

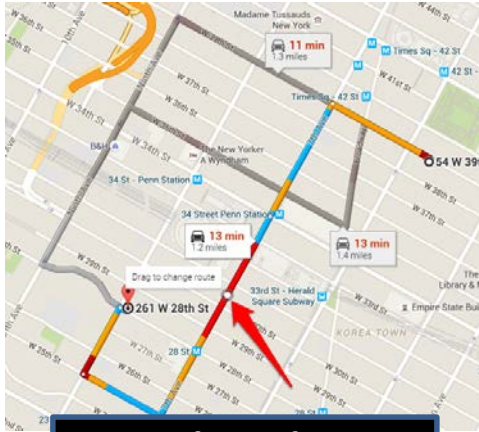
SEARCH METHODS IN AI

AUTOMATED PROBLEM SOLVING



Arijit Mondal & Partha P Chakrabarti
Indian Institute of Technology Kharagpur

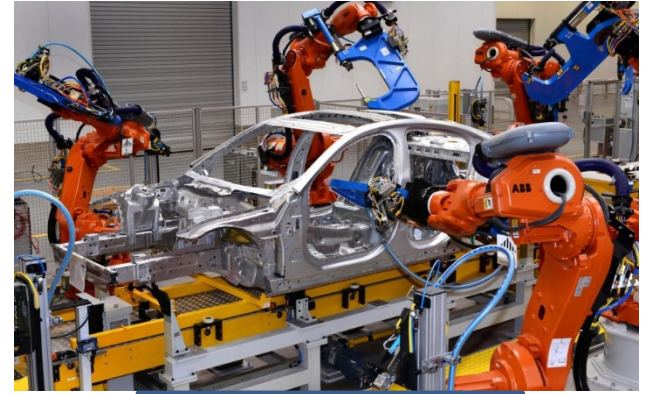
COMPLEX PROBLEMS & SOLUTIONS



Path Finding

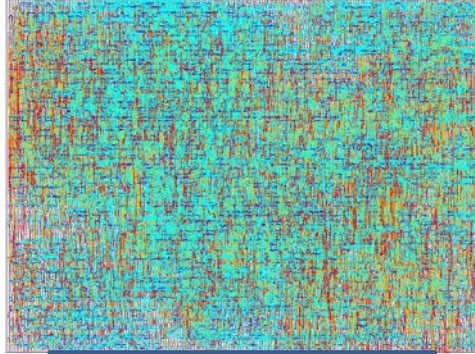


Chess Playing



Robot Assembly

COMPLEX PROBLEMS & SOLUTIONS



VLSI Chip Design

Example 1 - Prime Timetabl x

www.primetimetable.com/Application/?demo#id=69e5bf8d-74a5-4984-b9

Loading maker 78%

	Monday						Tuesday				
	1	2	3	4	5	6	1	2	3	4	5
5-A	PE	Mus		TW	Eng		Mat	His	Eng	Bio	Fre
5-B	Eng	Geo	PE	Fre	Mat		Bio	Eng	PE	Fre	Pal
5-C	Geo	Eng	Mat	Pal	Bio		Pal	Mat	Bio	Mus	Eng
5-D		TW		Mus	Eng	PE	Ger	Geo	Bio	Pal	Eng
7-A	Che	PE	Mat	Geo	Mus		Ger	Pal	Mat	Eng	PE
7-B	Geo	PE	His	Bio	Ger		Mus	PE	Che	Mat	His
7-C	PE	Phy	Geo	Mat	Fre			TW	Mat	Pal	Phy
7-D	Mus	Mat	Eng	Phy	His		PE	Che	Fre	Geo	Bio
7-E	Eng	Bio	Che	PE	Phy	Fre	Che	Mat	His	Phy	Eng

Time-Table Scheduling

In Exercises 43–46, evaluate the definite integral by hand. Then use a symbolic integration utility to evaluate the definite integral. Briefly explain any differences in your results.

