

SEARCH METHODS IN AI

AUTOMATED PROBLEM SOLVING

COMPLEX PROBLEMS & SOLUTIONS



Path Finding

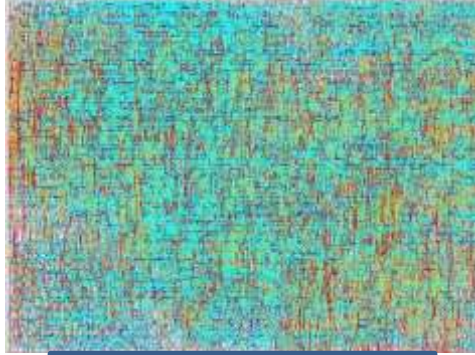


Chess Playing



Robot Assembly

COMPLEX PROBLEMS & SOLUTIONS

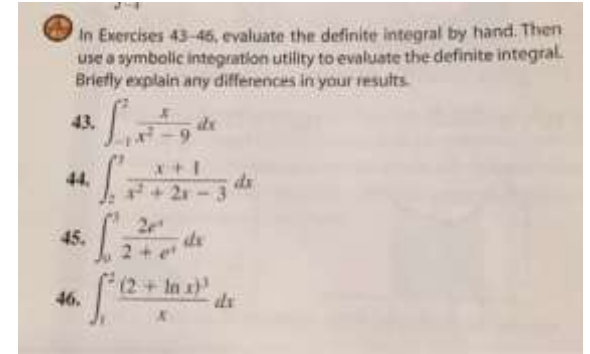


VLSI Chip Design

A screenshot of a web-based time-table scheduling application. The browser address bar shows the URL: www.primetimetable.com/Application/7demo?id=459e5b18d-74a5-4994-b9. The page displays a loading status of 70%. The main content is a grid showing a weekly schedule for Monday and Tuesday. The grid has columns for days (1-5) and rows for time slots (5-A, 5-B, 5-C, 5-D, 7-A, 7-B, 7-C, 7-D, 7-E). Each cell contains a colored box with a subject name, such as PE, Math, Eng, Sci, and History. The subjects are color-coded: PE (yellow), Math (blue), Eng (green), Sci (red), and History (purple).

	Monday					Tuesday				
	1	2	3	4	5	1	2	3	4	5
5-A	PE	Math		TW	Eng		Math	Math	Eng	Math
5-B	Eng	Sci	PE	Eng	Math		Math	Eng	PE	Math
5-C	Sci	Eng	Math	Math	Eng		Math	Math	Math	Eng
5-D		TW	Math	Eng	PE		Sci	Math	Math	Eng
7-A	Math	PE	Math	Math	Math		Math	Math	Eng	PE
7-B	Math	PE	Math	Math	Math		Math	PE	Math	Math
7-C	PE	Math	Math	Math	Math		TW	Math	Math	Math
7-D	Math	Math	Eng	Math	Math		PE	Math	Math	Math
7-E	Eng	Math	Math	PE	Math		Math	Math	Math	Eng

Time-Table Scheduling



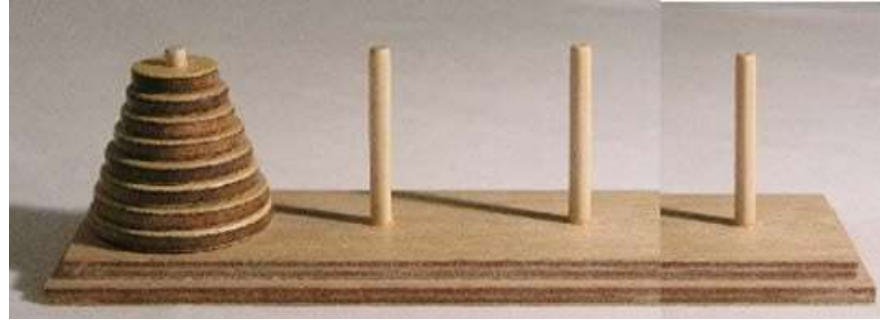
Symbolic Integration

AUTOMATED PROBLEM SOLVING BY SEARCH

- Generalized Techniques for Solving Large Classes of Complex Problems
- Problem Statement is the Input and solution is the Output, sometimes even the problem specific algorithm or method could be the Output
- Problem Formulation by AI Search Methods consists of the following key concepts
 - Configuration or State
 - Constraints or Definitions of Valid Configurations
 - Rules for Change of State and their Outcomes
 - Initial or Start Configurations
 - Goal Satisfying Configurations
 - An Implicit State or Configuration Space
 - Valid Solutions from Start to Goal in the State Space
 - General Algorithms which SEARCH for Solutions in this State Space
- **ISSUES**
 - Size of the Implicit Space, Capturing Domain Knowledge, Intelligent Algorithms that work in reasonable time and Memory, Handling Incompleteness and Uncertainty

