Knowledge Representation & Search

Punjab University College of Information Technology, Lahore

Introduction

- Search is the essential basis of human problem solving [Newell & Simon].
- According to Newell and Simon, intelligent activity, in either human or machine, is achieved through the use of:
  - Symbol patterns that represent significant aspects of a problem domain.
  - Operations on these patterns to generate potential solutions to problems.
  - Search to select a solution among these possibilities.
- The assumptions mentioned above form the well-known physical symbol system hypothesis (PSSH.)

Examples of PSS

Examples of physical symbol systems include:

- Formal Logic: the symbols are words like "and", "or", "not", "for all x" and so on. The expressions are statements in formal logic which can be true or false. The processes are the rules of logical deduction.
- Algebra: the symbols are "+", "×", "x", "y", "1", "2", "3", etc. The expressions are equations. The processes are the rules of algebra, that allow one to manipulate a mathematical expression and retain its truth.
- A digital computer: the symbols are zeros and ones of computer memory, the processes are the operations of the CPU that change memory.
- Chess: the symbols are the pieces, the processes are the legal chess moves, the expressions are the positions of all the pieces on the board.
- The physical symbol system hypothesis claims that both of the following are also examples of physical symbol systems:
  - Intelligent human thought: the symbols are encoded in our brains. The expressions are ‘thoughts’. The processes are the mental operations of thinking.
  - A running ‘AI’ program: The symbols are data. The expressions are more data. The processes are programs that manipulate the data.

State Space Search

- A State Space is a graphical representation of a problem.
- State Space includes all possible states of the problem including the solution state.
- By representing the problem as a state space graph we can use graph theory to analyze the structure and complexity of both the problem and the procedures used to solve it.
- State Space is also known as Solution Space or Problem Space

State Space Search

- A *state space* is represented by a four-tuple [**N,A,S,GD**], where:
- **N** is the set of nodes or states of the graph. These correspond to the states in a problem-solving process
- **A** is the set of arcs (or links) between nodes. These correspond to the steps in a problem-solving process.
- **S** is a nonempty subset of **N**, contains the start state(s) of the problem.
- **GD**, a nonempty subset of **N**, contains the goal state(s) of the problem. The states in **GD** are described using either:
  - A measurable property of the states encountered in the search.
  - A property of the path developed in the search.

