1 # EQ_{TM} is undecidable

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

$E_{TM}(M)$
{
Def $M_{\ddot\frown}$ = TM that always rejects
(immediately)

If $EQTM(M, M_{\ddot\frown})$
return true / "accept"
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} A_{CFG}

E_{DFA} 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

g^n config of tape

n places over tape R/W head can be

q machine in any of q states

A_{LBA} is decidable (Turing decidable)

$A_{LBA} = \{\langle M, w \rangle :$
M 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 $1 + g^n \cdot n \cdot g$ 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...