Def: Let $x,y$ be strings and $L$ be a language. We say that $x$ and $y$ are **indistinguishable by $L$** if for every $z$ the following holds: $xz \in L$ iff $yz \in L$. We write $x \equiv_L y$.

Note: this is an **equivalence** relation.

Examples: find the equivalence classes of $\equiv_L$:

$L_1 = \{ 0w \mid w \in \{0,1\}^* \}$
\[
\{ 0u \mid u \in \{0,1\}^* \} \quad \text{and} \quad \{ 1u \mid u \in \{0,1\}^* \} \quad \text{and} \quad \{ \lambda \}
\]

$L_2 = \{ w \in \{0,1\}^* \mid \text{sum of digits of } w \text{ is divisible by } 3 \}$
\[
\{ we \in \{0,1\}^* \mid 3 \mid \text{sodo } w \} \quad \text{and} \quad \{ wc \in \{0,1\}^* \mid \text{sodo } w \equiv 2 \pmod{3} \}
\]
\[
\{ wc \in \{0,1\}^* \mid \text{sodo } w \equiv 1 \pmod{3} \}
\]

$L_3 = \{ 0^k1^k \mid k > 0 \}$

ininitely many classes, e.g. $0^*$ each in a different class
Consider a DFA accepting $L$. Suppose that $x$ and $y$ end in the same state $q$. What can we say about $x, y$?

Claim: If $L$ is accepted by a DFA with $\leq k$ states, then $\equiv_L$ has at most $k$ equivalence classes.

Comment: If $L$ regular $\Rightarrow$ # equiv. classes of $\equiv_L$ is finite.
Claim: If $\equiv_L$ has $k$ equivalence classes, then $L$ can be accepted by a DFA with $k$ states.

Proof: Let the equivalence classes of $\equiv_L$ be $C_1, C_2, \ldots, C_k$.

We'll construct the DFA for $L$ as follows:

$$(Q, \Sigma, \delta, q_0, F)$$

where

- $Q = \{C_1, C_2, \ldots, C_k\}$
- $q_0 = [E]_L$
- $F = \{C_j \mid C_j \cap L \neq \emptyset\}$

where $E \in C_i$.

$\delta(C_x, a) = C_m$

where if $w \in C_x$

then $wa \in C_m$.

\[ \square \]
Thm [Myhill-Nerode]: $L$ is regular iff the number of equivalence classes of $\equiv_L$ is finite.

Using Myhill-Nerode to prove nonregularity:

$L_3 = \{ 0^k1^k \mid k > 0 \}$

done :)

$L_4 = \{ ww^R \mid w \in \{0,1\}^* \}$

E.g. $(01)^i$ for different $i$’s, different equiv. classes

$\Rightarrow \infty$ many
Claim: a DFA is minimal iff its number of states is the same as the number of equivalence classes of its language.

How to construct a min. DFA?

**Option 1:** Find all equivalent classes and follow construction from a couple of slides ago.

**Option 2:** Modify existing DFA into an equivalent smallest DFA.
Suppose we have a DFA - how to construct a corresponding minimal DFA?

1) Eliminate unreachable states
   (easy to do in "polynomial-time")
Suppose we have a DFA - how to construct a corresponding minimal DFA?

1. Remove unreachable states.

2. Identify states that correspond to the same equivalence class and merge them.
1. Remove unreachable states.
2. Identify equivalent states (and merge them):
   - construct graph with vertices = states
   - place edges between every accept and nonaccept state
1. Remove unreachable states.
2. Identify equivalent states (and merge them):
   - construct graph with vertices = states
   - place edges between every accept and nonaccept state
   - continue placing edges as follows while can:
     \[
     \{ 
     \text{for } q,r \in Q, q \neq r, \text{ place edge } (q,r) \\
     \text{if there exists } a \in \Sigma \text{ s.t. } \\
     (\delta(q,a),\delta(r,a)) \text{ is an edge.}
     \}
     \]
3. Find groups of states that do not have edges between them, merge them into a single state.
Minimimal DFA

1. Remove unreachable states.
2. Identify equivalent states (and merge them):
   - construct graph with vertices = states
   - place edges between every accept and nonaccept state
   - continue placing edges as follows while can:
     
     for \(q, r \in Q, q \neq r\), place edge \((q, r)\)
     
     if there exists \(a \in \Sigma\) s.t.
     
     \((\delta(q, a), \delta(r, a))\) is an edge.

   - merge all states that do not have edges between them into a single state