COL 352 Lecture 15 Feb 10

A special class of equivalence relation for strings in Σ^*

$x \sim y$, $x, y \in \Sigma^*$ 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

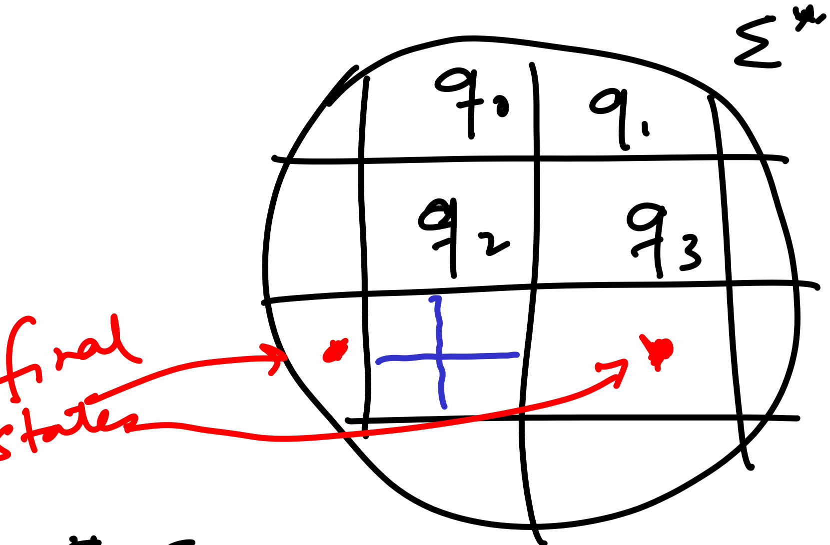
$$\hat{\delta}(q_0, w) = q'$$

$x \sim_M y$ 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 $x \sim y$ $y \sim z \Rightarrow x \sim z$



Any equivalence relation on a set partitions the elements into (disjoint) equivalence classes

Equivalence classes of $\sim_M = |Q|$

Can we achieve a reduction in the number of states (equivalence classes) for a given regular <u>language</u>

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 defn of the equivalence relation in the following way

$$\left. \begin{array}{l} x \sim_L y \text{ iff } \forall z \in \Sigma^* \\ \qquad x \cdot z \sim_L y \cdot z \end{array} \right]$$

The relation that we had defined on the basis of the machine does not force

$x \sim y$ even if $xz \sim yz$ $\forall z \in \Sigma^*$

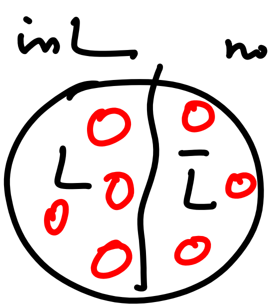
R_L : relation for a language L

Myhill Nerode Relation

$x \, R_L \, y$ iff $\forall z \in \Sigma^*$ either $x \cdot z$ and $y \cdot z$ are both in L or both are not in L.

R_M : for a specific DFA of L

<u>Claim</u> R_L is right invariant equivalence relation



Why is it equivalence?

reflexive, symmetric : obvious

transitive $x R_L y$ $y R_L u$

$\Rightarrow$ $x R_L u$

R I. if $x R_L y$ then $\forall z \in \Sigma^*$ $x \cdot z \, R_L \, y \cdot z$

We know that for all $z' \in \Sigma^*$

$x \cdot z'$ and $y \cdot z'$ are both in L
or not in L

from defn of $x R_L y$

We want to show that $\forall u \in \Sigma^*$

$x \cdot u \; R_L \; y \cdot u$ or in other words

$\forall z \in \Sigma^*$ $xu \cdot z$ and $yu \cdot z$ are both
in L or not in L

Choose $z' = uz$

> The equivalence classes of the reln 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 of R_L is finite.

For proof we will go thru an indirect construction using R_M

The following statements are equivalent

1. L is a regular language

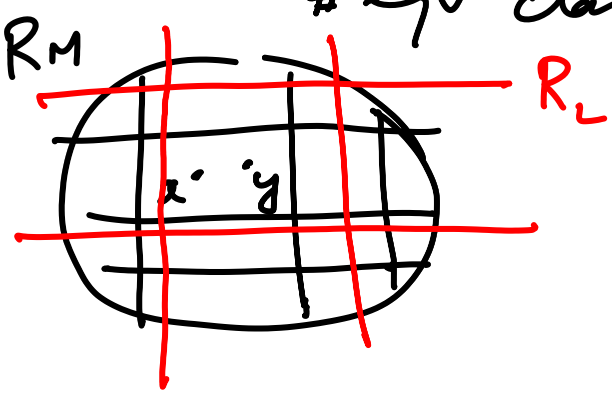
2. L is the union of some number of equivalence classes of a right invariant equivalence reln. of finite index (# equivalence classes is finite) R_M

3. R_L has finite index ✓

① ⇒ ② ⇒ ③ ⇒ ①

· whenever $x R_M y \Rightarrow x R_L y$

eqv classes of $R_L \leq$ # eqv classes of R_M



$x R_M y \Rightarrow \forall z \in \Sigma^*$

$x \cdot z \; R_M \; y \cdot z$ (property of right inv.)

So either both xz are in L or not in L

$\Rightarrow x R_L y$

③ $\Rightarrow$ ① : Given R_L with finite index, we will construct a DFA for L.

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

$M = (Q, q_0, F, \delta)$

Q: the eqv classes of R_L

q_0: $[\varepsilon]$

$\delta([x], a) = [x \cdot a]$, $a \in \Sigma$

Is it consistent? i.e. if $y \in [x]$ is $[ya] = [xa]$ (by R.I.)

F: A state $[x]$ is accepting state if $x \in L$

We must justify that M accepts exactly L

$\hat{\delta}([\varepsilon], x) \in F$ iff $x \in L$

$\hat{\delta}([\varepsilon], x) = [x]$ by our previous defn of $\hat{\delta}$

So $[x] \in F$ iff $x \in L$ Q.E.D