SEARCH METHODS IN AI

AUTOMATED PROBLEM SOLVING

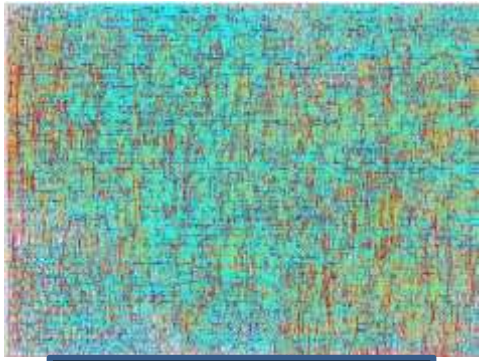
COMPLEX PROBLEMS & SOLUTIONS

Path Finding

Chess Playing

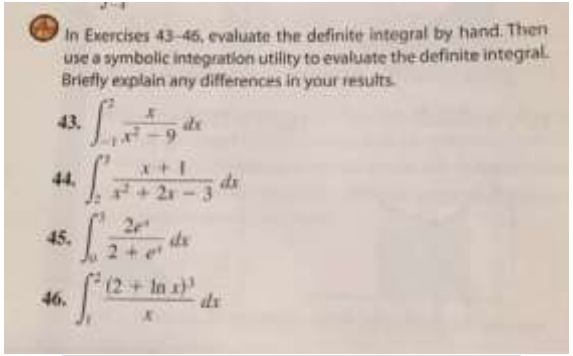
Robot Assembly

COMPLEX PROBLEMS & SOLUTIONS



VLSI Chip Design

Time-Table Scheduling



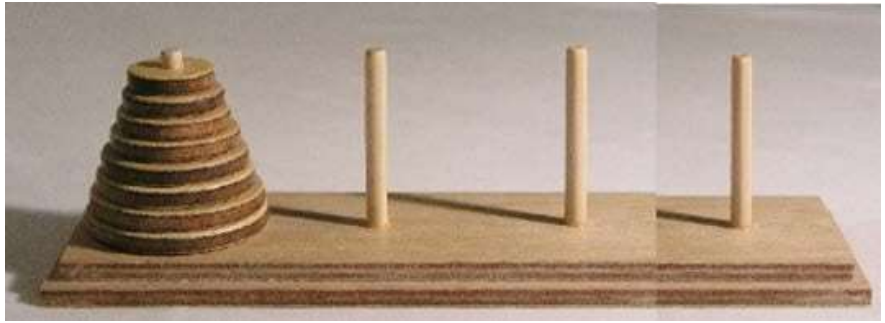
Symbolic Integration

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: <J1, J2>
- Rules:
  - Fill (J1): <J1, J2> to <3,J2>
  - Fill (J2): <J1, J2> to <J1, 5>
  - Empty (J1), Empty (J2): Similarly defined
  - Pour (J1, J2): <J1, J2> to <X,Y>, where
    - X = 0 and Y = J1 + J2 if J1+J2 ≤ 5,
    - Y = 5 and X = (J1+ J2) – 5, if J1+J2 > 5
  - Pour (J2, J2): Similarly defined
- Start: <0,0>, Goal: <1,0>

