Limits of Computation

Dr. Abdallah Mohamed

Acknowledgement: Original slides provided courtesy of Dr. Lawrence.

Key Points

1) Computers can demonstrate “artificial intelligence” but cannot yet mimic human creativity.

2) Game trees and search strategies are used to create the intelligence in games.

3) Scientists use big-Oh notation to analyze and compare the performance of algorithms.

4) There exists some problems where there is no efficient solution or no solution at all.

Do Computers have Limits?

We have seen that the computer gets it **power** by being able to perform **simple operations very fast**.

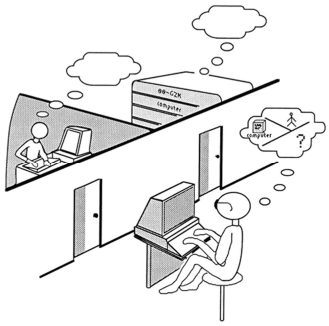
By combining these simple operations into larger software programs, a computer can perform complex tasks.

Two interesting questions:

- 1) Can computers do anything (everything)?
- 2) Can computers behave like humans?

Can Computers Think?

The ***Turing Test***



Source: www.alanturing.net

If the judge cannot identify the computer, the computer can be said to be intelligent.

Can Computers Think?

Alan Turing posed the ***Turing Test*** to evaluate if a computer can mimic a human. The Turing Test:

- Two identical rooms connected electronically to a judge who can type questions directed to the occupant of either room. A human being occupies one room, and the other contains a computer. The judge's goal is to decide, based on the questions asked and the answers received, which room contains the computer. If after a reasonable period of time the judge cannot decide for certain, the computer can be said to be intelligent.
- The computer is intelligent if it acts enough like a human to deceive the judge.

The test does not define thinking, intelligence, awareness or focus on any specific ability.

Can Computers Think? (2)

Computers have come closer to passing the test but not quite.

Computer advances:

- Better understanding and parsing of **natural language (ELIZA)**
- Recognize **semantics** in language and communication
- **Translate** to and from natural language realistically
- Improved computational power

Work to do:

- Computational power will go beyond that of the human brain in 50 years.
- Outstanding challenge is modeling **complexity and intelligence** in software.

Acting Intelligently?

Eliza is a system developed by MIT researcher Joseph Weizenbaum to carry on a conversation as though she were a psychotherapist.

- The software used word clues for answers but did not understand the meaning so it was not intelligent.

Example:

User: I'm depressed.
Doctor: Why are you depressed?
User: My mother is not speaking to me.
Doctor: Tell me about your mother.
User: She doesn't want me to major in engineering.
Doctor: No?
User: No, she wants me to go into medicine.

Survey
Computer Intelligence

Question: I believe a computer will behave like a human …

A) never
B) in my lifetime
C) within 50 years
D) within 20 years
E) within 10 years

Survey
Computers and Humanity

Question: It is a good thing if computers/robots become as intelligent as humans and develop/display emotions.

A) Strongly Agree
B) Agree
C) Neutral
D) Disagree
E) Strongly Disagree

Artificial Intelligence

Artificial intelligence (***AI***) refers to the ability of a computer to **mimic human intelligence** in certain situations.

To exhibit intelligence, the computer has to "**understand**" a complex situation and **reason** well enough to act on its understanding.

One example of AI is computer intelligence in playing games such as **chess** and **checkers**.

Game Intelligence

For a computer to play a game against a human opponent, it must make **intelligent decisions on** its moves.

Strategy games such as checkers and chess have been targeted games for computing "artificial intelligence".

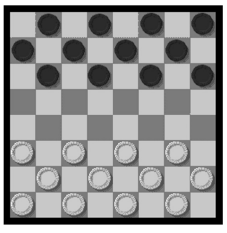
Even video games require the computer to determine **strategies**, even though the decision making is less complex.

- This includes games such as role-playing games and strategy/conquest games.

Checkers

The world champion is a program called **Chinook** created by researchers at the University of Alberta, Canada.