43. $\int_{-1}^2 \frac{x}{x^2 - 9} dx$

44. $\int_2^3 \frac{x + 1}{x^2 + 2x - 3} dx$

45. $\int_0^3 \frac{2e^x}{2 + e^x} dx$

46. $\int_1^2 \frac{(2 + \ln x)^3}{x} dx$

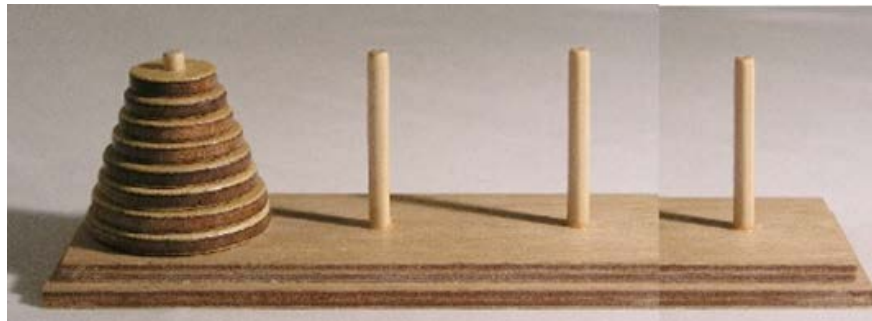
Symbolic Integration

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

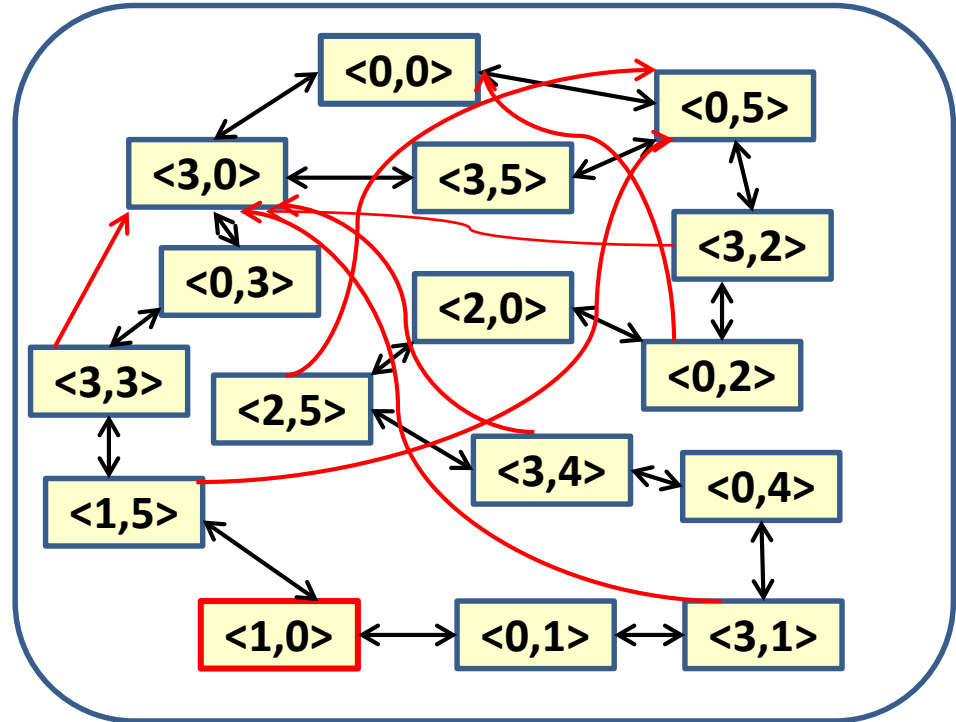


TWO JUG PROBLEM

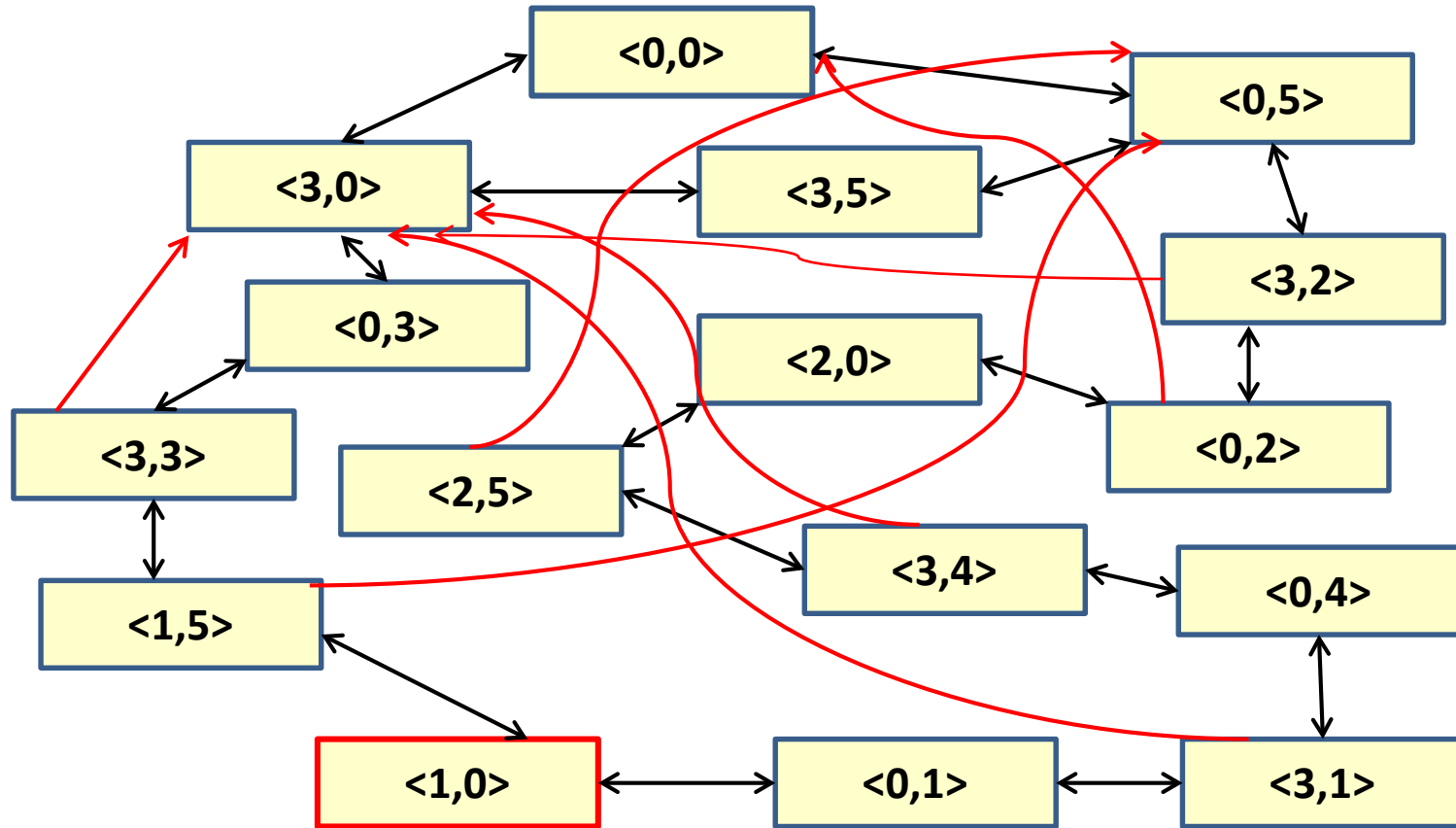
- There is a large bucket B full of water and Two (02) jugs, J1 of volume 3 litre and J2 of volume 5 litre. You are allowed to fill up any empty jug from the bucket, pour all water back to the bucket from a jug or pour from one jug to another. The goal is to have jug J1 with exactly one (01) litre of 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