Example of State Space Representation

- 8-puzzle problem

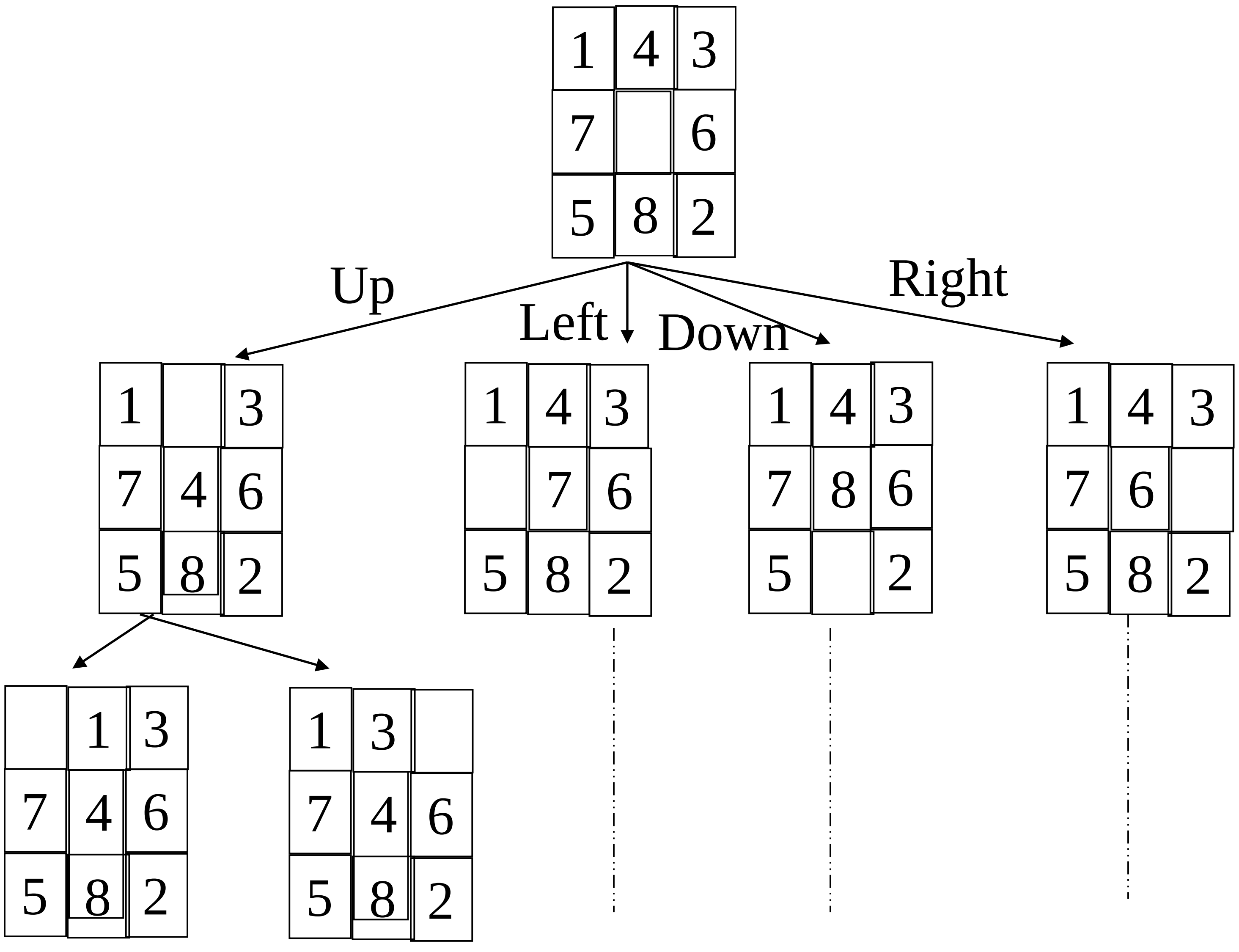
2	8	3
1	6	4
7		5

→

1	2	3
8		4
7	6	5

  - state description
    - 3-by-3 array: each cell contains one of 1-8 or blank symbol
  - two state transition descriptions
    - 8×4 moves: one of 1-8 numbers moves up, down, right, or left
    - 4 moves: one blank symbol moves up, down, right, or left

State Space Graph



Homework

- City of Konigsberg Problem and its graphical formulation
- Traveling Salesperson Problem
- Reference: Artificial Intelligence: Structures and Strategies for Complex Problem Solving 4th Ed. (George F. Luger, and William A. Stubblefield) pages 82-84, 91-92

Strategies for State Space Search

- A state space may be searched in two directions: from the given data of a problem instance towards a goal or from a goal back to the data.
  - Data Driven Search
  - Goal Driven Search

Data Driven Search

- Data Driven Search (Forward Chaining)
  - The Problem Solver begins with the given facts of the problem and a set of legal moves or rules for changing state. Search proceeds by applying rules to facts to produce new facts, which are in turn used by the rules to generate more new facts. This process continues until it generates a path that satisfies the goal condition.

Goal Driven Search

- Goal Driven Search (Backward Chaining)
  - It takes the goal that we want to achieve, see what rules or legal moves could be used to generate this goal and determine what conditions must be true to use them. These conditions becomes the new goals or sub-goals
  - It recalls the simple childhood trick of trying to solve a maze by working back from the finish to the start.

Which Strategy?

- In the final analysis, both **data-driven** and **goal-driven** problem solvers search the same state space graph; however, the order and actual number of states searched can differ. The preferred strategy is determined by the properties of the problem itself.