initial condition.
2. Every problem has a well defined objective.
3. Solution to every problem consist of a set of activities.each activity have a state of process.
4. previous knowledge and domain knowledge both are used as the resource during different state in the solution process.

General problem solving techniques

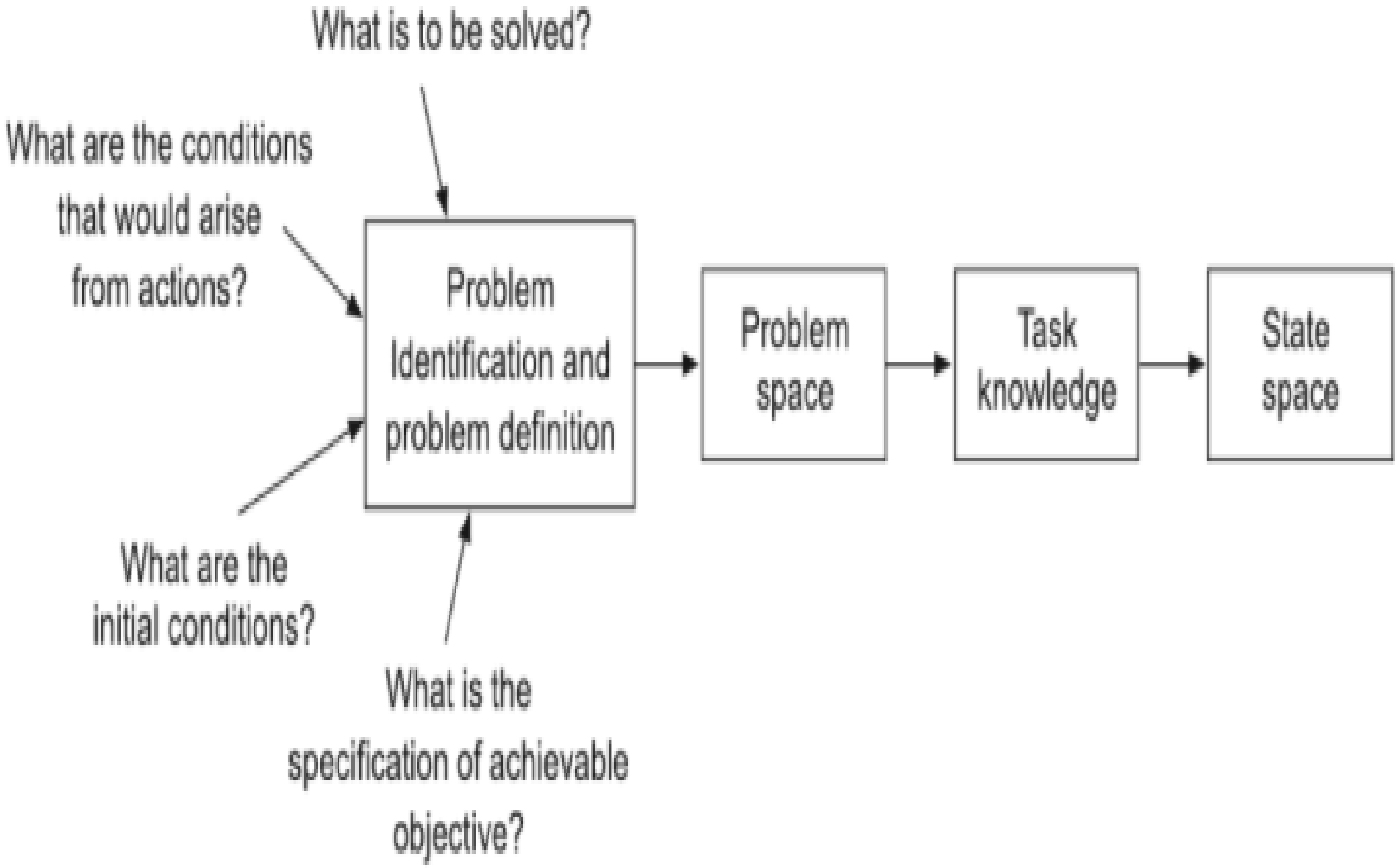
- Problem Definition
- Problem Analysis and representation
- Planning
- Execution
- Evaluating Solution
- Consolidating Gains

Formulating Problems

- First Step - Identification of problem in Problem solvıng process.
  - A problem statement can have description of data,method,procedure and algorithms that are used to solve it.

- The next step is Analysis and Representation of the task knowledge.
  - this is done using state space diagram.an operator is applied to a state to move to next state. this approach is aclled State-Space method.

Problem Formulating steps



What is to be solved?

What are the conditions that would arise from actions?

Problem Identification and problem definition

Problem space

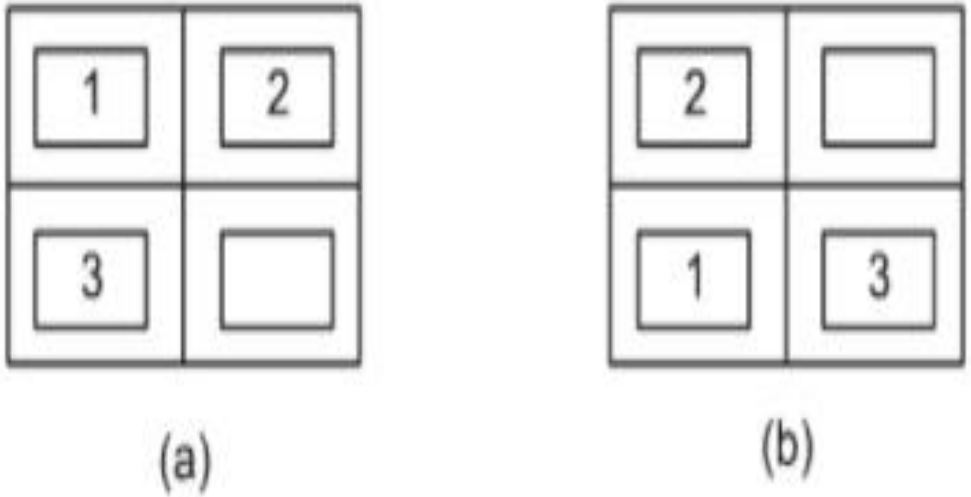
Task knowledge

State space

What are the initial conditions?

What is the specification of achievable objective?

EXAMPLE: Consider a problem where three cells in the four-cell board are filled single digits and one cell is left blank. The game is to change positions of the digi blank cell of the board to arrive at new board positions. The rule of the game is blank cell can change the position with a digit by horizontal or vertical movement. Figure 2.3 represents the initial and the final states of this game.



1	2
3	

(a)

2	
1	3

(b)

Figure 2.3 Initial and final state of the puzzle: (a) Initial state and (b) Final state.

