UNIT 13A

Image: http://www.wallsscreen.com/

AI: Games & Search Strategies

15110 Principles of Computing, Carnegie Mellon University

Artificial Intelligence

- Branch of computer science that studies the use of computers to perform computational processes normally associated with human intellect and skill.

- Some areas of AI:

  Game playing
  Robotics
  Natural language processing
  Knowledge representation
  Machine learning
  Music, Speech & Vision

A Cynic’s View

- AI is the study of how to get computers to do things we don’t understand.
  - “thinking”, “learning”, “creativity”, etc.
- When we do understand something, it’s no longer AI: it’s just programming.

Examples:

speech recognition,
computer chess and checkers,
robotics, …

Games and Search

• • •

A laboratory for artificial intelligence

Why Study Games?

- Thin end of the wedge for AI research
    - Characteristically **human activity**
    - Small so potentially **solvable**
    - Easy to **measure** success or failure
    - Solutions *might* tell us something about intelligence in general

Game Analysis

- For most games, the number of possible moves and potential outcomes is HUGE.

combinatorial
EXPLOSION

Example:

if there approximately 20 possible moves at each turn of a chess game, looking ahead 15 moves requires examining about 3.3 x 10^{19}☐ sequences, which would take ***years.***

Infeasibility

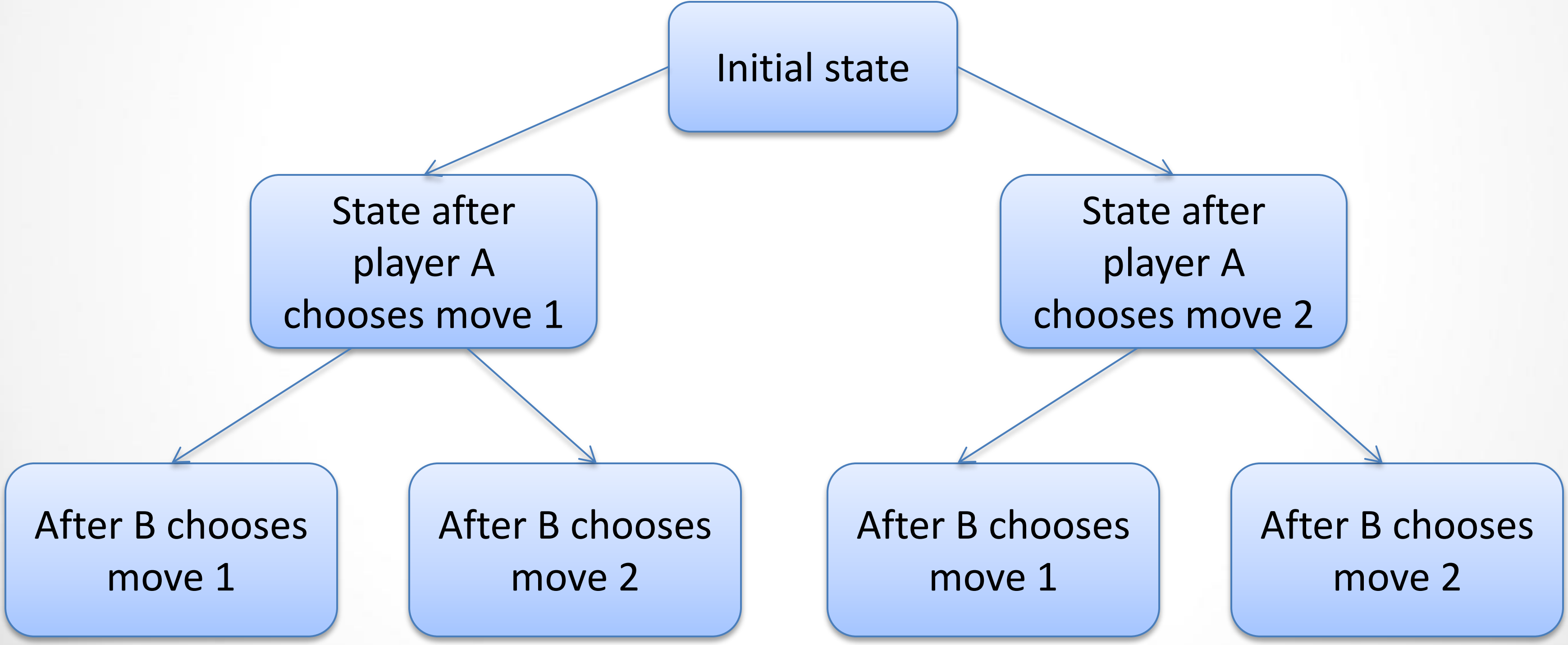
- A complete analysis of a chess game is *computationally infeasible*
- (What other problem have we seen lately that is computationally infeasible? Hint: in that case, infeasibility was a *desirable feature,* not a problem.)
- But human beings play chess, some very well!
    - *Idea:* computer techniques to deal with combinatorial explosion may reveal something important about intelligence

Managing combinatorial explosion

- An AI technique used to manage the explosion is the use of a *game tree*.
    - A tree is built with a **root node** representing the **current state** of the game.
    - **Child nodes** are generated representing the state of the game for each **possible move**.
    - The tree is propagated down, building **more child nodes** for moves allowed by the **next move**, etc.
    - **Leaves** are terminal (win/lose/draw) states of the game.
    - **Unlike the trees** you've studied before, game trees are not usually computed in their entirety (they are too large!)

Game Tree

- Imagine a two-player game with two possible moves at each point. *A* goes first:



