Computational power

- In today’s class, we will see that no attempt to extend the computational power of Turing machines yields a model of computation more powerful than the standard one-tape, one-head, deterministic Turing machine.

- By computational power, we mean what can be computed -- not how fast it can be computed. Your desktop may run faster than a Turing machine, but it can’t compute anything that a Turing machine can’t also compute.

Multiple tapes

- Consider a TM with k tapes and a separate head for each tape. It reads and writes on these tapes in parallel.

- We can show that this does not increase the computational power of a TM by showing that any multi-tape TM can be simulated by a standard, single-tape TM.

- The details of the simulation involve dividing a single tape into multiple tracks -- using an alphabet consisting of tuples, with one element for each track.

Multiple heads

- In this case, there is a single tape but k heads that can read/write at different places on the tape at the same time.

- We show that this does not increase the computational power of TMs by showing that a multiple-head TM can be simulated by a standard single-head TM.

- The simulation details are similar to those for a multi-tape TM.

Two-dimensional tapes

- A 2-dimensional tape is a grid that extends infinitely downward as well as to the right.

- The head can move in 4 directions: right, left, up, and down.

- This TM can also be simulated by a TM with a single, one-dimensional tape.

Random-access Turing machine

- Instead of accessing data on the tape sequentially, imagine a TM that has random-access memory and can go to any cell of the tape in one step. To allow this, the TM has registers that can store memory addresses.

- We can simulate this by a multi-tape TM in which one tape is used as memory and the extra tapes are used as registers.
Nondeterministic Turing machine

• A nondeterministic TM (NTM) has more than one transition with the same left-hand part, which means more than one transition can be taken in the same configuration.
• Nondeterminism allows a TM to have different outputs for the same input. This does not make sense when computing a function, but makes sense for language-recognition in the same way as before. A string is accepted if some computation leads to the halting state.

Nondeterminism and computational power

• Nondeterminism does not increase the computational power of a TM.
• We can show this by showing that any NTM can be simulated by a DTM using a technique that the book calls “dovetailing.”

Nondeterminism and efficiency

• Although nondeterminism does not increase the computational power of a TM, it lets it compute some things more efficiently by guessing the right thing to do.
• Although a DTM can always simulate a NTM, the DTM may be much more inefficient because it has to try all possibilities to find the right one.
• Surprisingly, the question whether a DTM can simulate an NTM efficiently is still unresolved. It is the famous question of whether P = NP.

Variations of TM that limit its power

• Restricting the amount of tape that a TM can use limits its computational power.
• Linear-bounded automaton (LBA) = the size of tape is no larger than the size of the input string. An LBA can accept the languages \{a^n b^n c^n | n ≥ 0\} and \{ww | w ∈ \{a,b\}^*\}, but not other languages accepted by TM. It is more powerful than a PDA but less powerful than a TM.
• Finite tape = no more powerful than finite automaton

Hierarchy of automata

Church-Turing Thesis

No model of digital computation is more powerful than a Turing machine.

By “more powerful,” we mean “can recognize languages that a TM cannot recognize.”

This is not something that can be proved. But everybody believes it because no one has been able to devise a more powerful model of computation.