Conversion Masters in IT (MIT)
AI as Representation and Search

(Representation and Search Strategies)

Lecture 002

Sandro Spina

Physical Symbol System Hypothesis

- Intelligent Activity is achieved through the use of :
  - Symbol patterns to represent **significant** aspects of a problem domain (eg. How do we describe a human being?)
  - Operations on these patterns to generate potential solutions to problems.
  - Search to select a solution from among these possibilities

- The physical symbol system hypothesis (Newell & Simon) implicitly distinguishes between **pattern formed by an arrangement of symbols** and the **medium used to implement them**. If intelligence derives only from the structure of a symbol system, then any medium that successfully implements the correct patterns and processes will achieve intelligence, regardless of whether it is composed of neurons or logic circuits.

Physical Symbol System Hypothesis (cont…)

- Problem solving as search
  - Problems are solved by searching amongst alternative choices
  - Eg. consider a game of tic-tac-toe: Given any board situation there are only a finite number of moves that a player can make. Starting with empty board there are 9 moves, then iteratively one less until we reach 0
  - Each board configuration is a node in a graph. The links in the graph represent legal moves. The different nodes are the different states (configurations) the board can be in. The resulting structure is referred to as the **state space graph**.

State Space Graph …

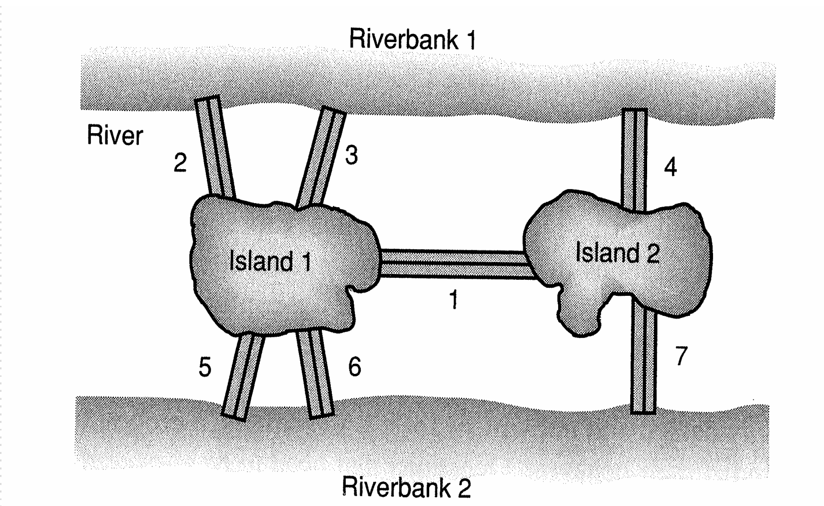
- … pg 43 Luger for diagram
- The significance of this structure is that by starting at the node representing a new board and moving along arcs until we get to a state representing a win or a tie, it is possible to trace the sequence of moves in any potential game.
- All possible games of tic-tac-toe are essentially all the different paths through the state graph.