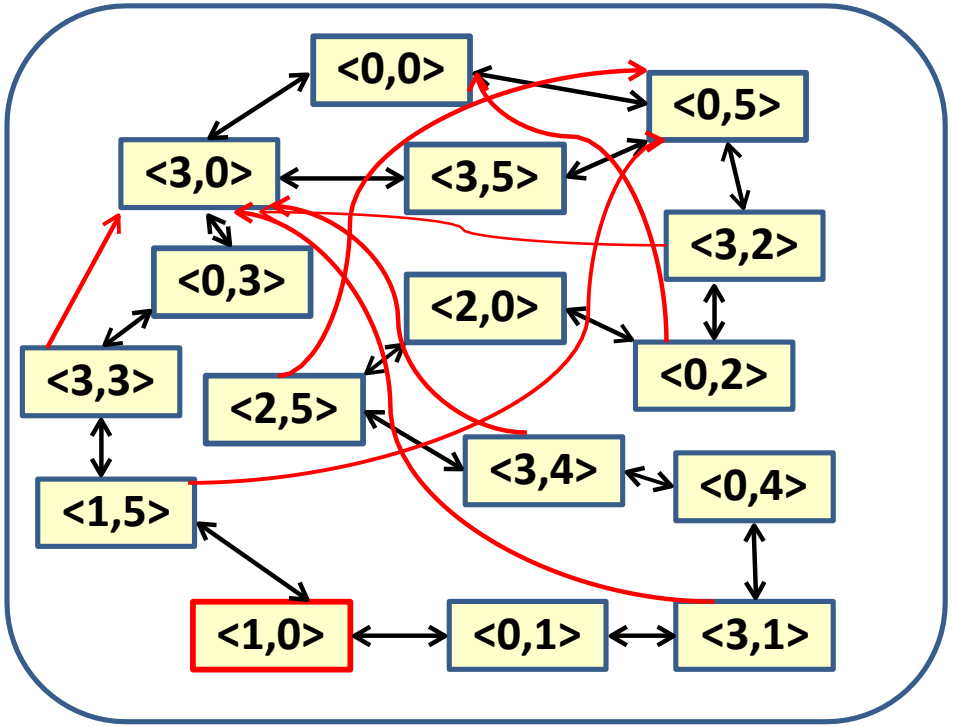
•Part of State Space Shown on the right

(Not all Links shown here)

