PHL342 Week 3 Notes
Prof: James John

Section One: Lecture (Turing Machines)

1. Questions
    a. Could a machine think?
    b. Could we be thinking machines?
2. **Artificial Intelligence and Question (a)**
    a. Those who answer "yes" to the question "Could a machine think?" believe in the possibility of "artificial intelligence" (AI).
    b. "Artificial Intelligence is the science of making machines do things that would require intelligence if done by men."
       — Marvin Minsky, Semantic *Information Processing* (1968)

*<u>*Trying to build a machine that can do the kinda thing we typically think requires intelligence if people do them.</u>*

    c. Example of "things that would **require intelligence** if done by men" (intelligent mental processes)
        i. reasoning
        ii. problem solving
        iii. decision-making
        iv. perception
        v. language understanding and production

Each of these is an example of <u>mental process</u>, i.e., a temporal sequence of <u>mental states</u> in which each state in the sequence is a cause of its successor.

3. The Twentieth-Century Computer Revolution
    a. The birth of computers and theoretical computer science in the last century radically altered the way we think about machines and thinking and, consequently, ushered in the first truly successful work in AI.
    b. These developments were so transformative that an ancient idea—that the mind might be a thinking apparatus of some kind—was reborn in computer terms.
4. **Computers and Question (b)**
    a. The upshot: those who wished to answer "yes" to the question "Could we be thinking machines?" now had, for the first time ever, a detailed and sophisticated way of developing their view: <u>The Computational Theory of Mind.</u>
    b. **The Computational Theory of Mind**: The mind is a computer. Hence, mental states and processes (reasoning, problem solving, etc.) are states and processes of a <u>naturally occurring, biologically implemented computational system</u>. (nOt hUmAn mAdE)
5. Summary
    a. The rise of computers in the twentieth century facilitated the first truly successful AI.
    b. Computer-based AI research in turn inspired the computational theory of mind.
    c. And the computational theory of mind became the foundation of modern cognitive science.

6. What is “Computation”? *What's so special of that?
    a. Turing and the Theory of Computation
        i. **Alan Turing** (1912-1954) was a British mathematician and logician. (He was on the team of Allied mathematicians that cracked the Nazi code in Word War II.)
        ii. Turing’s great theoretical achievement was to furnish a precise formal analysis of the notion of **computability**.
        iii. A computable function is a function for which there is an **algorithm** or **effective method** (a procedure that takes a finite number of purely mechanical steps) for calculating a solution.
    b. Computability, Algorithms, and the Idea of a “Mechanical” Procedure
        i. Logicians and mathematicians had long worked with a loose and informal understanding of computability.
            - Why loose and informal? Because the notion of a *purely mechanical* procedure was never precisely defined. The idea was that a set of instructions is mechanical, in the intended sense, if no creativity or insight is needed to follow them.
        ii. However, radical developments in the foundations of mathematics at the end of the nineteenth and beginning of the twentieth century forced upon logicians and mathematicians the need for a precise formal analysis.
    c. The Entscheidungsproblem
        i. For complex historical reasons, researchers in these fields needed to address a key issue (first raised by Hilbert and Ackermann and later refined by Gödel) known as the *Entscheidungsproblem* (or “decision problem”).
        ii. This is the problem of determining whether, for every statement S of first-order logic, S is universally valid in or derivable within (is a theorem of) first-order logic.
            - need a purely mechanical way to check
    d. Turing and the Entscheidungsproblem
        i. To deal adequately with this problem, a precise formal analysis of the notion of computability—of the notion of a **purely mechanical problem-solving procedure**—was needed. This is exactly what previous logic and mathematics had lacked.
        ii. In his ground-breaking 1936 article “On Computable Numbers, with an Application to the *Entscheidungsproblem*,” Turing supplied the needed formal analysis and proved that <u>there is no such algorithm for solving the decision problem</u>.
    e. **Turing Machines**: Turing’s formal analysis of computability is couched in terms of what he dubbed “<u>*computing machines*</u>”—or what later generations have come to call, in homage to Turing, “<u>*Turing machines.*</u>”
        i. A Turing machine is a very abstract kind of computer designed to explicate in precise terms the notion of a purely mechanical way of solving a problem.
            - **The key point:** With his Turing machine idea, Turing showed how to replace the earlier characterization of “purely mechanical” in terms of vague notions like creativity and

insight with a definition stated in terms of successive instances of rote following of simple rules.