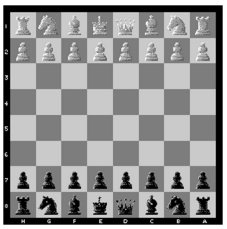
- http://www.cs.ualberta.ca/~chinook
  - ⇨ Get crushed by Chinook if you choose



- They also have a research group called GAMES (Game- playing, Analytical methods, Minimax search, and Empirical Studies)
  - ⇨ http://www.cs.ualberta.ca/~games
- Checkers was solved in 2005. Perfect play by both sides leads to a draw. There are 500 billion billion (5×10^{20}) positions.
- Group is working on poker players as well.

Chess

A computer program does not have world champion status, but defeated world champion Gary Kasparov in a regulation match in 1997.



Deep Blue

- Developed by IBM at a cost of millions of dollars.
  - ⇨ Powered by a RS/6000 massively parallel mainframe.
  - ⇨ Can evaluate 200 million board positions a second.
  - ⇨ http://www.research.ibm.com/deepblue/home/html/b.html

Jeopardy

IBM's Watson AI program competed on Jeopardy! in February 2011 against champions Ken Jennings and Brad Rutter.

- Watson: $77,147 Jennings: $24,000 Rutter: $21,000

Watson is a specialized program with a huge, self-contained database that **parses English, formulates queries to database, and filters and selects correct answer**.

- Database has 200 million unstructured pages.
- Watson consisted of 2,800 computers and terabytes of memory.
- Applied to medicine, banking, and research.

"Final Jeopardy!" question it got wrong in category U.S. Cities: *Its largest airport is named for a World War II hero, its second largest for a World War II battle.*

A Simple Game Tic-Tac-Toe

A good way to look at structures and algorithms that are capable of playing games of pure skill is by examining a simple game like Tic-Tac-Toe (also called x's and o's).

Tic-Tac-Toe

- A game of pure skill
  - ⇨ No element of chance
- Can program Tic-Tac-Toe by "looking" for forced moves, traps, and patterns.
  - ⇨ Careful case by case analysis
    - Can be done because of the relatively **few cases possible**

Game Playing Mini-max Strategy

The majority of game playing systems employ something called a ***mini-max strategy***.

- The basic idea in a mini-max strategy is that you determine a move which ***maximizes*** your potential to win the game and ***minimizes*** your opponent's potential to win the game.

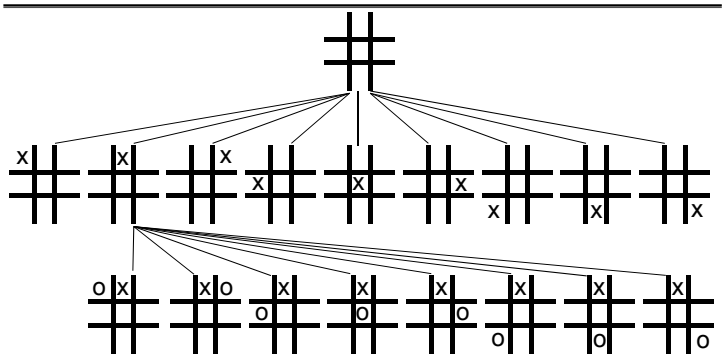
The mini-max strategy consists of three components:

- **Move generator** - determine **your possible moves**
- **Board evaluator** - evaluate the **desirability of each move**
- **Mini-max procedure** - determine an efficient way to search through all the possible moves that you can perform

All of these components use or work upon a ***game tree***.

- A game tree stores, and allows the mini-max procedure to manipulate the possible moves that can be made.

Move Generator Example



Note: Many branches are omitted.
The second level would actually contain 9 x 8 = 72 nodes.

Move Generator Tic-Tac-Toe

Creating a complete game tree starting from the empty state board for Tic-Tac-Toe turns out to be more complex than you might first expect:

- The game tree contains approximately **550,000** nodes.
  - ⇨ Easy for a computer to handle, but not insignificant.

