Introduction

• What is AI?
• The foundations of AI
• A brief history of AI
• The state of the art
• Introductory problems
What is AI?
What is AI?

- **Intelligence**: “ability to learn, understand and think” (Oxford dictionary)
What is AI?

<table>
<thead>
<tr>
<th>Thinking humanly</th>
<th>Thinking rationally</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acting humanly</td>
<td>Acting rationally</td>
</tr>
</tbody>
</table>
Acting Humanly: The Turing Test

- Alan Mathison Turing (1912-1954)
Turing Award

• A.M. Turing Award

ACM's most prestigious technical award is accompanied by a prize of $250,000. It is given to an individual selected for contributions of a technical nature made to the computing community. The contributions should be of lasting and major technical importance to the computer field. Financial support of the Turing Award is provided by the Intel Corporation and Google Inc.
Turing Award

2007 Clarke, Edmund M
Emerson, E Allen
Sifakis, Joseph

2006 Allen, Frances E

2005 Naur, Peter

2004 Cerf, Vinton
Kahn, Robert E

2003 Kay, Alan

2002 Adleman, Leonard M.
Rivest, Ronald L.
Shamir, Adi

2001 Dahl, Ole-Johan
Nygaard, Kristen

2000 Yao, Andrew Chi-Chih

1999 Brooks, Frederick P.

1998 Gray, Jim

1997 Engelbart, Douglas

1996 Pnueli, Amir

1995 Blum, Manuel

1994 Feigenbaum, Edward
Reddy, Raj

1993 Hartmanis, Juris
Stearns, Richard E

1992 Lampson, Butler W.

1991 Milner, A J

1990 Corbato, Fernando J.

1989 Kahan, William

1988 Sutherland, Ivan

1987 Cocke, John

1986 Hopcroft, John E
Tarjan, Robert E

1985 Karp, Richard M

1984 Wirth, Niklaus E

1983 Ritchie, Dennis M.
Thompson, Kenneth Lane

1982 Cook, Stephen A

1981 Codd, Edgar F.

1980 Hoare, C. Antony R.

1979 Iverson, Kenneth E.

1978 Floyd, Robert W

1977 Backus, John

1976 Rabin, Michael O.
Scott, Dana S

1975 Newell, Allen
Simon, Herbert A.

1974 Knuth, Donald E.

1973 Bachman, Charles W.

1972 Dijkstra, E. W.

1971 McCarthy, John

1970 Wilkinson, J. H.

1969 Minsky, Marvin

1968 Hamming, Richard

1967 Wilkes, Maurice V.

1966 Perlis, A. J.
Acting Humanly: The Turing Test

• “Computing Machinery and Intelligence” (1950)
Acting Humanly: The Turing Test

Imitation Game

Human Interrogator

Human

AI System
Acting Humanly: The Turing Test

• Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes.

• Anticipated all major arguments against AI in following 50 years.

• Suggested major components of AI: knowledge, reasoning, language, understanding, learning.
Thinking Humanly: Cognitive Modelling

• Not content to have a program correctly solving a problem.
  More concerned with comparing its reasoning steps to traces of human solving the same problem.

• Requires testable theories of the workings of the human mind: cognitive science.
Thinking Rationally: Laws of Thought

• Aristotle was one of the first to attempt to codify “right thinking”, i.e., irrefutable reasoning processes.

• Formal logic provides a precise notation and rules for representing and reasoning with all kinds of things in the world.

• Obstacles:
  – Representation of informal knowledge.
  – Computational complexity and resources.
Acting Rationally

• Acting so as to achieve one’s goals, given one’s beliefs.

• Does not necessarily involve thinking.

• Advantages:
  – More general than the “laws of thought” approach.
  – More amenable to scientific development than human-based approaches.
The Foundations of AI

• Philosophy (423 BC – present):
  – Logic, methods of reasoning.
  – Mind as a physical system.
  – Foundations of learning, language, and rationality.

• Mathematics (c.800 – present):
  – Formal representation and proof.
  – Algorithms, computation, decidability, tractability.
  – Probability.
The Foundations of AI

• Psychology (1879 – present):
  – Adaptation.
  – Phenomena of perception and motor control.
  – Experimental techniques.

• Linguistics (1957 – present):
  – Knowledge representation.
  – Grammar.
A Brief History of AI

• The gestation of AI (1943 – 1956):
  – 1950: Turing’s “Computing Machinery and Intelligence”.

• Early enthusiasm, great expectations (1952 – 1969):
  – Early successful AI programs: Samuel’s checkers,
    Newell & Simon’s Logic Theorist, Gelernter’s Geometry
    Theorem Prover.
  – Robinson’s complete algorithm for logical reasoning.
A Brief History of AI

• A dose of reality (1966 – 1974):
  – AI discovered computational complexity.