**even the most stupid and non-insightful imaginable thing can get the job done*

ii. **Warning**: A Turing machine is a thought experiment, an abstract logico-mathematical structure designed to make a point about computability.

- Thus, Turing machines are *NOT* physical objects, though of course they can be implemented by/instantiated in/realized in/built out of/... physical objects. (This point will matter later in a huge way.)

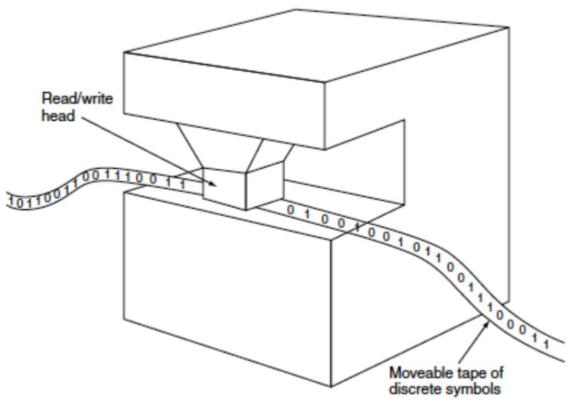
f. Think of a Turing machine as being organized as follows:

i. It has a programmable read/write "head" that can enter into a finite number of "machine states."

ii. It has an infinite length of "tape" that can move back and forth to the left or right under the head.

iii. The tape is divided into discrete squares that are either blank or display a "symbol."

iv. The head can read what is in the square beneath it at a given time. It can then either delete the symbol, write a new symbol, or move the tape one square to the right or left (or halt).



g. Turing Machine Behaviour

i. The behaviour of a Turing Machine is completely captured by a corresponding "machine table" that specifies exactly what the machine will do given its initial state and currently scanned tape square.

An Example (Walmsley 2012, following Penrose 1989)

If in state	and scanning	then print	change to	and move
S1	0	0	S1	R
S1	1	1	S2	R
S2	0	1	S1	HALT
S2	1	1	S2	R

…	0	0	0	0	1	1	1	0	0	0	0	…

…	0	0	0	0	1	1	1	1	0	0	0	…

Think of a machine table as encoding a set of five-place conditionals such as "If in state S1 and currently scanning a "1," then print "0," remain in S1 and move left."

h. A Major Upshot
    i. Turing then converted the informal model we have been visualizing into a rigorous mathematical formalism. In doing so, he showed that <u>any mechanical calculation involving symbols a human can perform can also be performed by a Turing machine with an appropriately configured machine table and tape.</u>
    ii. "**Turing's Thesis**": for any computable problem, there is a Turing machine that computes it. (—Known as the "**Church-Turing Thesis**" when combined with recognition of Alonzo Church's distinct but logically equivalent formal analysis of computability in terms of the lambda-calculus.)
    <u>**For any computable problems, any problems that humans can do in the purely mechanical way, there's a Turing machine that computes it.*</u>

i. Another Major Upshot
    i. Turing also showed that we can define what we today call a "universal Turing machine," <u>a Turing machine U that can replicate the behavior of any other Turing machine M by having M's machine table converted into a string of symbols on tape and fed to U as input.</u>
        - A universal Turing machine is just a mathematical abstract logical basis for the modern digital computer.
    ii. This is a tremendously big deal, as it is basically the concept of a general-purpose computing device—the abstract logico-mathematical basis for the modern **digital computer**.

j. How This Is Relevant to the Mind
    i. Modern computer-based AI seeks to analyze such processes (the activities we mentioned earlier that requires intelligence) in computational terms (i.e., as sequences of algorithmically computable operations) so as to develop "programs" for them that are implemented or "run" by human-made physical machines (like your laptop).
        - The result? <u>**Computing machines that reason/solve problems/etc. the way humans do.**</u>
    ii. The success of this research enterprise (together with additional considerations we will cover in a few weeks) *led to the idea that human mental processes are themselves nothing but computational processes.*

Section Two: Town Halls (Turing's Famous Test for Intelligence, the "Imitation Game")