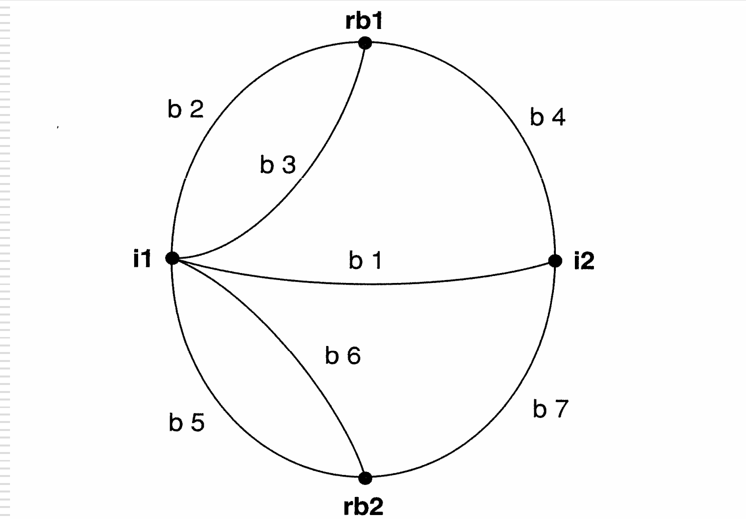
… State Space Search

- … Therefore one effective game strategy would be to search for paths that lead to the most wins and fewest loses and play in a way that always tries to force the game along one of these paths.
- Graph = { set of nodes S } + { transition relation SxS}
- *Graph theory* is our best tool for reasoning about the structure of objects and relations.

*Leonhard **Euler** – Konigsberg Bridge problem*

- Problem : Is there a walk around the city that crosses each bridge exactly once?





- Euler noted that unless a graph contained exactly zero or two nodes of odd degree, the walk was impossible!!

Graph Theory (basic definitions)

- A **graph** is a set of nodes and arcs that connect them.
- The nodes and arcs may be **labelled**. Arc labels are used to indicate that an arc represents a named relationship or to attach weights to an arc (as in TSP).
- A graph is **directed** if arcs have an associated directionality.
