From Search to Games

FEB
10
View Calendar

THIS DAY IN **HISTORY**

Sports

1996

Deep Blue beats Kasparov at chess

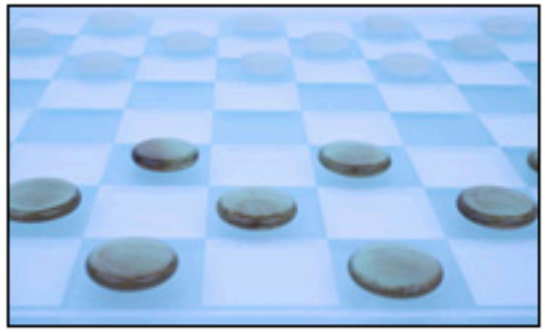
Computers Solve Checkers—It's a Draw

King me! Top computer scientist proves perfect play leads to draw, recounts battle for world championship, gets kinged

By JR Minkel | July 19, 2007

Jonathan Schaeffer's quest for the perfect game of checkers has ended. The 50-year-old computer scientist from the University of Alberta in Edmonton left human players in the dust more than a decade ago after a trial by fire against the greatest checkers champion in history.

And now, after putting dozens of computers to work night and day for 18 years—*jump, jump, jump*—he says he has solved the game—*king me!*. "The starting position, assuming no side makes a mistake, is a draw," he says.



© ISTOCKPHOTO/CHRISTOPHER O DRISCOLL

January 2015

Heads-up limit Texas hold 'em poker solved by University of Alberta scientists

Poker algorithm another step toward artificial intelligence

By Aleksandra Sagan, CBC News Posted: Jan 08, 2015 3:39 PM ET | Last Updated: Jan 08, 2015 3:39 PM ET

Scientists at the University of Alberta have essentially solved heads-up limit hold 'em poker with an algorithm they hope will lead to advances in artificial intelligence. (Shutterstock)

Readings for this class

- Chapter 5-5.4

Learning Objectives

- how to describe a game in AI terms
- basic concepts of game-theory
- how the minimax algorithm finds an optimal move
- benefits of alpha-beta pruning
- simple strategies for position evaluation functions

From problem formulation to game defn

- states: description of “world of interest”
- initial state
- successor function: generates set of legal next states from available actions
- goal test → terminal test: when is game over?
- path cost → utility function: value of terminal states

Why search won't work

- search for sequence of moves that leads to terminal state with positive utility (winning state)
- opponent might not be so cooperative!

Optimal Decisions in 2-player games

- solution: find strategy that leads to winning state regardless of what opponent does

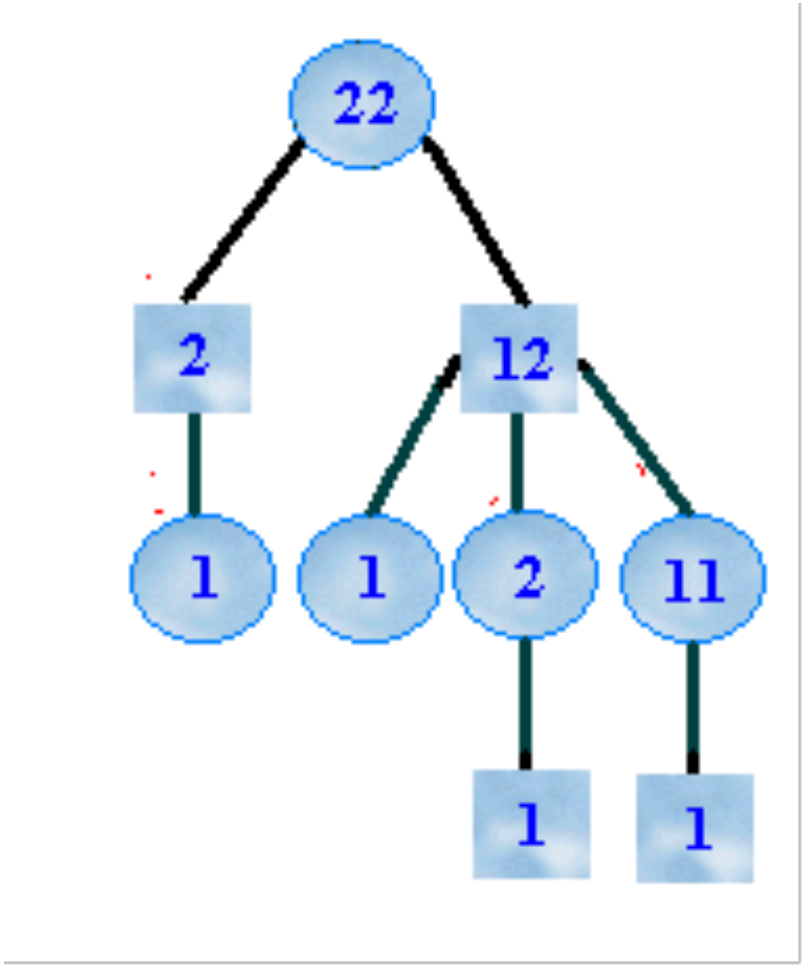
Minimax strategy for 2-player games

- generate whole game tree down to terminal nodes
- find value of each terminal state using **utility function**
- repeat
  - determine utility of parent nodes from children
    - MIN chooses move that minimizes utility
    - MAX chooses move that maximizes utility
- until we reach root
- choose move that leads to “best” value