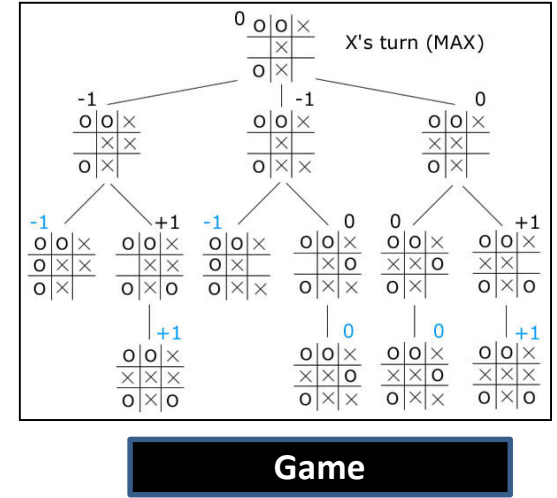
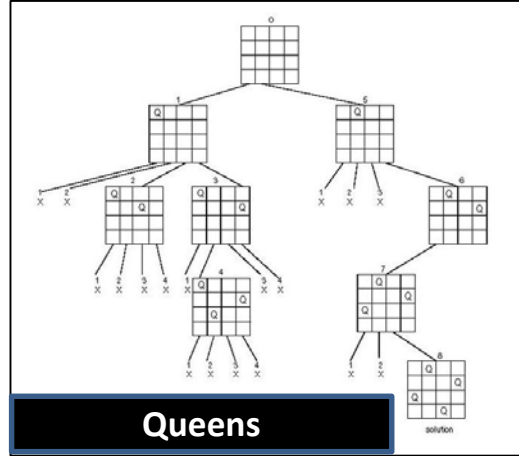
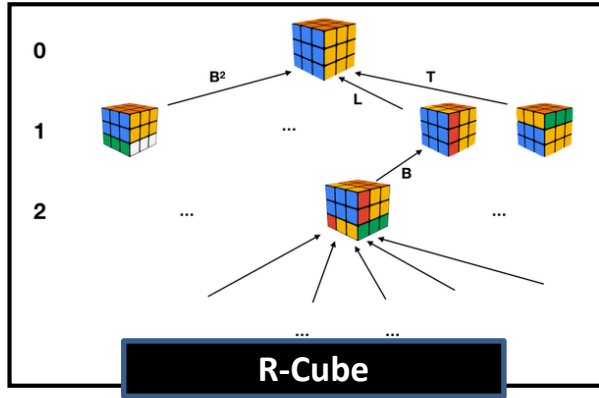
- There is a large bucket B full of water and Two (02) jugs, J1 of volume 3 litre and J2 of volume 5 litre. You are allowed to fill up any empty jug from the bucket, pour all water back to the bucket from a jug or pour from one jug to another. The goal is to have jug J1 with exactly one (01) litre of 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)



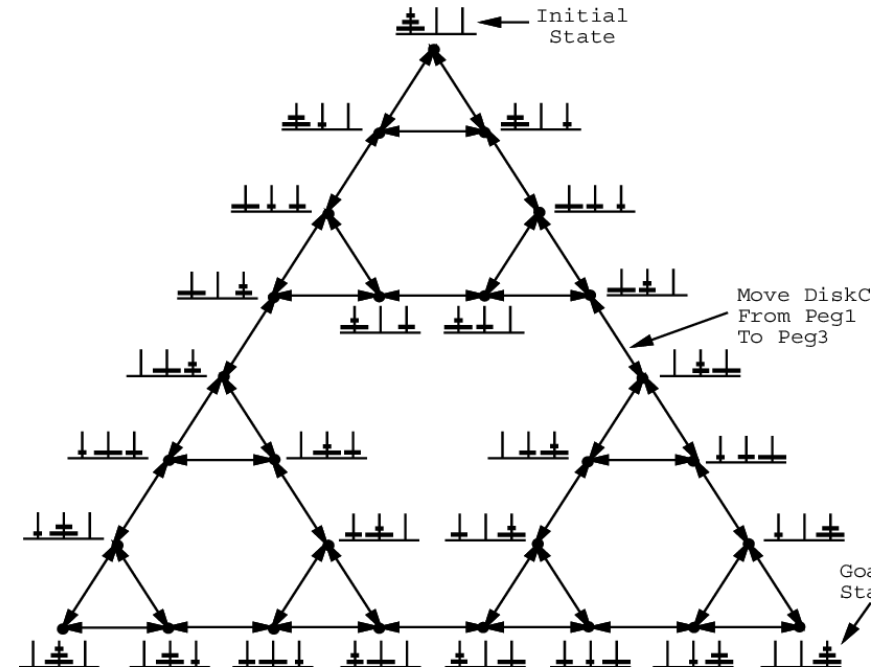
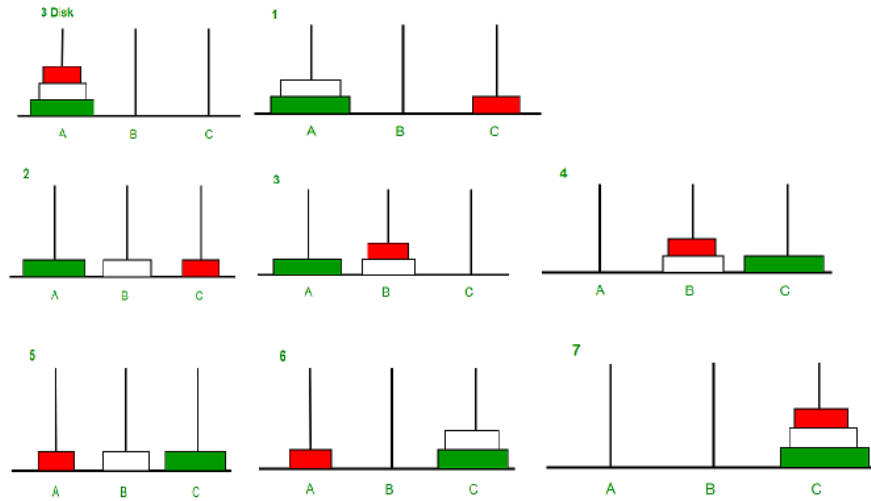
PART OF STATE SPACE