Initial state

State after player A chooses move 1

State after player A chooses move 2

After B chooses move 1

After B chooses move 2

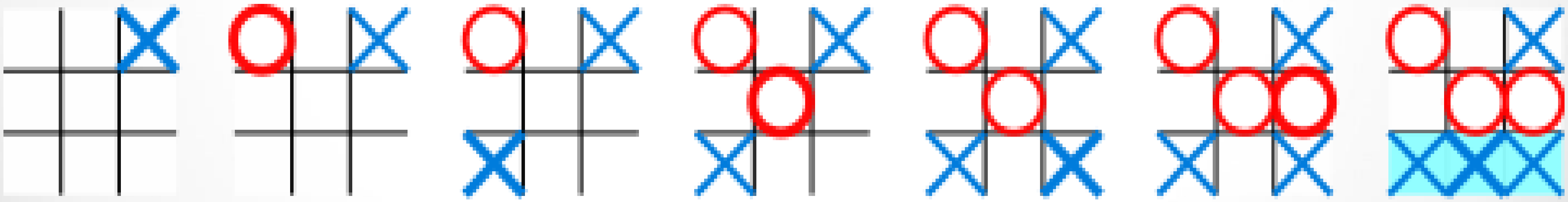
After B chooses move 1

After B chooses move 2

A **Small** Game Tree: Tic Tac Toe

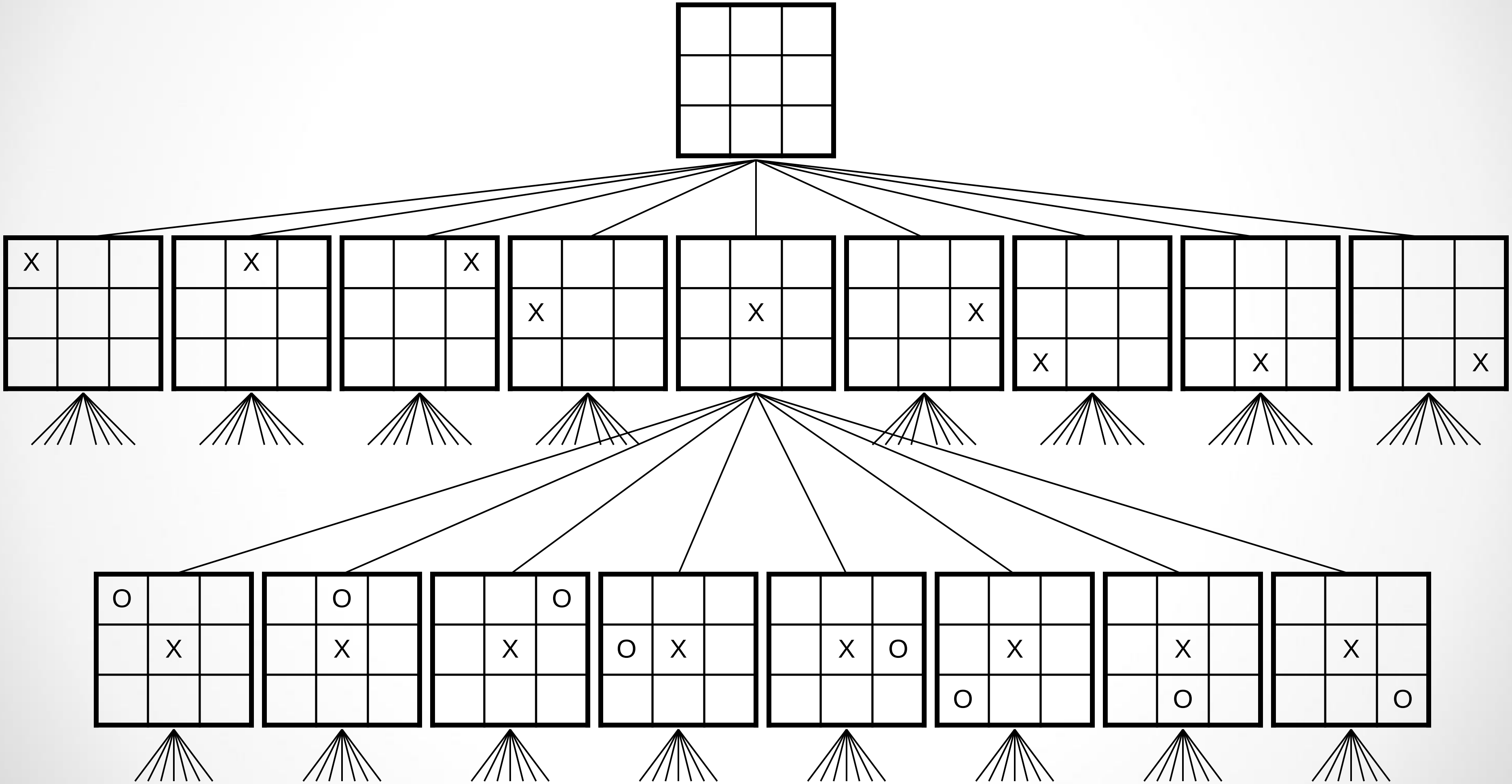
A pencil and paper game for two players, *X* and *O*, who take turns marking the spaces in a 3 × 3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.

The following example game is won by the first player, X:



Source: Wikipedia

Tic Tac Toe Game Tree



How Big is the Tic-Tac-Toe Tree?

- Assuming that all nine positions must be filled before the game ends, how big does this tree get?

How Big is the Tic-Tac-Toe Tree?

- Assuming that all nine positions must be filled before the game ends, how big does this tree get?

At most 9*8*7*6*5*4*3*2*1 = 9! = 362,880

How Big is the Tic-Tac-Toe Tree?

- Assuming that all nine positions must be filled before the game ends, how big does this tree get?

  At most 9*8*7*6*5*4*3*2*1 = 9! = 362,880

- Of course, in real Tic-Tac-Toe, a player can win without filling the whole board.
  - What is the first level of the tree where this can occur?

How Big is the Tic-Tac-Toe Tree?

- Assuming that all nine positions must be filled before the game ends, how big does this tree get?

  At most 9*8*7*6*5*4*3*2*1 = 9! = 362,880

- Of course, in real Tic-Tac-Toe, a player can win without filling the whole board.
  - What is the first level of the tree where this can occur?

    Ending on the 5th move. 3Xs, 2 Os

How Big is the Tic-Tac-Toe Tree?

- Assuming that all nine positions must be filled before the game ends, how big does this tree get?

  At most 9*8*7*6*5*4*3*2*1 = 9! = 362,880

- Of course, in real Tic-Tac-Toe, a player can win without filling the whole board.
  - What is the first level of the tree where this can occur? Ending on the 5th move. 3Xs, 2 Os
  - How big is this tree up to this level?

How Big is the Tic-Tac-Toe Tree?

- Assuming that all nine positions must be filled before the game ends, how big does this tree get?

  At most 9*8*7*6*5*4*3*2*1 = 9! = 362,880

- Of course, in real Tic-Tac-Toe, a player can win without filling the whole board.
  - What is the first level of the tree where this can occur?
    Ending on the 5th move. 3Xs, 2 Os
  - How big is this tree up to this level?
    9*8*7*6*5 = 15,120

Game Tree as Search Space

- We say we ***search* the game tree** as we try to compute a winning move
  - In reality AI programs generate a small part of the tree only!

- ***Search space*** or ***solution space*** — a powerful general idea in life as in computing

Tic-Tac-Toe Is Completely Searchable

- For the first move, pick a square at random.

  (Always picking the same square would be boring.)

- For subsequent moves:
  - Using recursion, grow the entire game tree with the current board state as the root (fewer than 362,000 nodes, *feasible*)

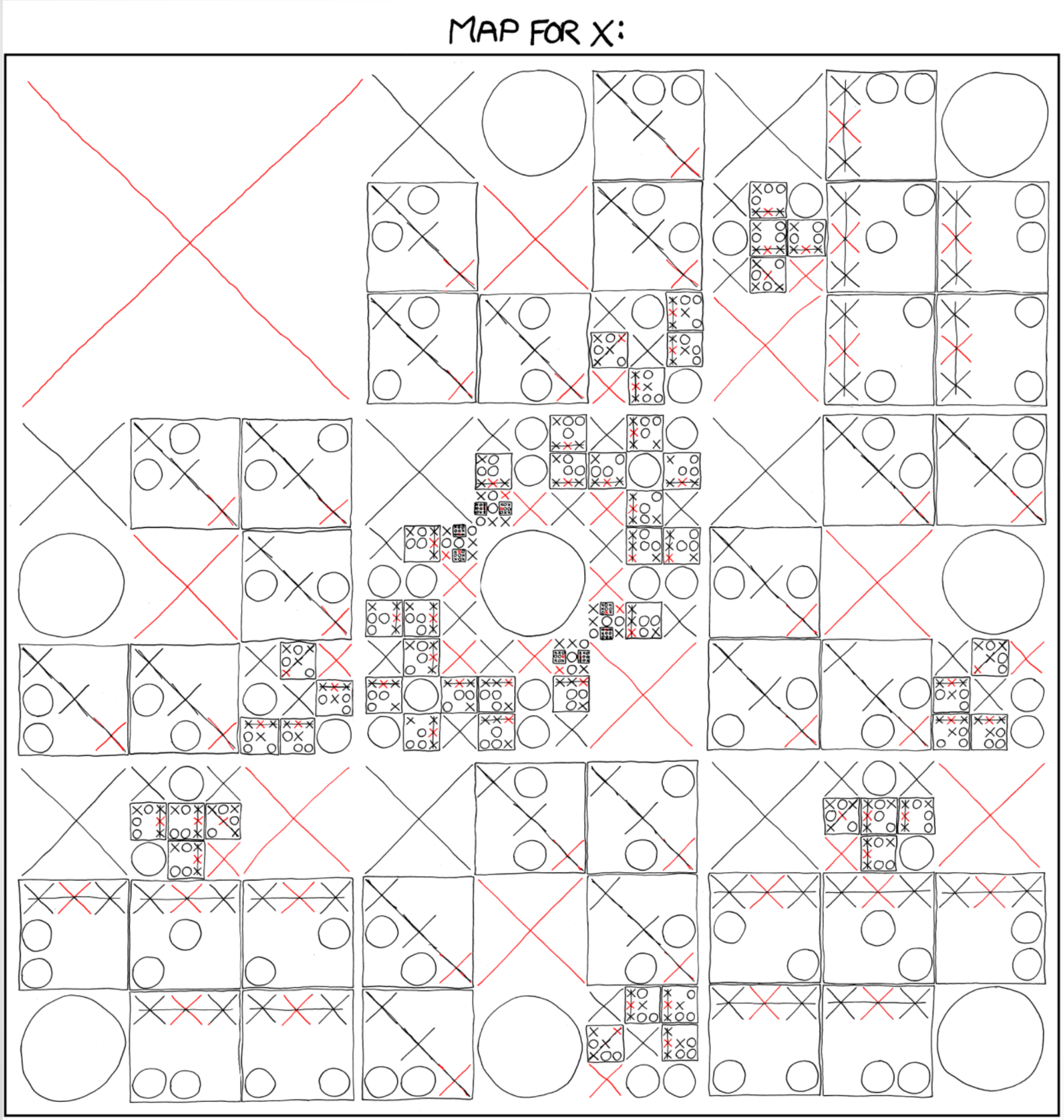
  - For each child of the root,
    count how many games below that child end in a win.