For more complex games, the complete game tree is effectively unmanageable because the number of possible nodes in the game tree is unbelievably large.

- Therefore, we will not want to construct the entire game tree when making a decision, but rather only construct and **search the most "promising" parts of the game tree.**

Heuristics and ***pruning*** are used to only evaluate the most likely beneficial moves.

Board Evaluator

The second component of a game system is the ***board evaluator*** which is responsible for determining if a given board position or state is advantageous for the player.

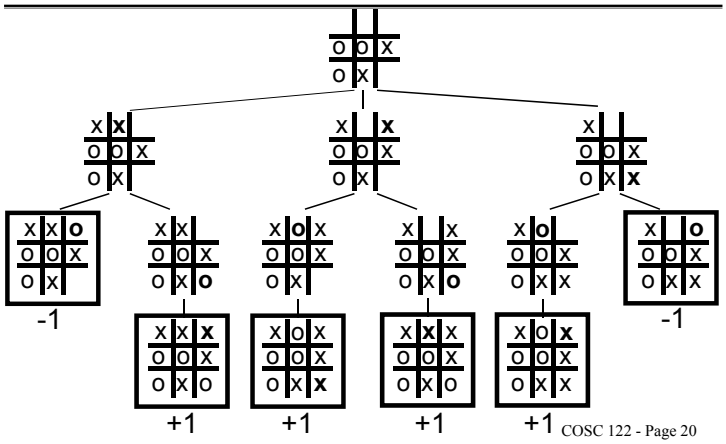
- The board evaluator **determines the good and bad moves**.

The move generator builds a game tree to get some insight as to **what might happen in future moves**:

- Future board scenarios are thus known by playing out moves, counter-moves, counter-counter-moves, etc.
- Future board scenarios are of no use if you have no mechanism to evaluate them.

The board evaluator determines when a sequence of moves (a path in the game tree) is advantageous for the player.

Board Evaluator Tic-Tac-Toe Example



Why did Deep Blue Win?

Deep Blue ended up winning due to increasing computation power.

- This **extra power** allowed the computer to **examine more possible moves** in the game tree.

The use of ***parallel computers*** that have multiple processors and memory allow for complex problems to be solved.

- The top 500 most powerful computers in the world have **thousands of processors** and are used for simulations of weather, military tests, and earthquakes.

Is Deep Blue intelligent?

- The search algorithm was "intelligent", but does it qualify as what we consider intelligence?

Survey Computer Games

Question: I have noticed an improvement in the intelligence/interactivity of the computer or computer characters in the games I play.

A) Yes

B) No

Survey Computer Games and Your Time

Question: I have spent more time this semester playing games than working on this course.

A) Yes

B) No

Survey Social Computer Games

Question: I confess to playing social games Zynga (Farmville, MafiaWars, etc.) or other Facebook or online games:

A) Never – What a waste of time!

B) Never – I love games but those are **NOT** games!

C) Once a month or less

D) Once a week

E) Daily or many times per day.

- Help me! I am addicted. I play all the time (**even during class**)!

Computer Creativity

Computers can run programs that **automatically generate music, art, and pictures.**

- The intelligence is still with the software - not the computer.
- The software is **encoding human intelligence**.

The underlying question is: **Is creativity algorithmic?**

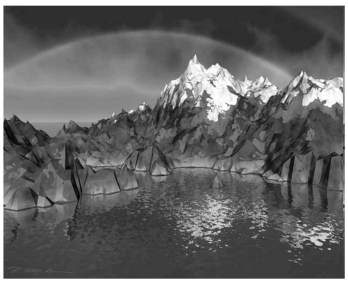
- If it is, computers may one day be creative.
- Many things that are creative are algorithmic.
  - ⇨Mathematics was once considered creative or inspired.

Creativity is sometimes inspiration but is also a lot of revision.

- Inspiration to create something totally new.
- Revision is modifying existing to produce something new. Algorithmic?
  - ⇨Many "new" advertising, research, etc. are based on revisions.