Solution

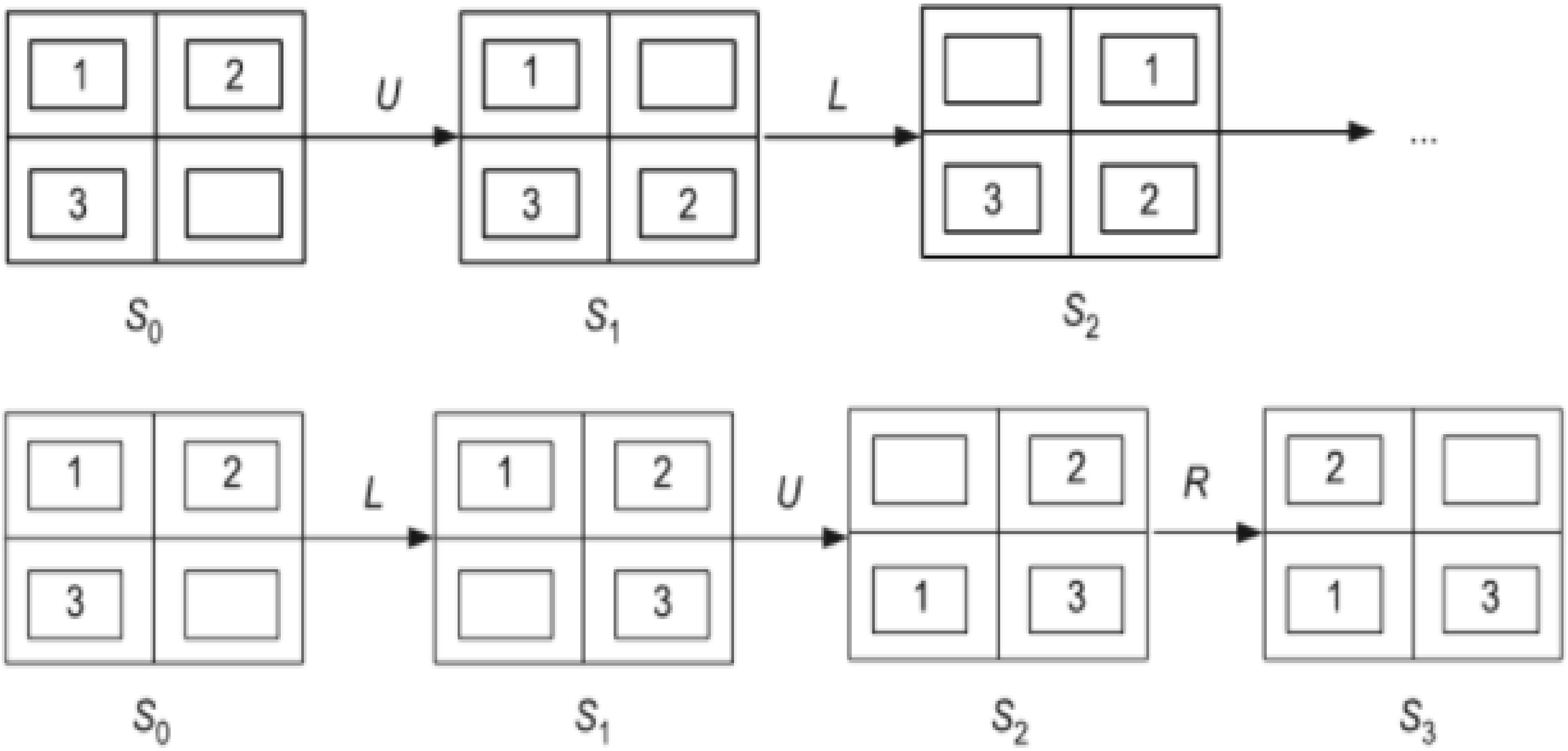
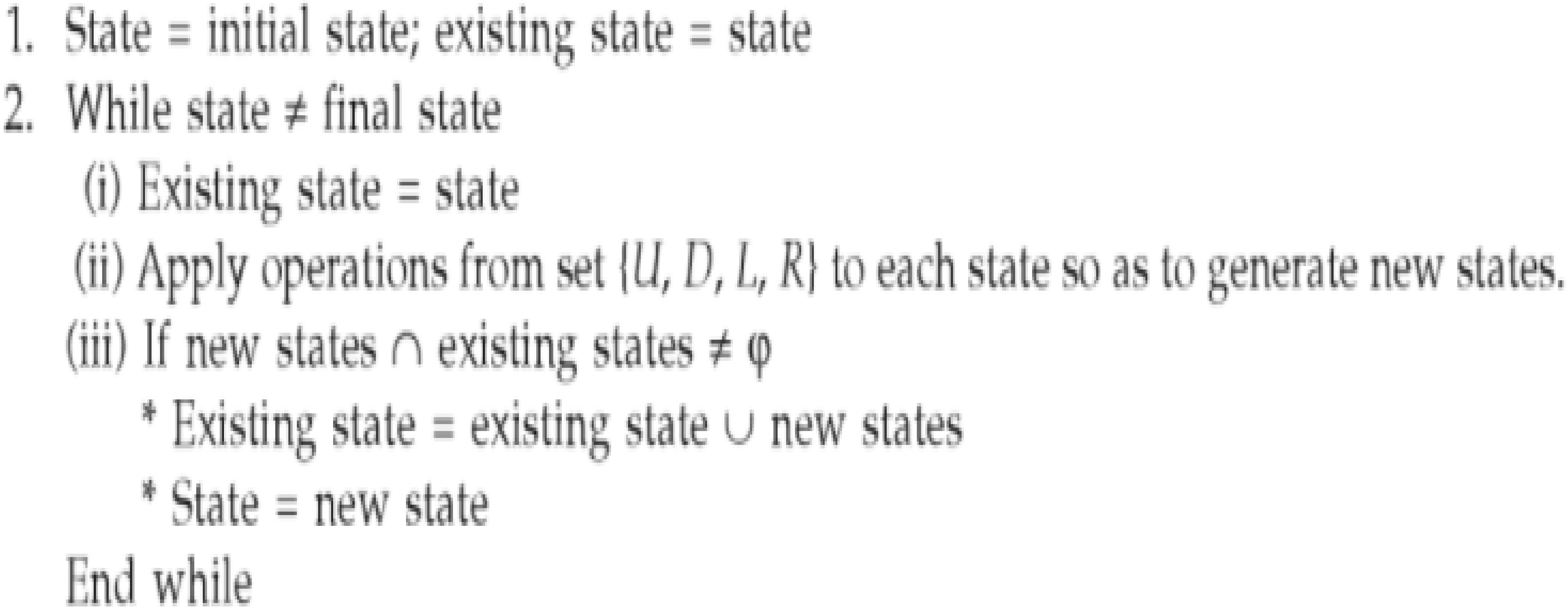
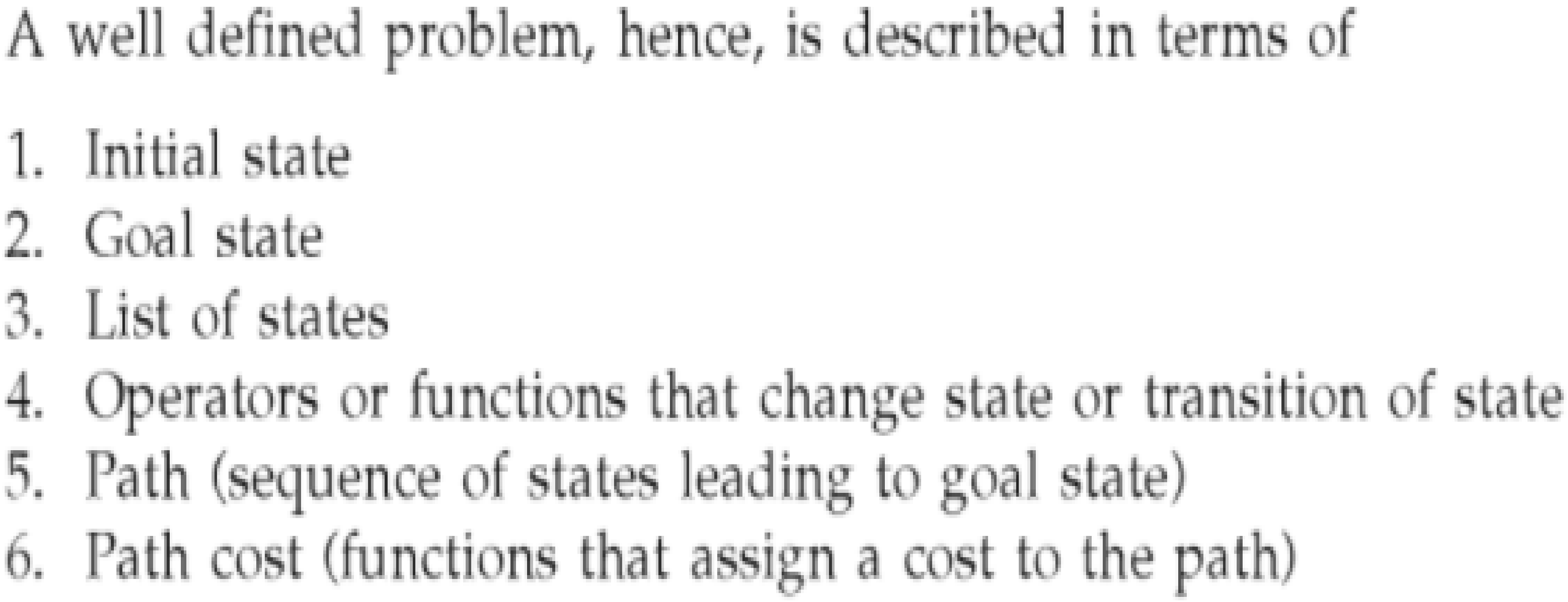


Figure 2.4 State transitions in the puzzle game.

Algorithm



```
1. State = initial state; existing state = state
2. While state $\neq$ final state
     (i) Existing state = state
     (ii) Apply operations from set $\{U, D, L, R\}$ to each state so as to generate new states.
     (iii) If new states $\cap$ existing states $\neq \varphi$
           * Existing state = existing state $\cup$ new states
           * State = new state
   End while
```



A well defined problem, hence, is described in terms of

1. Initial state
2. Goal state
3. List of states
4. Operators or functions that change state or transition of state
5. Path (sequence of states leading to goal state)
6. Path cost (functions that assign a cost to the path)

Problem Types and Characteristics

- Deterministic and observable - single state problem
- Non-Observable - multi state problem-no info about the state
- Non-deterministic or partially observable - new info is added and then operator acts on the state soluion space. Solution space is the tree structure of the states.
- Unknown state space - these are typically exploration problems. States and impact of the actions are not known.
  - Eg-searching address without map.

Problem Space and Search

- Problem is represented as the state space.
- Search plays the major role such as, search is a general algorithm that helps in finding the path in the state space.
- the search algorithm, makes use of control strategy like that of forward or backward search.
- There are two types of strategies
  - informed search and
  - uninformed seach.

Informed Search

- In this approach, a heuristic that is specific to the problem is used to control the flow of solution path.
- Heuristic is a skill based technique used to solve the problem.
- It is based on common sense, rule of thumb,educated guesses or intuitive judgement.
- When speedy process is required, informed search is preferred.

Uninformed Search

- It does not consider any specific nature of the problem.
- It is very simple and can be generalized to any problem.
- This strategy generates all possible states in the state space and checks for the goal state.
- But, this approach is time consuming, as the state space is large.
- This method is used to benchmark results of the algorithm.
- The search methods are evaluated on the basis of completeness, optimality, time and space complexities.

Issues in designing search problem

- State representation and identifying relationships among states
- Proper selection of forward and backward movement to determine optimal path to goal state
- Rule selection

Intelligent Agent

- **Intelligence** is a cummulative manifestation of different activities resulted from learning, sensing, understanding and knowledge augmentation.
- It has an association with environment.
- An **agent** is an entity that can preceive the information and act on that information to acheive the desired outcome.
- **Intelligent agent** is an agent that is capable of making decisions and act most logically in the scenario.

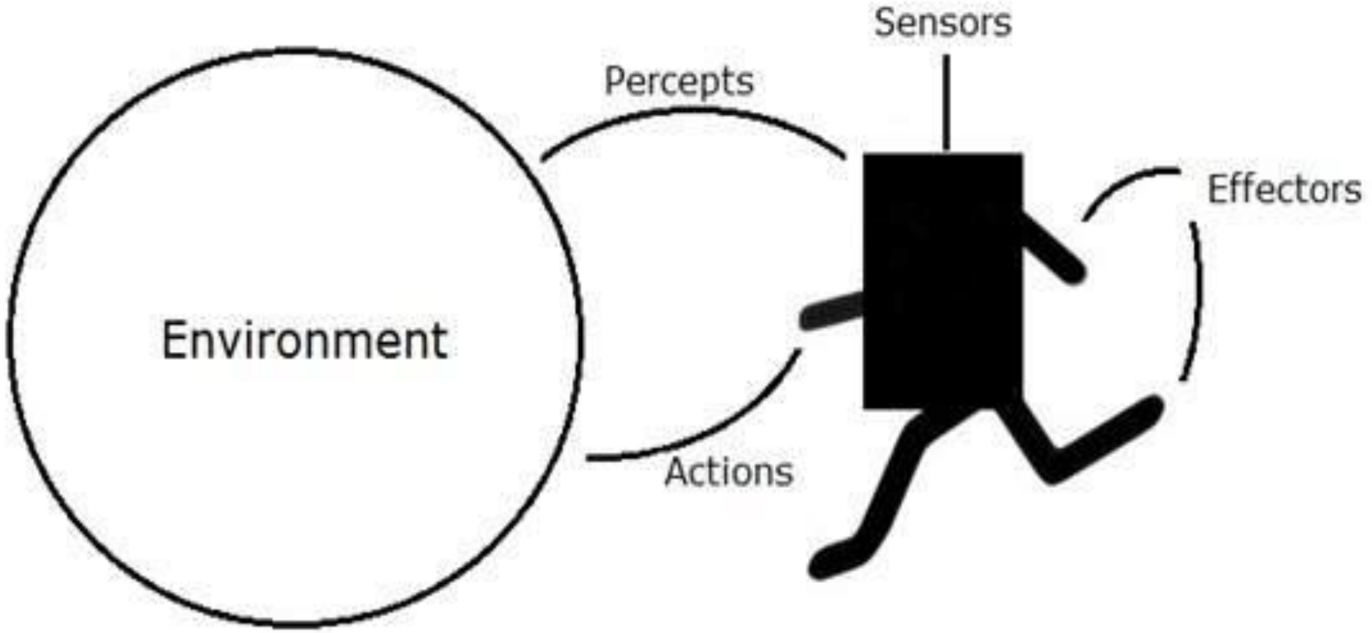
- An agent is anything that can perceive its environment sensors and acts upon that environment through effectors.

- A human agent has sensory organs such as eyes, ears, nose, tongue and skin parallel to the sensors, and other organs such as hands, legs, mouth, for effectors.

- A robotic agent replaces cameras and infrared range finders for the sensors, and various motors and actuators for effectors.

- A software agent has encoded bit strings as its programs and actions.

Agent and Environment



Agent Terminology

- **Performance Measure of Agent**
  - It is the criteria, which determines how successful an agent is.
- **Behaviour of Agent**
  - It is the action that agent performs after any given sequence of precepts.
- **Percept**
  - It is agent’s perceptual inputs at a given instance.
- **Percept Sequence**
  - It is the history of all that an agent has perceived till date.
- **Agent Function**
  - It is a map from the precept sequence to an action

What is Ideal Rational Agent?

- An ideal rational agent is the one, which is capable of doing expected actions to maximize its performance measure, on the basis of :
  - Its percept sequence
  - Its built-in knowledge base
- Rationality of an agent depends on the following:
  - The **performance measures**, which determine the degree of success.
  - Agent’s **Percept Sequence** till now.
  - The agent’s **prior knowledge about the environment**.
  - The **actions** that the agent can carry out.