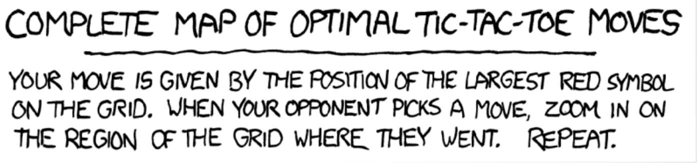
  - Pick the child with the most wins.

For the Enthusiast

- If you are interested in optimal strategies see the next 9 slides.

XKCD's Optimal Tic-Tac-Toe



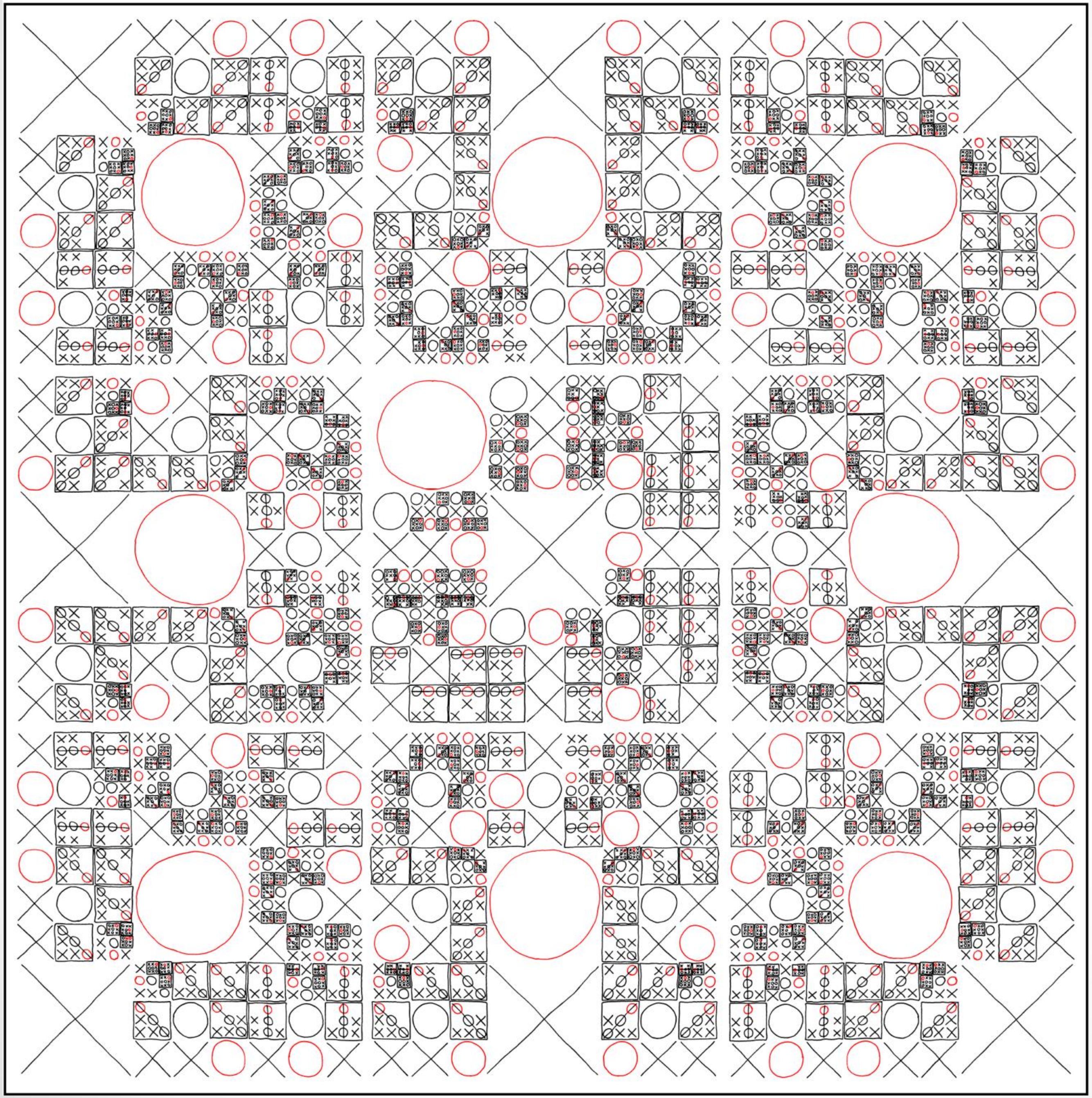


COMPLETE MAP OF OPTIMAL TIC-TAC-TOE MOVES

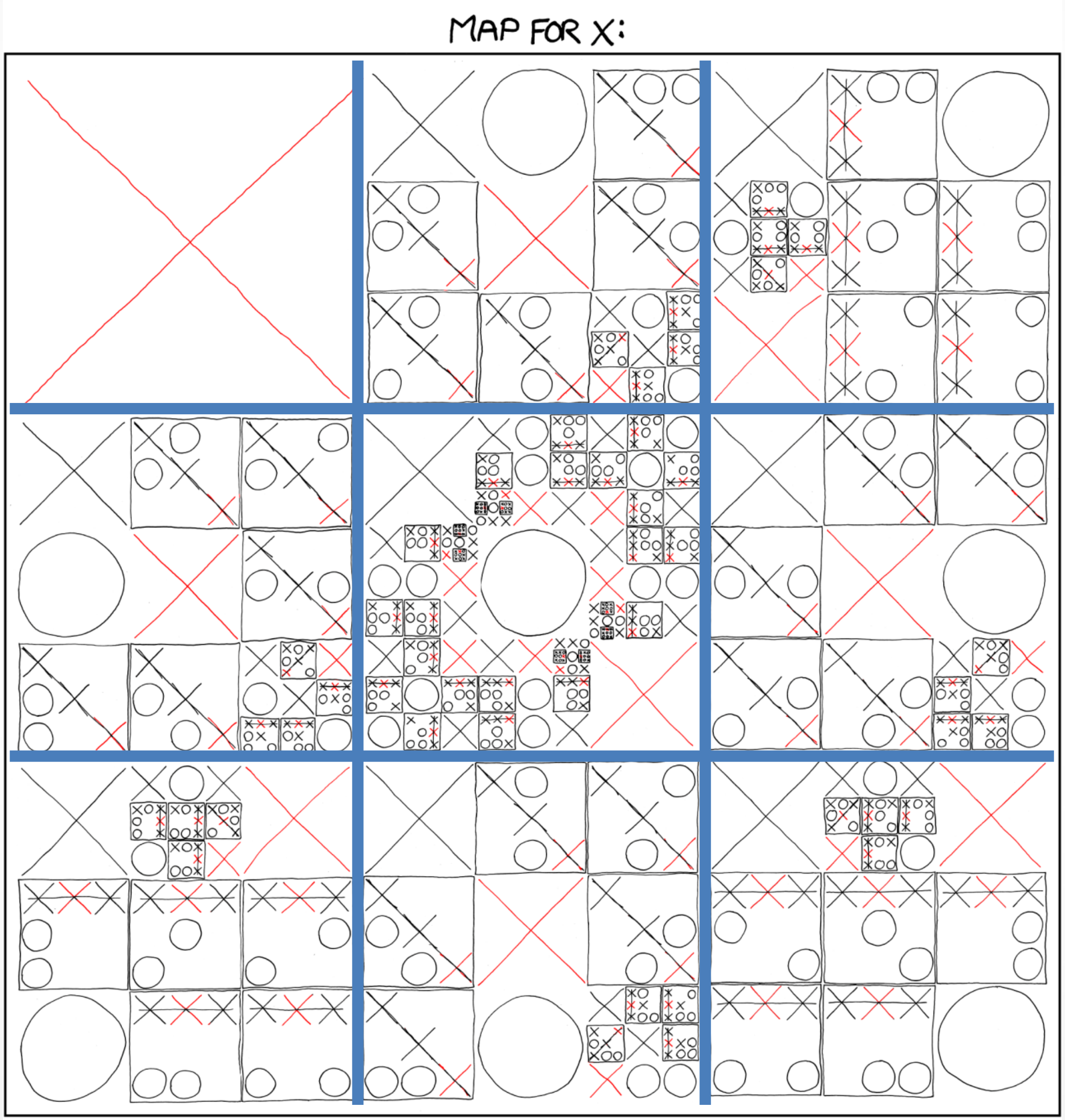
YOUR MOVE IS GIVEN BY THE POSITION OF THE LARGEST RED SYMBOL ON THE GRID. WHEN YOUR OPPONENT PICKS A MOVE, ZOOM IN ON THE REGION OF THE GRID WHERE THEY WENT. REPEAT.