TOWER OF HANOI

- Configuration or State
- Constraints or Definitions of Valid Configurations
- Rules for Change of State and their Outcomes
- Initial or Start Configurations
- Goal Satisfying Configurations
- An Implicit State or Configuration Space
- Valid Solutions from Start to Goal in the State Space
- General Algorithms which SEARCH for Solutions in this State Space

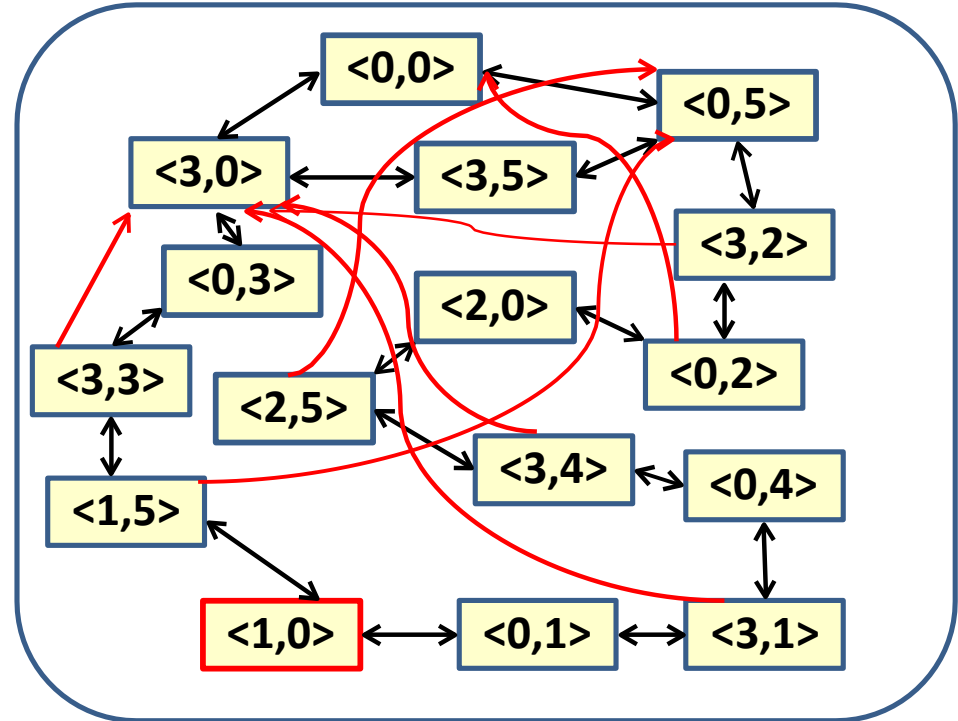


TWO JUG PROBLEM

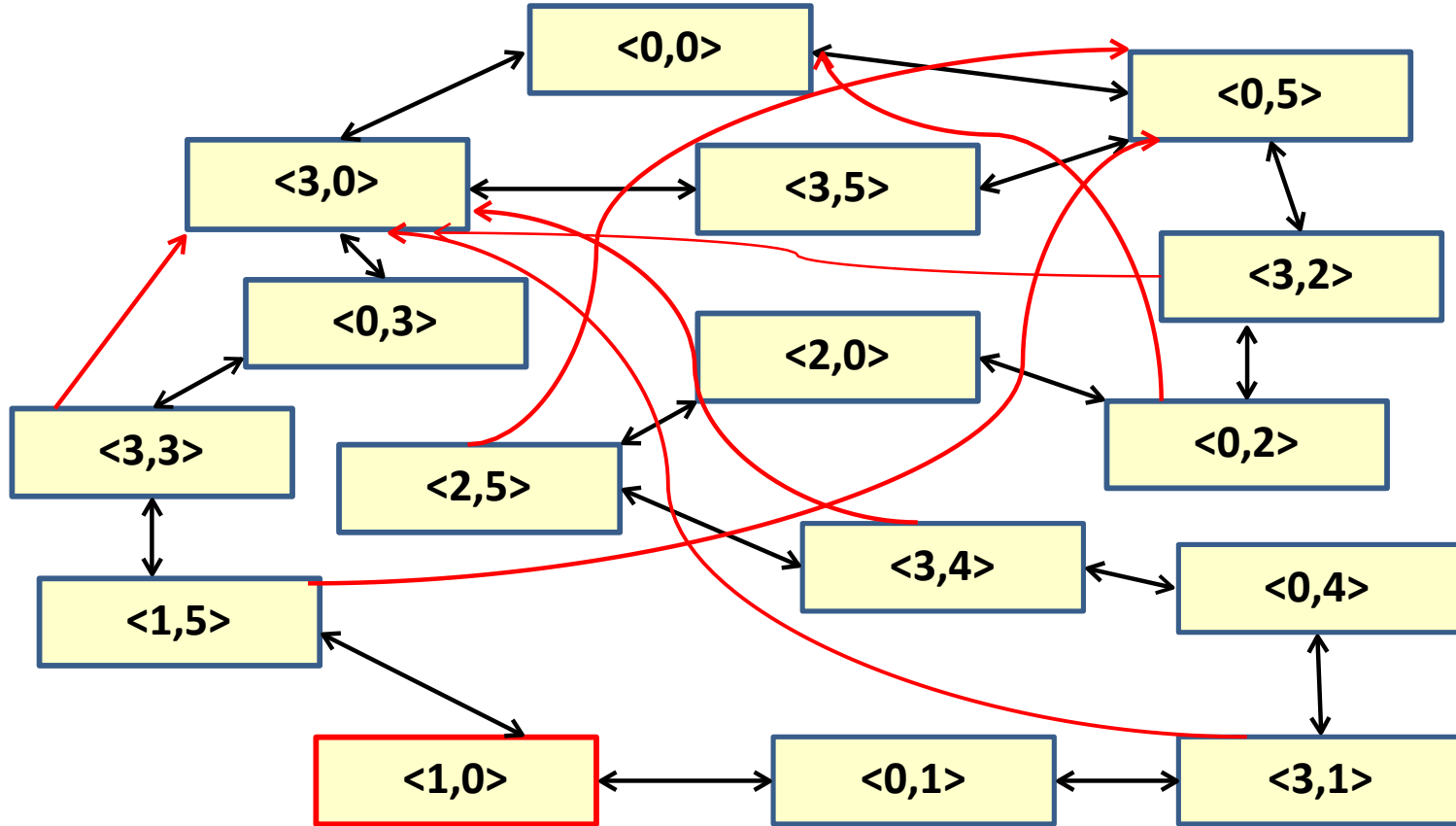
- There is a large bucket B full of water and Two(02) jugs, J1 of volume 3 litre and J2 of volume 5 litre. You are allowed to fill up any empty jug from the bucket, pour all water back to the bucket from a jug or pour from one jug to another. The goal is to have jug J1 with exactly one (01) litre of water
- State Definition: $\langle J1, J2 \rangle$
- Rules:
 - Fill (J1): $\langle J1, J2 \rangle$ to $\langle 3, J2 \rangle$
 - Fill (J2): $\langle J1, J2 \rangle$ to $\langle J1, 5 \rangle$
 - Empty (J1), Empty (J2): Similarly defined
 - Pour (J1, J2): $\langle J1, J2 \rangle$ to $\langle X, Y \rangle$, where
 - $X = 0$ and $Y = J1 + J2$ if $J1 + J2 \leq 5$,
 - $Y = 5$ and $X = (J1 + J2) - 5$, if $J1 + J2 > 5$
 - Pour (J2, J1): Similarly defined
- Start: $\langle 0, 0 \rangle$, Goal: $\langle 1, 0 \rangle$
- Part of State Space Shown on the right
(Not all Links shown here)