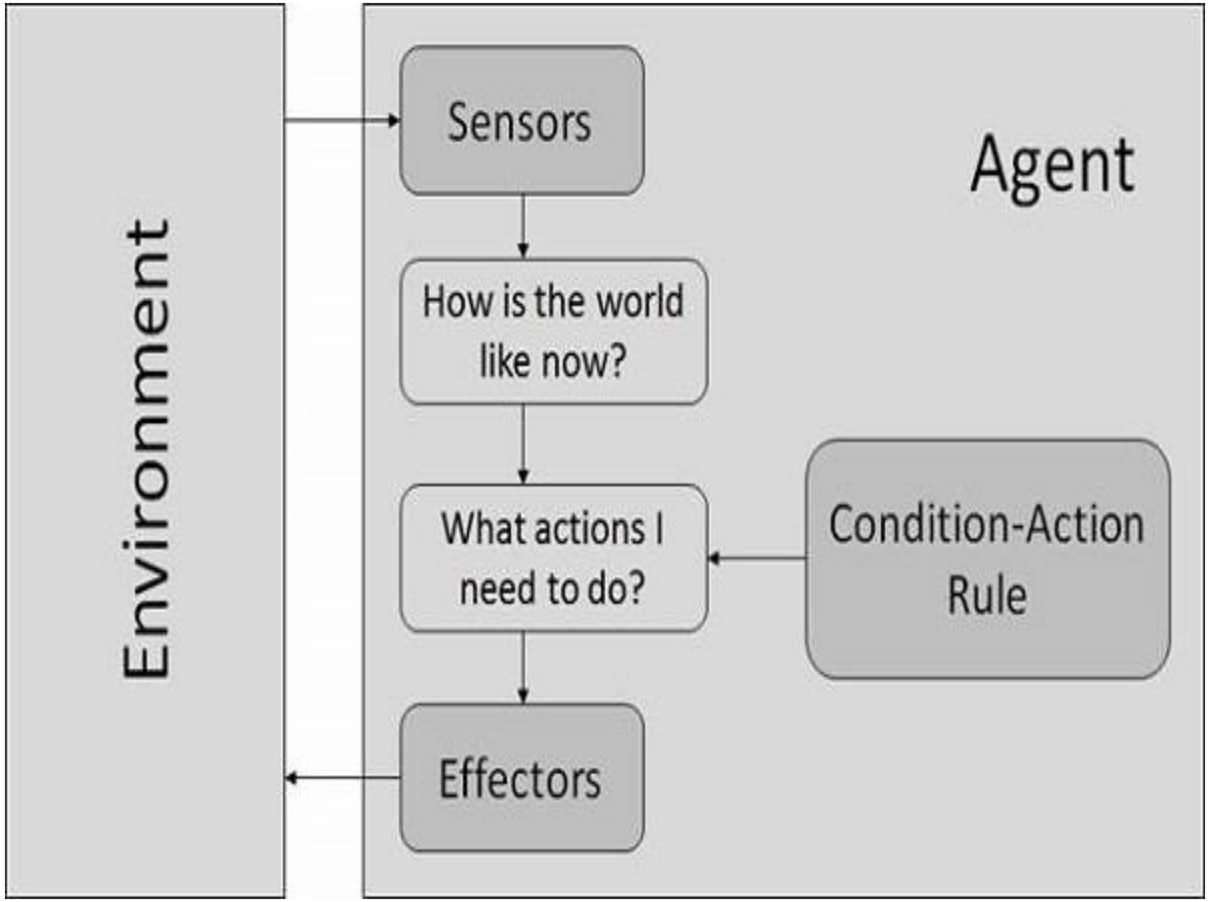
The Structure of Intelligent Agents

Agent’s structure can be viewed as −

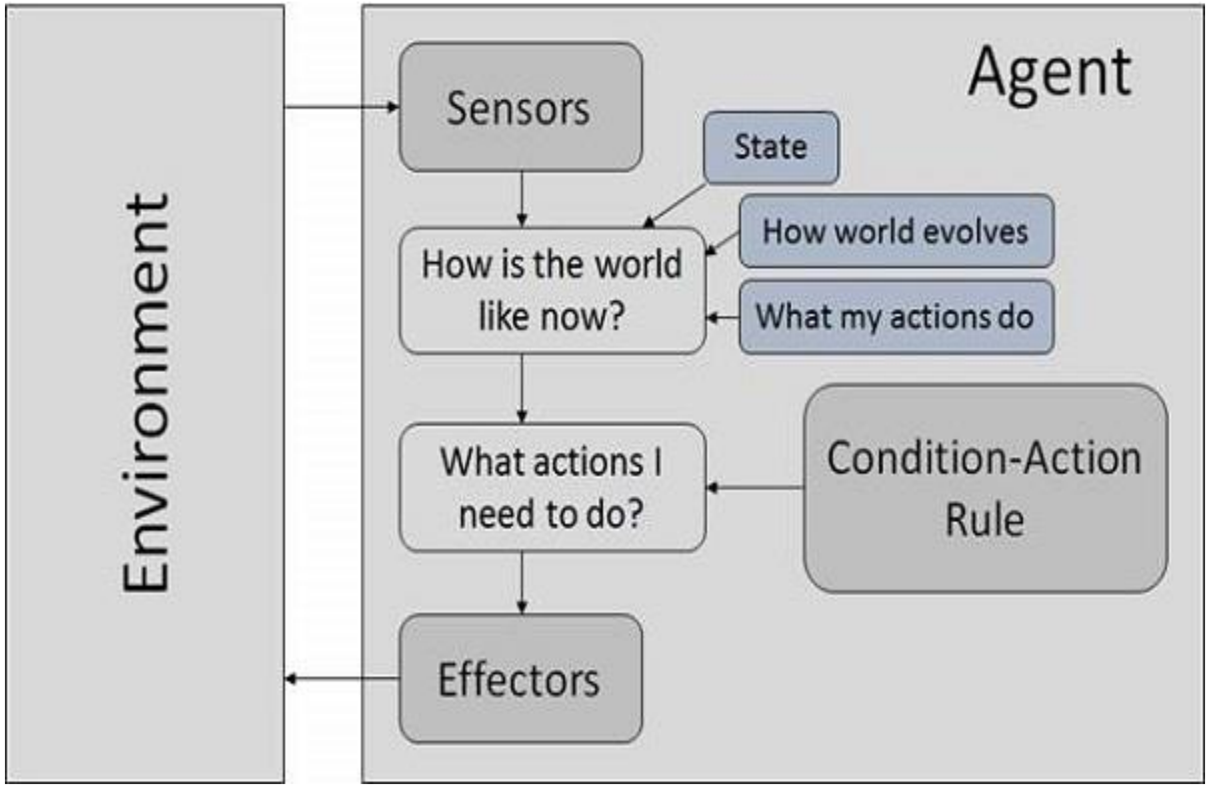
- Agent = Architecture + Agent Program
- Architecture = the machinery that an agent executes on.
- Agent Program = an implementation of an agent function.

- **Condition-Action Rule** − It is a rule that maps a state (condition) to an action.
- Simple Reflex Agents
  - They choose actions only based on the current percept.
  - They are rational only if a correct decision is made only on the basis of current precept.
  - Their environment is completely observable.

