Can a Machine Think?

Mathematics 15: Lecture 27

Dan Sloughter

Furman University

December 4, 2006

Alan Turing

- 1912 - 1954

Alan Turing

- 1912 - 1954
- Work during World War II

Alan Turing

- 1912 - 1954
- Work during World War II
  - The Germans coded and decoded messages on machines called Enigma machines.

Alan Turing

- 1912 - 1954
- Work during World War II
  - The Germans coded and decoded messages on machines called Enigma machines.
  - English mathematicians working at Bletchley Park deciphered the code using machines specifically designed to decode Enigma messages.

Alan Turing

- 1912 - 1954
- Work during World War II
  - The Germans coded and decoded messages on machines called Enigma machines.
  - English mathematicians working at Bletchley Park deciphered the code using machines specifically designed to decode Enigma messages.
  - Turing's work was instrumental in breaking the Enigma codes, a major factor in changing the course of the war.

Alan Turing

- 1912 - 1954
- Work during World War II
  - The Germans coded and decoded messages on machines called Enigma machines.
  - English mathematicians working at Bletchley Park deciphered the code using machines specifically designed to decode Enigma messages.
  - Turing's work was instrumental in breaking the Enigma codes, a major factor in changing the course of the war.
  - For example, ship convoys from the US to Britain were able to avoid the German submarines in the Atlantic, and the Allies were able to find and destroy many German submarines.

Alan Turing

- 1912 - 1954
- Work during World War II
  - The Germans coded and decoded messages on machines called Enigma machines.
  - English mathematicians working at Bletchley Park deciphered the code using machines specifically designed to decode Enigma messages.
  - Turing's work was instrumental in breaking the Enigma codes, a major factor in changing the course of the war.
  - For example, ship convoys from the US to Britain were able to avoid the German submarines in the Atlantic, and the Allies were able to find and destroy many German submarines.
- Before the war, Turing had worked on a machine to compute the zeros of the zeta function; after the war, he designs and builds one of the first programmable digital computers.

Alan Turing

- 1912 - 1954
- Work during World War II
  - The Germans coded and decoded messages on machines called Enigma machines.
  - English mathematicians working at Bletchley Park deciphered the code using machines specifically designed to decode Enigma messages.
  - Turing's work was instrumental in breaking the Enigma codes, a major factor in changing the course of the war.
  - For example, ship convoys from the US to Britain were able to avoid the German submarines in the Atlantic, and the Allies were able to find and destroy many German submarines.
- Before the war, Turing had worked on a machine to compute the zeros of the zeta function; after the war, he designs and builds one of the first programmable digital computers.
- Turing Prize in computer science is the equivalent of the Fields Medal in mathematics.

Alan Turing

- 1912 - 1954
- Work during World War II
  - The Germans coded and decoded messages on machines called Enigma machines.
  - English mathematicians working at Bletchley Park deciphered the code using machines specifically designed to decode Enigma messages.
  - Turing's work was instrumental in breaking the Enigma codes, a major factor in changing the course of the war.
  - For example, ship convoys from the US to Britain were able to avoid the German submarines in the Atlantic, and the Allies were able to find and destroy many German submarines.
- Before the war, Turing had worked on a machine to compute the zeros of the zeta function; after the war, he designs and builds one of the first programmable digital computers.
- Turing Prize in computer science is the equivalent of the Fields Medal in mathematics.
- Turing committed suicide in 1954.

Machines of the mind

- Turing described an ideal computing machine, which we now call a *Turing machine*.

Machines of the mind

- Turing described an ideal computing machine, which we now call a *Turing machine*.
  - Infinite paper tape divided into squares

Machines of the mind

- Turing described an ideal computing machine, which we now call a *Turing machine*.
  - Infinite paper tape divided into squares
  - Symbols written on the squares

Machines of the mind

- Turing described an ideal computing machine, which we now call a *Turing machine*.
  - Infinite paper tape divided into squares
  - Symbols written on the squares
  - A machine which could read the squares

Machines of the mind

- Turing described an ideal computing machine, which we now call a *Turing machine*.
  - Infinite paper tape divided into squares
  - Symbols written on the squares
  - A machine which could read the squares
  - Depending on the internal state of the machine and the symbol being read, the machine would

Machines of the mind

- Turing described an ideal computing machine, which we now call a *Turing machine*.
  - Infinite paper tape divided into squares
  - Symbols written on the squares
  - A machine which could read the squares
  - Depending on the internal state of the machine and the symbol being read, the machine would
    - ★ move forward or backward one square on the tape, or

Machines of the mind

- Turing described an ideal computing machine, which we now call a *Turing machine*.
  - Infinite paper tape divided into squares
  - Symbols written on the squares
  - A machine which could read the squares
  - Depending on the internal state of the machine and the symbol being read, the machine would
    - move forward or backward one square on the tape, or
    - erase the symbol on the current square and replace it with another.