- A **path** through a graph connects a sequence of nodes through successive arcs.
- A **rooted** graph has a unique node denoted the root node, such that there is a path from the root to all nodes within the graph.
  - Eg. The initial board configuration of a game is usually represented by the root node. From it all possible games can be played.
- A **tree** is a graph in which two nodes have at most one path between them.
  - Loop/cycle free graph

State Space (four-tuple [N,A,S,GD])

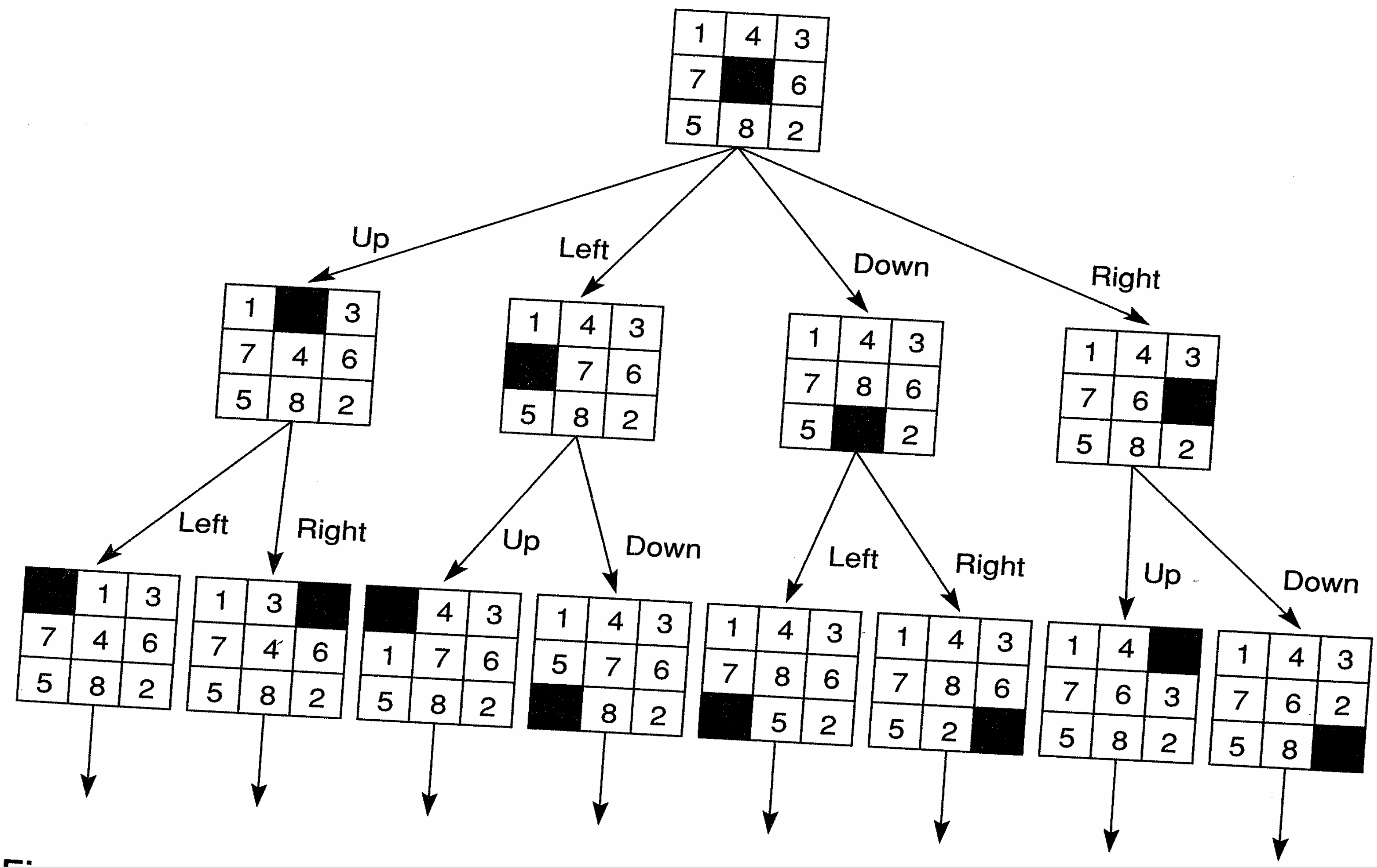
- **N** is a set of nodes or states of the graph. These correspond to the states in a problem-solving process.
- **A** is a set of arcs between nodes. These correspond to the steps in a problem-solving process.
- **S**, a nonempty subset of N, contains the start state(s) of the problem
- **GD**, a nonempty 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, for example, the transition costs for the arcs of the paths.
- A ***solution path*** *(what we want !!!)* is a path through this graph from a node S to a node in GD