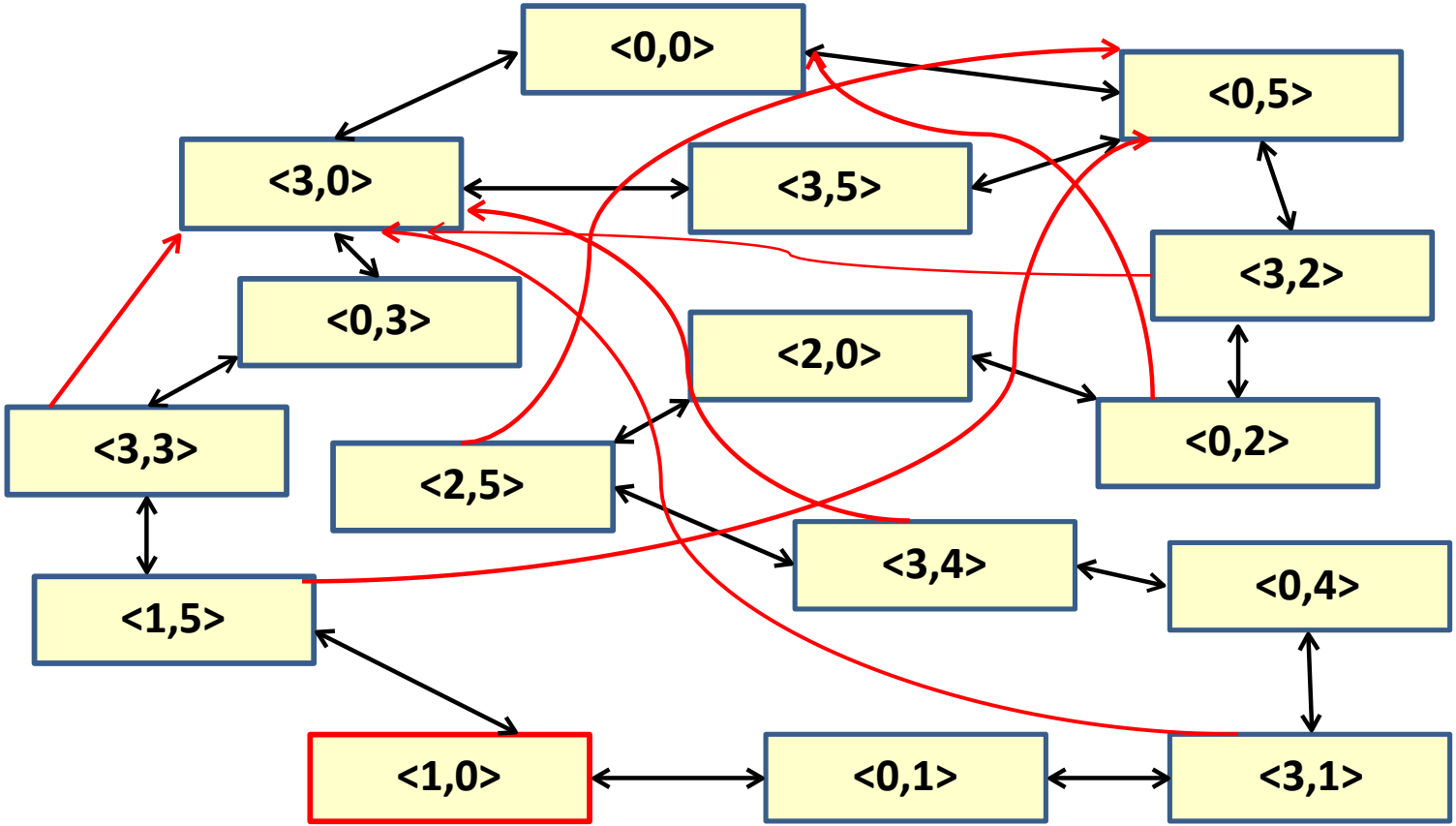
TWO JUG PROBLEM

- 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: <J1, J2>
- Rules:
  - Fill (J1): <J1, J2> to <3,J2>
  - Fill (J2): <J1, J2> to <J1, 5>
  - Empty (J1), Empty (J2): Similarly defined
  - Pour (J1, J2): <J1, J2> to <X,Y>, where
    - X = 0 and Y = J1 + J2 if J1+J2 ≤ 5,
    - Y = 5 and X = (J1+ J2) – 5, if J1+J2 > 5
  - Pour (J2, J2): Similarly defined
- Start: <0,0>, Goal: <1,0>
- Part of State Space Shown on the right

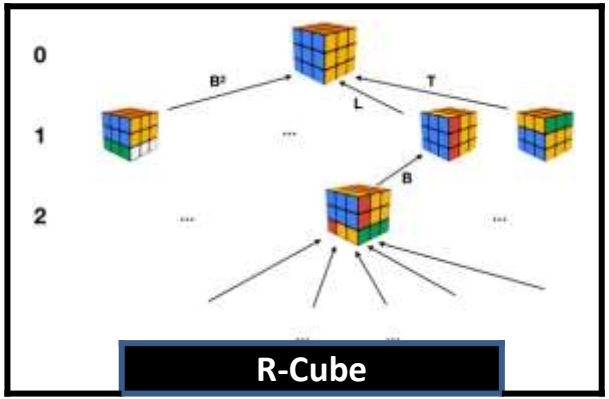
(Not all Links shown here)