http://xkcd.com/832

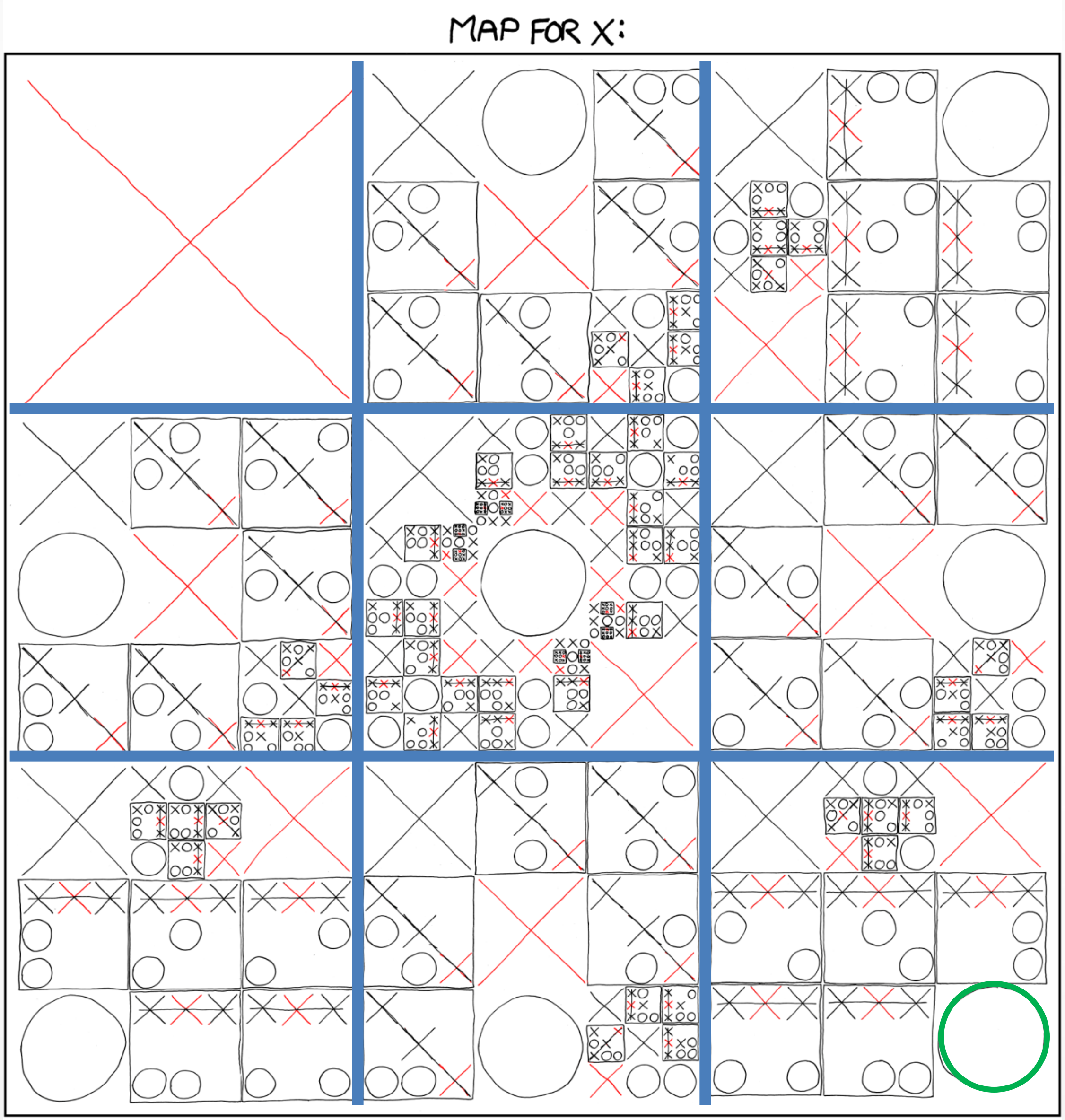
MAP FOR O:



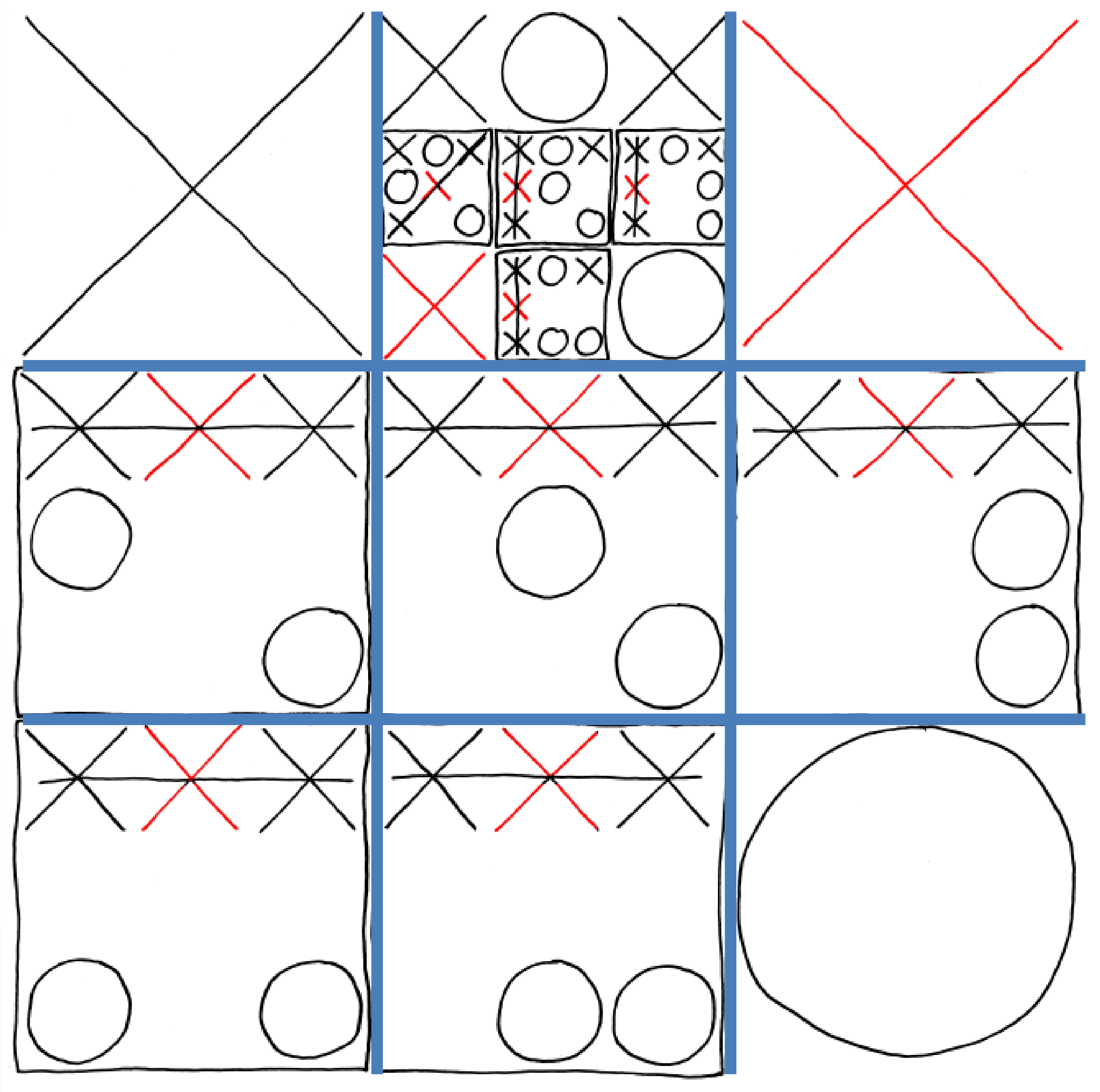
X Move 1



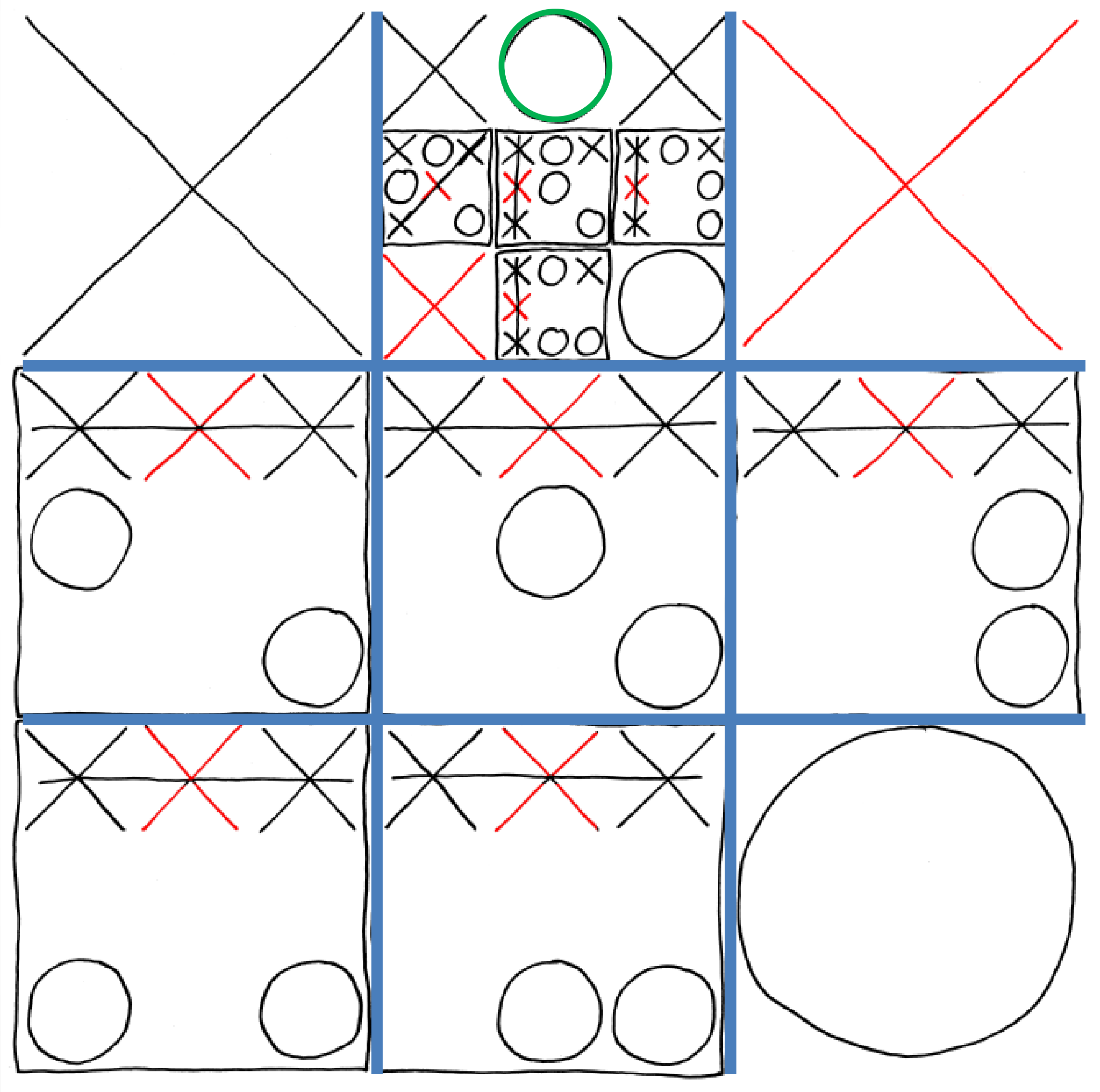
O Move 1 (non-optimal play)