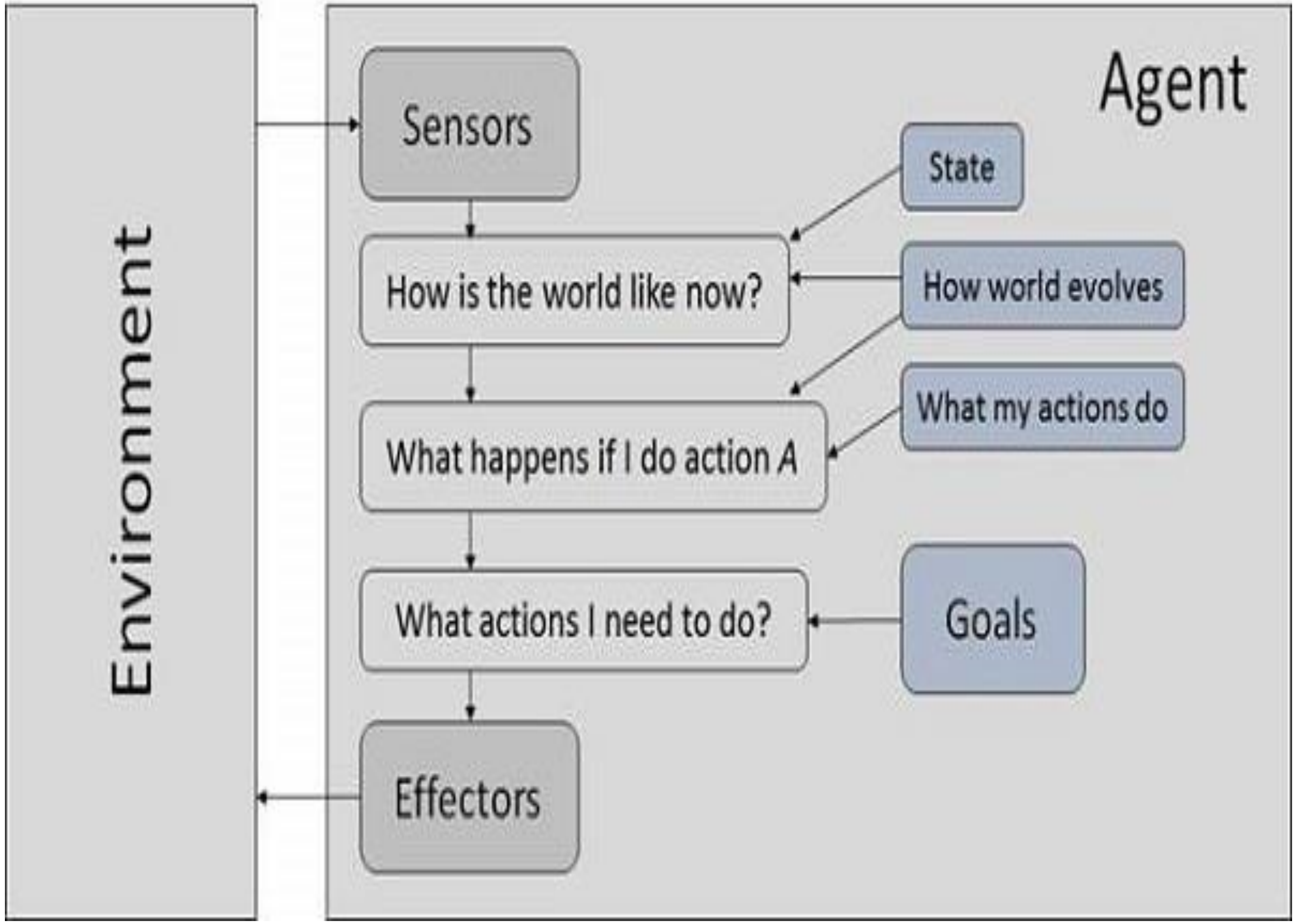
Simple Reflex Agents



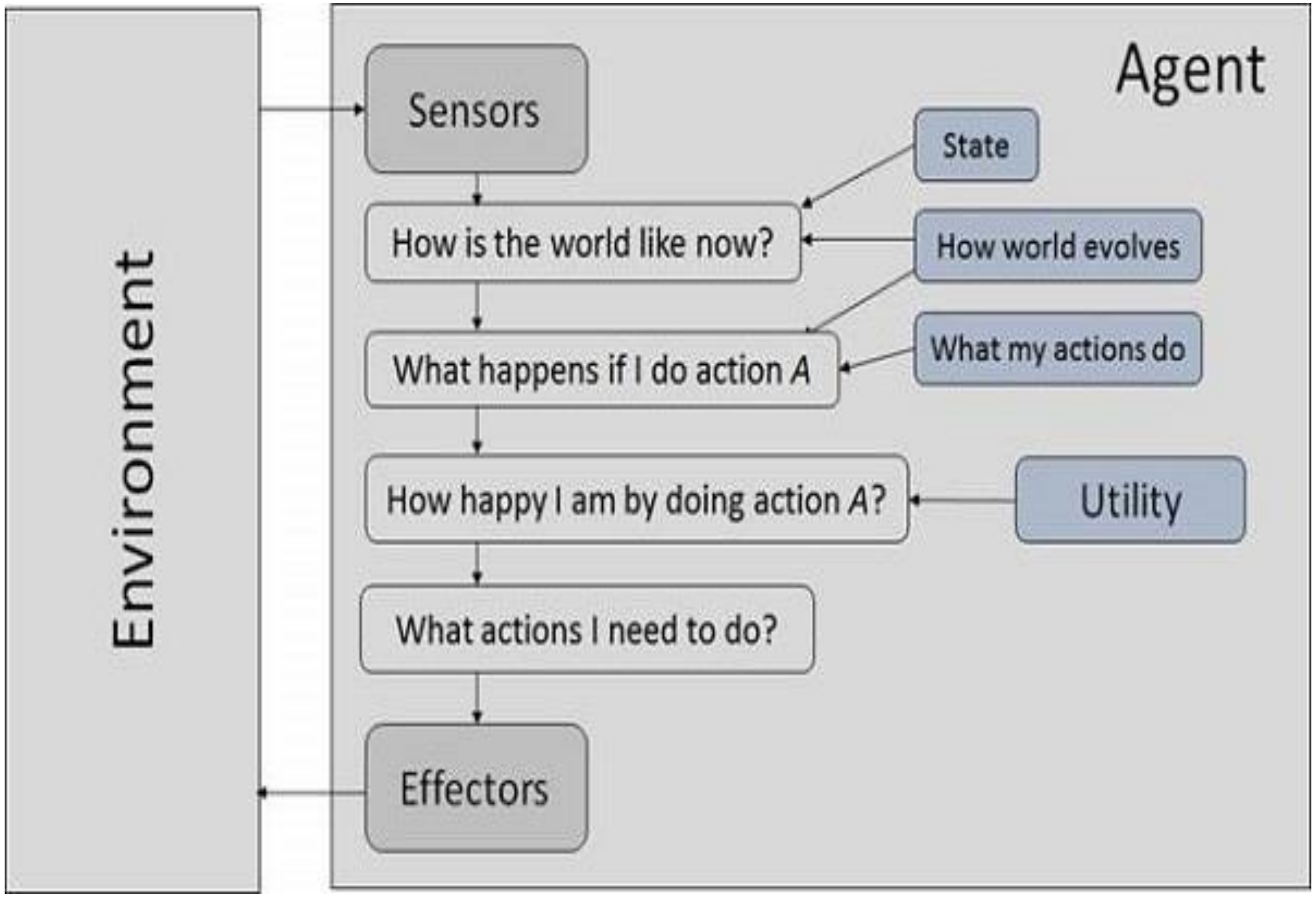
Model Based Reflex Agents



Goal Based Agents


Environment
Agent
Sensors
State
How is the world like now?
How world evolves
What my actions do
What happens if I do action A
What actions I need to do?
Goals
Effectors

Utility Based Agents


Environment
Agent
Sensors
State
How is the world like now?
How world evolves
What happens if I do action A
What my actions do
How happy I am by doing action A?
Utility
What actions I need to do?
Effectors

Flexibilty and Intelligent Agents

- Intelligence demands flexibility. Hence, agents need to be flexible.
- The flexibility equips the agent to negotiate with dynamic scenarios.
- To exhibit the required intelligence, we expect certain properties associated with flexibility from an intelligent agent.
    - **Responsive**
    - **Pro-active**
    - **Social**

- Other properties an intelligent agent should have a

  - **Mobility**
  - **Veracity**
  - **Benevolence**
  - **Rationality**
  - **Learning**

Task Environment and its properties

- The Task Environment is the environment in which the task takes place.
- Clearly defining task environment along with the desired behavior and task is necessary for appropriate design of the agent program.
- PEAS - performance measure, environment, agents actuators, sensors must be specified for design of an intelligent agent.

- For example, Auto-door operating mechanism, The description is:
- P: Timelins of the operation, electricity usage, smooth operations, noise generated, and efficiency
- E: Both sides of the door, tiles and object belonging to it
- A: Motors which push and pull the doors and indicators
- S: cameras on both sides, along with the mechanism deployed for sensing the obstacles.

- So, the PEAS description plays a core part in designing an agent.

Environment Types

- Observable / Partially Observable
- Deterministic / Stochastic
- Discrete / Continuous
- Episodic / Sequential
- Static / Dynamic
- Single agent / Multiple agents

- **Fully observable vs. Partially observable**
  - – A task environment is effectively fully observable, if the agent are able to detect all the aspects that are relevant to its choice of action.
- **Deterministic vs. stochastic**
  - – If a next state of the environment is completely determined by an agent, and any variations are excluded, then the environment is deterministic. Otherwise, it is stochastic.
- **Episodic vs. sequential**
  - – Episodic environment is divided into atomic episodes, each of which consist of agent perceiving and performing a single action. Next episode is independent from actions taken in the previous episode. In contrast, in sequential environment, each decision can affect all the future decisions.

- **Static vs. Dynamic**
  - If an environment is changing while an agent is deliberating, dynamic. Static environments does not change over time. Semidynamic environments does not change, but an agent’s performance score does.
- **Discrete vs. Continuous**
  - Describes a state of the environment, the way time is being handled, and to the percepts and action of an agent. Chess game is discrete (finite number of states, discrete set of actions). Taxi driving is continuous.
- **Single agent vs. multiagent**
  - Either an agent is acting in the environment solely, or engage into certain relationships with other agents, distinguishing them from other objects of the environment (by identifying that its own perormance depends on other agent’s performance). Multiagent environment can be competitive, cooperative, or partially both.

Other aspects of Intelligent Agent

- As the complexity of the system increases, it may require multiple agents.
- problems to be solved in better way, agent with dynamic learning capability is required.
- some complex and practical scenarios in real life can be handled only by multi-agent system with co-operative learning capability.
- some drawbacks include
  - 1. No overall system controllers
  - 2. No global perspective

Constraint satisfaction problems (CSP)

- **Constraint satisfaction problems (CSPs)** are mathematical questions defined as a set of objects whose state must satisfy a number of constraints or limitations.
- CSPs represent the entities in a problem as a homogeneous collection of finite constraints over variables, which is solved by constraint satisfaction methods.
- It is a search procedure that operates in a space of constraints.
- Any problem in the world can mathematically be represented as CSP.
- The solution is typically a state that can satisfy all the constraints.

- A constraint satisfaction problem (CSP) consists of
  - a set of variables,
  - a domain for each variable, and
  - a set of constraints.
- The aim is to choose a value for each variable so that the resulting possible world satisfies the constraints; we want a model of the constraints.

Constraint

- It is mathematical/logical relationship among the attributes of one or more objects.
- It is important to know the type of constraint.
  - Unary Constraint - single variable.
  - Binary Constraint - two variable.
  - Higher order Constraint - 3 or more variables.
- Constraints can restrict the values of variables.

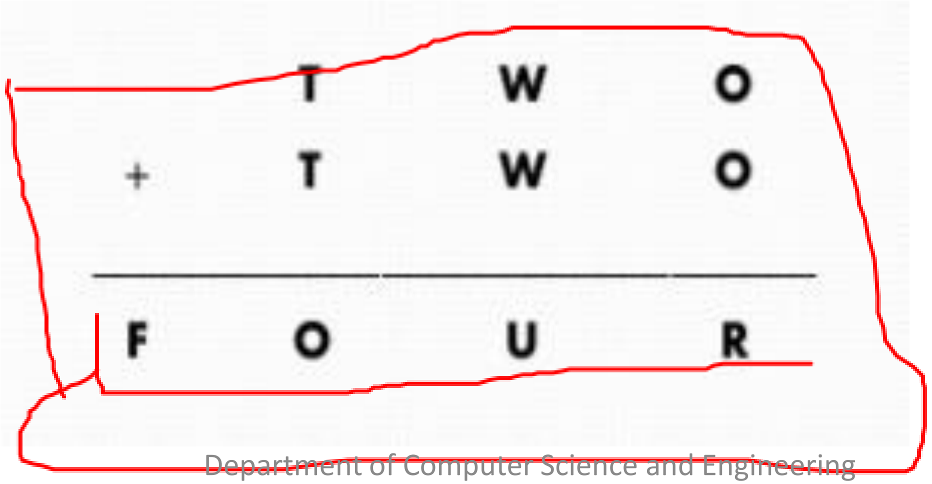
Real-World CSPs

- Assignment problems
  - e.g., who teaches what class
- Timetabling problems
  - e.g., which class is offered when and where?
- □ Transportation scheduling
- □ Factory scheduling

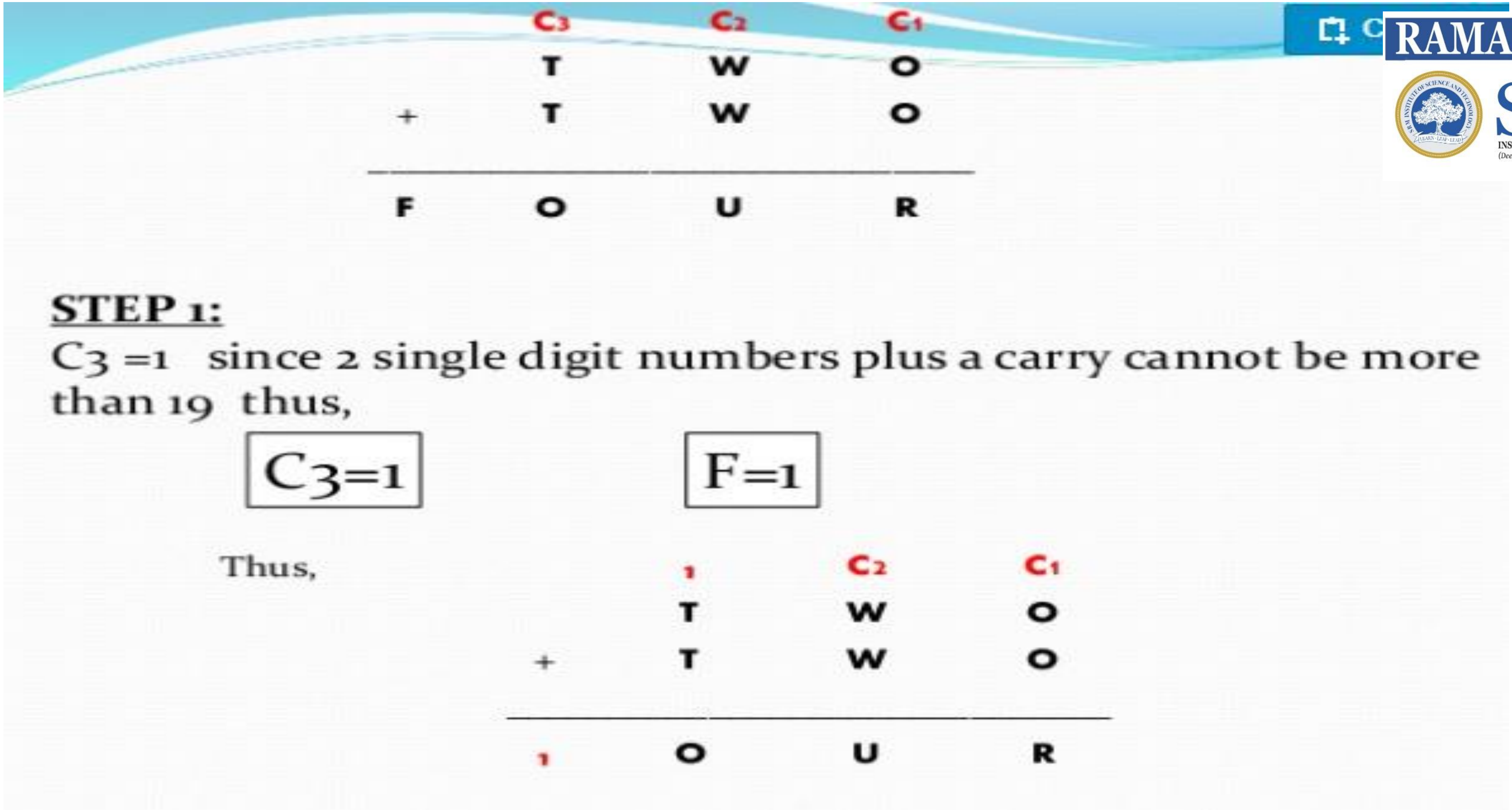
Crypt Arithmetic Puzzles

- Constraints:

1. Variables: can take values from 0-9
2. No two variables should take same value
3. The values should be selected such a way that it should comply with arithmetic properties.