Machines of the mind

- Turing described an ideal computing machine, which we now call a *Turing machine*.
    - Infinite paper tape divided into squares
    - Symbols written on the squares
    - A machine which could read the squares
    - Depending on the internal state of the machine and the symbol being read, the machine would
        - move forward or backward one square on the tape, or
        - erase the symbol on the current square and replace it with another.
- Turing shows that such a machine could perform any possible computation.

Example: solving equations

- Consider Fermat's equation: $x^n + y^n = z^n$, where n is a positive integer.

Example: solving equations

- Consider Fermat's equation: $x^n + y^n = z^n$, where n is a positive integer.
- Question: Given an n, does this equation have a solution in integers?

Example: solving equations

- Consider Fermat's equation: $x^n + y^n = z^n$, where n is a positive integer.
- Question: Given an n, does this equation have a solution in integers?
- We could create a Turing machine T which would search for solutions when given an input n by working through a list of integers.

Example: solving equations

- Consider Fermat's equation: $x^n + y^n = z^n$, where n is a positive integer.
- Question: Given an n, does this equation have a solution in integers?
- We could create a Turing machine T which would search for solutions when given an input n by working through a list of integers.
- Then $T(n)$ would halt when $n = 1$ (having found a solution, such as $x = 1$, $y = 1$, and $z = 2$), and when $n = 2$ (having found a solution such as $x = 3$, $y = 4$, and $z = 5$).

Example: solving equations

- Consider Fermat's equation: $x^n + y^n = z^n$, where n is a positive integer.
- Question: Given an n, does this equation have a solution in integers?
- We could create a Turing machine T which would search for solutions when given an input n by working through a list of integers.
- Then $T(n)$ would halt when $n = 1$ (having found a solution, such as $x = 1$, $y = 1$, and $z = 2$), and when $n = 2$ (having found a solution such as $x = 3$, $y = 4$, and $z = 5$).
- But T would not halt for any other input.

The halting problem

- Suppose we list all Turing machines: T_1, T_2, T_3, $\ldots$.

The halting problem

- Suppose we list all Turing machines: T_1, T_2, T_3, ...
- Does there exist an algorithm which can tell when $T_n(p)$ will not halt?

The halting problem

- Suppose we list all Turing machines: T_1, T_2, T_3, $\ldots$.
- Does there exist an algorithm which can tell when $T_n(p)$ will not halt?
- That is, suppose $A(n, p)$ is a procedure which halts only if $T_n(p)$ does not halt.

The halting problem

- Suppose we list all Turing machines: $T_1, T_2, T_3, \ldots$.
- Does there exist an algorithm which can tell when $T_n(p)$ will not halt?
- That is, suppose $A(n, p)$ is a procedure which halts only if $T_n(p)$ does not halt.
- Now consider $A(n, n)$: If $A(n, n)$ stops, then $T_n(n)$ does not stop.

The halting problem

- Suppose we list all Turing machines: T_1, T_2, T_3, ...
- Does there exist an algorithm which can tell when $T_n(p)$ will not halt?
- That is, suppose $A(n, p)$ is a procedure which halts only if $T_n(p)$ does not halt.
- Now consider $A(n, n)$: If $A(n, n)$ stops, then $T_n(n)$ does not stop.
- But now $A(n, n)$ is a Turing machine: it's a computational procedure which takes an input n and may or may not stop with some output. So $A(n, n) = T_k(n)$ for some k.

The halting problem

- Suppose we list all Turing machines: T_1, T_2, T_3, ...
- Does there exist an algorithm which can tell when $T_n(p)$ will not halt?
- That is, suppose $A(n, p)$ is a procedure which halts only if $T_n(p)$ does not halt.
- Now consider $A(n, n)$: If $A(n, n)$ stops, then $T_n(n)$ does not stop.
- But now $A(n, n)$ is a Turing machine: it's a computational procedure which takes an input n and may or may not stop with some output. So $A(n, n) = T_k(n)$ for some k.
- We now have: If $A(k, k)$ stops, then $T_k(k)$ does not stop; that is, if $T_k(k)$ stops, then $T_k(k)$ does not stop.