TWO JUG PROBLEM

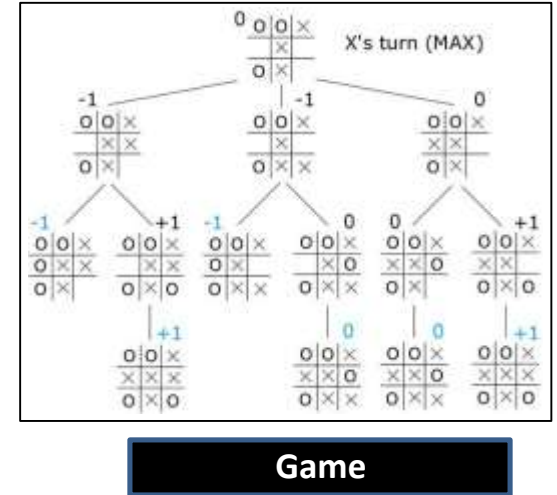
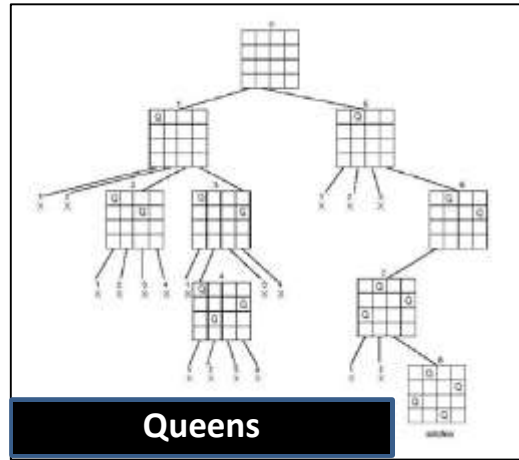
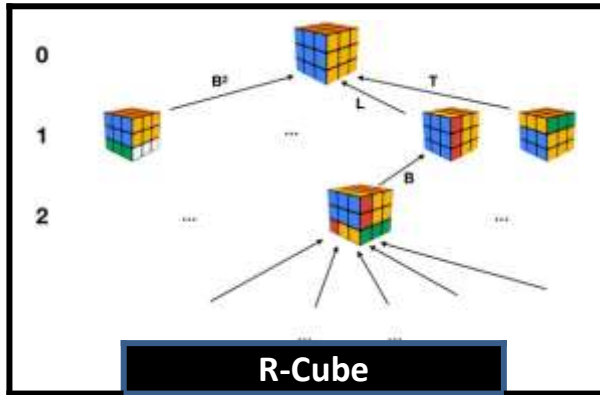
- There is a large bucket B full of water and Two(02) jugs, J1 of volume 3 litre and J2 of volume 5 litre. You are allowed to fill up any empty jug from the bucket, pour all water back to the bucket from a jug or pour from one jug to another. The goal is to have jug J1 with exactly one (01) litre of water
- State Definition: $\langle J1, J2 \rangle$
- Rules:
 - Fill (J1): $\langle J1, J2 \rangle$ to $\langle 3, J2 \rangle$
 - Fill (J2): $\langle J1, J2 \rangle$ to $\langle J1, 5 \rangle$
 - Empty (J1), Empty (J2): Similarly defined
 - Pour (J1, J2): $\langle J1, J2 \rangle$ to $\langle X, Y \rangle$, where
 - $X = 0$ and $Y = J1 + J2$ if $J1 + J2 \leq 5$,
 - $Y = 5$ and $X = (J1 + J2) - 5$, if $J1 + J2 > 5$
 - Pour (J2, J1): Similarly defined
- Start: $\langle 0, 0 \rangle$, Goal: $\langle 1, 0 \rangle$
- Part of State Space Shown on the right
(Not all Links shown here)



