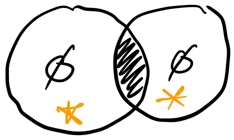
Deciding & Recognizing Languages — Sipser pp. 197-199

Decide $EQ_{DFA} = \{\langle D_1, D_2 \rangle \mid D_1$ and D_2 are DFAs and $L(D_1) = L(D_2)\}$

$L(D_1)$ $L(D_2)$



M = "On input w:

1. Check that input encodes DFAs D_1, D_2, reject if not.
2. Build DFAs for $\overline{L(D_1)}$ and $\overline{L(D_2)}$ using our trick for complements.
3. Using our tricks/procedures for $\cup$ and $\cap$, we'll build a DFA for $L' = (L(D_1) \cap \overline{L(D_2)}) \cup (L(D_2) \cap \overline{L(D_1)})$
4. $L' = \emptyset$ if and only if $L(D_1) = L(D_2)$

Simulating our TM for E_{DFA} on L': L' empty $\Rightarrow$ accept.

Decide $E_{CFG} = \{\langle G \rangle \mid G$ is a CFG and $L(G) = \emptyset\}$.

Idea 1: try generating all strings? convert to a PDA and test all strings? X

M = "on input w:

1. Check to make sure input encodes a grammar G.
2. Repeat until the set of marked symbols stops increasing:
   Mark all terminals.
   Mark all variables that produce a string of only marked symbols.
3. Accept if and only if the start variable is unmarked.

$S \to AA \mid A0A$
$A \to 11 \mid 00 \mid B$
$B \to S$

Recognize $\overline{E_{TM}} = \{\langle M \rangle \mid M$ is a TM that accepts at least one string.$\}$

$M_0 =$ "On input ω:

1. Check that the input encodes a TM, and reject otherwise.
2. Enumerate all strings over the input alphabet. Say that $s_1, s_2, s_3 \cdots$ is a sequence containing all strings in Σ^*.
3. For $i > 0$, i increasing:
   - simulate strings 1 through i on M for i steps
   - accept if any simulation accepts."