		T	W	O
+		T	W	O
	F	O	U	R



$$\begin{array}{ccccc} & & C_3 & C_2 & C_1 \\ & & T & W & O \\ + & & T & W & O \\ \hline & F & O & U & R \end{array}$$

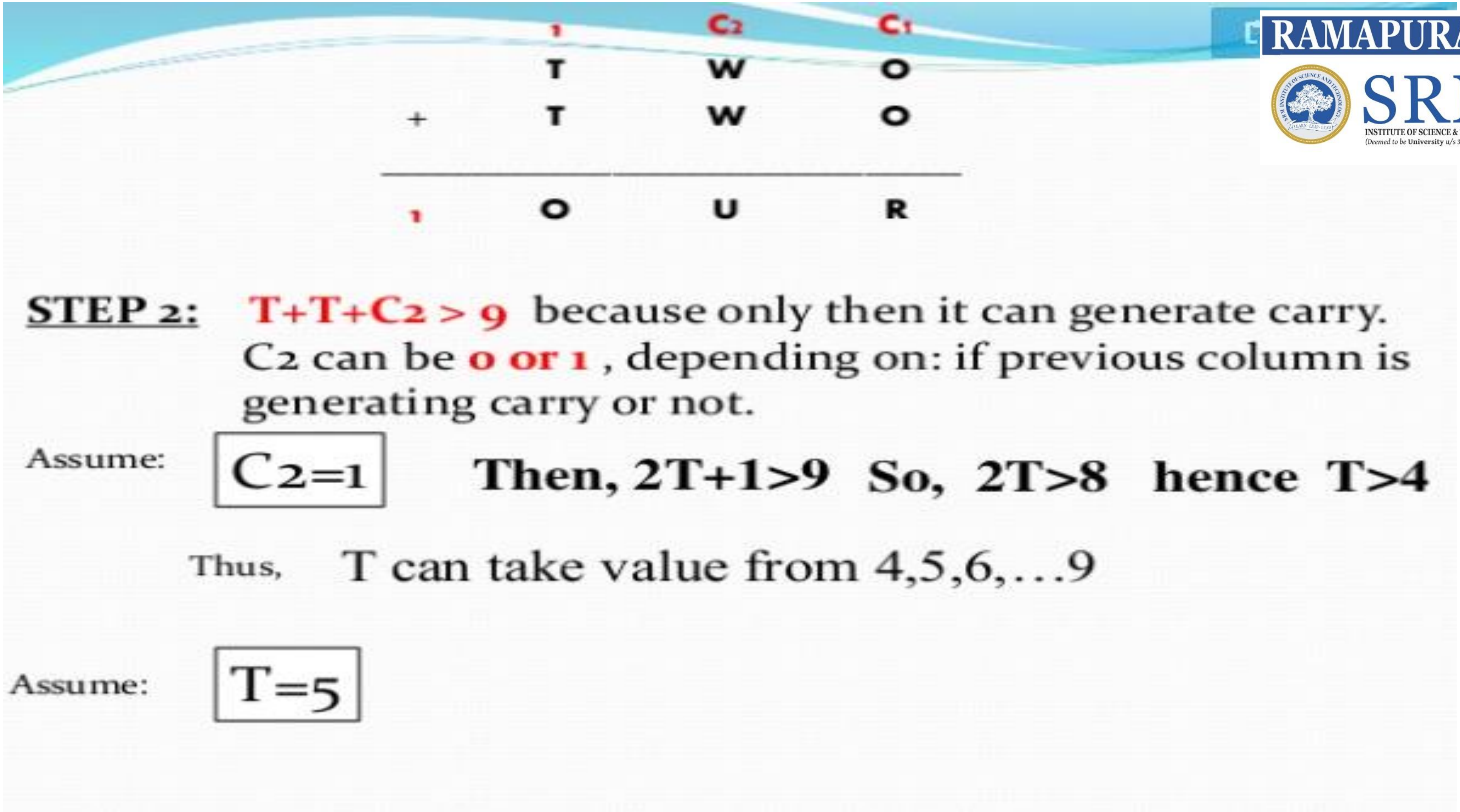
STEP 1:

C3 =1 since 2 single digit numbers plus a carry cannot be more than 19 thus,

$\boxed{C3=1}$ $\boxed{F=1}$

Thus,

$$\begin{array}{ccccc} & & 1 & C_2 & C_1 \\ & & T & W & O \\ + & & T & W & O \\ \hline & 1 & O & U & R \end{array}$$



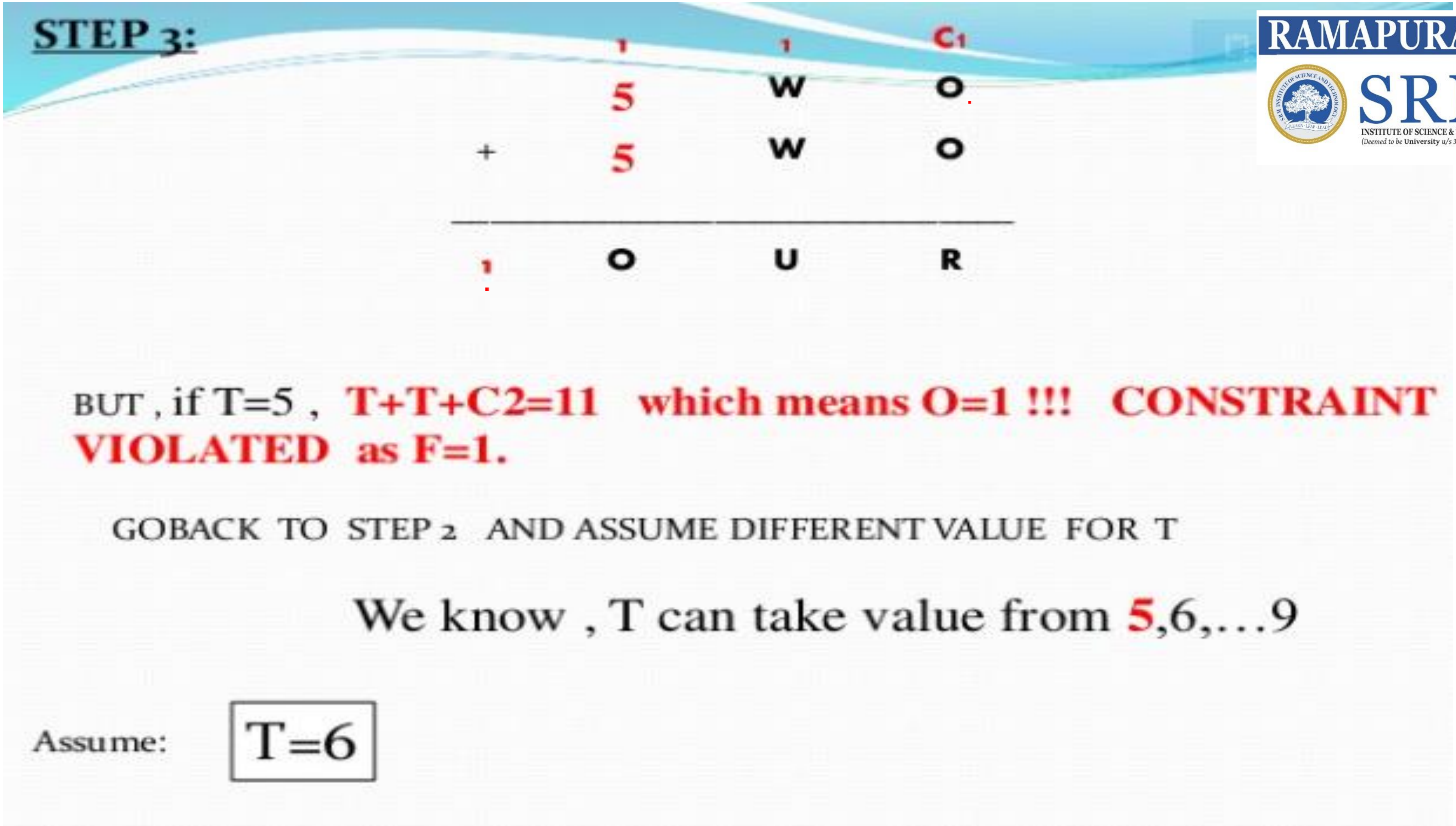
$$
\begin{array}{cccc}
 & 1 & C_2 & C_1 \\
 & T & W & O \\
+ & T & W & O \\
\hline
1 & O & U & R
\end{array}
$$

STEP 2: T+T+C2 > 9 because only then it can generate carry. C2 can be **0 or 1** , depending on: if previous column is generating carry or not.

