Consider a DFA accepting \( L \). Suppose that \( x \) and \( y \) end in the same state \( q \). What can we say about \( x, y \)?

We will say that such \( x, y \) are indistinguishable w.r.t. \( L \).

\[ \begin{align*}
  x^2 & \text{ have to end up in the same state} \\
  y^2 & \text{ in particular} \\
  x^2 & \text{ is accepted iff} \\
  y^2 & \text{ is accepted} \\
  \forall z \in \Sigma^* : xz \in L \text{ iff } yz \in L
\]
Myhill-Nerode Thm

Def: Let $x, y$ be strings and $L$ be a language. We say that $x$ and $y$ are **indistinguishable by $L$** if there 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.

Equivalence classes partition $\Sigma^*$

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

$L_1 = \{ 0w \mid w \in \{0,1\}^* \}$

- $01 \equiv_{L_1} 00011$ indistinguishable bec. $\forall z \in \{0,1\}^*$: $01z \in L_1$ as well as $00011z \in L_1$.
- Observe: all strings starting with a 0 are $\equiv_L$ with one another.
- $[0] = \{0,00,000,\ldots\}$ the set of all strings indisting. from 0, i.e. all strings beg. with 0
- $11 \equiv_{L_1} 1100$ indistinguishable bec. $\forall z \in \{0,1\}^*$: $11z \in L_1$ as well as $1100z \in L_1$.
- $[1] = \{1,11,111,\ldots\}$ the set of all strings starting w. 1
- $[\varepsilon] \equiv_{L_1}$

Diagram:

- $0 \sim [0]$ or $[0] \sim 0$ (equivalence class 0)
- $1 \sim [1]$ or $[1] \sim 1$ (equivalence class 1)
- $\varepsilon \sim [\varepsilon]$ or $[\varepsilon] \sim \varepsilon$ (equivalence class $\varepsilon$)

---

[Problem 1.52, pages 91, 97-8]
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 $\equiv_L$:

$$L_2 = \{ w \in \{0,1\}^* \mid \text{sum of digits of } w \text{ is divisible by 3} \}$$

Idea: $\mod 3 = 0 \equiv 0$

$$\begin{align*}
\epsilon \in L_2 & \quad \text{all strings with \# of ones } \mod 3 = 0 \\
[1] \in L_2 & \quad = 1 \\
[11] \in L_2 & \quad = 2
\end{align*}$$
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 \( \equiv_L \):

\[
L_3 = \{ 0^k1^k \mid k > 0 \}
\]

As discussed on the board: \( \forall i \neq j: 0^i \not\in L_3 0^j \) bec. for \( z = 1^i \):

\[
0^i 2 \in L_3 \\
0^i z \notin L_3
\]

So we have equivalence classes:

\[
[0]_{L_3} \\
[00]_{L_3} \\
[000]_{L_3} \\
[0000]_{L_3} \\
\vdots
\]

\( \infty \)-many equiv. classes (plus other equiv. classes) but this suffices to say that a language is not regular.
Claim: If $L$ is accepted by a DFA with $\leq k$ states, then $\equiv_L$ has $\leq k$ equivalence classes.
Claim: If $\equiv_L$ has $k$ equivalence classes, then $L$ can be accepted by a DFA with $k$ states.
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 😊

∞ many equivalence classes
Thm [Myhill-Nerode]: \( L \) is regular iff the number of equivalence classes of \( \equiv_L \) is finite.

Using Myhill-Nerode to prove nonregularity:

\[
L_4 = \{ \text{ww}^R \mid w \in \{0,1\}^* \}
\]
Thm [Myhill-Nerode]: $L$ is regular iff the number of equivalence classes of $\equiv_L$ is finite.

Claim: a DFA is minimal iff its number of states is the same as the number of equivalence classes of its language.
Suppose we have a DFA - how to construct a corresponding minimal DFA?

1. **remove unreachable states**
   - run DFS/BFS from the starting state, remove unvisited states
Suppose we have a DFA - how to construct a corresponding minimal DFA?

1. Remove unreachable states.

Observe:

- $E$ (corresponding eq. class to $q_0$) and
- $a$ (corresponding to $q_1$)

are distinguishable.

(Rec. $z = E \in E \cdot z \cdot e \subseteq L$)

More generally:

for any accepting state $q$

and any non-accepting state $p$

we know that the corresponding eq. classes

must be different (take $z = E$)
1. Remove unreachable states.
2. Identify equivalent states (and merge them):
   - construct graph with vertices = states
   - place edges between every accept and nonaccept state

for every pair of states that have not been connected yet \( (p, q) \)
- if there is a symbol \( \Sigma \) s.t. after taking \( \Sigma \)-transitions from \( p \) and \( q \) we get to two states that are already connected,
  draw an edge between \( p \) and \( q \).
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.
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