1. Questions
    a. Turing's "Imitation Game"
    b. Can (digital) computing machines think?
2. What Intelligence Is vs. How We DetectIntelligence
    a. In his great 1950 article "Computing Machinery and Intelligence," Turing proposed a way of using conversational abilities as a method for testing the intelligence of a given computing machine.
        - He focused on digital computers.
3. The "Imitation Game"

a. Set-up:
    i. A human judge or interrogator tries to determine, going on nothing but written responses to questions, which of two parties they are conversing with is a human and which is a computing machine.
    ii. Imagine that the questions and answers are conveyed via text messaging of some sort.
b. Turing's (First) Confident Prediction
    i. "I believe that in about fifty years' time it will be possible to programme computers...to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning" (Turing, 442).
    ii. A second, less confident prediction came later: "Oh, at least 100 years, I should say" (quoted in Copeland 2000)).
c. The Basic Idea
    i. Can a computing machine—a digital computer, appropriately programmed—fool a human judge into thinking they are having a (text message) conversation with a real human being?
    ii. Assume that the judge is allowed **ample time and plenty of trials**.
d. Some Misunderstandings of Turing's Aims
    i. He is not arguing for a behaviorist definition of "thinking."
        - Behaviorism: movements in thoughts. They argue that talks of minds, mind states and processes, intelligence and consciousness are hopelessly vague, ambiguous and thus unscientific.
        - Thus, study the mind should be replaced with talk of empirically observable behavior.
        - Do not ask as a psychologist what thinking is, instead you should ask questions like what kinda stimuli will cause you to respond in which sorts of way.
          *He is some kinda behaviorist, but he also believes there is a perfectly good answer to the question what is intelligence. It's a answer might not have nothing to do intrinsically with any kinds of central connection to behavior. He thinks we are currently not at a position to know what that analysis is. if we want to think scientifically, we should bypass all of that and try to deal with a rather more trackable question, that is, whether one machine can pass the imitation game.
    ii. Neither does he think that substantial amounts of AI research should go towards building machines that succeed at the game.
        - The Imitation game is a thought experiment as well.
        - What Turing is trying to do, is in terms of this game, what do we mean by intelligence, what sorts of things would we take as signs of intelligence.
    iii. Strictly speaking, he doesn't need to claim that his preferred question "can a digital computer succeed at the game?" is equivalent to the original question "can a digital computer think?"
e. So What Is the "Turing Test" Good For?

        i. The test helps us to address questions about <u>machine intelligence in an unbiased way</u> by enabling us to ignore matters that are irrelevant to judgments of intelligence.
        ii. Though the test isn't a definition of intelligence and **isn't a plausible necessary condition for intelligence**, it does, say some defenders, give us this much: *passing the test makes it <u>prima facie</u> reasonable to believe that there is intelligence present <u>on the level of a normal adult human being</u>.*
            - Is it plausible to believe that fooling judages is necessary for intelligence? No. It is unecessary.
            - Different kinda intelligence that cannot relate our way of frames questions.
            - If a machine can be intelligent and yet fails to pass the test because its intelligence is asort from ours that it cannot engage question and answer. Will there be a non-linguistic form of language?
    f. Turing In Defense of Machine Intelligence and His Game
        i. Turing considers a number of objections to his belief that machines (digital computers) can (and perhaps one day will) think and be genuinely intelligent.
            - We will ignore objection 9, the objection from extra-sensory perception.
4. Objections to the "Imitation Game"
    a. **Objection 1: the Theological Objection**
        i. A soul is necessary for thinking; no human-made machine has a soul; therefore no human-made machine can think.
        ii. Turing's Replies:
            - Theological arguments are typically inconclusive.
            - In any case, surely we have souls because our brains have the right structure. So making a machine with that structure ought to bring in a soul as well.
    b. **Objection 2: the "Heads in the Sand" Objection**
        i. "The consequences of machines thinking would be too dreadful. Let us hope and believe that they cannot do so" (Turing, 444).
        ii. Turing's Reply:
            - this isn't really an objection!
            - But why, exactly, would intelligent computers be "too dreadful"? Turing focuses on an <u>existential reason</u>. Can you think of others?
                - We are special because of our intelligence. Will it be taken if we made others to become the same.
                - Will there be AI with sophisticated minds without the tendency to take away what's so special about us.
            - *enable human made machines to replace humans for jobs. (social disruption)
    c. **Objection 3: The Mathematical Objection (Page 444)**
        i. Gödel's **First Incompleteness Theorem**: Some Background

