Lecture 4
Announcements

Late policy
eTurnin
Office hours M 2:30, W 12:30, Th 5:00
Midterm Fri 5/7, probably
**Nondeterministic Turing Machines**

\[ \delta : Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{L,R\}) \]

Accept if *any* path leads to \(q_{\text{accept}}\); reject otherwise, (i.e., *all* halting paths lead to \(q_{\text{reject}}\))
Simulating an NTM

Key issue: avoid getting lost on $\infty$ path

Key Idea: *breadth*-first search

$$\text{tree arity } \leq |Q| \times |\Gamma| \times |\{L,R\}| \quad (3 \text{ in example})$$
A TM "Enumerable"
L Turing recognizable iff a TM enumerates it

\[ \iff \]: Run enumerator, compare each “output” to input; accept if they match (reject by not halting if input never appears)

\[ \implies \]: The “obvious” idea: enumerate \( \Sigma^* \), run the recognizer on each, output those that are accepted.

[Oops, doesn’t work...]
L Turing recognizable iff a TM enumerates it

(⇒): A better idea—“dovetailing”:

For i = 0, 1, 2, 3, ... :

At stage i, run the recognizer for i steps on each of the first i strings in \( \Sigma^* \), output any that are accepted.
Encoding things

\[ G = (V, \Sigma, R, S); \quad <G> = (S, A, B, \ldots, (a, b, \ldots), (S \rightarrow aA, S \rightarrow b, A \rightarrow cAb, \ldots), S) \]

or

\[ <G> = ((A_0, A_1, \ldots, (a_0, a_1, \ldots), (A_0 \rightarrow a_0 A_1, A_0 \rightarrow a_1, A_1 \rightarrow a_2 A_1 a_1, \ldots), A_0) \]

\[ \Sigma = ? \]

\[ DFA \quad D = (Q, \Sigma, \delta, q_0, F); \quad <D> = (...) \]

\[ TM \quad M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r); \quad <M> = (...) \]

\[ ... \]
Decidability

Recall: L *decidable* means there is a TM recognizing L *that always halts*.

Example:

“The acceptance problems for DFAs”

\[ A_{DFA} = \{ <D,w> \mid D \text{ is a DFA} \& w \in L(D) \} \]
Some Decidable Languages

The following are decidable:

\( A_{\text{DFA}} = \{ <D,w> \mid D \text{ is a DFA} \& w \in L(D) \} \)

pf: simulate \( D \) on \( w \)

\( A_{\text{NFA}} = \{ <N,w> \mid N \text{ is an NFA} \& w \in L(N) \} \)

pf: convert \( N \) to a DFA, then use previous as a subroutine

\( A_{\text{REX}} = \{ <R,w> \mid R \text{ is a regular expr} \& w \in L(R) \} \)

pf: convert \( R \) to an NFA, then use previous as a subroutine
\( \text{EMPTY}_{\text{DFA}} = \{ <D> \mid \text{D is a DFA and } L(D) = \emptyset \} \)

pf: is there no path from start state to any final state?

\( \text{EQ}_{\text{DFA}} = \{ <A,B> \mid \text{A & B are DFAs s.t. } L(A) = L(B) \} \)

pf: equal iff \( L(A) \oplus L(B) = \emptyset \), and \( x \oplus y = (x \cap y^c) \cup (x^c \cap y) \), and regular sets are closed under \( \cup \), \( \cap \), complement

\( \text{A}_{\text{CFG}} = \{ <G,w> \mid \ldots \} \)

pf: see book

\( \text{EMPTY}_{\text{CFG}} = \{ <G> \mid \ldots \} \)

pf: see book
$EQ_{CFG} = \{ <A,B> | A \& B \text{ are CFGs s.t. } L(A) = L(B) \}$

This is NOT decidable