Last time:
- Equivalence of variants of the Turing machine model
  a. Multi-tape TMs
  b. Nondeterministic TMs
  c. Enumerators
- Church-Turing Thesis
- Notation for encodings and TMs

Today:
- Decision procedures for automata and grammars

Will have mini chat-breaks (experiment)
A TM has 3 possible outcomes for each input $w$:

1. **Accept** $w$ (enter $q_{acc}$)
2. **Reject** $w$ by halting (enter $q_{rej}$)
3. **Reject** $w$ by looping (running forever)

$A$ is **$T$-recognizable** if $A = L(M)$ for some TM $M$.
$A$ is **$T$-decidable** if $A = L(M)$ for some TM decider $M$.

$\langle O_1, O_2, ..., O_k \rangle$ encodes objects $O_1, O_2, ..., O_k$ as a single string.

Notation for writing a TM $M$ is

$M = \text{“On input } w$,

[English description of the algorithm]”
Acceptance Problem for DFAs

Let $A_{DFA} = \{(B, w) \mid B \text{ is a DFA and } B \text{ accepts } w\}$

Theorem: $A_{DFA}$ is decidable

Proof: Give TM $D_{A-DFA}$ that decides $A_{DFA}$.

$D_{A-DFA} =$ “On input $s$

1. Check that $s$ has the form $(B, w)$ where $B$ is a DFA and $w$ is a string; reject if not.

2. Simulate the computation of $B$ on $w$.

3. If $B$ ends in an accept state then accept. If not then reject.”

Shorthand: On input $(B, w)$

$D_{A-DFA} = \{Q = \{q_0, \ldots, q_k\}, \Sigma = \{0, 1\}, \delta = \cdots, q_0, F = \cdots\}$, $w = 01101$

input tape contains $(B, w)$

$q_i, k$

work tape with current state and input head location
Acceptance Problem for NFAs

Let $A_{NFA} = \{ (B, w) \mid B \text{ is a NFA and } B \text{ accepts } w \}$

Theorem: $A_{NFA}$ is decidable
Proof: Give TM $D_{A-NFA}$ that decides $A_{NFA}$.

$D_{A-NFA} =$ “On input $\langle B, w \rangle$

1. Convert NFA $B$ to equivalent DFA $B'$.
2. Run TM $D_{A-DFA}$ on input $\langle B', w \rangle$. [Recall that $D_{A-DFA}$ decides $A_{DFA}$]
3. Accept if $D_{A-DFA}$ accepts.
   Reject if not.”

New element: Use conversion construction and previously constructed TM as a subroutine.
Let $E_{DFA} = \{ \langle B \rangle \mid B$ is a DFA and $L(B) = \emptyset \}$

Theorem: $E_{DFA}$ is decidable
Proof: Give TM $D_{E-DFA}$ that decides $E_{DFA}$.

$D_{E-DFA} =$ “On input $\langle B \rangle$  [IDEA: Check for a path from start to accept.]
1. Mark start state.
2. Repeat until no new state is marked:
   Mark every state that has an incoming arrow from a previously marked state.
3. Accept if no accept state is marked.
   Reject if some accept state is marked.”
Let $\mathcal{E}_\text{ DFA} = \{ \langle A, B \rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B) \}$

**Theorem:** $\mathcal{E}_\text{ DFA}$ is decidable

**Proof:** Give TM $D_{\text{EQ-DFA}}$ that decides $\mathcal{E}_\text{ DFA}$.

$D_{\text{EQ-DFA}} =$ "On input $\langle A, B \rangle$ [IDEA: Make DFA $C$ that accepts $w$ where $A$ and $B$ disagree.]

1. Construct DFA $C$ where $L(C) = L(A) \cap L(B) \cup L(A) \cap L(B)$.

2. Run $D_{\text{EQ-DFA}}$ on $\langle C \rangle$.

3. Accept if $D_{\text{EQ-DFA}}$ accepts. Reject if $D_{\text{EQ-DFA}}$ rejects."

---

**Check-in 7.1**

Let $\mathcal{E}_\text{ REX} = \{ \langle R_1, R_2 \rangle \mid R_1 \text{ and } R_2 \text{ are regular expressions and } L(R_1) = L(R_2) \}$

Can we now conclude that $\mathcal{E}_\text{ REX}$ is decidable?

a) Yes, it follows immediately from things we’ve already shown.

b) Yes, but it would take significant additional work.

c) No, intersection is not a regular operation.
Teach at Splash!

esp.mit.edu/splash20

Splash is an annual teaching and learning extravaganza, brought to you by MIT ESP!

