Game-Playing Algorithms (2)

CSCI 5582, Fall 2007
Assignments

- To read: Chapter 6, sections 1-4
- Problem Set 1 due next week
Game-Playing: the Essential Arsenal

• The idea of a “minimax search”
• Avoiding useless search: the alpha-beta algorithm
• Other ways of pruning the search tree
• Other central issues in game-playing
In Scheme:

(define (minimax gamestate myfunction otherfunction level)
  (cond ((= level 0) (getvalue gamestate))
    (else
      (let* ((successors (getnextmoves gamestate))
              (othervals (map (lambda (successor)
                               (minimax successor otherfunction
                                myfunction (- level 1)))
                     successors))
        (apply myfunction othervals)))))
(define (alpha-beta alpha beta level max?)
  (cond ((= level 0) (static-value node))
    (max? (let ((children (find-next-moves node #t)))
       (max-look-at-each-child-node
        children alpha beta level)))
    (else (let ((children (find-next-moves node #f)))
       (min-look-at-each-child-node
        children alpha beta level))))

(define (max-look-at-each-child-node children alpha beta level)
  (cond ((null? children) alpha)
    ((>= alpha beta) alpha)
    (else (let ((next-value
             (alpha-beta
              alpha beta (- level 1) (first children) #f)))
       (max-look-at-each-child-node
        (rest children)
        (max next-value alpha)
        beta level)))))

(define (min-look-at-each-child-node children alpha beta level)
  (cond ((null? children) beta)
    ((>= alpha beta) beta)
    (else (let ((next-value
             (alpha-beta
              alpha beta (- level 1) (first children) #f)))
       (min-look-at-each-child-node
        (rest children)
        alpha (min next-value beta) level)))))
Calling alpha-beta with starting values of a (very low) alpha value, a (very high) beta value, a depth of 5, the starting game state, and from the point of view of the maximizer.

(alpha-beta -1000 1000 5 initial-game-state #t)
Popular Strategies for Game Players

• Forward pruning
• Tapered search
• Iterative deepening
Other ideas in game search

• How do you evaluate a game state, anyhow?
• The notion of “quiescence”
• The “horizon effect”
• Dealing with chance moves (e.g., dice)
How do people play games?

• Chase and Simon experiment on chess-players’ memory
• Do we perform search?
• The chess program as Turing test
Logic in AI

• The appeal of logic as a representation of knowledge
• Logic as “applied” (in contrast to “cognitive”) AI
• A variety of logics
Propositional Logic: the Basic Lingo

• Atomic propositions (P, Q, etc.) with binary truth values
• Connectives AND, OR, NOT, IMPLIES, XOR, IFF
• Values of compound sentences are determined as functions of the truth values of individual propositions (truth tables)
• Rules of inference: Modus Ponens, And-Elimination, various equivalence forms (like De Morgan’s rules)