STATE SPACES



3 DISK, 3 PEG TOWER of HANOI STATE SPACE



STATE SPACES

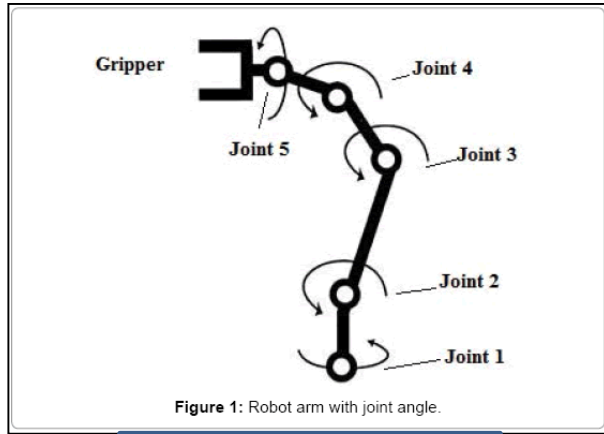


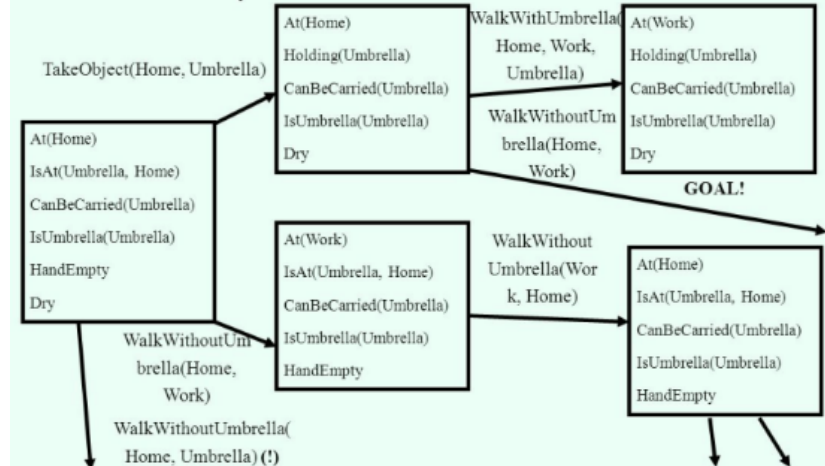
Figure 1: Robot arm with joint angle.