Problem 1 : Tic-Tac-Toe (State Space)

Problem 1 : Tic-Tac-Toe (… search)

- **GD** = { N | board state has 3 Xs in a row,column or diagonal } … assuming we are playing for X
- Note that there are 3^9 ways to arrange {blank,X,O} however …
  - The transition function (**A**) will actually give you a subset of these.
- We have a graph here and not a tree really … why?
  - Because some states can be reached by different paths (level three and deeper), however
  - There are no cycles in the state space, because the directed arcs of the graph do not allow a move to be undone.
- Directed Acyclic Graph (DAG)
- 9! different paths … for chess we have 10^{120} paths while checkers 10^{40} paths. For these spaces we need hueristics.

Problem 2 : The 8-puzzle (State Space)



Problem 2 : The 8-puzzle (… search)

- 15 (or 8) different numbered tiles are fitted into 16 (or 9) spaces on a grid. One space is left blank so that tiles can be moved around to form different patterns. The goal is to find a series of moves of tiles into the blank space that places the board in a goal configuration.

- How do I represent the board? It is much simpler to think in terms of "moving the blank space" in order to simplify the definition of **legal** move rules, which are:
  - Move the blank up or right or down or left without moving it off the board.

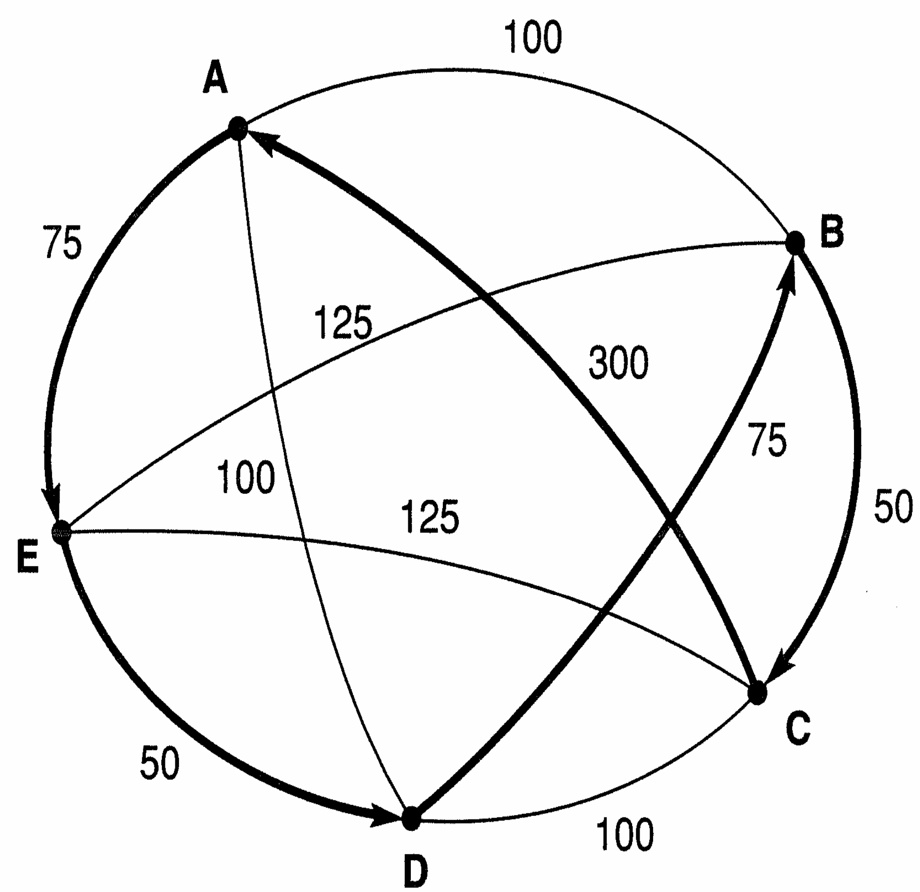
Problem 2 : The 8-puzzle (… cont)

- As with tic-tac-toe, the state space is a graph with most states having multiple parents, however
  - Cycles are possible !!! (implications ??)
- The GD (goal description) of the state space is a particular state or board configuration. Search terminates when this board is found (unique)
- Then path from the start to the goal is the desired series of moves (multiple ones)
- Translate to decision problems … does there exist a path with a maximum of N transitions