Computer Generated Art

These pictures are generated from algorithms.



Source: Ken Musgrave - http://www.kenmusgrave.com

The Universality Principle

In theory, all computers have **the same ability to compute** as they use the same basic functions.

- This is called the ***Universality Principle***.

In practice, differences in computer hardware, software, and operating systems make it **impossible to run all software on all computers and to run it efficiently**.

Examples:

- programs require processing speed that hardware cannot achieve
- operating systems support different features
- processors encode instructions differently in hardware

Six basic instructions: Add, Subtract, Set_to_One, Load, Store, and Branch_On_Zero.

Why are some programs faster than others?

Recall that an ***algorithm*** is a sequence of steps to solve a problem.

The performance of an algorithm when implemented on a computer depends on the **approach used to solve the problem and the actual steps taken.**

- Example: algorithms for data sorting have different efficiencies.
  - ⇨http://www.sorting-algorithms.com/
  - ⇨http://bl.ocks.org/andrewringler/raw/3809399/

Although faster hardware makes all algorithms faster, algorithms that solve the same problem can be compared in **a hardware-independent way** using ***big-Oh*** notation.

Algorithms
Best and Worst Case

Very few algorithms have the exact same performance every time because **the performance of an algorithm typically depends on the size of the inputs it processes.**

The ***best case*** performance of the algorithm is the most efficient execution of the algorithm on the **"best" data inputs**.

The ***worst case*** performance of the algorithm is the least efficient execution of the algorithm on the **"worst" data inputs**.

The ***average case*** performance of the algorithm is the average efficiency of the algorithm on **the set of all data inputs**.

Best, worst, and average-case analysis typically express **efficiency in terms of the input size of the data**.

- The input size is often a function of ***n***.

Algorithms
Big-Oh Notation

Big-Oh notation is a mechanism for quickly **communicating the efficiency of an algorithm.**

- Big-Oh notation **measures the worst case performance** of the algorithm by ***bounding*** the formula expressing the efficiency.

In big-Oh notation:

- The performance is specified as **a function of *n* which is the size of the problem**.
  - ⇨e.g. *n* may be the size of an array, or the number of values to compute
- Only the most significant expression of ***n*** is chosen:
  - ⇨e.g. If the method performs $n^3 + n^2 + n$ **steps**, it is $O(n^3)$.
  - ⇨**Significance ordering**: 2^n , n^5 , n^4 , n^3 , n^2 , $n*log(n)$, n , $log(n)$
- Constants are ignored for big-Oh:
  - ⇨e.g. If the method performs $5*n^3 + 4*n^2$ steps, it is $O(n^3)$.

Algorithms
Common Big-Oh Notation Values

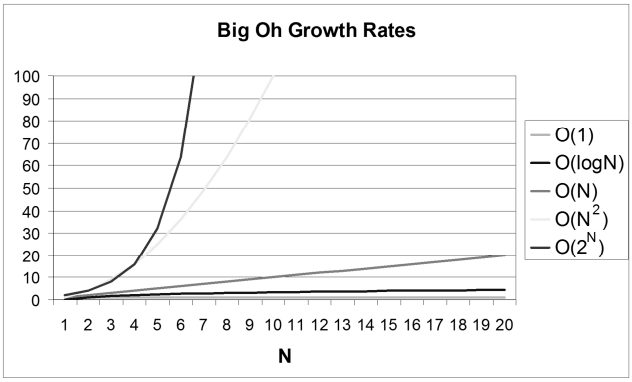
There are certain classes of functions with common names:

- constant = O(1)
- logarithmic = O(*log n*)
- linear = O(*n*)
- quadratic = O(n^2)
- exponential = O(2^n)

These functions are listed in order of fastest to slowest.

- For example, for large values of *n*, an algorithm that is considered O(*n*) is faster than an algorithm that is O(2^n).
- Big-Oh notation is useful for specifying the growth rate of the algorithm execution time.
  - ⇨How much longer does it take the algorithm to run if the input size is doubled?