Manipulator Arm

Forward state-space search

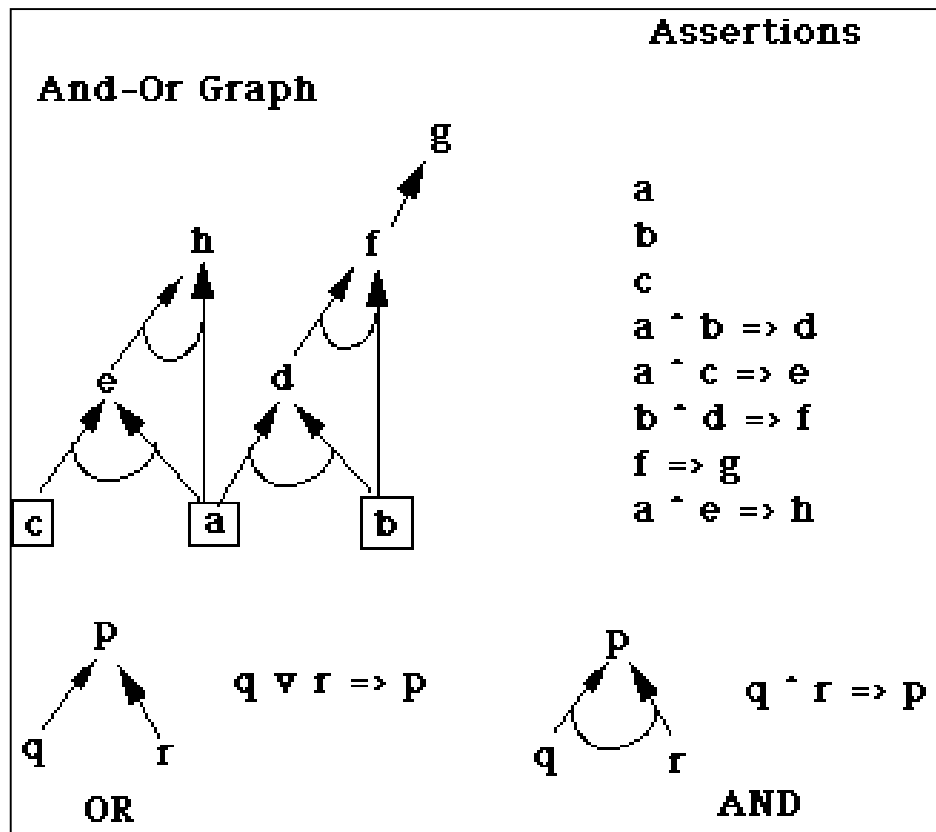
(progression planning)