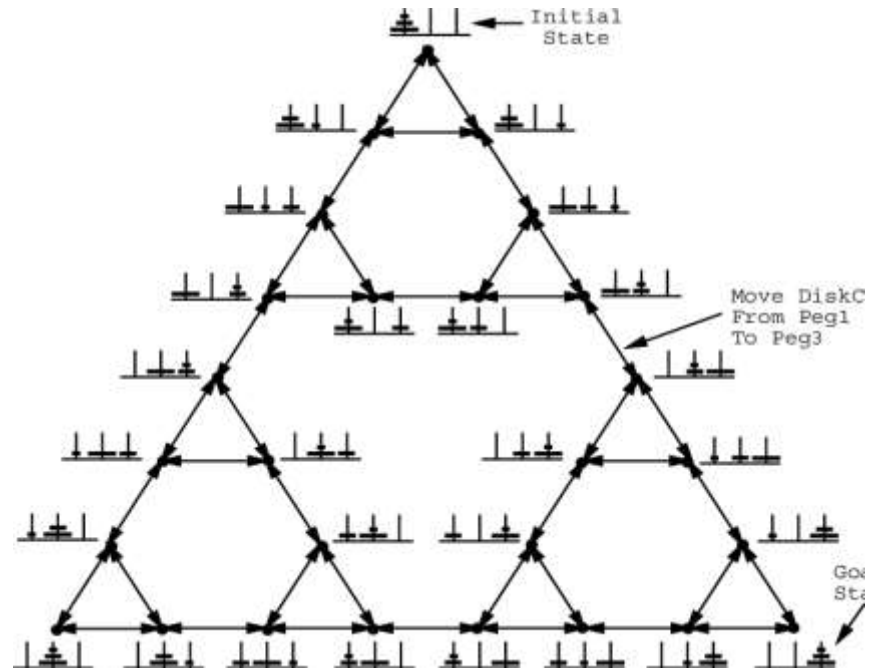
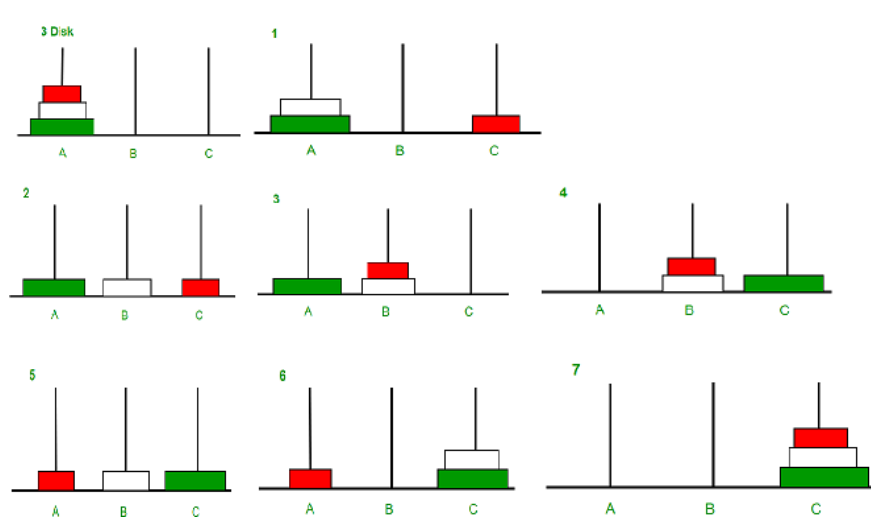
PART OF STATE SPACE



