CS39N
The Beauty and Joy of Computing

Lecture #4 : Computational Game Theory

2009-09-14

UC Berkeley
Computer Science
Lecturer SOE
Dan Garcia

CHECKERS SOLVED IN 2007!

A 19-year project led by Prof Jonathan Schaeffer, he used dozens (sometimes hundreds) of computers and AI to prove it is, in perfect play, a … draw! This means that if two Gods were to play, nobody would ever win!

www.cs.ualberta.ca/~chinook/

Computational Game Theory

- **History**
- **Definitions**
  - Game Theory
  - What Games We Mean
  - Win, Lose, Tie, Draw
  - Weakly / Strongly Solving
- **Gamesman**
  - Dan's Undergraduate R&D Group
  - Demo!!
- **Future**

www.eecs.berkeley.edu/Research/Areas/

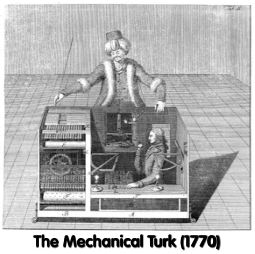
Computer Science … A UCB view

- **CS research areas:**
  - Artificial Intelligence
  - Biosystems & Computational Biology
  - Computer Architecture & Engineering
  - Database Management Systems
  - Graphics
  - Human-Computer Interaction
  - Operating Systems & Networking
  - Programming Systems
  - Scientific Computing
  - Security
  - Theory
  - …

en.wikipedia.org/wiki/The_Turk

The Turk (1770)

- **A Hoax!**
- **Built by Wolfgang von Kempelen**
  - to impress the Empress
- **Could play a strong game of Chess**
  - Thanks to Master inside
- **Toured Europe**
  - Defeated Benjamin Franklin & Napoleon!
- **Burned in an 1854 fire**
  - Chessboard saved…

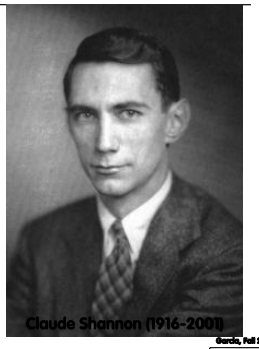


The Mechanical Turk (1770)

en.wikipedia.org/wiki/Claude_Shannon#Shannon.27s_computer_chess_program

Claude Shannon's Paper (1950)

- **The "Father of Information Theory"**
  - Founded the digital computer
  - Defined fundamental limits on compressing/storing data
- **Wrote "Programming a Computer for Playing Chess" paper in 1950**
  - C. Shannon, *Philos. Mag.* 41, 256 (1950).
  - All chess programs today have his theories at their core



Claude Shannon (1916-2001)

en.wikipedia.org/wiki/Deep_Blue_(chess_computer)

Deep Blue vs Garry Kasparov (1997)

- **Kasparov World Champ**
- **1996 Tournament**
  - First game DB wins a classic!
  - But DB loses 3 and draws 2 to lose the 6-game match 4-2
  - In 1997 Deep Blue upgraded, renamed "Deeper Blue"
- **1997 Tournament**
  - GK wins game 1
  - GK resigns game 2
    - even though it was draw!
  - DB & GK draw games 3-5
  - Game 6 : 1997-05-11 (May 11th)
    - Kasparov blunders move 7, loses in 19 moves. Loses tournament 3 ½ - 2 ½
    - GK accuses DB of cheating. No rematch.
- **Defining moment in AI history**

IBM's Deep Blue vs Garry Kasparov

www.cs.berkeley.edu/~ddgarcia/eyawtkagtbwata

What is "Game Theory"?

Combinatorial
- Sprague and Grundy's 1939 Mathematics and Games
- Board games
- Nim, Domineering, dots and boxes
- Film: *Last Year in Marienbad*
- Complete info, alternating moves
- Goal: Last move

Computational
- R. C. Bell's 1988 Board and Table Games from many Civilizations
- Board games
- Tic-Tac-Toe, Chess, Connect 4, Othello
- Film : *Searching for Bobby Fischer*
- Complete info, alternating moves
- **Goal: Varies**

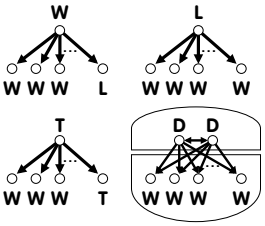
Economic
- von Neumann and Morgenstern's 1944 *Theory of Games and Economic Behavior*
- Matrix games
- Prisoner's dilemma, auctions
- Film : *A Beautiful Mind* (about John Nash)
- **Incomplete** info, simultaneous moves
- Goal: Maximize payoff

What "Board Games" do you mean?

