W8L2 decidability continued

Wednesday, March 25, 2020 8:42

Languages:

$A_{DFA} = \{ < D, w > \mid D$ is a DFA, w is a string, and D accepts $w\}$ → decider simulates D on w

$E_{DFA} = \{ < D > \mid D$ is a DFA, and $L(D) = \emptyset\}$ → decider run a DFS on D looking for accept states
- if we find one, reject
- if never find one, accept

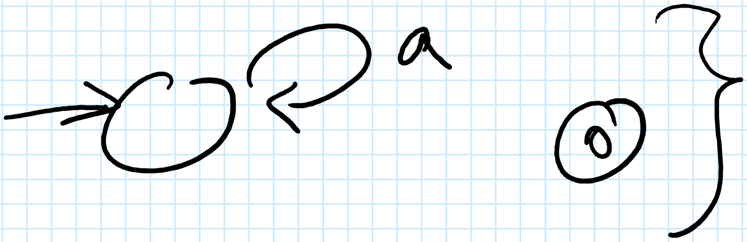
$EQ_{DFA} = \{< M_1, M_2 > \mid M_1$ and M_2 are both DFAs,
and $L(M_1) = L(M_2)\}$

$A_{REX} = \{< R, w > \mid R$ is a regular expression, and matches string $w\}$

$E_{REX} = \{< R > \mid R$ is a regular expression and $L(R) = \emptyset\}$

EQ_{REX}=$\{< R_1, R_2 > \mid R_1$and R_2 are regular expressions
and $L(R_1) = L(R_2)\}$

$A_{NFA}, E_{NFA}, EQ_{NFA}$ defined similarly, but for NFAs.



DFA D

$L(D) = \emptyset$

A_{CFG} $= \{< G, w > \mid G$ is a grammar, w is a string, G generates $w\}$

E_{CFG} and EQ_{CFG} are defined similarly.

$A_{PDA}, E_{PDA}, EQ_{PDA}$ also defined similarly.

What about Turing machines?

$L_{TM} = \{ < M > \mid M$ is a Turing machine$\}$ ← decider checks formatting & that the 7-tuple elements are all there

A countable language