Assume: $\boxed{C2=1}$ **Then, 2T+1>9 So, 2T>8 hence T>4**

Thus, T can take value from 4,5,6,…9

Assume: $\boxed{T=5}$



STEP 3:

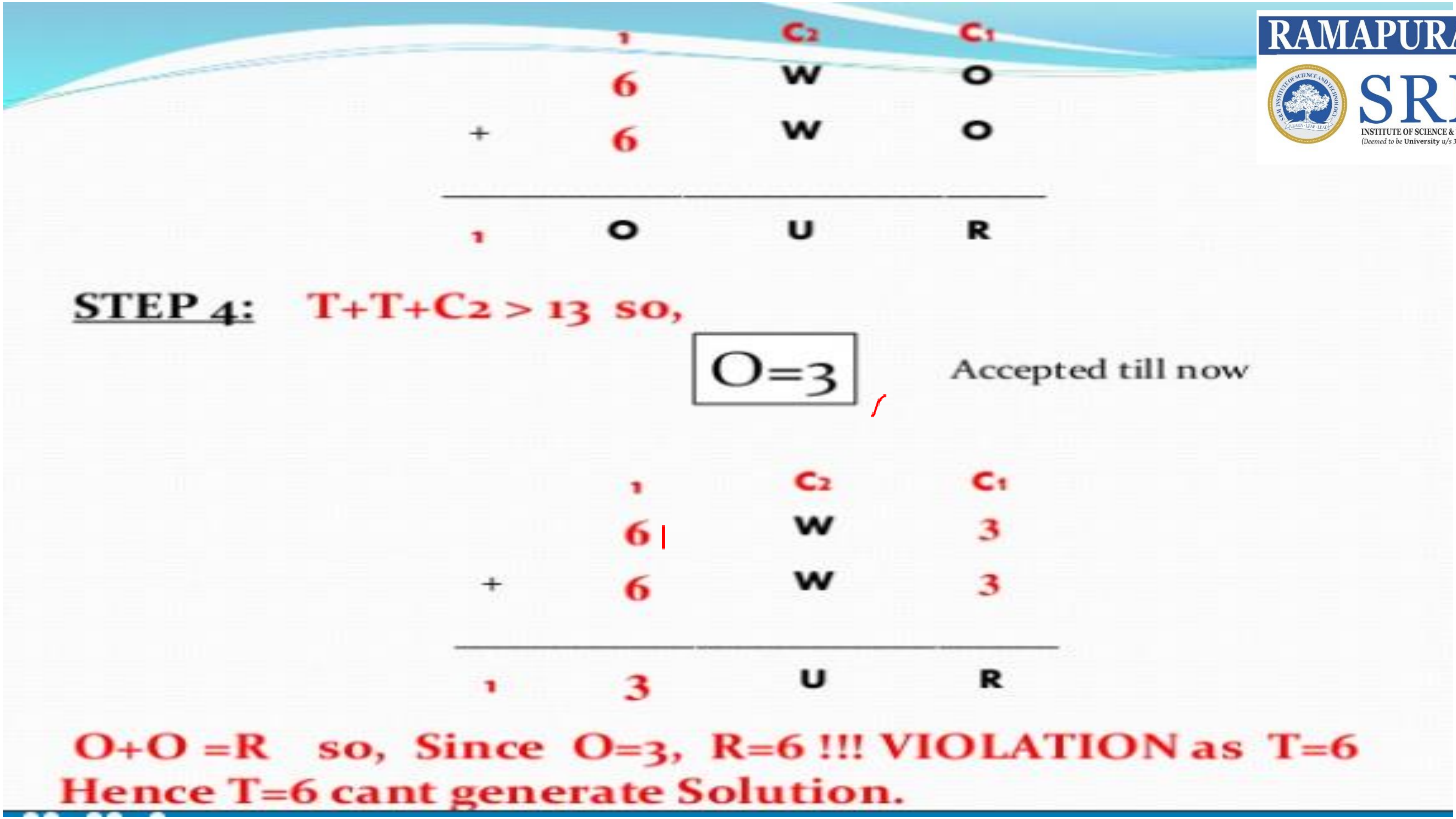
	1	1	C1
	5	W	O
+	5	W	O
1	O	U	R

BUT , if T=5 , **T+T+C2=11 which means O=1 !!! CONSTRAINT VIOLATED as F=1.**

GOBACK TO STEP 2 AND ASSUME DIFFERENT VALUE FOR T

We know , T can take value from **5**,6,…9

Assume: $\boxed{T=6}$



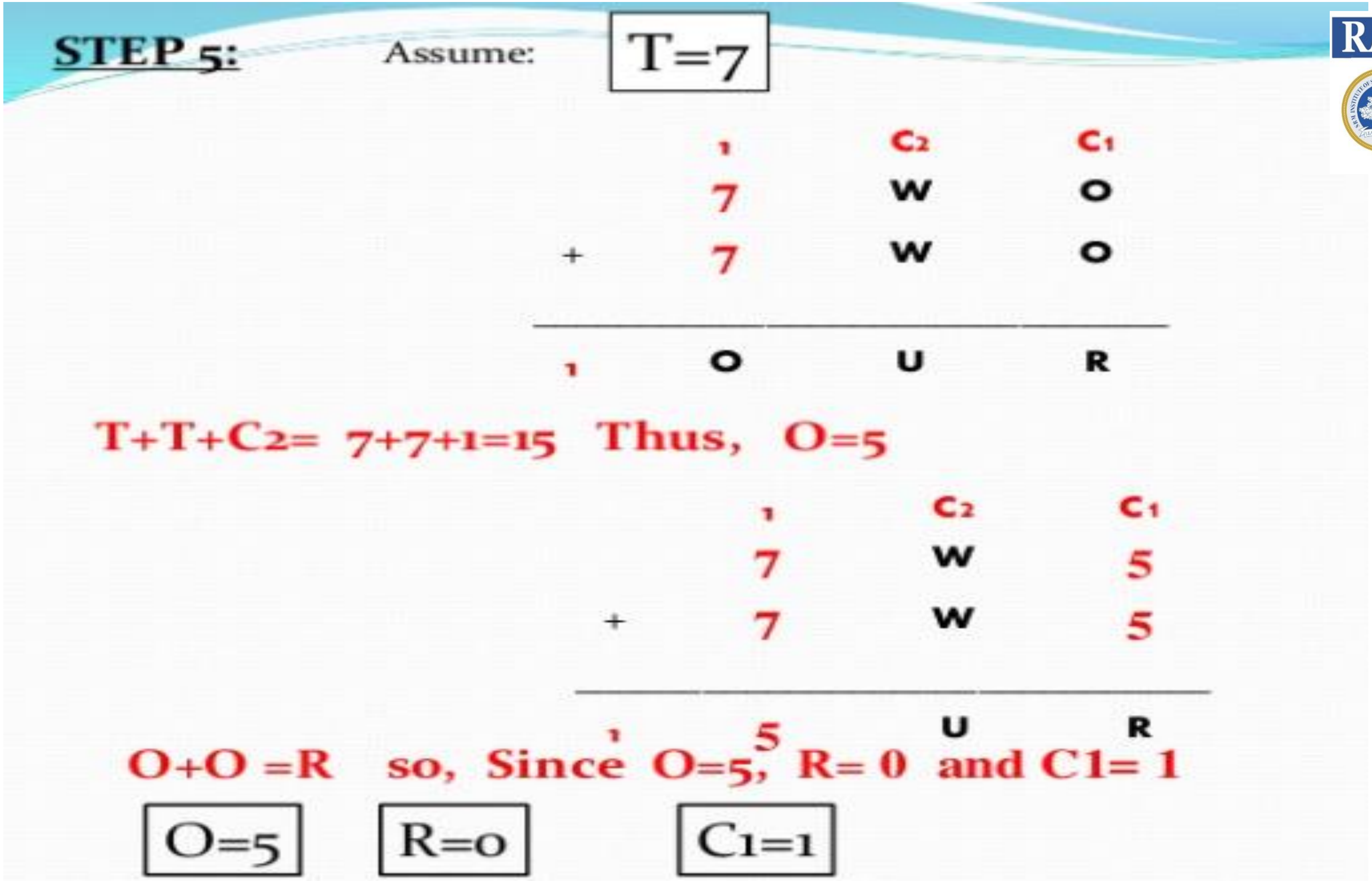
$$\begin{array}{ccccc} & 1 & C_2 & C_1 \\ & 6 & W & O \\ + & 6 & W & O \\ \hline 1 & O & U & R \end{array}$$

STEP 4: T+T+C2 > 13 so,

$\boxed{O=3}$ Accepted till now

$$\begin{array}{ccccc} & 1 & C_2 & C_1 \\ & 6 & W & 3 \\ + & 6 & W & 3 \\ \hline 1 & 3 & U & R \end{array}$$

O+O =R so, Since O=3, R=6 !!! VIOLATION as T=6
Hence T=6 cant generate Solution.



STEP 5: Assume: $\boxed{T=7}$

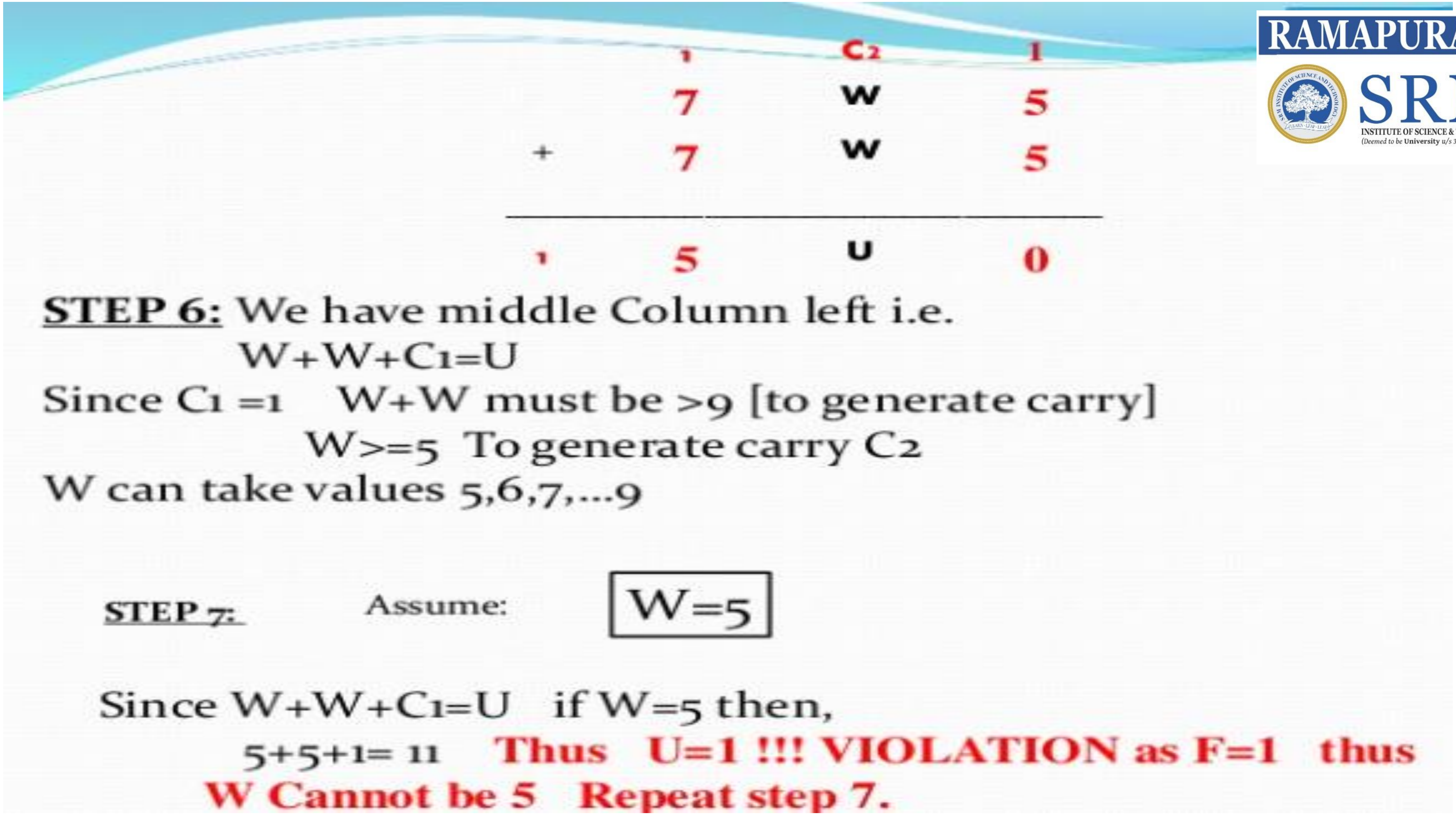
$$\begin{array}{cccc} & 1 & C_2 & C_1 \\ & 7 & W & O \\ + & 7 & W & O \\ \hline 1 & O & U & R \end{array}$$

$T+T+C_2= 7+7+1=15$ Thus, $O=5$

$$\begin{array}{cccc} & 1 & C_2 & C_1 \\ & 7 & W & 5 \\ + & 7 & W & 5 \\ \hline 1 & 5 & U & R \end{array}$$

$O+O = R$ so, Since $O=5$, $R= 0$ and $C1= 1$

$\boxed{O=5}$ $\boxed{R=0}$ $\boxed{C_1=1}$



$$\begin{array}{cccc} & 1 & C_2 & 1 \\ & 7 & W & 5 \\ + & 7 & W & 5 \\ \hline 1 & 5 & U & 0 \end{array}$$

STEP 6: We have middle Column left i.e.