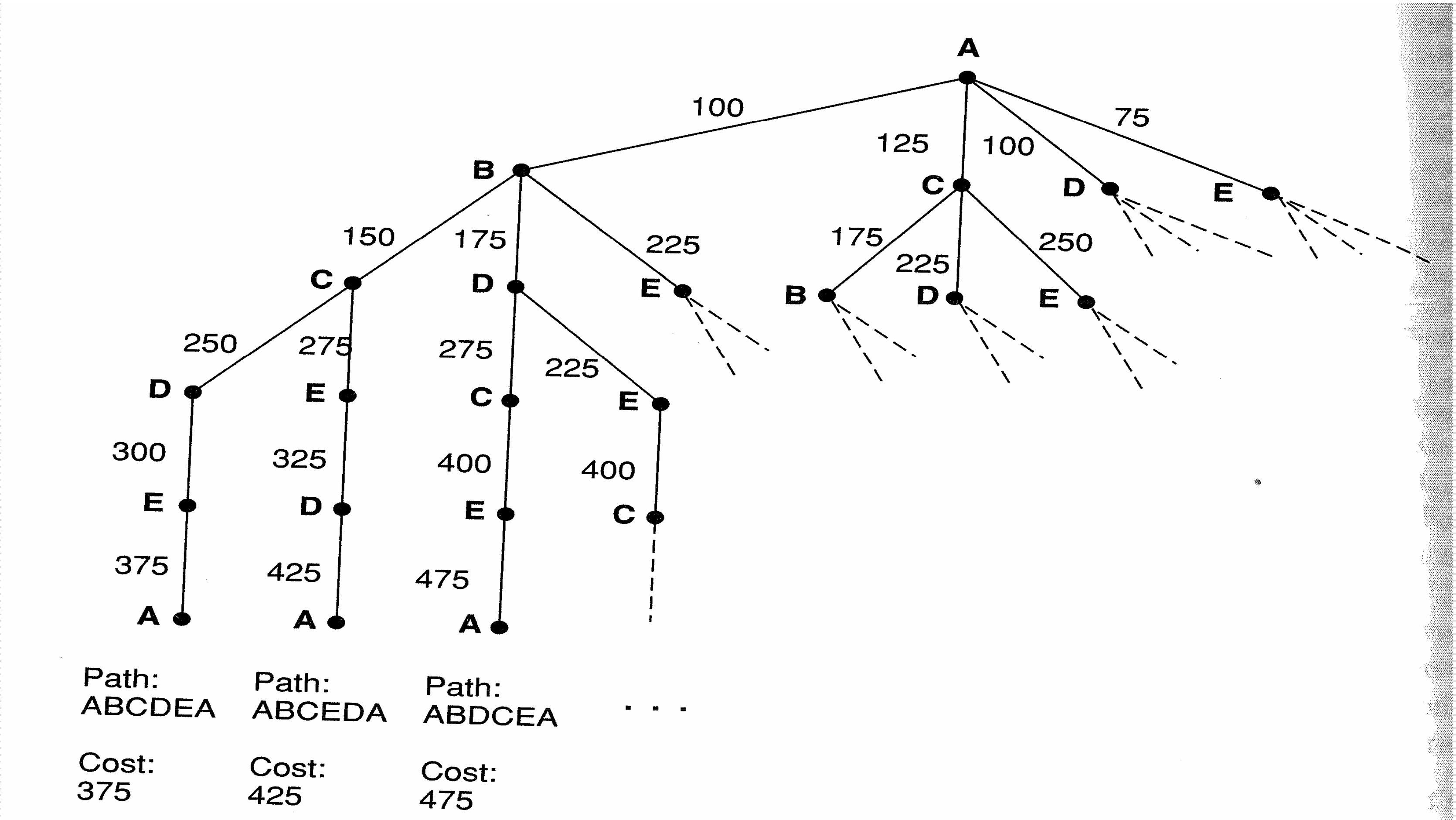
Problem 3 : Travelling Salesman Problem

- *Definition:* Suppose a salesperson has N cities to visit and then must return home. What is the shortest path for the salesperson to travel, visiting each city, and then returning to the home city.
- The arcs are labelled with the distance between the cities it connects.
- The GD requires a complete circuit with minimum cost.
- This is different from what we've seen so far because the GD is a property of the entire path rather than a single state!

Problem 3 : Travelling Salesman Problem (Search)



Problem 3 : Travelling Salesman Problem



Problem 3 : Travelling Salesman Problem (a simple heuristic)

- Rule “go to the closest unvisited city”
- Rule “go to nearest neighbour”
- Very efficient but will not give you the optimal path !!

Strategies (Data vs Goal driven search)

- *Data-Driven search* (forward chaining): problem solver begins with the given facts of the problem and a set of legal moves or rules for changing state

- *Goal-Driven search* (backward chaining): take the goal that we want to solve, then see what rules or legal moves could be used to generate this goal and determine what conditions must be true to use them.

Depth / Breadth First Searches

- (describe using 8-puzzle game)

Best First Search (start using heuristics)

- Hill climbing strategies expand the current state in the search and evaluate its children. The best child is selected for further expansion.
- Search halts when it reaches a state that is better than an of its children.
- Problem of become stuck in a local maxima!! Eg Genetic algorithms