**When?** November 14-15

**Where?** Virtual

**What?** Teach anything! Any topic, length, or class size!

**Who?** Teach thousands of curious and motivated high schoolers
Acceptance Problem for CFGs

Let $A_{\text{CFG}} = \{(G, w) | G \text{ is a CFG and } w \in L(G)\}$

**Theorem:** $A_{\text{CFG}}$ is decidable

**Proof:** Give TM $D_{A-\text{CFG}}$ that decides $A_{\text{CFG}}$.

$D_{A-\text{CFG}} =$ “On input $(G, w)$
1. Convert $G$ into CNF.
2. Try all derivations of length $2|w| - 1$.
3. Accept if any generate $w$.
   Reject if not.

Recall Chomsky Normal Form (CNF) only allows rules:
A $\to BC$
B $\to b$

**Lemma 1:** Can convert every CFG into CNF.
Proof and construction in book.

**Lemma 2:** If $H$ is in CNF and $w \in L(H)$ then every derivation of $w$ has $2|w| - 1$ steps.
Proof: exercise.

**Check-in 7.2**
Can we conclude that $A_{\text{PDA}}$ is decidable?

a) Yes.
b) No, PDAs may be nondeterministic.
c) No, PDAs may not halt.
Emptiness Problem for CFGs

Let $E_{\text{CFG}} = \{ \langle G \rangle | G \text{ is a CFG and } L(G) = \emptyset \}$

Theorem: $E_{\text{CFG}}$ is decidable

Proof:

$D_{E-\text{CFG}} = \text{“On input } \langle G \rangle \text{ [IDEA: work backwards from terminals]}

1. Mark all occurrences of terminals in $G$.
2. Repeat until no new variables are marked
   - Mark all occurrences of variable $A$ if
   - $A \rightarrow B_1B_2 \cdots B_k$ is a rule and all $B_i$ were already marked.
3. Reject if the start variable is marked.
   Accept if not.”

$S \rightarrow RTa$
$R \rightarrow Tb$
$T \rightarrow a$
Equivalence Problem for CFGs

Let $EQ_{\text{CFG}} = \{(G, H) | G, H \text{ are CFGs and } L(G) = L(H) \}$

Theorem: $EQ_{\text{CFG}}$ is NOT decidable
Proof: Next week.

Let $AMBIG_{\text{CFG}} = \{(G) | G \text{ is an ambiguous CFG} \}$

Check-in 7.3
Why can’t we use the same technique we used to show $EQ_{\text{DFA}}$ is decidable to show that $EQ_{\text{CFG}}$ is decidable?

a) Because CFGs are generators and DFAs are recognizers.
b) Because CFLs are closed under union.
c) Because CFLs are not closed under complementation and intersection.
Acceptance Problem for TMs

Let \( A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \} \)

Theorem: \( A_{TM} \) is not decidable
Proof: Thursday.

Theorem: \( A_{TM} \) is \( T \)-recognizable
Proof: The following TM \( U \) recognizes \( A_{TM} \)
\[
U = \text{“On input } \langle M, w \rangle \text{ } \\
1. \text{ Simulate } M \text{ on input } w. \\
2. \text{ Accept if } M \text{ halts and accepts.} \\
3. \text{ Reject if } M \text{ halts and rejects.} \\
4. \text{ Reject if } M \text{ never halts.”} \]

\( U \) inspired the concept of a stored program computer.

Von Neumann said \( U \) inspired the concept of a stored program computer.
Quick review of today

1. We showed the decidability of various problems about automata and grammars:
   \( A_{\text{DFA}}, A_{\text{NFA}}, E_{\text{DFA}}, E_{\text{QDFA}}, A_{\text{CFG}}, E_{\text{DFA}} \)

2. We showed that \( A_{\text{TM}} \) is \( T \)-recognizable.