$W+W+C_1=U$

Since $C_1 = 1$ $W+W$ must be >9 [to generate carry]

$W>=5$ To generate carry C_2

W can take values 5,6,7,...9

STEP 7: Assume: $\boxed{W=5}$

Since $W+W+C_1=U$ if $W=5$ then,

$5+5+1= 11$ **Thus U=1 !!! VIOLATION as F=1 thus W Cannot be 5 Repeat step 7.**

STEP 8: Assume: W=6

Since W+W+C1=U if W=6 then,

6+6+1= 13 **Thus U=3 which is Accepted**

U=3

THUS AT THIS STATE SINCE ALL THE VARIABLES HAVE BEEN ASSIGNED VALUES WHICH COMPLY WITH CONSTRAINTS GIVEN, WE HAVE REACHED FINAL STATE!!

S E

K

M I C K

+ J A C K

J O H N

1) O - 1

2) same

3) A

T, W, O, F, U, R

0 1 1 01

	1	1	1
+	T 7	W 6	O 5
	T 7	W 6	O 5
1	5	3	0
F	O	U	R

Constraint Domain

- It describes different constrainers, operators, arguments, variables and their domains.
- It consists of:

1. Legal set of operators
2. Set of variables
3. Set of all types of functions
4. Domain variables
5. Range of variables

constraint domain is five-tuple and represented as

D={var,f,O,dv,rg}

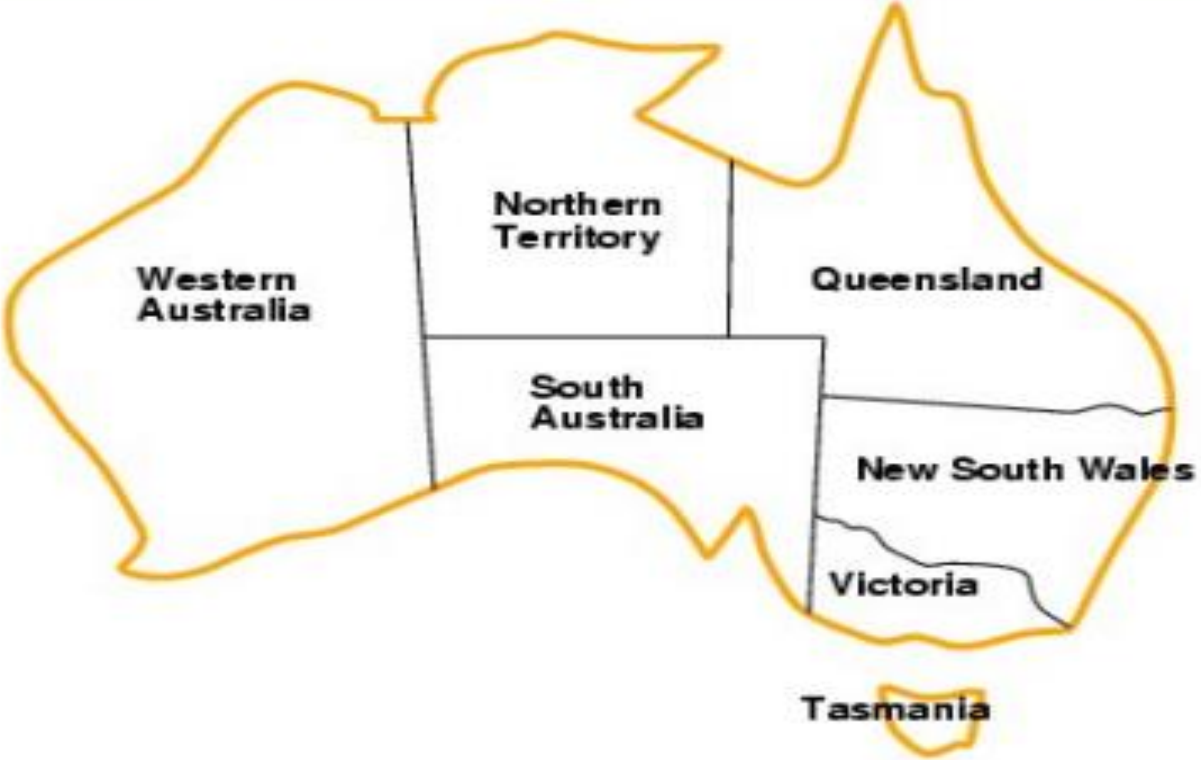
- A constraint without conjuction is reffered as primiti constraint.
- A conjuction of primitive constraints is called as non-primitive constraints or generic constraints.

- The constraint problem can be visualisec as a constraint graph.
- nodes represents the groups and the arcs define the constraint
- One of the prime benefits is the easier representation of problem in the form of a standard pattern.

CSP as a Search Problem

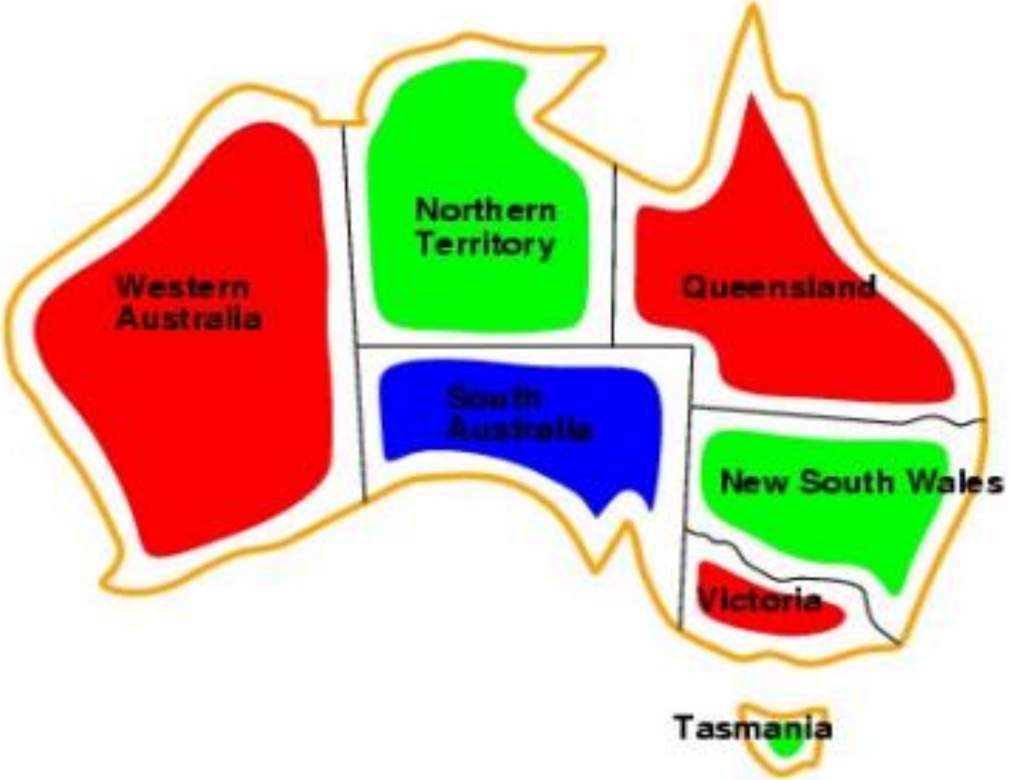
- Initial state:
  - {} – all variables are unassigned
- Successor function:
  - a value is assigned to one of the unassigned variables with no conflict
- Goal test:
  - a complete assignment
- Path cost:
  - a constant cost for each step
- Solution appears at depth n if there are n variables

Example: Map Coloring



- Variables: $X = \{WA, NT, Q, NSW, V, SA, T\}$ ✓
- Domains: D_i = {red, green, blue} ✓
- Constraints: adjacent regions must have different colors

Solution: Complete and Consistent Assig

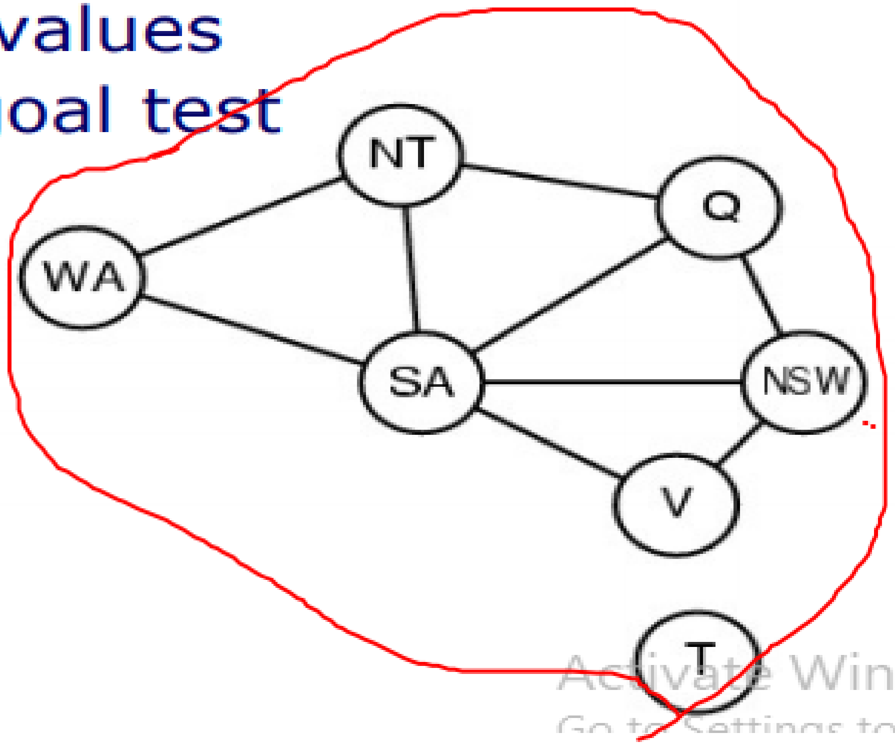


- Variables: X = {*WA, NT, Q, NSW, V, SA, T* }
- Domains: D_i = {red, green, blue}
- Constraints: adjacent regions must have different colors
- Solution? {WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = red}.