Big Oh Growth Rates



Big-Oh Exercise

1) What is the Big-Oh for the following formulas?

- a) $4*n^3 + 3*n^2 + 6*n$
- b) $n + n*log(n)$

Big Oh Notation

Question: What is the big Oh for the following formula:

$$4*n^2 + 3*n^4 + 6*n$$

A) O(n^2)
B) O($3n^4$)
C) O(n^4)
D) O(n)

Big-Oh Notation

Question: What is the big Oh notation for the formula:

$$4*n^2 + 3*n^4 + 6*n + 5*n*log(n)$$

A) O($4*n^2$)
B) O(n^4)
C) O($6*n^4$)
D) O($5*n*log(n)$)
E) O($n*log(n)$)

Best/Worst/Average Case

Question: Assuming your mark is given between 0 and 100. How many of the three values (best case, worst case, average case) do you know for sure regardless who is in the class?

A) 0
B) 1
C) 2
D) 3

Best/Worst Case Finding a Song

Question: You have 1000 songs on your music player and want a particular song. You "search" for your song by pressing the random button (pick a random song) until your song comes up. How many times do you have to press the random button until you find your song? **Assume that the randomize feature can return the same song more than once.**

A) best case = 1, worst case = 1000

B) best case = 1, worst case = 500

C) best case = 1, worst case = forever

How Hard Can a Problem Be?

There exists problems that no computer can solve efficiently.

These problems are called ***NP-complete problems*** and are considered ***intractable***.

- The only way to solve the problem is to try all possible solutions to find the best.
- Even the most powerful computers cannot solve large examples of these problems.
- Example problem: Travelling salesman problem - find best route between *n* cities.

How Hard Can a Problem Be? (2)

Even worse, there exist problems that have been proven to be unsolvable regardless of the computer speed.

There exist no algorithms at all for such ***unsolvable problems***.

An example is the Halting Problem that has the simple task of asking if a given program will always stop (halt) or will it run forever.

Computers in the Future

The future of IT is bright. There are many technologies being developed that are migrating from the research labs into use.

- **Software agents** – Can software be your personal butler?
- **Robots** – When we build robots, what would you want it to do?
- **Self-healing and adapting** – Can our systems fix themselves?
- **Wearable computers** – Can we embed computers in clothing and glasses? In our eyes and brains?
- **Language translation** – Can we have the universal translator?
- **Personal Life Databases** – Can we record all of our life information and moments (text, images, sound, video)?
  - ⇨What would that look like? Would you want that?
- **Automatic driving cars** – Our cars will do the driving (probably better than us). They already know where they are going…

Computers in the Future (2)

Some challenges:

- Information overload
  - ⇨If we can get data from everywhere at any time, do we get too much?
  - ⇨Can we trust the data we get?
  - ⇨How about our privacy and security?
- Always-on society
  - ⇨Our technology has trained us to be always available for communication.
  - ⇨Is that good? Are we actually more productive that way? More human?
- Pace of innovation
  - ⇨Technology has sped up society and business. Everything changes rapidly. Innovation may not always be good.
- Essence of Humanity
  - ⇨If everything is automated and computerized around us, do we lose the essence of being human?
  - ⇨Are we ready for the ability to alter human DNA and lifestyles?

Conclusion

Computers do not yet mimic human creativity although they demonstrate "***artificial intelligence***" in many domains.

- One of these domains is game playing where intelligence is provided by game trees and search strategies.

Computer scientists compare algorithms independently of hardware using ***big-Oh notation***.

NP-complete problems are problems where no efficient solution exists. ***Unsolvable problems*** are problems where it is proven no solution at all exists.

Objectives

- Explain the Turing Test in your own words.
- Define: artificial intelligence
- List and briefly explain the three components of game playing using game trees and the mini-max strategy.
- Define: Universality Principle
- Be able to convert a formula in *n* into big-Oh notation.
- Compare and contrast: best case, worst case, average case
- Compare and contrast: NP-complete problem, unsolvable problem