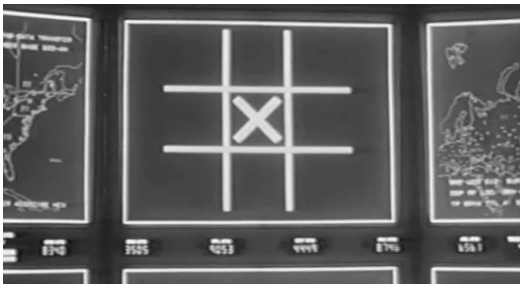
- **No chance, such as dice or shuffled cards**
- **Both players have complete information**
  - No hidden information, as in Stratego & Magic
- **Two players (Left & Right) usually alternate moves**
  - Repeat & skip moves ok
  - Simultaneous moves not ok
- **The game can end in a pattern, capture, by the absence of moves, or …**

Basic Definitions

- **Games are graphs**
  - Position are nodes
  - Moves are edges
- **We strongly solve game by visiting every position**
  - "Playing" every game ever
- **Each position is (for player whose turn it is)**
  - Winning (∃ losing child)
  - Losing (All children winning)
  - Tieing (!∃ losing child, but ∃ tieing child)
  - Drawing (can't force a win or be forced to lose)



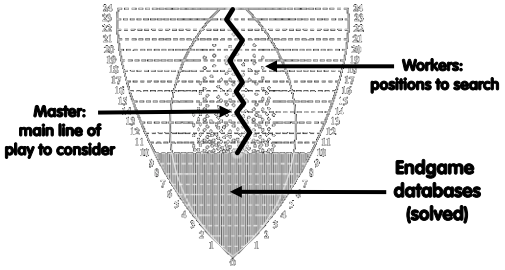
What did you mean "strongly solve"?



Wargames (1983)

Thanks to Jonathan Schaeffer for this slide…

Weakly Solving A Game (Checkers)



Log of Search Space Size

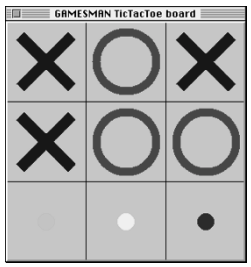
Example: 1,2,…,10

- **Rules (on your turn):**
  - Running total = 0
- **Rules (on your turn):**
  - Add 1 or 2 to running total
- **Goal**
  - Be the FIRST to get to 10
- **Example**
  - Ana: "2 to make it 2"
  - Bob: "1 to make it 3"
  - Ana: "2 to make it 5"
  - Bob: "2 to make it 7" → photo
  - Ana: "1 to make it 8"
  - Bob: "2 to make it 10" I WIN!

7 ducks (out of 10)

Example: Tic-Tac-Toe

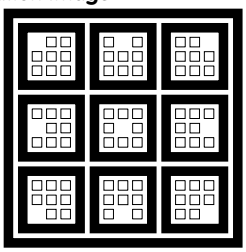
- **Rules (on your turn):**
  - Place your X or O in an empty slot on 3x3 board
- **Goal**
  - If your make 3-in-a-row first in any row / column / diag, win
  - Else if board is full with no 3-in-row, tie
- **Misére is tricky**
  - 3-in-row LOSES
  - Pair up and play now, then swap who goes 1st



Values Visualization for Tic-Tac-Toe

Tic-Tac-Toe Answer Visualized!

- **Recursive Values Visualization Image**
- **Misére Tic-tac-toe**
  - Outer rim is position
  - Inner levels moves
  - Legend
    - ☐ Lose
    - ☐ Tie
    - ☐ Win



Misére Tic-Tac-Toe 2-ply Answer

Computational Game Theory

- **Large games**
  - Can theorize strategies, build AI systems to play
    - Using "Endgame databases"
  - Can study endgames, smaller version of orig
    - Examples: Quick Chess, 9x9 Go, 6x6 Checkers, etc.
  - Can put 18 years into a game [Schaeffer, Checkers]
- **Small-to-medium games**
  - Can have computer strongly solve and…
    - Play against it and teach us strategy
    - Allow us to test our theories on the database, analysis
    - Analyze human-human game and tell us where we erred!
  - Big goal: *Hunt Big Game* – those not solved yet
  - I wrote GAMESMAN in 1988 (almost 20 yrs ago!), the basis of my GamesCrafters research group

GamesCrafters.berkeley.edu

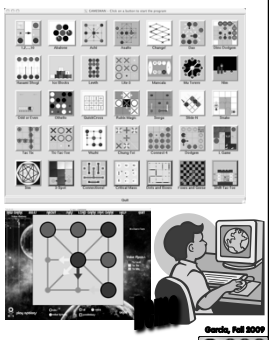
GamesCrafters

- **Undergraduate Computational Game Theory Research Group**
- **250+ students since 2001**
  - We now average 20/semester!
  - They work in teams of 2+
- **Most return, take more senior roles (sub-group team leads)**
  - Maximization (bottom-up solve)
  - Oh, DeepaBlue (parallelization)
  - GUI (graphical interface work)
  - Retro (GUI refactoring)
  - Architecture (core)
  - New/ice Games (add / refactor)
  - Documentation (games & code)

GamesCrafters

- **Projects span CS areas**
  - AI : Writing "intelligent" players
  - DB: How do we store results?
  - HCI: Implementing interfaces
  - Graphics: Values visualizations
  - SE: Lots of SE juice here, it's big!
    - Defining & implementing APIs
    - Managing open source SW
  - OS: We have our own VM
    - Also eHarmony & net DB
  - PL: We're defining languages to describes games and GUIs
  - THY: Lots of combinatorics here: position & move hash functions
- **Perennial Cal Day favorite!**
- **"Research and Development can be fun?!"**

Lines of Code:	
8K	Java
80K	Tcl/Tk
155K	C



Future

- **Board games are exponential in nature**
  - So has been the progress of the speed / capacity of computers!
  - Therefore, every few years, we only get to solve one more "ply"
- **One by one, we're going to solve them and/or beat humans**
  - We'll never solve some
    - E.g., hardest game : Go

174089650659031927907185238070564367946027245026354119482811870680105167618464984116279288988714938612096988816320780613754987181355093129514803369660572893075468180597603

Go's search space ~ 3^{361}