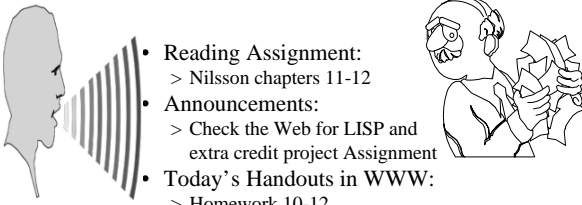


EEL5840: Elements of Machine Intelligence

SubjuGator

Announcements



- Reading Assignment:
 - > Nilsson chapters 11-12
- Announcements:
 - > Check the Web for LISP and extra credit project Assignment
- Today's Handouts in WWW:
 - > Homework 10-12
 - > Outline for Class 26
 - > www.mil.ufl.edu/eel5840
 - > Software and Notes

University of Florida
EEL 5840 - Class #26 - Fall 2009
© Dr. A. Armitage Armitage

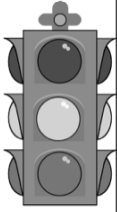
1

EEL5840: Elements of Machine Intelligence

SubjuGator

Today's Menu

- Adversarial Search
- MINIMAX Game Searches



University of Florida
EEL 5840 - Class #26 - Fall 2009
© Dr. A. Armitage Armitage

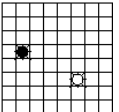
2

EEL5840: Elements of Machine Intelligence

SubjuGator

Adversarial Search

- Adversarial Search
 - > When there are multiple agents, if we lack knowledge of how agents might act, we can do little more than use a sense/plan/act architecture that does not plan too far into the future.
 - > However, when knowledge permits, an agent can construct plans that explicitly consider actions of other agents. This is especially true in situations where agents take turns acting.
 - > Example:
 - Robots (B & W) can each move into an adjacent cell in their row/col
 - They move alternatively, and when it is their turn they must move
 - Suppose W has the goal to capture B (by moving to the cell where B is)
 - B's goal is to prevent this from happening



University of Florida
EEL 5840 - Class #26 - Fall 2009
© Dr. A. Armitage Armitage

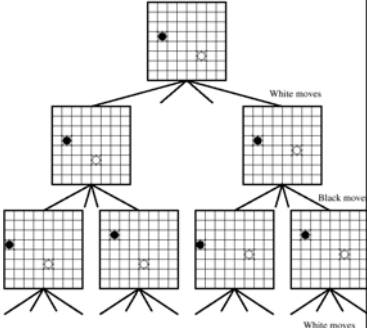
3

EEL5840: Elements of Machine Intelligence

SubjuGator

Adversarial Search

- Example
 - > W plans by constructing a search tree in which, at alternate levels, B's moves are considered also.
 - > Is it possible for an agent to find a move such that no matter what the opponent does, they will succeed?



University of Florida
EEL 5840 - Class #26 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence
 SubjuGator
Adversarial Search

- Adversarial Search
 - The examples considered will be instances of two agent, perfect information, zero sum games. The agents (players) move in turn until either one of them wins or the result is a draw.
 - Each player has a complete and perfect model of the environment and of its own and the opponent's actions—neither player has perfect knowledge of what the opponent will actually do in any situations
 - Examples are chess, checkers, Go, Tic-Tac-Toe and Nim.
 - Iconic Representations naturally suggest themselves in setting up state-space descriptions for many games such as chess.
 - Search trees are constructed as before, although different techniques are used for selecting the first move

University of Florida
 EEL 5840 - Class #2 - Fall 2009
 © Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence
 SubjuGator
Adversarial Search

- Adversarial Search
 - Example: Grundy's Game
 - Two players have in front of them a single pile of n objects, say, pennies. The first player divides the original stack into two stacks that *must* be unequal. Each player alternatively thereafter does the same to some *single* stack when it is their turn to play. The game proceeds until every stack has either just one penny or two—at which point continuation becomes impossible. The first player who first cannot play is the loser.
 - Suppose $n=7$ and we call out two players MAX and MIN and let MIN go first.