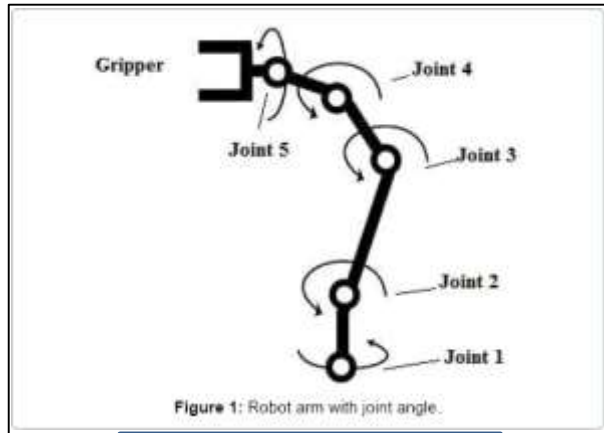
STATE SPACES



3 DISK, 3 PEG TOWER of HANOI STATE SPACE



STATE SPACES

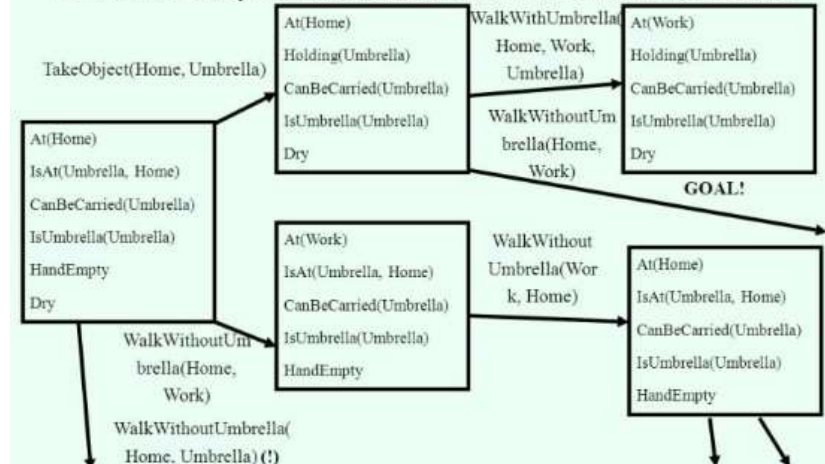


Manipulator Arm

Forward state-space search

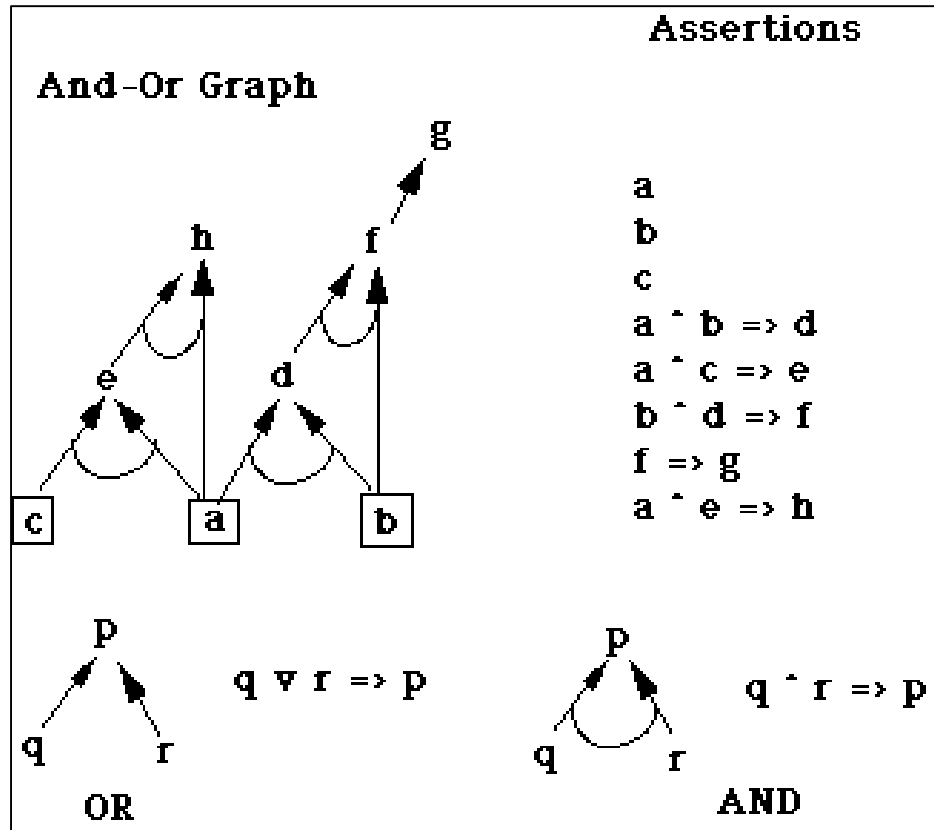
(progression planning)

- Successors: all states that can be reached with an action whose preconditions are satisfied in current state