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



Planning

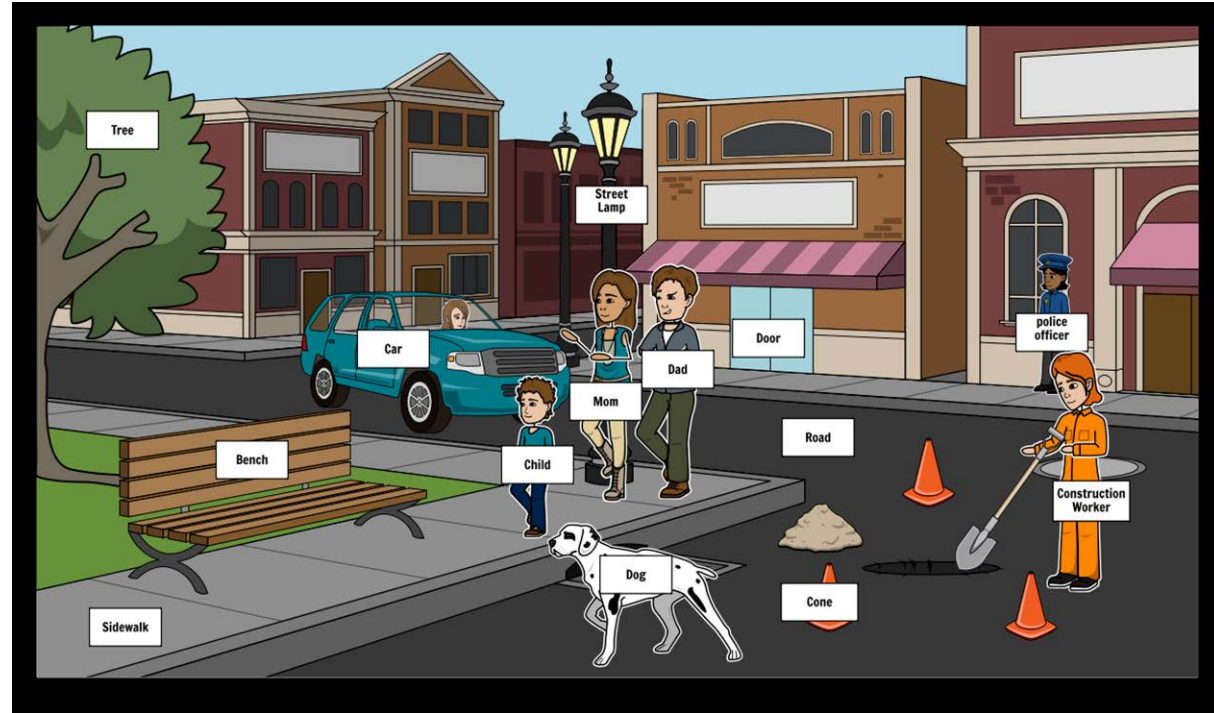
AND / OR STATE SPACES



CONSISTENT LABELLING BY CONSTRAINT SATISFACTION

$$\begin{array}{r} \text{B O B} \\ \times \text{B O B} \\ \hline \text{M E O Y} \\ \text{M I L O} \\ \hline \text{M E O Y} \\ \hline \text{M A R L E Y} \end{array}$$

Cryptarithmic



Scene Analysis

CONSISTENT LABELLING BY CONSTRAINT SATISFACTION

1		2		3
	4		5	
6		7		
8				

Instructions

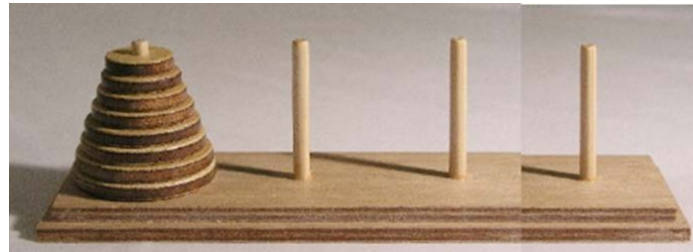
- fill in words from the list

List of Words

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

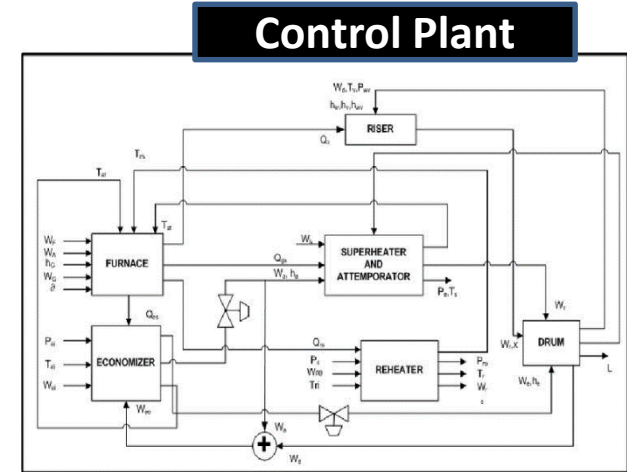
STATES, SPACES, SOLUTIONS, SEARCH

- **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**



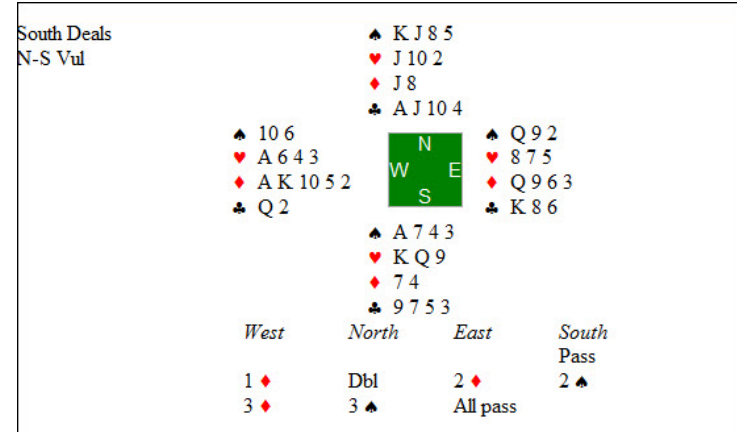
STATES, SPACES, SOLUTIONS, SEARCH

- **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**



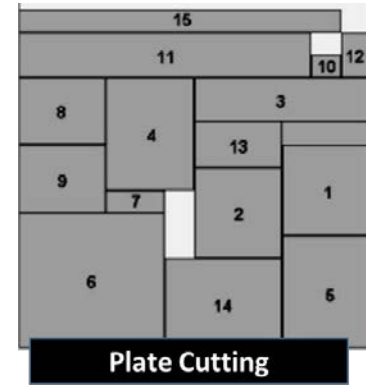
STATES, SPACES, SOLUTIONS, SEARCH

- **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