State Minimization (Ch. 2.5)

Why?

Algorithm

- Remove states that are not reachable
- Identify states that are indistinguishable
  These states form a new state

Definition Two states p and q are indistinguishable if for all $w \in \Sigma^*$

\[
\delta^*(q, w) \in F \Rightarrow \delta^*(p, w) \in F
\]
\[
\delta^*(p, w) \notin F \Rightarrow \delta^*(q, w) \notin F
\]

Definition Two states p and q are distinguishable if $\exists w \in \Sigma^*$ s.t.

\[
\delta^*(q, w) \in F \Rightarrow \delta^*(p, w) \notin F \text{ OR } \\
\delta^*(q, w) \notin F \Rightarrow \delta^*(p, w) \in F
\]

Algorithm for Minimizing Number of States in DFA

1. Remove states that are not reachable.
2. Group all nonfinal states together as indistinguishable.
3. Group all final states together as indistinguishable.
4. Repeat until no more states are distinguishable.
   (a) Apply symbol to a group and split group if states are distinguishable.
Example:
Example:
Analysis of Algorithms: (Ch 2.6)

1. Given DFA $M$ and string $w$, is $w \in L(M)$?

2. Given NFA $M$ and string $w$, is $w \in L(M)$?

3. Given an NFA $M_1$, create a DFA $M_2$ s.t. $L(M_1) = L(M_2)$

4. Given a DFA $M_1$, create a minimal state DFA $M_2$ s.t. $L(M_1) = L(M_2)$

5. Given a NFA $M_1$, create a minimal state DFA $M_2$ s.t. $L(M_1) = L(M_2)$