• Knowledge-based systems (1969 – 1979):
  – 1969: DENDRAL by Buchanan et al..
  – 1979: PROSPECTOR by Duda et al.
A Brief History of AI

• AI becomes an industry (1980 – 1988):
  – Expert systems industry booms.
  – 1981: Japan’s 10-year Fifth Generation project.

• The return of NNs and novel AI (1986 – present):
  – Mid 80’s: Back-propagation learning algorithm reinvented.
  – Expert systems industry busts.
  – 1988: Novel AI (ALife, GAs, Soft Computing, …).
  – 2003: Human-level AI back on the agenda.
The State of the Art

• Computer beats human in a chess game.
• Computer-human conversation using speech recognition.
• Expert system controls a spacecraft.
• Robot can walk on stairs and hold a cup of water.
• Language translation for webpages.
• Home appliances use fuzzy logic.
• ......
Introductory Problem: Tic-Tac-Toe

X X X
O

Cao Hoang Tru
CSE Faculty - HCMUT
13 February, 2009
Introductory Problem: Tic-Tac-Toe

Program 1:

1. View the vector as a ternary number. Convert it to a decimal number.

2. Use the computed number as an index into Move-Table and access the vector stored there.

3. Set the new board to that vector.
Introductory Problem: Tic-Tac-Toe

Comments:

1. A lot of space to store the Move-Table.

2. A lot of work to specify all the entries in the Move-Table.

3. Difficult to extend.
Introductory Problem: Tic-Tac-Toe

<table>
<thead>
<tr>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Introductory Problem: Tic-Tac-Toe

Program 2:

Turn = 1 Go(1)
Turn = 2 If Board[5] is blank, Go(5), else Go(1)
Turn = 3 If Board[9] is blank, Go(9), else Go(3)
Turn = 4 If Posswin(X) ≠ 0, then Go(Posswin(X))

.......
Introductory Problem: Tic-Tac-Toe

Comments:

1. Not efficient in time, as it has to check several conditions before making each move.

2. Easier to understand the program’s strategy.

3. Hard to generalize.
Introductory Problem: Tic-Tac-Toe

How to check for a possible win?
Introductory Problem: Tic-Tac-Toe

\[
\begin{array}{ccc}
8 & 3 & 4 \\
1 & 5 & 9 \\
6 & 7 & 2 \\
\end{array}
\]

\[15 - (8 + 5)\]
Introductory Problem: Tic-Tac-Toe

Comments:

1. Checking for a possible win is quicker.

2. Human finds the row-scan approach easier, while computer finds the number-counting approach more efficient.
Introductory Problem: Tic-Tac-Toe

Program 3:

1. If a position is a win, give it the highest rating.

2. Otherwise, consider all the moves the opponent could make next. Assume the opponent will make the move that is worst for us. Assign the rating of that move to the current position.

3. The best position is then the one with the highest rating.
Introductory Problem: Tic-Tac-Toe

Comments:

1. Require much more time to consider all possible moves.

2. Could be extended to handle more complicated games.
Introductory Problem: Question Answering

“Mary went shopping for a new coat. She found a red one she really liked. When she got it home, she discovered that it went perfectly with her favourite dress”.

Q1: What did Mary go shopping for?

Q2: What did Mary find that she liked?

Q3: Did Mary buy anything?
Introductory Problem: Question Answering

Program 1:

1. Match predefined templates to questions to generate text patterns.
2. Match text patterns to input texts to get answers.

“What did X Y” “What did Mary go shopping for?”

“Mary go shopping for Z”

Z = a new coat
# Introductory Problem: Question Answering

## Program 2:

Structured representation of sentences:

<table>
<thead>
<tr>
<th>Event2:</th>
<th>Thing1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance: Finding</td>
<td>instance: Coat</td>
</tr>
<tr>
<td>tense: Past</td>
<td>colour: Red</td>
</tr>
<tr>
<td>agent: Mary</td>
<td></td>
</tr>
<tr>
<td>object: Thing 1</td>
<td></td>
</tr>
</tbody>
</table>
Introductory Problem: Question Answering

Program 3:
Background world knowledge:

C finds M

C leaves L

C buys M

C leaves L

C takes M
What is AI?

Not about what human beings can do!

About how to instruct a computer to do what human beings can do!
AI = Algorithms + Intelligence
AI = Algorithms + Intelligence

The Thinker
Auguste Rodin (1840-1917)
Homework

Reading