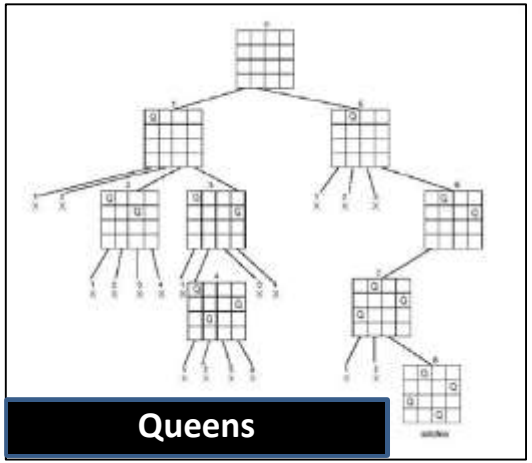


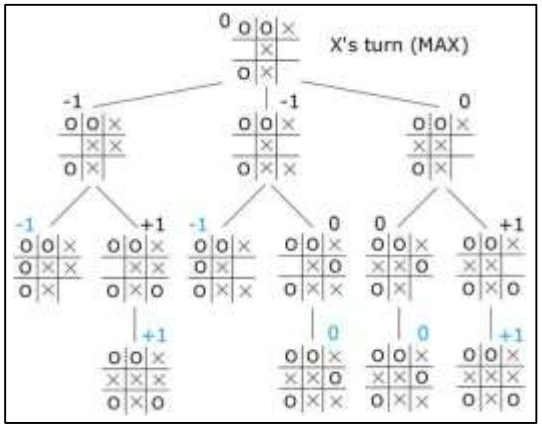
PART OF STATE SPACE



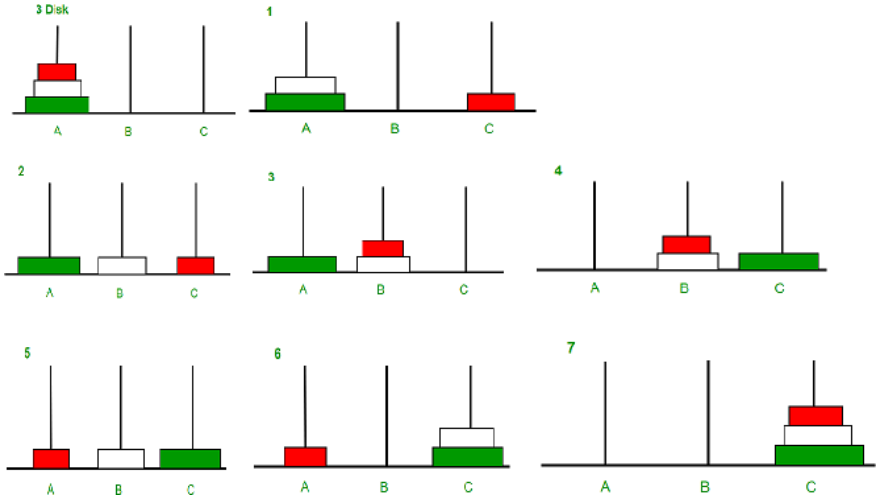
STATE SPACES

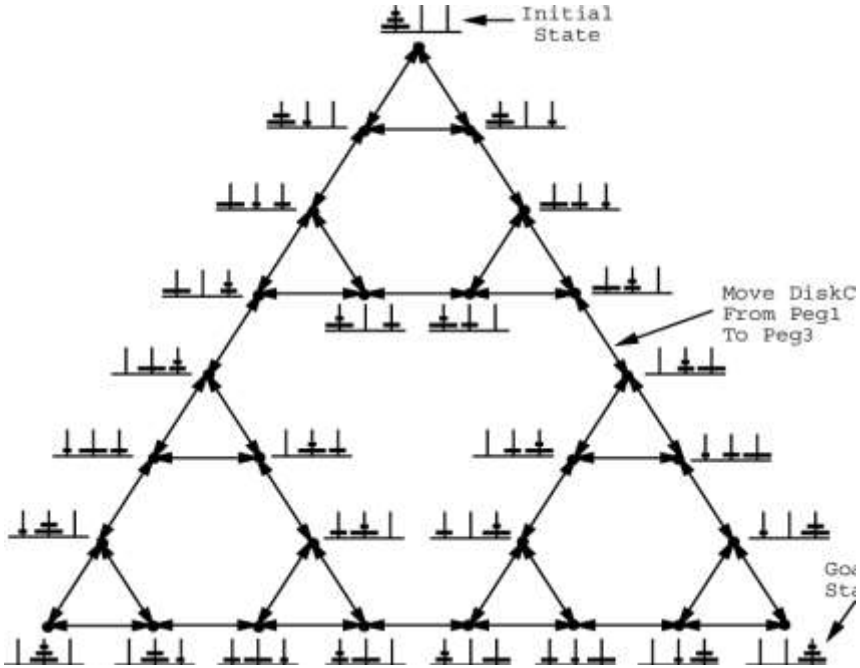