University of Florida
 EEL 5840 - Class #2 - Fall 2009
 © Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence
 SubjuGator
Adversarial Search


Example: Grundy's Game

University of Florida
 EEL 5840 - Class #2 - Fall 2009
 © Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence
 SubjuGator
Adversarial Search

Example: Grundy's Game $n=8$

University of Florida
 EEL 5840 - Class #2 - Fall 2009
 © Dr. A. Aravamudan




EEL5840: Elements of Machine Intelligence

Adversarial Search

- **Minimax Procedure**
 - > Let the two players be named MAX and MIN. MAX plays first so our task is to find the best move for MAX.
 - > Even-numbered nodes will be those when it is MAX's move next. These are called MAX nodes. Similarly odd-numbered nodes are MIN's move next. These are called MIN nodes.
 - > A ply of depth k in a game tree consists of the nodes of depths $2k$ and $2k+1$. The extent of search in a game tree is given in terms of ply depth. (The start node is defined to be at depth 0)
 - > Because the size of most games makes it infeasible for search (10^{40} nodes for chess) we use methods similar to limited horizon search.
 - > We can use BFS, DFS or Heuristic Search with a modified termination condition (time limit, depth, memory, etc.)

University of Florida
EEL 5840 - Class #2 - Fall 2009
© Dr. A. Aravamudan

9




EEL5840: Elements of Machine Intelligence

Adversarial Search

- **Minimax Procedure**
 - > After search terminates we extract an estimate of the best-first move by using a static-evaluation function which measures the "worth" of the leaf nodes of the search tree.
 - > By convention, game positions favorable to MAX cause the evaluation function to have a positive value; positions favorable to MIN have a negative value; and values near zero correspond to game positions not particularly favorable to either
 - > In the MINIMAX procedure we assume that if MAX were to choose among the tip nodes of a k -ply search tree he would prefer that node having the largest evaluation. Since MAX can indeed pick that node when it is his time to play the backed-up value of a MAX node parent of MIN tip nodes is equal to the maximum of the static evaluation of the tip nodes.

University of Florida
EEL 5840 - Class #2 - Fall 2009
© Dr. A. Aravamudan

10




EEL5840: Elements of Machine Intelligence

Adversarial Search

- **Minimax Procedure**
 - > If MIN were to choose among the tip nodes of a k -ply search tree he would prefer that node having the smallest evaluation. Since MIN could indeed pick that node when it is his time to play the backed-up value of a MIN node parent of MAX tip nodes is equal to the minimum of the static evaluation of the tip nodes.
 - > After the parent of all tip nodes have been assigned backed-up values, we back up values another level, MAX picks the successor of MIN nodes having the largest backed-up value, MIN would choose that successor on MAX nodes with the smallest value.
 - > We continue to backup values until we reach the start node where MAX can now pick the best available move.

University of Florida
EEL 5840 - Class #2 - Fall 2009
© Dr. A. Aravamudan

11



EEL5840: Elements of Machine Intelligence

Adversarial Search

- **Minimax Procedure**
 - > Why should this work? The backed-up values of the start node's successors are more reliable measures of the ultimate relative worth of these positions than are the values that would be obtained by directly applying the static evaluation function to these positions. If we "look ahead" we should get a better view of the game and by doing so we depend on features occurring near the end of the game as opposed to the beginning.
 - > We illustrate the method with Tic-Tac-Toe with a depth bound of two, conduct a BFS search until all the nodes at level two are generated, and then we apply the static evaluation function $e(p)$.
 If p is not a winning position for either player

$$e(p) = \sum_{\text{MAX}} (\text{available rows, cols, diag}) - \sum_{\text{MIN}} (\text{available rows, cols, diag})$$

$$e(p) = \infty \text{ if } p \text{ is a win for MAX} \quad e(p) = -\infty \text{ if } p \text{ is a win for MIN}$$

University of Florida
EEL 5840 - Class #2 - Fall 2009
© Dr. A. Aravamudan

12

EEL5840: Elements of Machine Intelligence

SubjuGator

- Minimax Procedure
 - > If $p = \begin{matrix} _ & 0 & _ \\ _ & x & _ \\ _ & _ & _ \end{matrix}$
 - $e(p) = \sum(2,2,2) - \sum(2,2,2) = 6 - 4 = 2$
 - > Symmetries are used in generating successor positions

University of Florida
EEL 5840 - Class #2 - Fall 2009
© Dr. A. Aravamudan

13

EEL5840: Elements of Machine Intelligence

SubjuGator

University of Florida
EEL 5840 - Class #2 - Fall 2009
© Dr. A. Aravamudan

14

EEL5840: Elements of Machine Intelligence

SubjuGator

University of Florida
EEL 5840 - Class #2 - Fall 2009
© Dr. A. Aravamudan

15

EEL5840: Elements of Machine Intelligence

SubjuGator

The End!

University of Florida
EEL 5840 - Class #2 - Fall 2009
© Dr. A. Aravamudan

16