- A <u>formal system</u> is a logical language with rules specifying *which* arrangements of its vocabulary count as expressions of the system, *which* sequences of expressions count as "well-formed formulae," and *which* sequences of wffs (well-formed formulas) count as proofs.
- In typical cases, there are axioms such that proofs from the axioms end in wffs that are theorems of the system.
- The propositional and predicate calculus are examples of formal systems.

ii. More background to Gödel

- <u>Consistency</u>: A formal system is consistent if for some wff it does not imply both it and its negation .
- <u>Completeness</u>: A formal system is complete (with respect to some property P) if all wff with P can be derived in (are theorems of) the system.

1. In classical deductive logic, a **consistent** theory is one that does not entail a contradiction. 2. In mathematical logic and metalogic, a formal system is called **complete** with respect to a particular property if every formula having the property can be derived using that system, i.e. is one of its theorems; otherwise the system is said to be **incomplete**.

  - **Example**: If a formal system sufficient for expressing arithmetic were complete, then every truth of arithmetic that can be stated in the system would be derivable (provable) within it.
  - Math may be somehow reducible to logic.
  - **Example**: G:"I will never order this sentence".G creates a problem and its truth/falsity depends on future action. There is no way to truly utter G. WHat Godel did was construct arithmetical equivalence, self-referential arithmetical statements that are equivalent to statement G. We can see that G is true but because of the self-referentiality of the statement, there is no way to prove from the axioms you can state within the formal system.
    i. Self-referential sentences that cannot be proven from axioms within the system that they are a part of.
    ii. Godel sentences are not self-inferential in the exact way G is in.
  - **Example:** "This sentence is false"

iii. Gödel's First Incompleteness Theorem

- In any consistent formal system S sufficient for expressing arithmetic, there must exist sentences (so-called Gödel sentences) that cannot be proven in S but which are true on the standard interpretation of S.

- Upshot: No formal system with the resources for capturing arithmetic can be both consistent and complete.

Gödel said that *every non-trivial (interesting) formal system is either incomplete or inconsistent*:

1. There will always be questions that cannot be answered, using a certain set of axioms;
2. You cannot prove that a system of axioms is consistent, unless you use a different set of axioms .

iv. John Lucas's notorious argument:

"Gödel's theorem must apply to cybernetical machines, because it is of the essence of being a machine, that it should be a concrete instantiation of a formal system. It follows that given any machine which is consistent and capable of doing simple arithmetic, there is a formula which it is incapable of producing as being true—i.e., the formula is unprovable-in-the-system—but which we can see to be true. It follows that no machine can be a complete or adequate model of the mind, that <u>minds are essentially different from machines</u> (1961, 113).

v. An Interpretation of Lucas's Reasoning
   - Given Gödel's results, any machine instantiating a formal system that is consistent and capable of doing arithmetic will be such that there are statements that the machine can neither prove nor disprove but that human mathematicians can see to be true.
   - If human mathematicians can see statements to be true that machines instantiating formal systems can neither prove nor disprove, then human mathematical reasoning—hence, human intelligence—is not the instantiation of a formal system.
   - Thus, human intelligence is not the instantiation of a formal system— and "minds are essentially different from machines."

vi. But !!!
   - Step 2 is false: we might be able to see the truth of the relevant statements even if our intelligence does involve the instantiation of a formal system.
   - Perhaps we are inconsistent formal systems. As Turing himself argued in a different article, <u>there is no guarantee that we are consistent in our reasoning</u>, though we may often seem to ourselves to be. Inconsistency could arise through reliance on heuristics in our thought.

[This Lecture is finished. The unfinished one will be covered next week]

Discussion:

1. How the Turing test is supposed to avoid bias when it comes to evaluating intelligence.

- He thinks that computers do not look like people will lend the judges to make biased decisions.
- The very choice of a conversation frame might bias our judgement. views about physical forms may bias for instance.