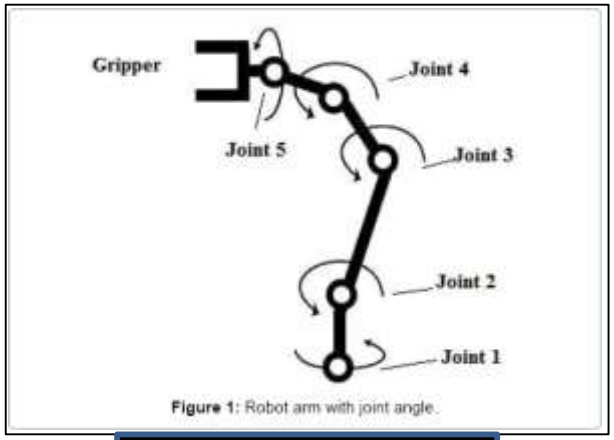


3 DISK, 3 PEG TOWER of HANOI STATE SPACE

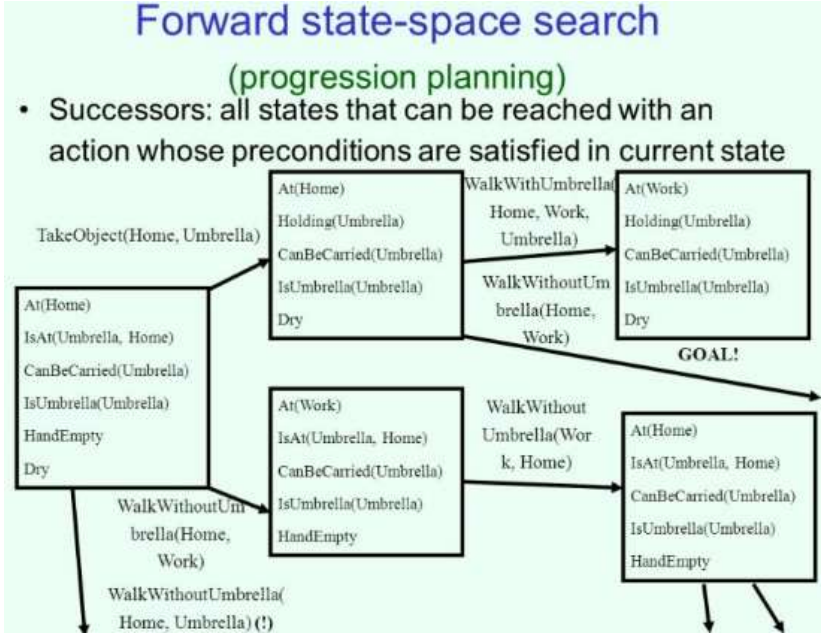




STATE SPACES

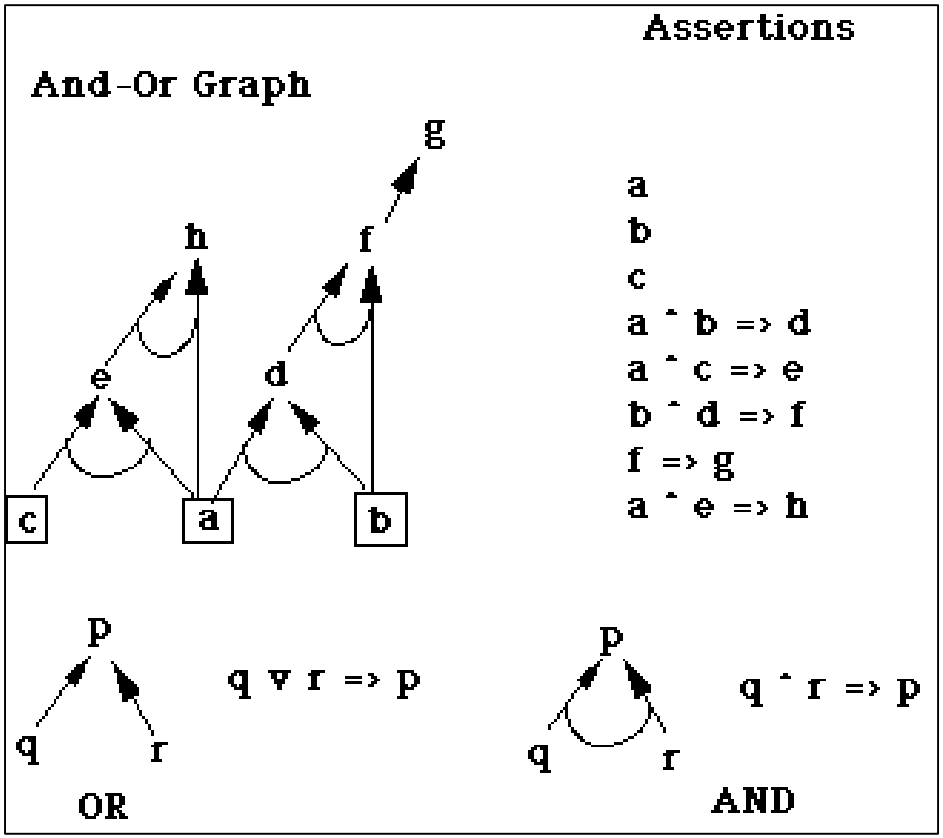


Manipulator Arm



Planning