Planning

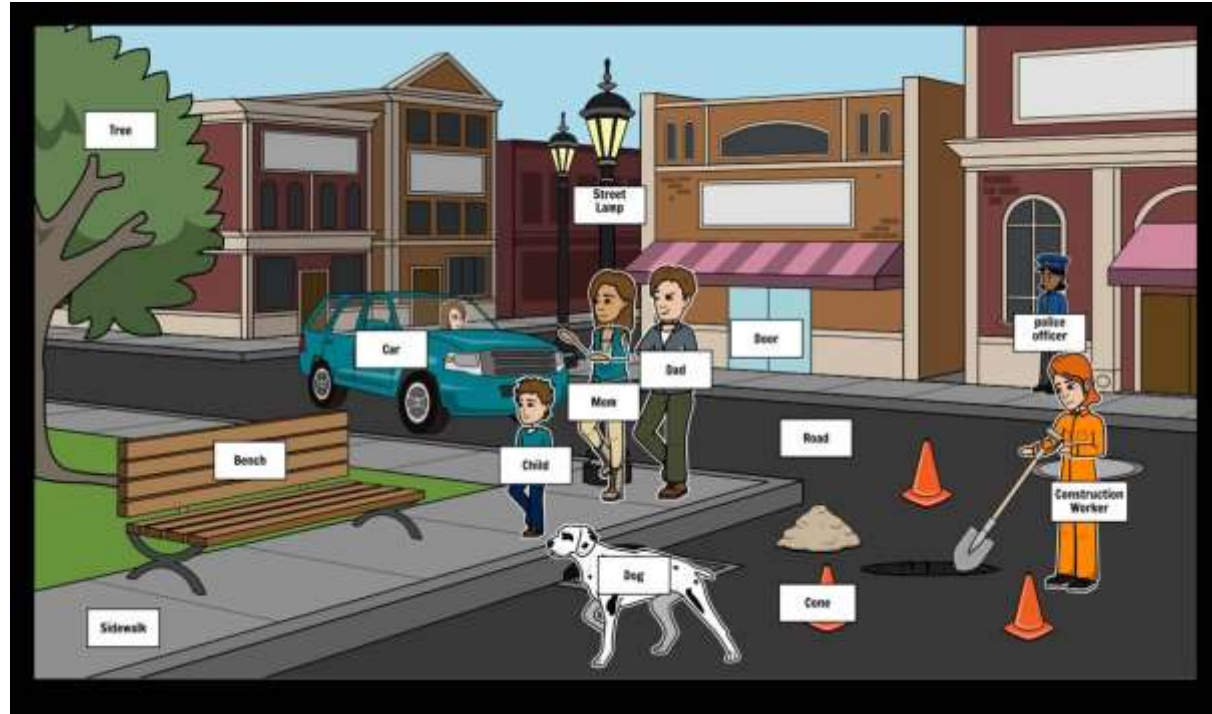
AND / OR STATE SPACES



CONSISTENT LABELLING BY CONSTRAINT SATISFACTION

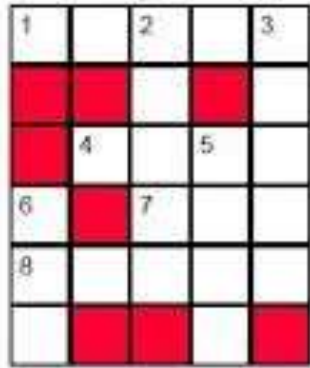
```
      B O B  
    x B O B  
  -----  
    M E O Y  
  M I L O  
M E O Y  
-----  
M A R L E Y
```

Cryptarithmic



Scene Analysis

CONSISTENT LABELLING BY CONSTRAINT SATISFACTION



Instructions

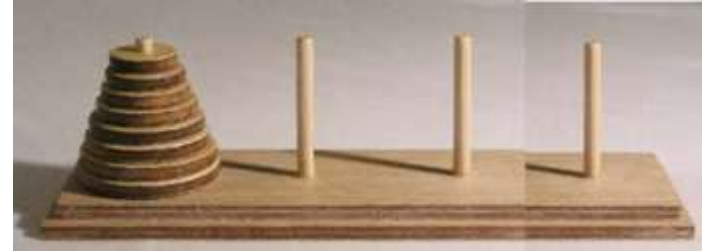
- fill in words from the list

List of Words

- Alt
- Ale
- Eel
- Hike
- Hoses
- Keel
- Knot
- Laser
- Lee
- Line
- Sails
- Sheet
- Steer
- Tie