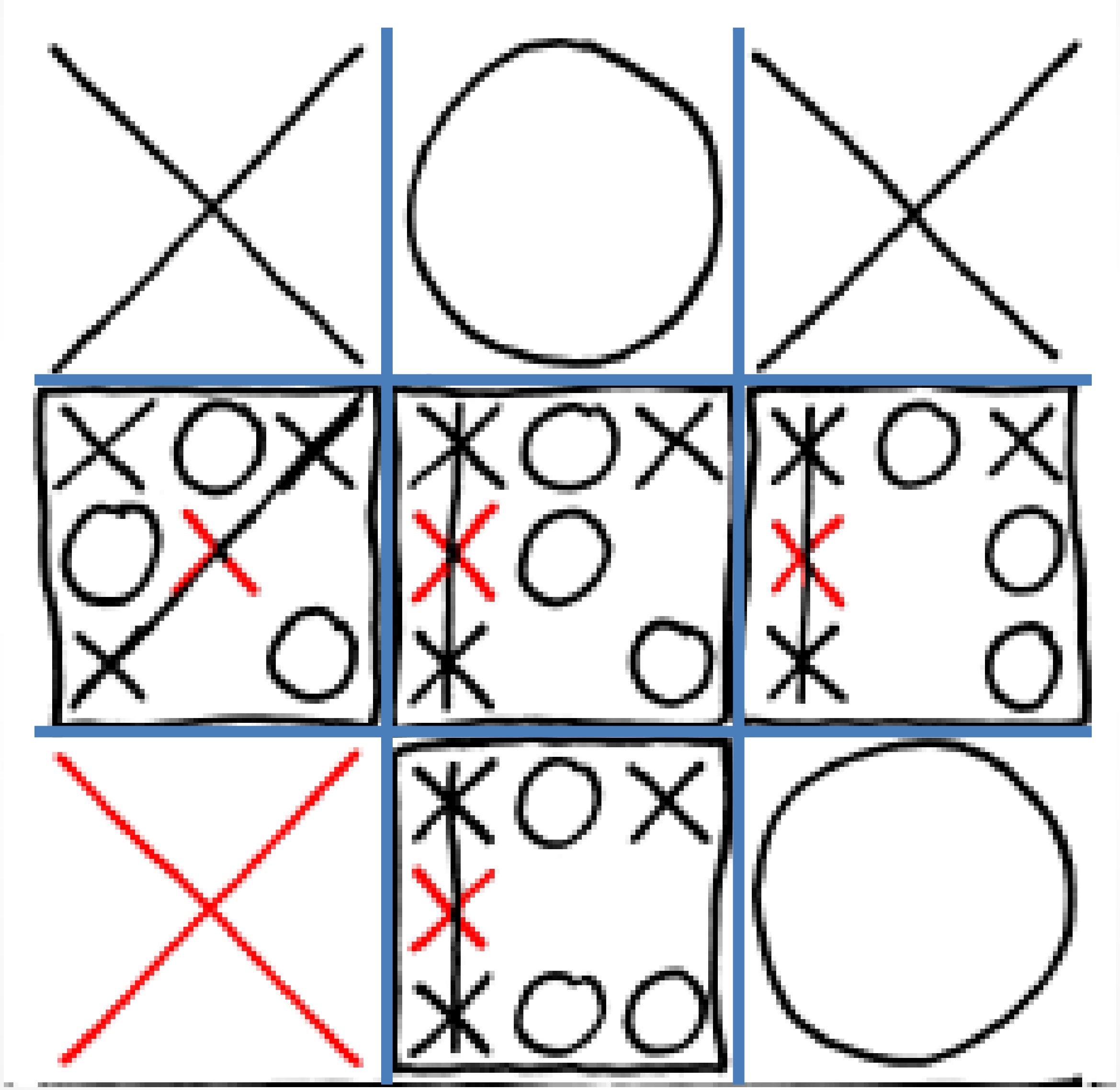
X Move 2



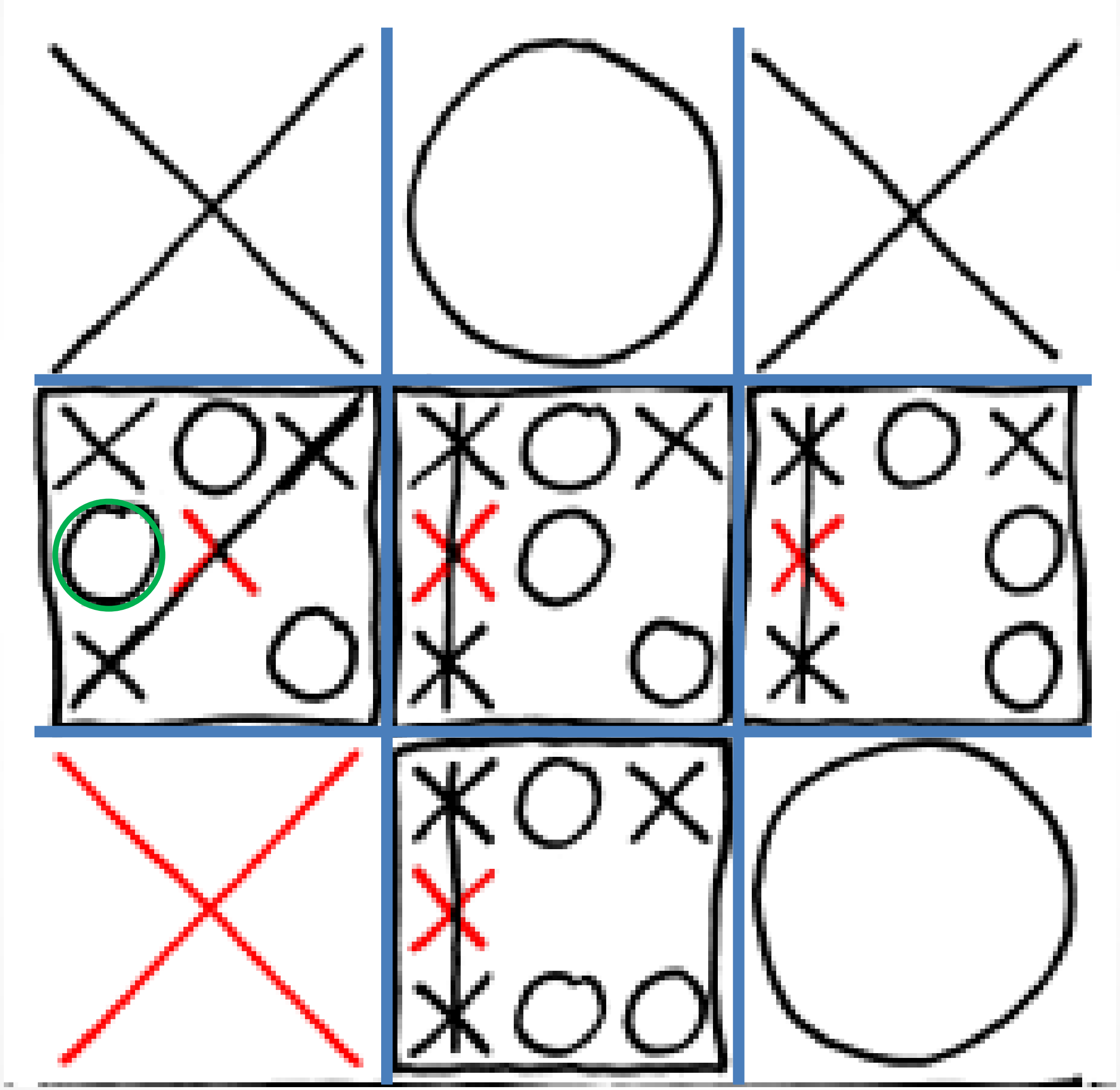
O Move 2



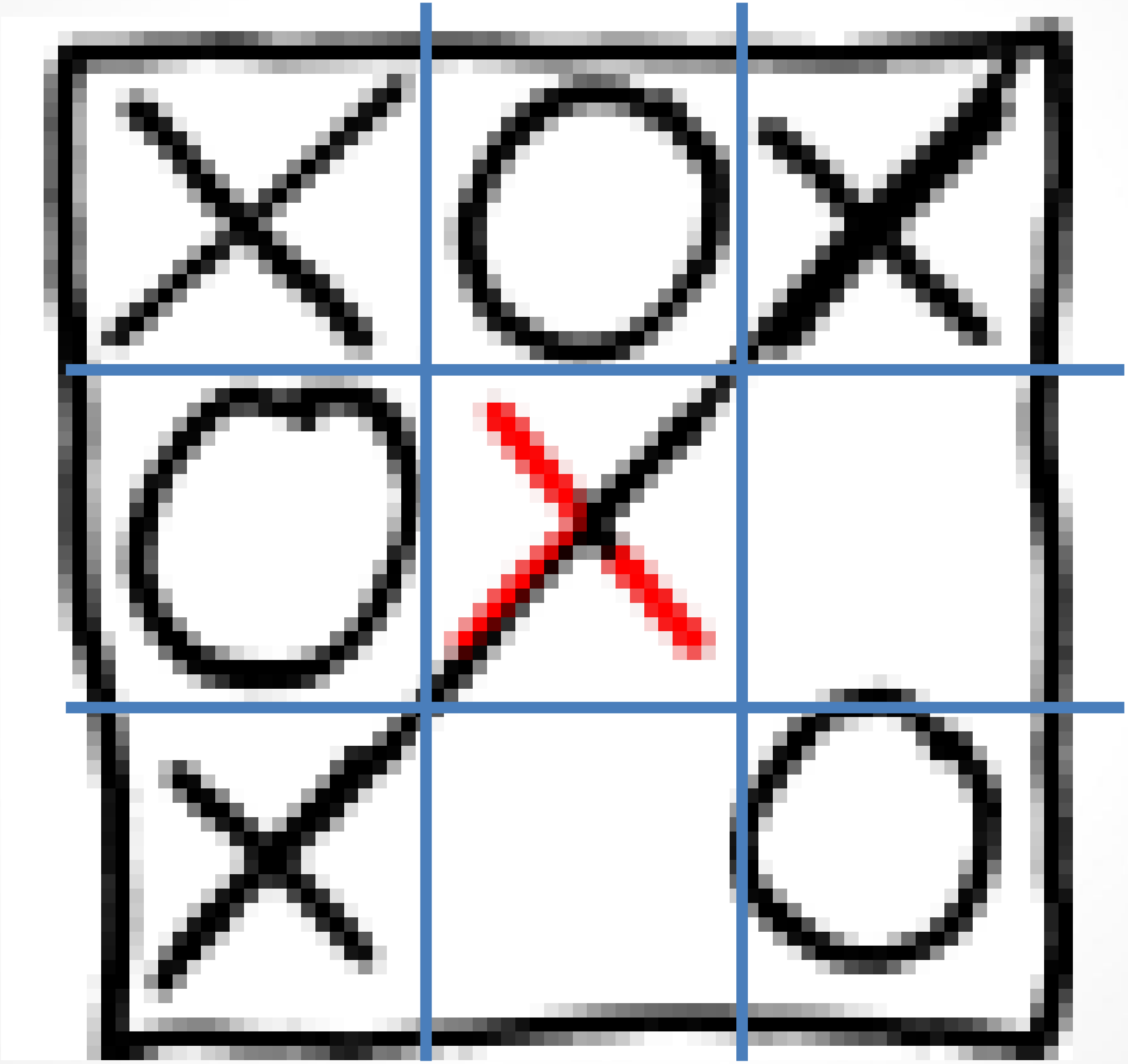
X Move 3



O Move 3



X Move 4 To Win



Search and heuristics

• • •

Coping with combinatorial explosion

Dealing With **Huge** Game Trees

- How does a computer program that plays **chess** or **go** deal with the huge size of the game trees that can be generated?
    - **Chess:** size of game tree: ~35^{100} (about 10^{150}), beyond hope, even for fastest computers!
    - **Go:** much bigger (one estimate is 10^{360})
- These programs use <u>heuristics</u> to reduce the part of the tree that must be examined

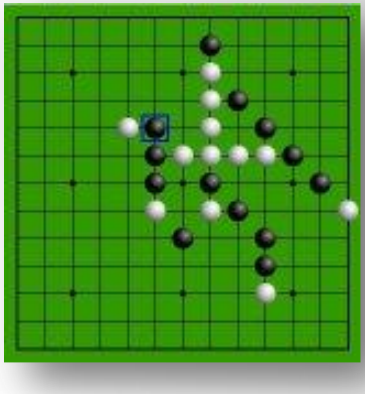
Heuristics

- Human thought is not purely deductive. It often relies on approximations or analogies.

- A <u>heuristic</u> is a “**rule of thumb**” that may not always be correct but usually gives useful results.

- Heuristic algorithms are helpful because they can find a **reasonably good solution** to a problem **without requiring excessive search**.

Heuristics in game play



- Consider the game of Pente.
    - Players alternate, placing their stones on the game board at the intersection of lines, one at a time.
    - The object of the game is to be the first player to either **get five stones in a straight line** or **capture 5 pairs of stones** of the other player.
    - "Capture" means two adjacent stones surrounded by opponent stones. Captured stones are removed.

