CS 401 Artificial Intelligence

State Space Search

Lecture 2

Faculty of Computer Science
National University of Computers & Emerging Sciences
Lahore.

Structures & Strategies for State Space Search

- In Today's Lecture:
  - What are States and State Spaces?
  - Problems and their State Space Representation
  - Graph Theoretical Concepts
  - Searching Through State Space for Goal(s)

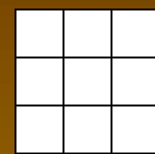
Problem Solving and State Space

In solving many AI problems, we may be faced with answering questions like:

- Is the problem solver guaranteed to find a solution to our problem?
- Will the solution procedure (which usually a search for the solution) always terminate, or can it become caught in an infinite loop?
- When a solution is found, is it guaranteed to be "optimal"?
- What is the complexity of the search process in terms of time/storage requirements?
- Many problems in A.I. can be considered to be what is called a state space search, but what are state spaces? State spaces are best described by an example.

Tic-Tac-Toe

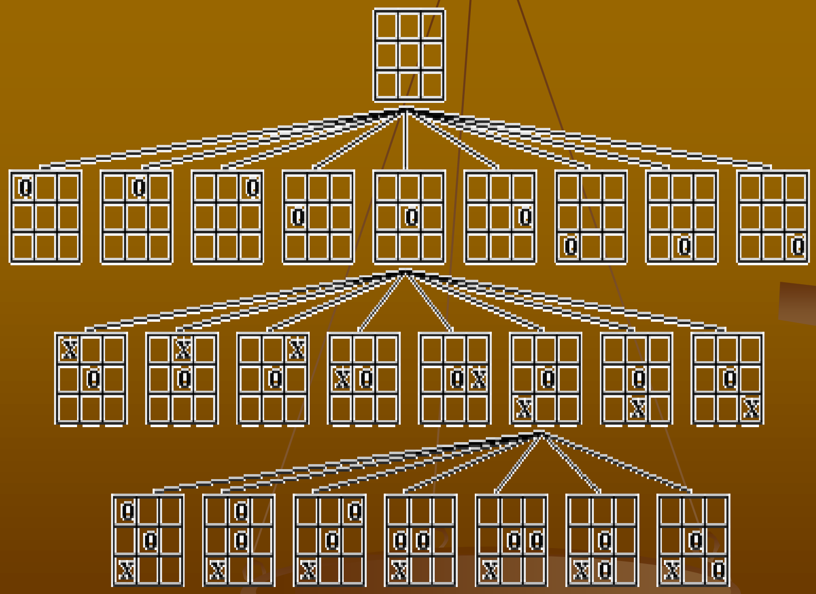
- Consider the game of Noughts and Crosses (tic-tac-toe).



- There are nine spaces that can be occupied by either a space, a nought or a cross.
- Each square has a given state and the playing area as a whole has a given state defined by the states of the squares comprising it.
- Assume you have just started a new game and your opponent has placed a nought in the centre square. The board is now in a state which is your *Initial State* and your *Goal State* is to have three crosses in a line without three noughts in a line.
- From your *Initial State* you need to find the next state that will bring you closest to your *Goal State* and this is where search comes in.

Tic-Tac-Toe (continued…)

- Looking at this game as a whole, we can see that there is a reasonably large number of possible states, some of which represent a win, some a lose and some are mid-game.
- From any given state only a small subset of the possible states can be arrived at in one move and this subset will be defined by the *rules* and the current state.
- A graph can be constructed that starts with the empty board - the first node - which then branches along the nine possible moves from this state to give the nine possible states that can be arrived at. For each of these new nodes, more branches may be drawn and new nodes added until the board is full or a game is won.



Tic-Tac-Toe (continued …)

- The picture shows a graph representing just three moves and the game tree produced just by following that game, as you can imagine it gets pretty big if you draw all the possible nodes for each level of the game.
- If the empty board is level 0 and the board after the first move is level 1 then by level 4 the tree has widened to 3024 nodes!
- This is just a very simple game yet it demonstrates very well the way that the number of states to search explodes as you seek to look further and further down the tree.
- The entire set of possible states makes up the *State Space* for the game and this is smaller than the total possible number of nodes in the tree, why?
- The reason is that many of the nodes in the tree are identical!
- Although they are arrived at by a different route, the actual position of the noughts and crosses can be the same.
- Two ways of arriving at the same state:

Game One

1 O at Top Left

2 X at Top Right

3 O at Centre

Game Two

1 O at Centre

2 X at Top Right

3 O at Top Left

Tic-Tac-toe (continued …)

- The end results are identical so end up at the same state in the state space, but they are different nodes on the game tree.
- Each state in the game can be simply represented by a 3 x 3 array of which each entry can hold one of three values - 0, 1, and 2, and these represent an empty space, a nought and a cross respectively.
- The rules of the game are simple:
  1. Moves must alternate between players - one player makes a move then the other player does.
  2. A nought or a cross may only be placed on an empty space.
  3. A win is accomplished if a line of three noughts or crosses is achieved.
  4. The game is drawn when the board is filled without rule 3 being satisfied.
- So using the state representation and the set of rules, how can a computer play Noughts and Crosses?