Constraint Graph

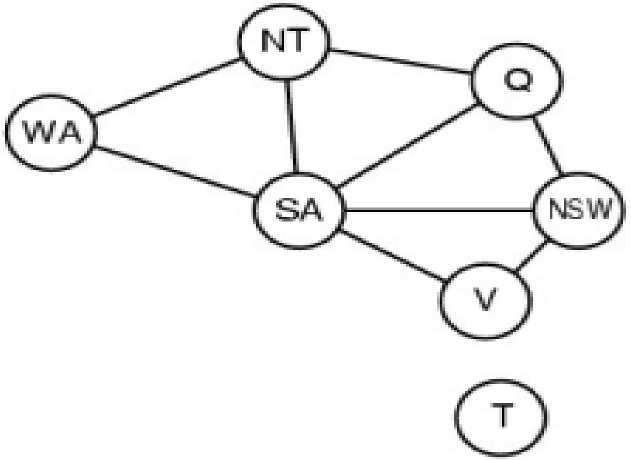
RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

- Constraint graph: nodes are variables, arcs are constraints
- Binary CSP: each constraint relates two variables
- CSP conforms to a standard pattern
  - a set of variables with assigned values
  - generic successor function and goal test
  - generic heuristics
  - reduce complexity



CSP Solvers Can be Faster

- CSP solver can quickly eliminate large part of search space
- If {SA = blue}
- Then 3^5 assignments can be reduced to 2^5 assignments, a reduction of 87%



- In a CSP, if a partial assignment is not a solution, we can immediately discard further refinements of it

Backtracking Search for CSP

- Assignment of value to any additional variable withi constraint can generate a legal state (Leads to successor state in search tree).
- Nodes in a branch backtracks when there is no options are available.
- Backtracking allows to go to the previous decision-making node to eliminate the invalid search space with respect to constraints.
- Heuristics plays a very important role here.
- If we are in position to determine which variables should be assigned next, then backtracking can be improved.

Algorithm for Backtracking

```
Pick initial state
R = set of all possible states
Select state with var assignment
Add to search space
check for con
If Satisfied
    Continue
Else
    Go to last Decision Point (DP)
    Prune the search sub-space from DP
    Continue with next decision option
If state = Goal State
    Return Solution
Else
    Continue
```

Role of Heuristic

- Heuristics help in deciding the initial state as subsequent selected states.
- Selection of a variable with minimum number of possible values can help in simplifying the search.
- This is called as Minimum Remaining Values Heuristic (MRV) or Most Constraint Variable Heuristic.
- The notion of selection is to detect a failure at an early stage.
- It restricts the most search which ends up in same variable (which would make the backtracking ineffecctive).

- MRV cannot have hold on initial selection process.
- Node with maximum constraint is selected over other unassigned variables - Degree Heuristics.
- By degree heuristics, branching factor cannot be reduced.
- Selection of variables are considered not the values for it.
- So, the order in which the values of particular variable can be arranged is tackled by least constraining value heuristic.

ForwardChecking

- To understand the forward checking, we shall see 4 Queens problem.
- If an arrangement on the board of a queen x, hampers the position of queens x+1, then this forward check ensures that the queen x should not be placed at the selected position and a new position is to be looked upon.

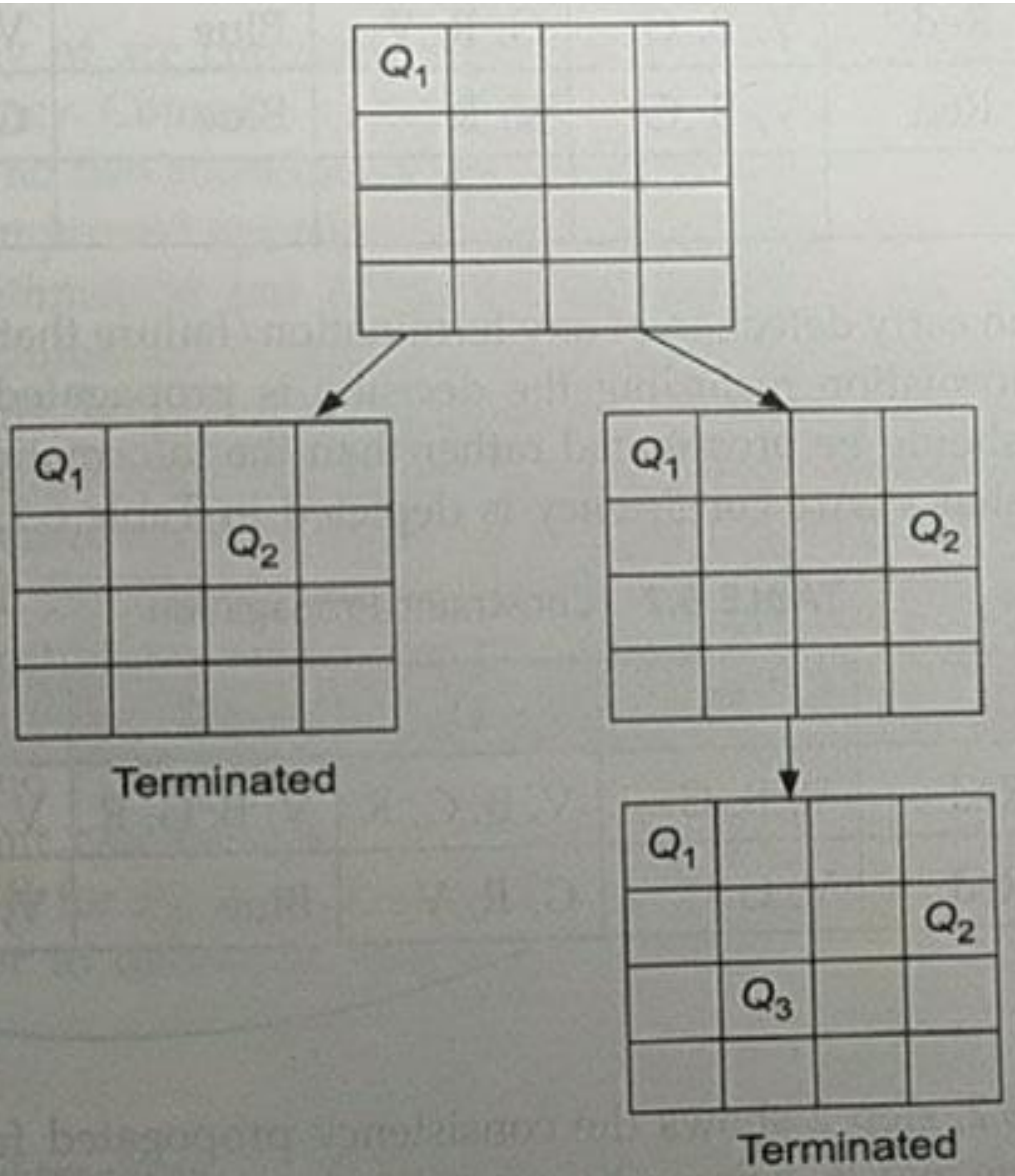


Figure 6.8 Forward checking for four-queens.

- Q1 and Q2 are placed in row1 and 2 in the left sub-tree, so, search is halted, since No positions are left for Q3 and Q4.
- Forward Checking keeps track of the next moves that are available for the unassigned variables.
- The search will be terminated when there is no legal move available for the unassigned variables.

Constraint Propagation

- There is no early detection of any termination/failure that would possible occur even though the information regarding the decision is propagated.
- Constraint should be propagated rather than the information.

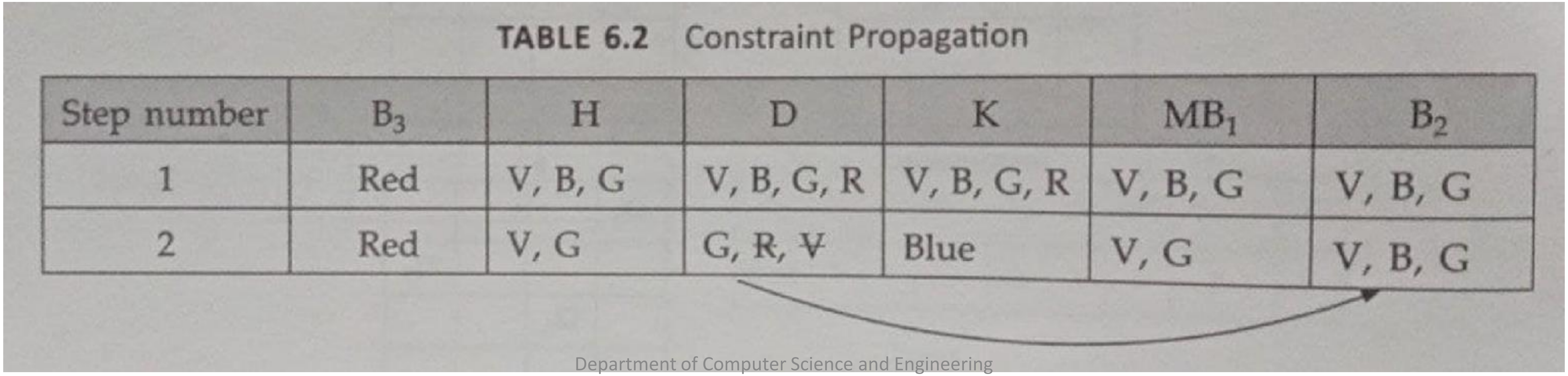
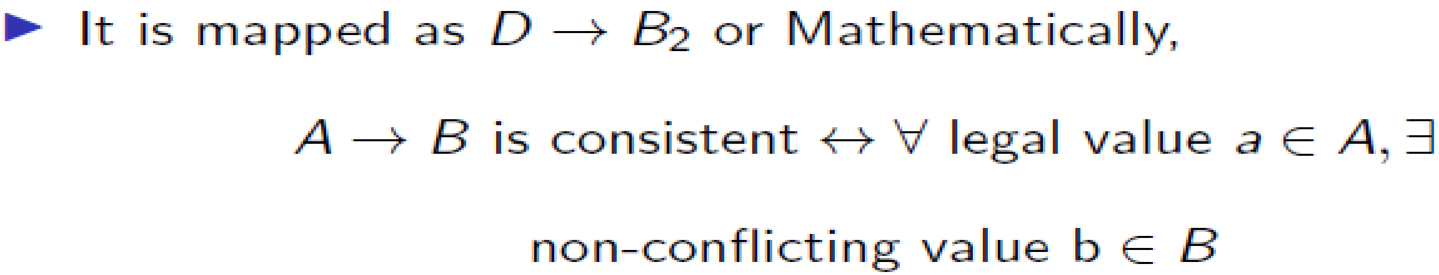


TABLE 6.2 Constraint Propagation

Step number	B_3	H	D	K	MB_1	B_2
1	Red	V, B, G	V, B, G, R	V, B, G, R	V, B, G	V, B, G
2	Red	V, G	G, ~~R~~, ~~V~~	Blue	V, G	V, B, G

- Step 2 shows the consistency propagated from Dt
- Since D is can have only G value and B2 being adjacent to it, the arc is drawn.



► It is mapped as $D \rightarrow B_2$ or Mathematically,

$$A \rightarrow B \text{ is consistent} \leftrightarrow \forall \text{ legal value } a \in A, \exists$$

$$\text{non-conflicting value b} \in B$$

- Failure detection can take place at early stage

Constraint Propagation

- Algorithm for arc assignment is:
  - Let C be the variable which is being assigned at a given instance
  - X will have some value from D{} where D is domain
  - For each and every assigned variable, that is adjacent to X, Say X'
    1. Perform forward check (remove values from domain D that conflict the decision of the current assignment)
    2. For every other variable X'' that are adjacent or connected to X' ;

       i Remove the values from D from X'' that can't be taken as further unassigned variables

       ii Repeat step 2, till no more values can be removed or discarded
  - Inconsistency is considered and constraints are propagated in Step (2)

Intelligent Backtracking

- Conflict set is maintained using forward checking and maintained.
- Considering the 4 Queens problem,Conict needs to be detected by the user of conflict set so that a backtrack can occur
- Backtracking with respect to the conflict set is called as conflict-directed backjumping
- Backjumping approach can't actually restrict the earlier committed mistakes in some other branches

THANK YOU....