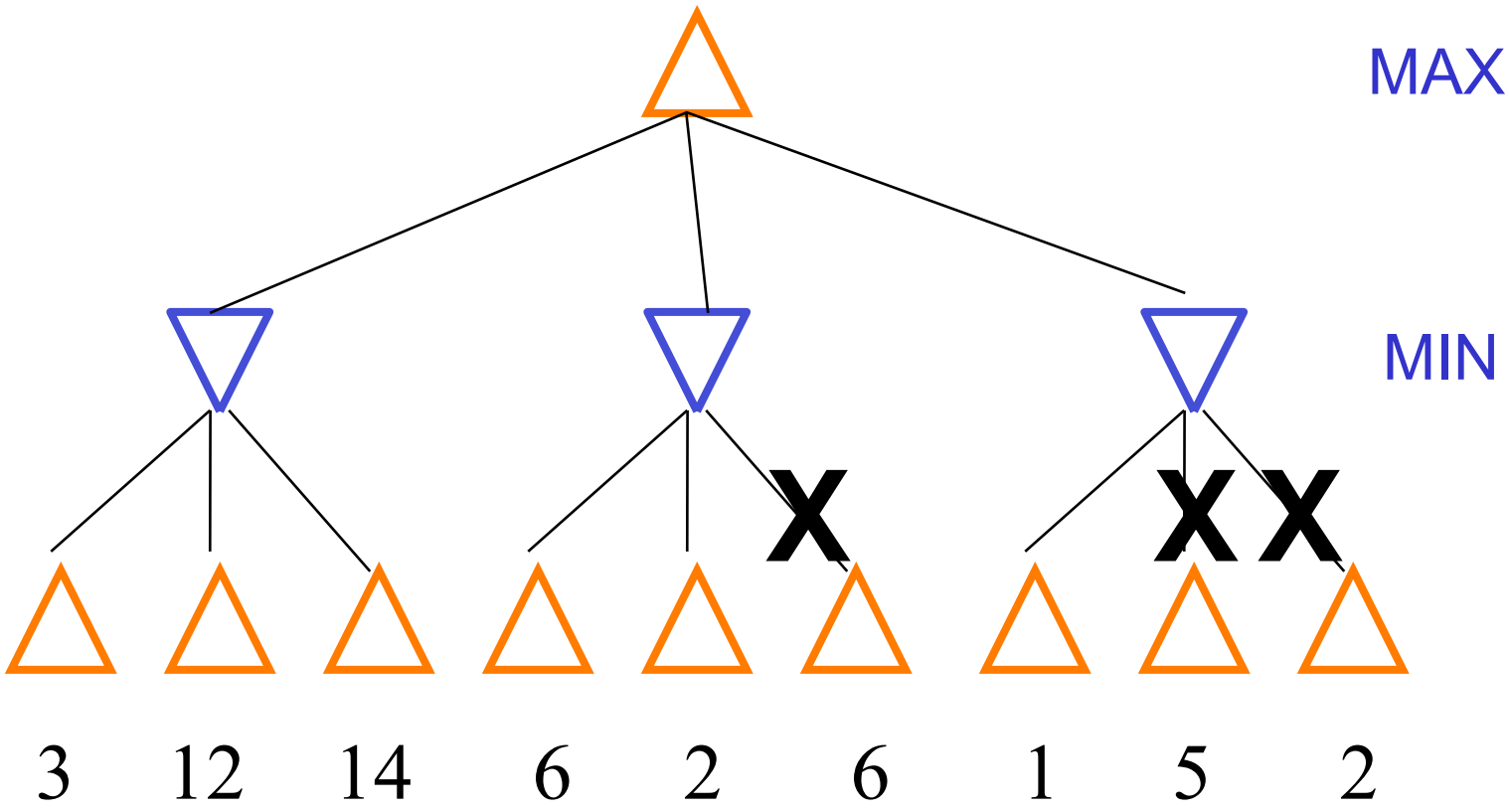
MiniMax algorithm

```
minmax(u) {   // u is node to evaluate
   if u is a leaf return score of u;
     else if u is a min node
           for all children of u: v1, .. vn
                return min{ minmax(v1),..., minmax(vn)}
     else  // u is a max node
      for all children of u: v1, .. vn
                return max{ minmax(v1),...,minmax(vn)}
}
```

NIM - player to take last stick loses



Improved search with alpha-beta pruning



stops evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move

Alpha-Beta pseudocode

```
// from https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning
// initial call is: alphabeta(origin, depth, -∞, +∞, TRUE)

function alphabeta(node, depth, α, β, maximizingPlayer)
    if depth = 0 or node is a terminal node
        return the heuristic value of node
    if maximizingPlayer
        v := -∞
        for each child of node
            v := max(v, alphabeta(child, depth - 1, α, β, FALSE))
            α := max(α, v)
            if β ≤ α  break (* β cut-off *)
        return v
    else
        v := ∞
        for each child of node
            v := min(v, alphabeta(child, depth - 1, α, β, TRUE))
            β := min(β, v)
            if β ≤ α   break (* α cut-off *)
        return v
```

Problem

- usually impossible to explore entire state space (e.g., chess search tree has approx. 35^{100} nodes)
- infeasible to make optimal decision
- solution: use heuristic position evaluators – an estimate of utility of states based on insight

Position Evaluators

- game specific: have to be creative
- what are the determining factors in the goodness of a game state (utility)?
  - e.g., chess:
    - Sum of point value of pieces
    - Control of centre of board
    - Pawn structure
    - Defence of king
    - Mobility of pieces

Homework

- Read **before next class**:
  - Ch. 2
  - Brooks, A Robust Layered Control System for a Mobile Robot