The Traveling Salesman

- Given a map with a number of cities on and given that you can get to each city directly from any other city on the map with the distance for each of these, what is the minimum distance needed to visit every city only once?
- Well we can draw a tree again, picking any city at random to start with and branch out to other cities from this city.
- At each node we keep a record of cities visited and the total distance traveled thus far.
- In this case, the tree is finite again as in Tic-Tac-Toe, you may not visit a city twice, so the tree can only be as many levels deep as there are cities to visit.
- A variation on the problem can be insoluble; if cities are not always reachable from other cities, i.e. you have to go via C to get from A to B, then there exist maps for which there is NO solution.
  So how do we solve this one?
- These are all difficult problems of different types and they each require some form of search algorithm to come to a sensible solution, if any solution exists! We will look at some of these different techniques and try to ascertain how to decide which type of search to use.
- To formalize the approach, we can use Graph Theoretical Concepts. Graph theory can be used to "analyze the structure and complexity" of both the "problem" and the "procedures used to solve it".

Graph Theory (Definitions)

- A graph consists of:
  - A set of "nodes" N_1, N_2, …, N_n, … which need not be finite.
  - A set of "arcs" that connect pairs of nodes. (Arcs are often described as an ordered pair of nodes; e.g. the arc (N_3,N_4) connects nodes N_3 with N_4).
- A "labeled graph" has one or more descriptors (labels) attached to each node that distinguish that node from any other node in the graph.
- In the state space model of problem solving
  - The nodes of a graph are taken to represent discrete "states" in a problem solving process (such as the result of logical inferences or configurations of a game board).
  - The arcs of the graph represent transitions between states. (These transitions correspond to logical inferences or legal moves of a game).
- In "Expert Systems”, states describe our knowledge of a problem instance at some stage of a reasoning process.
- Expert knowledge in the form of “if … then” rules, allows us to generate new information; the act of applying a rule is represented as an arc between states.
- A “directed graph” has an indicated direction for traversing each arc. For example, a directed graph might have (N_3,N_4) as an arc but not (N_4,N_3). Thus a “path” could go from node N_3 to N_4 but not from N_4 to N_3.

Graph Theory Definitions (continued …)

- If a directed arc connects N_j to N_k, then N_j is called the "parent" of N_k and N_k is called the child of N_j. If the graph also contains an arc (N_j,N_l), then N_k and N_l are called "siblings".
- An ordered sequence of nodes $[N_1,N_2, \ldots, N_n]$, where each N_i, N_{i+1} in the sequence represents an arc (N_i,N_{i+1}), is called a "path" of length n-1 in the graph.
- A "rooted graph" has a unique node N_s from which all paths in the graph originate. That is, the root has no parent in the graph. (The state space graphs of the most games are usually rooted graphs, with the start of the game at the root).
- A "tip" or "leaf" node is a node that has no children.
- On rooted graph, a node is said to be the "ancestor" of all nodes positioned after it (or to its right) and a "descendant" of all nodes before it (or to its left).

Graph Theory Definitions (continued …)

- A path that contains any node more than once, is said to contain a “cycle” or “loop”.
- A “tree” is a graph in which there is a unique path between every pair of nodes. (The paths in a tree contain no cycles).
- The edges in a rooted tree are directed away from the root.
- Each node in a rooted tree has a unique parent.
- Two nodes in a graph are said to be “connected” if a path exists that includes them both.

State Space (Formal Definition)

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 (links) between nodes. These correspond to the steps in a problem solving process.
- S is a nonempty subset of N, that contains the start state(s) of the problem.
- GD is a nonempty subset of N, that 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.

A "solution path" is a path through this graph from a node in S to a node in GD.

There are 9! Paths in tic-tac-toe.
There are 10^{40} paths in the game of checkers.
There are 10^{120} paths in the game of chess.

Exhaustive search through all these paths may not be possible!

Home Work

Read Chapter 3 of George F. Luger