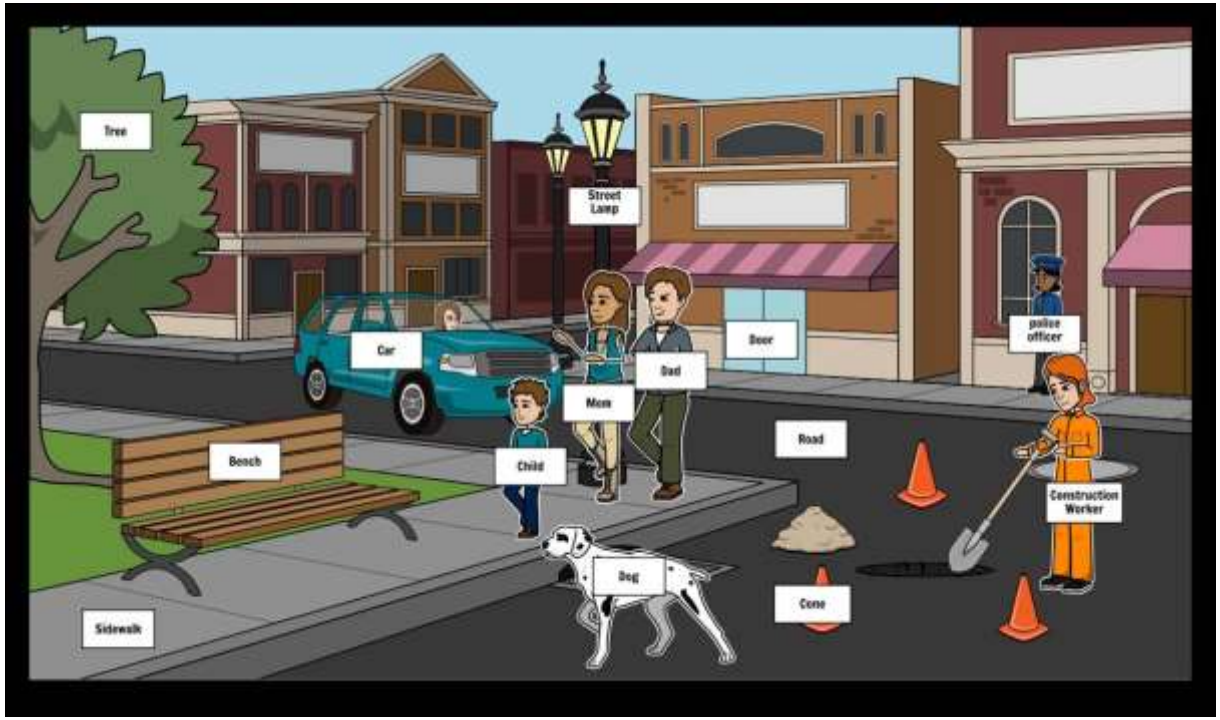
AND / OR STATE SPACES



CONSISTENT LABELLING BY CONSTRAINT SATISFACTION

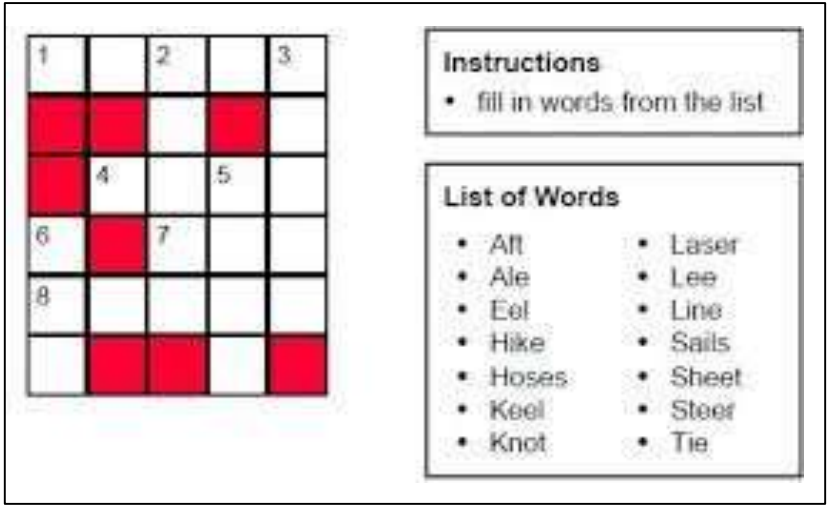
```
    B O B
x   B O B
  M E O Y
M I L O
M E O Y
M A R L E Y
```

