$E_{\text{TM}}$ is undecidable

$E_{\text{TM}} = \{ \langle M_1, M_2 \rangle : 
\text{ } M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2) \}$

\[
E_{\text{TM}}(M) \\
\{ \\
\text{Def } H_{\bar{a}} = \text{TM that always rejects (immediately)} \\
\text{if } E_{\text{TM}}(M, H_{\bar{a}}) \\
\text{return true } \text{"accept"} \\
\text{else reject}
\}
CompSci 162
Spring 2023 Lecture 19:
Computational Histories
Reductions via Computational Histories

- Computational History of a Turing Machine
  - Accepting Computational History
  - Rejecting Computational History
- These are finite sequences.
Linear Bounded Automata

- Like a Turing Machine
- R/W head restricted to input region

Which deciders that we saw can be LBAs?

\[ A_{DFA} \quad A_{CFG} \]
\[ E_{DFA} \quad E_{CFG} \]
# Configurations for LBAs

- Let $M$ be an LBA
  - $q$ states
  - $g$ symbols in tape alphabet
  - Tape length $n$

- $M$ has $qng^n$ distinct configurations
$A_{LBA}$ is decidable (Turing decidable)

$A_{LBA} = \{ \langle M, w \rangle : M \text{ is an LBA that accepts string } w \}$

Simulate $M$ on $w$ until one of:

- $M$ accepts. Then we accept.
- $M$ rejects. Then we reject.
- We perform it going steps. Then reject
  (b/c we know we are in an infinite loop)
$E_{LBA}$ is undecidable

$$E_{LBA} = \{\langle M \rangle : M \text{ is an LBA where } L(M) = \emptyset \}$$

A TM $(M, w)$ decider:

- build LBA $B$ that recognizes accepting computational histories on $M$
- if $E_{LBA}(B)$, reject
- else accept

See Ed Discussion for more...