Conclusion

- To avoid a contradiction, we must conclude that $T_k(k)$ does not stop.

Conclusion

- To avoid a contradiction, we must conclude that $T_k(k)$ does not stop.
- This means that we know $T_k(k)$ does not stop, but no computational procedure can tell us that it does.

The Turing test

- Question: Can a machine think?

The Turing test

- Question: Can a machine think?
- To answer, one must first say what we mean by "machine" and what we mean by "think."

The Turing test

- Question: Can a machine think?
- To answer, one must first say what we mean by "machine" and what we mean by "think."
- Machine: programmable digital computer, in part because it can mimic any other type of machine.

The Turing test

- Question: Can a machine think?
- To answer, one must first say what we mean by "machine" and what we mean by "think."
- Machine: programmable digital computer, in part because it can mimic any other type of machine.
- Thinking: we must say a machine can think if an interrogator cannot distinguish the replies from the machine and a person.

Objections

- Theological: only creatures with immortal souls can think.

Objections

- Theological: only creatures with immortal souls can think.
- Heads in the Sand: the consequences of machines being able to think are too dire to consider.

Objections

- Theological: only creatures with immortal souls can think.
- Heads in the Sand: the consequences of machines being able to think are too dire to consider.
- Mathematical: results of Gödel and and Turing show the limitations of formal reasoning abilities.

Objections

- Theological: only creatures with immortal souls can think.
- Heads in the Sand: the consequences of machines being able to think are too dire to consider.
- Mathematical: results of Gödel and and Turing show the limitations of formal reasoning abilities.
- Consciousness: a machine cannot have self-awareness.

Objections

- Theological: only creatures with immortal souls can think.
- Heads in the Sand: the consequences of machines being able to think are too dire to consider.
- Mathematical: results of Gödel and and Turing show the limitations of formal reasoning abilities.
- Consciousness: a machine cannot have self-awareness.
- Disabilities: there are numerous human traits that a machine could never share.

Objections

- Theological: only creatures with immortal souls can think.
- Heads in the Sand: the consequences of machines being able to think are too dire to consider.
- Mathematical: results of Gödel and and Turing show the limitations of formal reasoning abilities.
- Consciousness: a machine cannot have self-awareness.
- Disabilities: there are numerous human traits that a machine could never share.
- Lady Lovelace's objection: a machine cannot originate ideas.

Objections

- Theological: only creatures with immortal souls can think.
- Heads in the Sand: the consequences of machines being able to think are too dire to consider.
- Mathematical: results of Gödel and and Turing show the limitations of formal reasoning abilities.
- Consciousness: a machine cannot have self-awareness.
- Disabilities: there are numerous human traits that a machine could never share.
- Lady Lovelace's objection: a machine cannot originate ideas.
- Continuity of the nervous system: the human nervous system appears to have a continuum of states.

Objections

- Theological: only creatures with immortal souls can think.
- Heads in the Sand: the consequences of machines being able to think are too dire to consider.
- Mathematical: results of Gödel and and Turing show the limitations of formal reasoning abilities.
- Consciousness: a machine cannot have self-awareness.
- Disabilities: there are numerous human traits that a machine could never share.
- Lady Lovelace's objection: a machine cannot originate ideas.
- Continuity of the nervous system: the human nervous system appears to have a continuum of states.
- Informality of behaviour: there is no set of rules which describe what a man should do in every conceivable set of circumstances.

Objections

- Theological: only creatures with immortal souls can think.
- Heads in the Sand: the consequences of machines being able to think are too dire to consider.
- Mathematical: results of Gödel and and Turing show the limitations of formal reasoning abilities.
- Consciousness: a machine cannot have self-awareness.
- Disabilities: there are numerous human traits that a machine could never share.
- Lady Lovelace's objection: a machine cannot originate ideas.
- Continuity of the nervous system: the human nervous system appears to have a continuum of states.
- Informality of behaviour: there is no set of rules which describe what a man should do in every conceivable set of circumstances.
- Extra-sensory perception