Cryptarithmetic



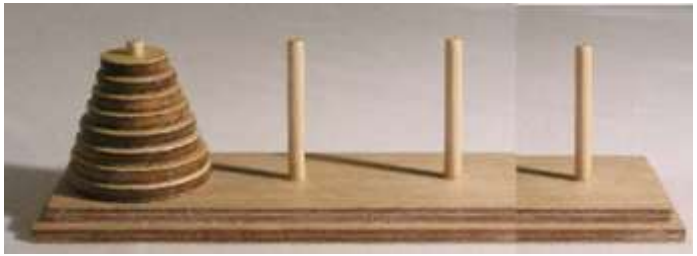
Scene Analysis

CONSISTENT LABELLING BY CONSTRAINT SATISFACTION



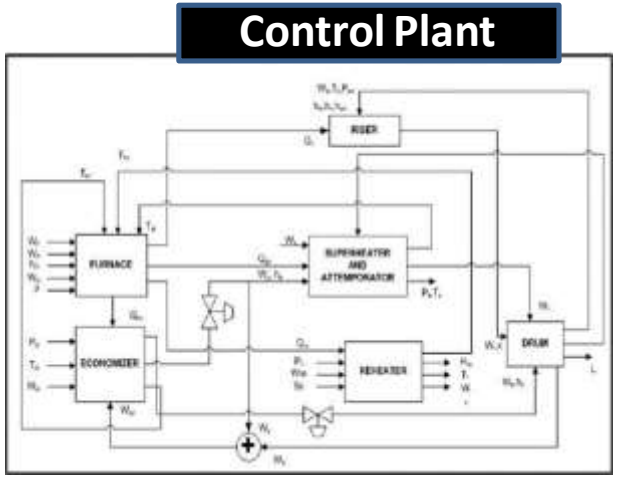
STATES, SPACES, SOLUTIONS, SEARCH

- 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



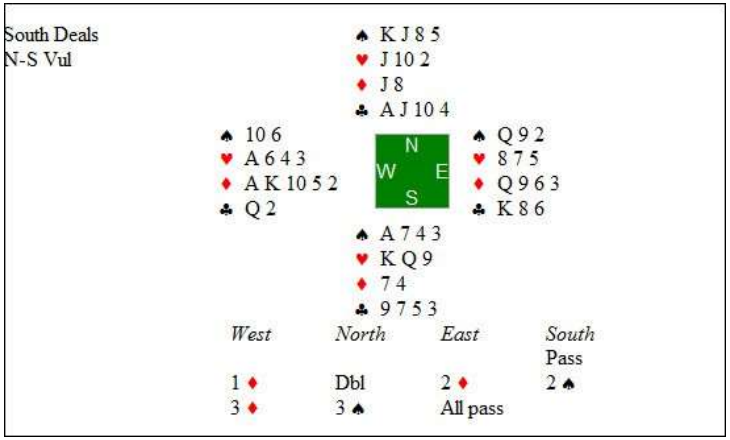
STATES, SPACES, SOLUTIONS, SEARCH

- 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



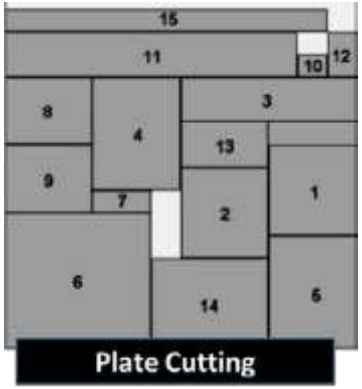
STATES, SPACES, SOLUTIONS, SEARCH

- 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