COMS W3261 — Lecture 11, Part 1:

Reductions + Time Complexity

Announcements: HW #6, due Monday, 8/9 @ 11:59 PM EST (Late through 8/13)

Final Exam on Tues/Weds. Available from 12:01 AM EST on 8/10 — 11:59 PM on 8/11.

— Use any 12 hours.

Review Session: 1-4pm on Monday 8/9 (in-person CS Lounge)
5-8pm (virtual)

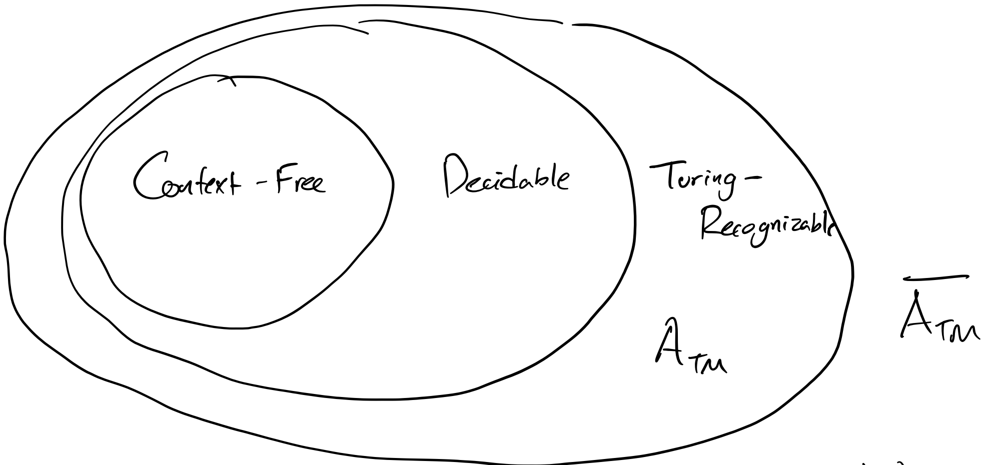
Readings: 5.1 (Undecidability + reductions)
7.1, 7.2, 7.3 Big-O, Big-Ω, Time Complexity, P and NP.

Today:
1. Review
2. Reductions + more examples of undecidable languages
3. Big-O notation, time complexity
4. P, NP
5. Wrap-Up

1. Review

Decidable Languages.

- $A_{DFA} = \{\langle A, w\rangle \mid A \text{ is a DFA, } A \text{ accepts } w\}$
- A_{NFA}, A_{REX}, A_{CFG} all decidable.
- $E_{DFA} = \{\langle A\rangle \mid A \text{ is a DFA, } L(A) = \emptyset\}$
- $EQ_{DFA} = \{\langle A, B\rangle \mid A, B \text{ DFAs, } L(A) = L(B)\}$.



- A set is <u>countable</u> if it admits a 1-to-1 mapping to $\mathbb{N} = 1, 2, 3, \ldots$
  - The set of TMs is countable
  - The set of <u>infinite</u> binary strings is uncountable
    - $\Rightarrow$ the set of Languages over Σ is uncountable
    - $\Rightarrow$ some languages must not be Turing-recognizable. ✪

- $A_{TM} = \{\langle A, w\rangle \mid A \text{ is a TM, } A \text{ accepts } w\}$.
- A_{TM} is recognizable, but not decidable. (If A_{TM} <u>were</u> decidable, we could create a "paradox machine." X)
- $\overline{A_{TM}}$ is unrecognizable.
  (If A_{TM} and $\overline{A_{TM}}$ both recognizable $\rightarrow$ A_{TM} would be decidable. X)

---

2. Reductions and More Undecidable Languages.

Idea: In CS, we build up solutions to hard problems using subroutines.

Prove: "I can do A if I can do B"

Prove: "I can do B."

$\therefore$ "I can do A"

"A <u>reduces</u> to B"

Prove: "A is hard" (undecidable, unrecognizable.)
Prove: "If I could solve B, I could solve A."

$\therefore$ "B must be hard."

<u>Example</u>. The Halting Problem.

<u>Theorem</u>. $HALT_{TM} = \{\langle M, w\rangle \mid M$ is a TM and M halts on $w\}$ is undecidable.

<u>Proof</u>. (By <u>reduction</u> from A_{TM}, by contradiction.) We show that if $HALT_{TM}$ were decidable, A_{TM} would be decidable.

Assume some TM R decides $HALT_{TM}$. Given R, we can build a decider for A_{TM} as follows:

M_1 = "On input $\langle M, w\rangle$, an encoding of a TM M and a string w,

1. Run R on $\langle M, w\rangle$. If R rejects, we reject.
2. Simulate M on w, accept/reject if M accepts/rejects."

guaranteed to halt!

This contradicts the undecidability of A_{TM}, so $HALT_{TM}$ must be undecidable. ∎

— Is $HALT_{TM}$ recognizable? (Yes – simulate)

— Is $\overline{HALT_{TM}}$ recognizable? (No – if so, this would imply decidability.)

Moral: No infinite-loop detectors. ☹

Example 2. $E_{TM} = \{ \langle M \rangle \mid M \text{ is a TM}, L(M) = \emptyset \}$ is undecidable.

Proof – by reduction. "If E_{TM} decidable $\longrightarrow$ A_{TM} decidable." X.

Suppose E_{TM} is decided by some TM S. The following TM would decide A_{TM}:

T = "On input $\langle M, w \rangle$, where M is a TM:

- Use $\langle M \rangle$ to build a new TM M' that rejects all strings $x \neq w$, and on w simulates $M(w)$ and accepts/rejects if $M(w)$ accepts/rejects.
- Simulate S on $\langle M' \rangle$. If $\langle M' \rangle \in E_{TM}$, reject. If $M' \notin E_{TM}$, accept."

Why does this work?

Say $M(w)$ accepts. Then $L(M') = \{w\}$, $\langle M' \rangle \notin E_{TM}$.

Say $M(w)$ not accept. Then $L(M') = \emptyset$, $\langle M' \rangle \in E_{TM}$. ∎

Example. $EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are TMs and } L(M_1) = L(M_2) \}$ is undecidable.

Proof. We'll show that if EQ_{TM} were decidable, E_{TM} would be decidable. Assume EQ_{TM} is decided by some TM R.

S = "On input $\langle M \rangle$, where M is a TM:

1. Run R on $\langle M, M_1 \rangle$, where M_1 is some TM that rejects all inputs.

$L(M) = L(M_1)$ iff. $L(M) = \emptyset$.

2. Accept/reject if R accepts/rejects.

$\therefore$ EQ_{TM} cannot be decided. ∎

Rice's Theorem: Let P be some *language* of TM descriptions (e.g., input $\langle M \rangle$), such that:

(1) P contains some but not all TM descriptions (nontrivial)

(2) P captures some property of the language recognized by the input TM. (If $L(M_1) = L(M_2)$, $\langle M_1 \rangle \in P \leftrightarrow \langle M_2 \rangle \in P$.)

Then P is undecidable.

Takeaway: All nontrivial properties of TMs are undecidable!

Next up: Time Complexity.