DFA/NFA Minimization
Minimizing DFAs

The Idea:
Distinguishable Strings

- Given a language $L$, strings $x, y$ are **distinguishable** by $L$ if there is string $z$ such that $xz \in L$ and $yz \notin L$ (or the other way round).

- Equivalently, strings $x, y$ are **indistinguishable** if for every string $z$, $xz \in L \iff yz \in L$.

- Given a DFA $M$ and a state $q$, let $L_M(q) = \{ w \mid q \xrightarrow{w} q_f, q_f \in F \}$, i.e., the set of strings leading $M$ to acceptance from $q$. $L_M(q)$ is called the language associated with state $q$.

- If $x$ and $y$ are distinguishable by $L$, any DFA accepting $L$ must reach different states upon reading $x$ and $y.$
Minimal DFA and Distinguishability

Distinguishable strings must be associated with different states.
Indistinguishable strings may end up in the same state.

**DFA minimal ⇔ Every pair of states is distinguishable**
Distinguishable States

Two states $q$ and $r$ are distinguishable if $\exists z_1, \ldots, z_k$

Indistinguishability is an equivalence relation, which partitions the set of states into equivalence classes.
Finding (In)distinguishable States

Phase 1: If $q$ is accepting and $q'$ is rejecting
Mark $(q, q')$ as distinguishable (X)

Phase 2: If $(q, q')$ are marked
Mark $(r, r')$ as distinguishable (X)

Phase 3: Unmarked pairs are indistinguishable
Merge them into groups
An Example

(Phase 1) $q_{11}$ is distinguishable from all other states
(Phase 2) Looking at \((r, r') = (q_\epsilon, q_0)\), Neither \((q_0, q_{00})_{\text{input} \ 0}\) nor 
\((q_1, q_{01})_{\text{input} \ 1}\) are distinguishable
(Phase 2) Looking at \((r, r') = (q_\epsilon, q_1), (q_1, q_{11})\) input 1 is distinguishable.
(Phase 3) Merge states into groups (also called *equivalence classes*)

Minimized DFA:
Why It Works?

Why have we found all distinguishable pairs?

Because we work backwards!
Theorem 1

*Every regular language has a single minimal automaton (up to isomorphism).*

However, minimal NFAs are not unique as the following examples show.
Another way of Characterizing Regular Languages – Residuals of Languages

- The **residual** of a language $L \subseteq \Sigma^*$ with respect to a word $w$ is the language
  \[ L^w = \{ u \in \Sigma^* \mid wu \in L \} \]

- A language $L' \subseteq \Sigma^*$ is a residual of $L$ if $L' = L^w$ for some $w \in \Sigma^*$.

- We define
  \[ R_L : xR_L y \overset{\text{def}}{=} (\forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L) \].

  $R_L$ is an equivalence relation. Note that $xR_L y \Leftrightarrow L^x = L^y$.

- Let $A$ be a DFA. The language recognized by $A$ with $q$ the initial state, denoted by $L_A(q)$, is a residual of $L(A)$.

---

**Theorem 2 (Myhill-Nerode Theorem)**

A language is regular iff it has finitely many residuals.
Canonical DFA of a Regular Language

Let $L \subseteq \Sigma^*$ be a regular language, the canonical DFA of $L$ $M_L = (Q_L, \Sigma, \delta_L, q_{0L}, F_L)$ is

- $Q_L$ is the set of residuals of $L$, i.e., $Q_L = \{ L^w \mid w \in \Sigma^* \}$
- $\delta_L(R, a) = L^{wa}$, where $R = L^w$, for some $w$, where $R \in Q_L$ and $a \in \Sigma$
- $q_{0L} = L^\epsilon = L$
- $F_L = \{ R \in Q_L \mid \epsilon \in R \}$

Example: $L = a^*b^* \subseteq \{a, b\}^*$

- $Q_L = \{ Q_1, Q_2, Q_3 \}$, where
  - $Q_1 = a^*b^* (= L^\epsilon)$, $Q_2 = b^* (= L^{ab})$, $Q_3 = \emptyset (= L^{aba})$
- $q_{0L} = Q_1$
- $F_L = \{ Q_1, Q_2 \}$
- $\delta_L(Q_1, a) = Q_1$, $\delta_L(Q_1, b) = Q_2$, $\delta_L(Q_2, a) = Q_3$, $\delta_L(Q_2, b) = Q_2$, $\delta_L(Q_3, a \mid b) = Q_3$. 

(NTU EE) Finite Automata Spring 2020 14 / 17
Theorem 3

If $L$ is regular, then $M_L$ is the unique minimal DFA up to isomorphism recognizing $L$. 
State partition for DFAs

The quotient of a DFA (NFA) \( M = (Q, \Sigma, \delta, q_0, F) \) w.r.t. a partition \( P \) is \( M_P = (Q_P, \Sigma, \delta_P, q_{0P}, F_P) \), where

- \( Q_P = P = \{B_1, \ldots, B_n\} \),
- \((B, a, B') \in \delta_P \) iff \((q, a, q') \in \delta \) for some \( q \in B \) and \( q' \in B' \),
- \( q_{0P} \) is the block containing \( q_0 \),
- \( F_P \) is the set of blocks that contain some state of \( F \).
The notion of a quotient can be used for NFA minimization except that the definition of a partition is slightly different.

- A block $B$ containing $q_1, q_2$ can be split via an input $a$ and a block $B'$ if $\delta(q_1, a) \cap B' \neq \emptyset$ and $\delta(q_2, a) \cap B' = \emptyset$

Theorem 4

Given an NFA $M$ and a number $k$, deciding if there is another NFA $M'$ equivalent to $M$ with at most $k$ states is PSPACE-complete (polynomial-space complete).