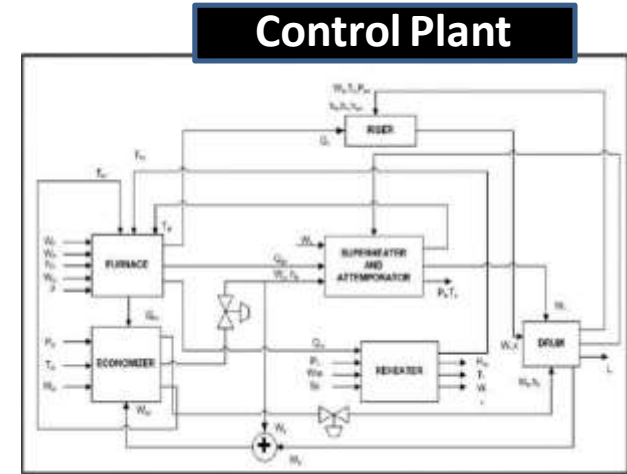
STATES, SPACES, SOLUTIONS, SEARCH

- **States**
 - Full / Perfect Information and Partial Information States
- **State Transformation Rules**
 - Deterministic Outcomes
 - Non-Deterministic / Probabilistic Outcomes
- **State Spaces As Generalized Games**
 - Single Player: OR Graphs
 - Multi-Player: And / Or, Adversarial, Probabilistic Graphs
- **Solutions**
 - Paths
 - Sub-graphs
 - Expected Outcomes
- **Costs**
- **Sizes**
- **Domain Knowledge**
- **Algorithms for Heuristic Search**



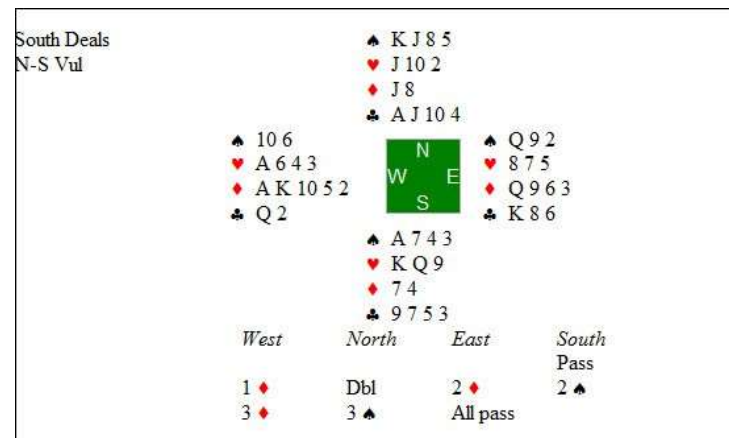
STATES, SPACES, SOLUTIONS, SEARCH

- **States**
 - Full / Perfect Information and Partial Information States
- **State Transformation Rules**
 - Deterministic Outcomes
 - Non-Deterministic / Probabilistic Outcomes
- **State Spaces As Generalized Games**
 - Single Player: OR Graphs
 - Multi-Player: And / Or, Adversarial, Probabilistic Graphs
- **Solutions**
 - Paths
 - Sub-graphs
 - Expected Outcomes
- **Costs**
- **Sizes**
- **Domain Knowledge**
- **Algorithms for Heuristic Search**



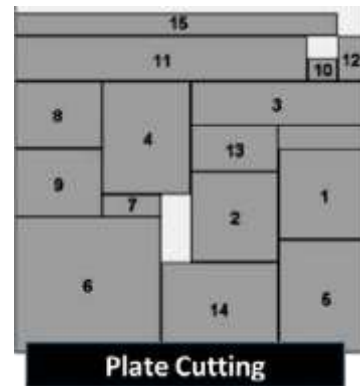
STATES, SPACES, SOLUTIONS, SEARCH

- **States**
 - Full / Perfect Information and Partial Information States
- **State Transformation Rules**
 - Deterministic Outcomes
 - Non-Deterministic / Probabilistic Outcomes
- **State Spaces As Generalized Games**
 - Single Player: OR Graphs
 - Multi-Player: And / Or, Adversarial, Probabilistic Graphs
- **Solutions**
 - Paths
 - Sub-graphs
 - Expected Outcomes
- **Costs**
- **Sizes**
- **Domain Knowledge**
- **Algorithms for Heuristic Search**



STATES, SPACES, SOLUTIONS, SEARCH

- **States**
 - Full / Perfect Information and Partial Information States
- **State Transformation Rules**
 - Deterministic Outcomes
 - Non-Deterministic / Probabilistic Outcomes
- **State Spaces As Generalized Games**
 - Single Player: OR Graphs
 - Multi-Player: And / Or, Adversarial, Probabilistic Graphs
- **Solutions**
 - Paths
 - Sub-graphs
 - Expected Outcomes
- **Costs**
- **Sizes**
- **Domain Knowledge**
- **Algorithms for Heuristic Search**



Thank you