*<u>*If you are worried that the Turing Test isn't even a decent way to line down a plausible reason to think something is intelligent, how can you do better?</u>*

2. Block provides a proof to show that passing the Turing Test cannot be sufficient for intelligence. You can devise a system that is a massive lookup table. One that basically has any possible conversation you can ever have with humans and code it in it. And a very quick algorithm for scanning the lookup table and locating appropriate responses to any given human input. This kinda machine is conceivable and thus passable. BUT it is not intelligent.

3. The Turing Test depends on the judges. Block thinks that even though Turing was trying to make the question more specific, he still cannot be specified enough to consider the judge's influences on the result. Unless we get cleared in the judge, the test isn't any better than the ambiguous question Turing began with. IF SO, why not just focus on that and forget about the imitation game and Turing Test.

---

vii. An Important Qualification: the Need for Heuristics

- Example 1: Newell and Simon's Logic Theory Machine: it found proofs of theorems by showing how they can be derived from axioms via permitted inference rules.
    - This program did NOT work by mere brute force, by way of an algorithm that does nothing but search through all possible proof solutions.
    - The Reason Why Not:

        "Suppose that the rules of inference are such that at any instant, a string of symbols could be changed in ten different ways. This means that if we want the machine to check all possible 6-step 'proof chains,' it would have to go through one million (i.e., 106) different possibilities. We can see that such a 'combinatorial explosion' [...] means that exhaustive search through the space of possible symbolic transformations would have been prohibitively time-consuming" (Walmsley 2012, 52-53).
- Example 2: Deep Blue and AI work on chess-playing computers.
    - Again, this program did NOT work by mere brute force, by way of an algorithm that does nothing but search through all possible moves at any given moment of play.
    - The Reason Why Not:

> Though “perfect” or “exhaustive” chess is a theoretical possibility, “the numbers involved in chess make this approach practically unfeasible. There are somewhere around 10120 40-move chess games (Shannon, 1950). But current estimates put the total number of atoms in the known universe at around 1080, so even if you used one atom per tree branch, there wouldn’t be enough matter in the universe to represent all of the possible games. Further, current calculations put the age of the universe at less than 1025 nanoseconds, so even if you had been evaluating one tree branch every nanosecond since the big bang, you wouldn’t yet have had time to calculate your first move” (ibid, 56-57).

- Heuristics
  - A heuristic is a “rule of thumb” for solving a problem that, while not guaranteeing a solution, nevertheless makes one fairly likely. Newell et al. 1957: a heuristic is “a process that may solve a given problem, but offers no guarantee of doing so” (ibid, 114).
  - The Logic Theory Machine’s programmed heuristics were based on observation of how actual logicians and mathematicians construct proofs.
  - Deep Blue’s programmed heuristics were worked out in consultation with several leading chess grandmasters.
- The Reason Why Heuristics Matter in the Present Context: If human intelligence is computational and yet involves reliance on heuristics, then the algorithms it employs may on occasion leave us with contradictory—hence, inconsistent—beliefs.

d. Objection 6: Lady Lovelace’s Objection
   i. Lovelace, in her memoir, on Babbage’s “Analytical Engine”: “The Analytical Engine has no pretensions to originate anything. It can do whatever we know how to order it to perform” (from Turing, 450).
   ii. What, exactly, is the objection here? (What does it have to do with whether computers can do new things or with whether they can surprise us?)
   iii. Parting Questions
      (1) According to the "heads in the sand" objection, "[t]he consequences of thinking machines would be too dreadful." Why think such a thing? How many different reasons can you come up with? How should someone untroubled by AI respond?
      (2) What, exactly, is the "argument from consciousness" against thinking machines? How plausible is it?
      (3) Of all of the objections Turing considers, "Lady Lovelace's objection" has received the most attention. What do you think

the objection is supposed to be? How plausible is Turing's reply?

Section Three: Tutorial

1. In general, a **necessary** condition is one which must be present in order for another condition to occur, while a **sufficient** condition is one which produces the said condition.
2. Questions discuss: Is the Turing Test a necessary and sufficient condition for human intelligence?
3. Notice: the judge itself is not subjective. results depend on the performance of the judge.