(i) Is there a minimum state machine for $L$?

(ii) How are all minimum state machines for $L$ related?

What happens if we club states - say $[A,C]$ and $[B,D]$?

What if we club $[A,B]$ and $[C,D]$?
When and how can we club states together and obtain a legitimate DFA for language $L$?

Two strings $x, y \in \Sigma^*$ will behave similarly in future if $\delta(q_0, x) = \delta(q_0, y)$ since $\delta(q_0, xz) = \delta(q_0, yz)$ for all $z \in \Sigma^*$.

So states club together strings whose future behavior w.r.t. concatenation is same — in particular either both are accepted or both are rejected.

This reln is an equivalence reln since it satisfies $xR_my \Rightarrow xR_mx$ and $yR_mz \Rightarrow xR_my$.
A special class of equivalence relation for strings in $\Sigma^*$

$x \sim y$, $x, y \in \Sigma^*$ are "equivalent" under a right invariant property if $\forall z \in \Sigma^*$, $x \sim y \Rightarrow x \cdot z \sim y \cdot z$

Intuition: Given a DFA, $M$, all string $w \in \Sigma^*$ such that

$\hat{\delta}(q_0, w) = q'$

$x \sim_M y$ if $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$

This equivalence reln $\sim_M$ is right invariant since $\hat{\delta}(q_0, x \cdot z) = \hat{\delta}(q_0, y \cdot z) \forall z$

Why is $\sim_M$ transitive?

If $x \sim_M y \sim_M z \Rightarrow x \sim_M z$
Any equivalence relation on a set partitions the elements into (disjoint) equivalence classes.

Can we achieve a reduction in the number of states (equivalence classes) for a given regular language?

1. What is the min no. of equivalence classes – related to minimum state DFA for a language $L$?

2. Is this machine “unique”?

We want to tighten our definition of the equivalence relation in the following way:

$x \sim_L y \iff \forall z \in \Sigma^* [x.z \sim_L y.z]$
The relation that we had defined on the basis of the machine does not force $x \sim y$ even if $xz \sim yz$, $z \in \Sigma^*$

$R_L$ : relation for a language $L$

$R_L(y, y)$ if $xz \in L^*$ either $x, z$ and $y, z$ are both in $L$ or both are not in $L$.

Claim: $R_L$ is a right invariant equivalence relation

$R_I :$ if $x R_L y, \forall z \in \Sigma^*$ $x z \sim y z$

$R_M$ : for a specific DFA $\delta_L$
We know that for all $z' \in \Sigma^*$, $x \cdot z'$ and $y \cdot z'$ are both in $L$ or not in $L$ for $R$, $xR_y$.

We want to show that $\forall u \in \Sigma^*$ $x \cdot u \cdot R_y \cdot u \cdot y$ or in other words $\forall z \in \Sigma^*$ $x \cdot u \cdot z$ and $y \cdot u \cdot z$ are both in $L$ or not in $L$.

Choose $z' = u \cdot z$

---

The equivalence classes $I$ the rela $R_L$ for a regular language correspond to the states of min state DFA

---

Myhill-Nerode Theorem
A language $L$ is regular iff the no. of equivalence classes $I$ $R_L$ is finite.
For proof we will go through an
indirect construction using $R_M$. The following statements are equivalent:

1. $L$ is a regular language
2. $L$ is the union of some
   number of equivalence classes of
   a right invariant equivalence reln.
   of finite index (equivalence classes)
3. $R_L$ has finite index

$\forall x R_M y \Rightarrow x R_L y$ 
\[ \# \text{equiv classes of } R_L \leq \# \text{equiv classes of } R_M \]

So either both $x \notin L$ or not in $L$

$\Rightarrow x R_L y$
\[ \Theta \Rightarrow \Xi \]  

Given \( R \) with finite index, we will construct a DFA for \( L \).

Let \([x]\) denote the equivalence class of any string \( x \in \Sigma^* \).

\[ M = (Q, \Sigma, \delta, q_0, F) \]

\( \equiv \) \( \text{class of } R \)

\( \exists \]

\[ \delta([x], a) = [xa] \]

Is it consistent? \( \forall a \in \Sigma \), if \( y \in [x] \) is \( [ya] = [xa] \) (by \( R \)).

\( F \): A state is accepting state of \( x \in L \) if \( \hat{\delta}([\Xi], x) \in F \) iff \( x \in L \)

\( [x] \) by our previous definition \( \hat{\delta} \)

So \( [x] \in F \) iff \( x \in L \) \( q.e.d. \)