Pente

Black Wins With 5 In A Row

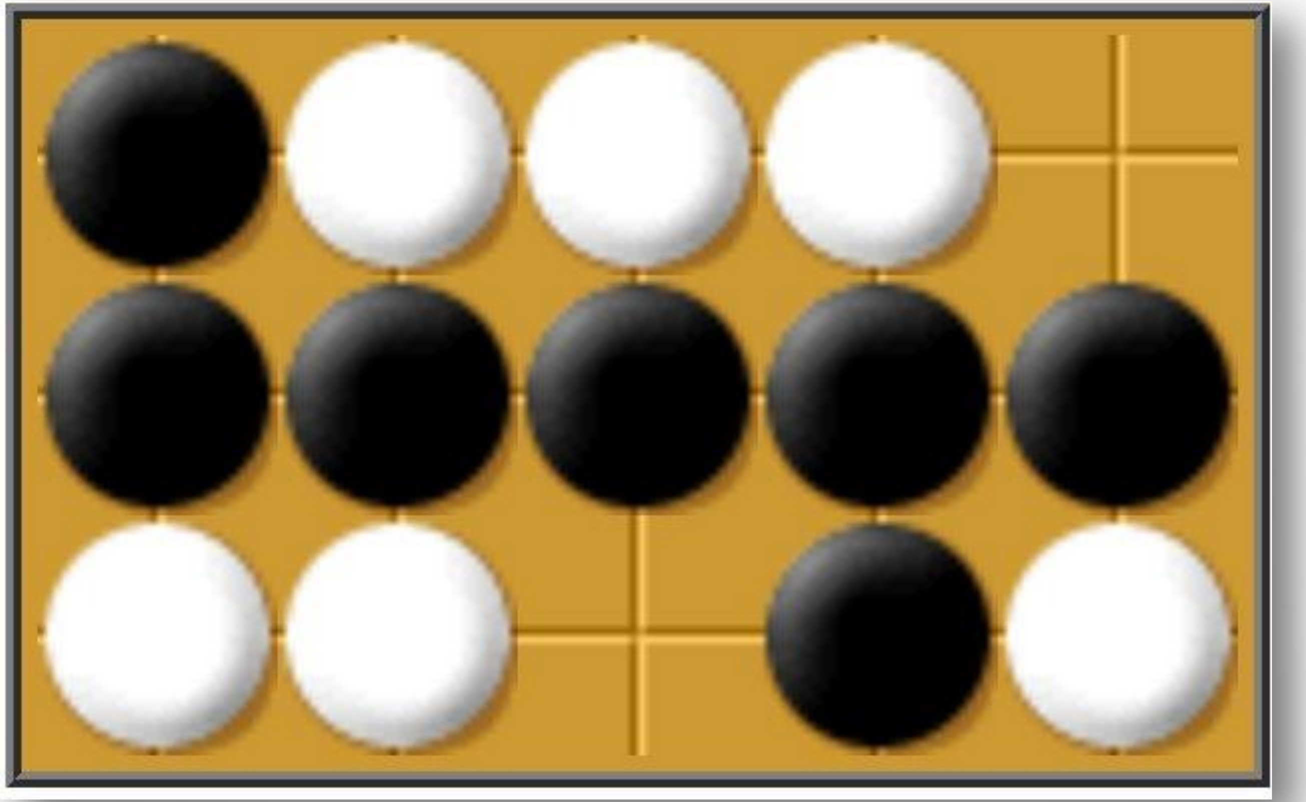
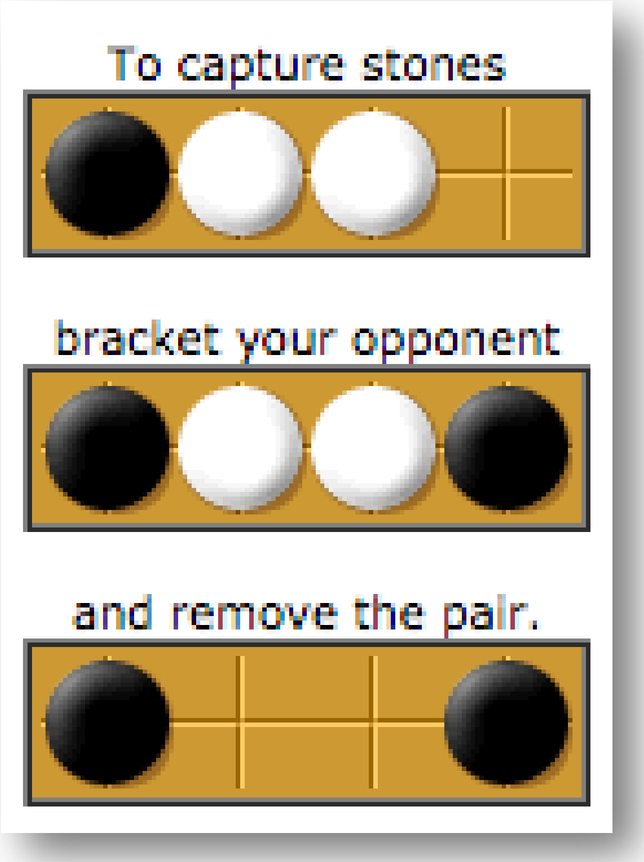


Image http://www.pente.net/

Capture in Pente



To capture stones

bracket your opponent

and remove the pair.

image: http://www.pente.net/

Pente

Black To Move; White Will Win

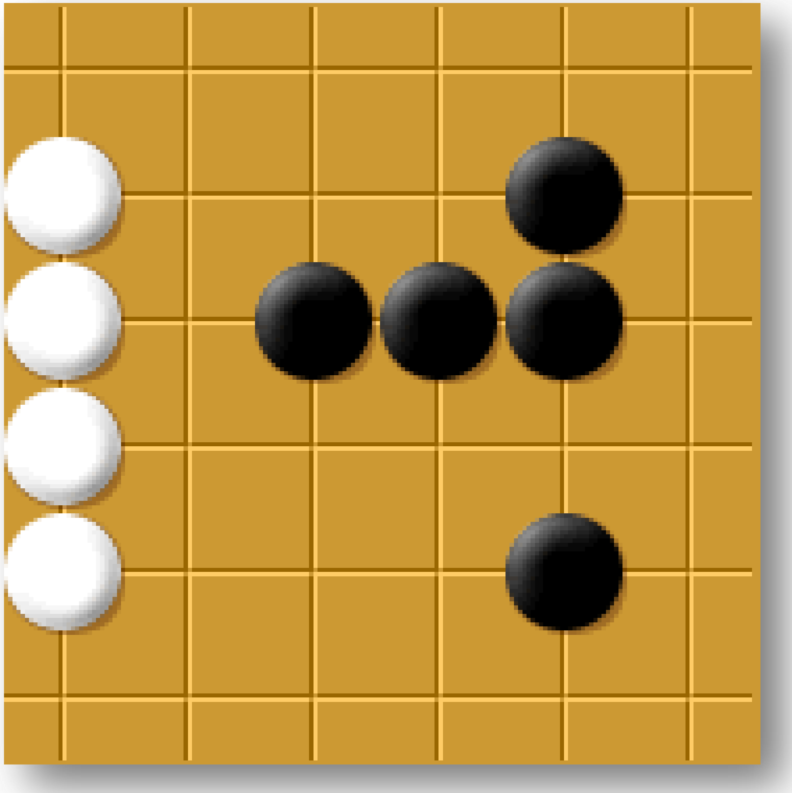


Image: http://www.pente.net/

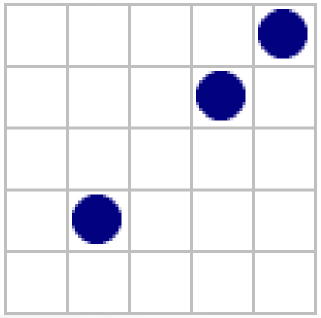
Computer Pente

- At each turn, what are the possible moves?
  - Not the *sensible* moves, all the *legal* ones

- If we were write a computer program to play Pente against a human player, how does the computer calculate its moves when there are a huge amount of possibilities to consider?

Heuristics

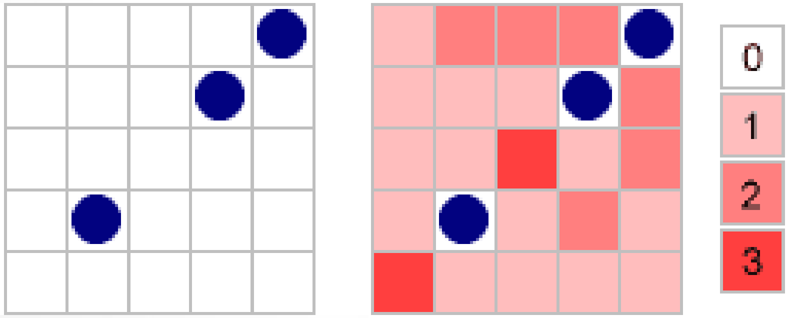
- Important board features to look for in Pente:
  - Check for *x*-pieces in a row.
  - Check for capture possibilities.
  - Check for *x*-pieces in a row for opponent.
  - Check for opponent’s potential to make a capture.
- Where is the opponent likely to put a stone?



http://www.generation5.org/content/2000/boardai.asp

Heuristics

- Important board features to look for in Pente:
  - Check for *x*-pieces in a row.
  - Check for capture possibilities.
  - Check for *x*-pieces in a row for opponent.
  - Check for opponent's potential to make a capture.
- Compute a feature score and choose squares with high value:



***influence mapping example*:**

Code each cell based on number of opponent pieces in same row, column and diagonal.

http://www.generation5.org/content/2000/boardai.asp

Search and problem solving

• • •

not just for games!

