**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.

- Reflexive: $x \equiv_L x$ \checkmark
- Symmetric: $x \equiv_L y \Rightarrow y \equiv_L x$ \checkmark
- Transitive: $x \equiv_L y \land y \equiv_L w \Rightarrow x \equiv_L w$ \checkmark

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

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

- **equiv. class of $\equiv_L$, containing 0:**
  
  $[0]_{\equiv_L} = \{ 0w \mid w \in \{0,1\}^* \}$

- **equiv. class of $\equiv_L$, containing 1:**
  
  $[1]_{\equiv_L} = \{ 1w \mid w \in \{0,1\}^* \}$

  $[\varepsilon]_{\equiv_L} = \{ \varepsilon \}$

**e.g.**

- $0 \not\equiv_L 1$ need to find $z \in \Sigma^+$ s.t. $xz \in L$ and $yz \not\in L$ or vice versa

  - e.g. $z = 0$  
  
  **equiv. classes of $\equiv_{L_1}$**

  - $0 \equiv_{L_1} 00$
  
  - $1 \equiv_{L_1} 11010$
**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 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 $\equiv_L$:

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

- $0 \not\equiv_L 1$, e.g., $2 \in \{0,1\}$
- $0 \equiv_L 0 \not\in L_L$
- $1 \equiv_L 1 \in L_L$

$\begin{align*}
[0]_{\equiv_L} &= \{ z \in \{0,1\}^* \mid z \equiv_L 0 \} \\
[1]_{\equiv_L} &= \{ z \in \{0,1\}^* \mid z \equiv_L 1 \} \\
[11]_{\equiv_L} &= \{ z \in \{0,1\}^* \mid z \equiv_L 1 \}
\end{align*}$
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 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 $\equiv_L$:

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

$0 \not\equiv_{L_3} 00 \quad z=1$
$0 \not\equiv_{L_3} 000 \quad z=1$
$00 \not\equiv_{L_3} 000 \quad z=11$

$[0]_{L_3}$
$[00]_{L_3}$
$[000]_{L_3}$
$\ldots$

$0^k \not\equiv_{L_3} 0^l \quad k \neq l$

Ex. $z=1^k \quad 0^k z \in L_3$

$0^k \equiv_{L_3} 0^l \quad k=l$

$0^k z \not\in L_3$

$0^{3k} z \not\in L_3$
Myhill-Nerode Thm

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 \( \leq k \) equivalence classes.
Claim: If $\equiv_L$ has $k$ equivalence classes, then $L$ can be accepted by a DFA with $k$ states.

Construction:

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA where:

- $Q = \{ [w]_{\equiv_L} \mid w \in \Sigma^* \}$ has $k$ elements, i.e., finite.
- $q_0 = [\varepsilon]_{\equiv_L}$
- $F = \{ [w]_{\equiv_L} \mid w \in L \}$ notice that if $w \in L$ and $u \not\in L$ then $w \#_L u$ because $\varepsilon \#_L \varepsilon$ which $[w]_{\equiv_L}$ contains only strings in $L$

$\delta([w]_{\equiv_L}, \sigma) = [w\sigma]_{\equiv_L}$

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

Using Myhill-Nerode to prove nonregularity:

\( L_3 = \{ 0^k1^k \mid k > 0 \} \)

we already did it! (b/c we showed that \( L_3 \) has \( \infty \) many equiv. classes)

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

\( 0 \notin L_4 \) \( 0^k \notin L_4 \) \( 0^k \notin L_4 \)

\( [\epsilon], [00], ... \)

\( \infty \) many equiv. classes of \( \equiv_{L_4} \)

\( \Rightarrow L_4 \) non-reg.

\( \square \)
Claim: a DFA is minimal iff its number of states is the same as the number of equivalence classes of its language.

idea: if \#states = \#equiv. classes:

if 3 smaller DFA, then we would get a smaller \#equiv. classes from the states

⇒ hence, DFA minimal

if DFA minimal:

we know \#states ≥ \#equiv. classes, hence minimal implies

\#states = \#equiv. classes

(this is possible due to the claim 2 slides back)

□
Suppose we have a DFA - how to construct a corresponding minimal DFA?

1. remove unreachable states (e.g. by using BFS/DFS from the start state to find all reachable states → remove the others)
Suppose we have a DFA - how to construct a corresponding minimal DFA?

1. Remove unreachable states.

2. Idea: Identify pairs of states that correspond to different equiv. classes
   => the others (those that correspond to the same class) will be merged
1. Remove unreachable states.
2. Identify equivalent states (and merge them):
   - construct graph with vertices = states
   - place edges between every accept and nonaccept state

3. we will draw an edge between $p$ and $q$ if:
   $\exists \sigma \in \Sigma$ s.t.:
   $\delta(p, \sigma)$ consist of a pair
   $\delta(q, \sigma)$ consist of states already
   connected by
   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.

   - at the end:
     let's look at the complement of the graph
     merge states connected in the complement
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 ∈ Q, q ≠ r, place edge (q, r) if there exists a ∈ Σ s.t.
     (δ(q, a), δ(r, a)) is an edge.
   - merge all states that do not have edges between them into a single state