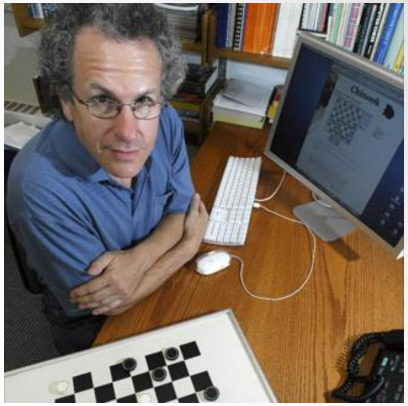
State Space Search

- Searching game trees is a form of *state space search*
- *"Good old-fashioned AI"*: idea that *any* intelligent behavior can be modeled as search in a state space tree
- Successes:
  - Chess, checkers, etc.
  - Route-finding for airline travel planning
  - Layout for integrated circuits in electronics
  - Automated manufacturing processes
  - Symbolic integration
  - Proving theorems

“Deep Blue”

- IBM's "Deep Blue" computer beats Gary Kasparov in a chess match in 1997.
- Heuristics values:
  - The value of each piece. (1 for pawn up to 9 for queen)
  - The amount of control each side has over the board.
  - The safety of the king.
  - The quickness that pieces move into fighting position.
- For more info:
  - http://www.research.ibm.com/deepblue/home/html/b.html
- Is Deep Bl

“Chinook”



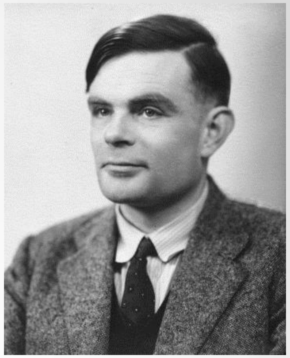
- Created by computer scientists from the University of Alberta to play checkers (draughts) in 1989.
- In 2007, the team led by Jonathan Schaeffer announced that Chinook could never lose a game. Checkers had been solved:

  the best a player playing against Chinook can achieve is a draw.
- Chinook's algorithms featured:
  - a library of opening moves from games played by grandmasters
  - a deep search algorithm
  - a good move evaluation function (based on piece count, kings count, trapped kings, player’s turn, “runaway checkers”, etc.)
  - an end-game database for all positions with eight pieces or fewer. and other minor factors.
- Is Chinook intellige

Philosophical foundations

what is intelligence?

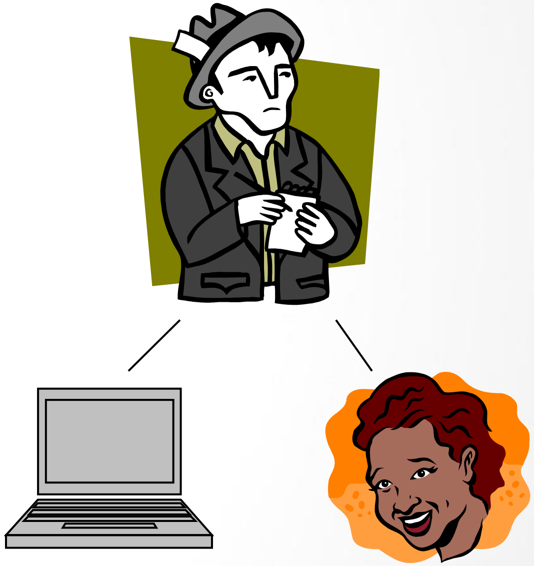
The Imitation Game (Turing Test)



Turing in publishes Computing Machinery and Intelligence in 1950.

A human interrogator communicates with two players via teletype.

- One player is human, the other a machine.
- Each tries to convince the interrogator that they are the human.



If the interrogator can't tell which player is human, the machine must be thinking.

Hypothetical Conversation from Turing’s Paper

- **Interrogator:** In the first line of your sonnet which reads “Shall I compare thee to a summer’s day,” would not “a spring day” do as well or better?
- **Computer:** It wouldn’t scan.
- **Interrogator:** How about “a winter’s day”? That would scan all right.
- **Computer:** Yes, but nobody wants to be compared to a winter’s day.

- **Interrogator:** Would you say Mr. Pickwick reminded you of Christmas?
- **Computer:** In a way.
- **Interrogator:** Yet Christmas is a winter's day, and I do not think Mr. Pickwick would mind the comparison.
- **Computer:** I don't think you're serious. By a winter's day one means a typical winter's day, not a special one like Christmas.

Dickens character

What features of intelligence does this hypothetical conversation exhibit? Is this the same "intelligence" used to play chess?

ELIZA

- One of the first programs to build a conversation between a human and a computer, created by Joseph Weizenbaum in 1966
- Computer responses were created based on *scripts*.
- Most famous example was DOCTOR which simulated a Rogerian psychotherapist.

ELIZA

H: My father wouldn’t buy me a puppy.

C: Tell me more about your family.

H: I am sad I never got a puppy.

C: Can you explain what made you sad you never got a puppy?

H: I am sad we never had a chance to play fetch.

C: I am sorry to hear you are sad we never had a chance to play fetch.

ELIZA

- Mid-60's "trick" program fooled unsuspecting users:

User: Men are all alike.

Eliza: In what way?

User: They're always bugging us about something or other.

Eliza: Can you think of a specific example?

ELIZA exposed

- Part of a conversation with a modern-day version:

  User: You are a machine.

  Eliza: You have your mind on machines, it seems.

  User: Clever you!

  Eliza: Are you sure that clever me?

- But this is just a toy—what if someone is really trying?

Loebner Prize

A chatterbot written by someone really trying to win the prize:

JUDGE: What music do you like?

ZOE: I'm not really into music. What's your favorite song?

JUDGE: What is your favorite song?

ZOE: That's very interesting. Tell me more. I don't think I have a favorite. What is your favorite play?

JUDGE: I like Waiting for Godot. What is your favorite play?

ZOE: I like Waiting for Godot, too! I don't think I have a favorite. What is your favorite play?

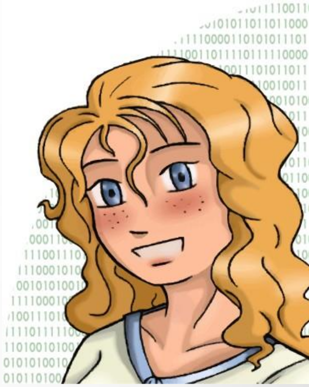
A chat with a recent winner

Human: what will the weather be like in san jose for thanksgiving?

Mitsuku: I hope it's sunny in san jose for thanksgiving today.

Human: do you know what the weather will be like on that day?

Mitsuku: I don't know where you live to tell you what the weather will be.



Where do you live and I'll tell you the weather forecast.

Human: san jose california

Mitsuku: I hope it's sunny in san jose california today.

Chatbot at http://mitsuku.com/

Eliza and the Turing Test

- Human (natural) language is very complex.

“You said I wouldn’t be afraid of horses
if I visited a state fair, but when I was there
I was chased by a pig, and now I’m having
nightmares about bacon.”

Natural language processors (NLPs) must use **parsers** to break a sentence down into its constituent parts (noun, verb, etc.) to determine an appropriate response.

Eliza and the Turing Test

- Human (natural) language is very complex.

  "The lions were lying down in the grass."

  A: Go on.

  "They were quiet"

  A: Why were the lions quiet?

  Natural language processors (NLPs) must have **memory** to associate words (e.g. "they") to words from previous sentences.

Eliza and the Turing Test

- Human (natural) language is very complex.

  "The elephant was huge."
  A: How big was it?

  "The cat fell into the well."
  A: Was it able to get out?

  NLPs must use **semantics** to assign meaning to certain phrases in order to generate appropriate responses.

Eliza and the Turing Test

- Human (natural) language is very complex.

  "The dog ran across the backyard while I waited on my patio."
  A: What happened when the dog reached the patio?

  NLPs must establish **inferences** to add information to sentences that is not given but is assumed.

AI and Common Sense

- “Common sense” knowledge a huge challenge
- We take ours for granted but how can a computer “have” or “simulate” it?
- *Watson* (we’ll see more next time about this)

Summary

- Artificial Intelligence
- Games described as (large) decision trees
- Heuristics to guide search
- Would-be definitions of intelligence
- Turing Test
- Next time, an introduction to statistical methods: Watson and machine learning

Next time

Watson and common sense knowledge

Happy Carnival!