

Lecture 9



Our goals:

- To prove that if a language L has a finite number of equivalence classes then it is finite
- To find FA recognizing L
- This FA will have the fewest possible states.



REMINDER

- Let L be a language Σ^* and $x \in \Sigma^*$.
- $L/x = \{z \in \Sigma^* \mid xz \in L\}$
- x, y are distinguishable strings with respect to L if $L/x \neq L/y$.

I_L indistinguishability relation on Σ^* defined by $xI_L y$ if and only if $L/x = L/y$



- For a language L , $[x]$ denotes the equivalence class containing x .
- *Claim.* If $[x] = [y]$ then $[xa] = [ya]$ for any $a \in \Sigma$
- Proof: $[x] = [y]$ means that for any string $z' \in \Sigma^*$, xz' and yz' are either both in L or both not in L . Thus for any string $z \in \Sigma^*$, xaz and yaz are either both in L or both not in L . Hence,

$[xa] = [ya]$



Lemma

- I_L is right invariant with respect to concatenation. In other words, for any $x, y \in \Sigma^*$ and any $a \in \Sigma$, if $xI_L y$, then $xaI_L ya$.



Theorem

- Let $L \subseteq \Sigma^*$, and let Q_L be the set of equivalence classes of I_L on Σ^* . If Q_L is finite, then $M_L = (Q_L, \Sigma, q_0, A_L, \delta)$ is a FA accepting L , where
 - $q_0 = [\Lambda]$,
 - $A_L = \{q \in Q_L \mid q \cap L \neq \emptyset\}$,
 - $\delta : Q_L \times \Sigma \rightarrow Q_L$ is defined by $\delta([x], a) = [xa]$.
- Furthermore, M_L has the fewest states of any FA accepting L



Proof

- First we prove that for any $x, y \in \Sigma^*$

$$\delta^*([x], y) = [xy]$$



- Induction on y .
- Basis: $\delta^*([x], \Lambda) = [x \Lambda]$ (by def. of FA, $\delta^*([x], \Lambda) = [x]$)
- Induction step

$$\begin{aligned} \delta^*([x], ya) &= \delta(\delta^*([x], y), a) \quad (\text{by definition of } \delta^*) \\ &= \delta([xy], a) \quad (\text{by the induction hypothesis}) \\ &= [xya] \quad (\text{by the definition of } \delta) \end{aligned}$$



- $\delta^*([x], y) = [xy]$

implies

$$\delta^*(q_0, x) = \delta^*([\Lambda], x) = [x]$$



- By definition, $A_L = \{q \in Q_L \mid q \cap L \neq \emptyset\}$

Thus x is accepted by M_L , iff

$$[x] \cap L \neq \emptyset.$$

- $[x] \cap L \neq \emptyset \Leftrightarrow x \in L$. Why?

Because

- $x \in L$ and $x \in [x] \Rightarrow [x] \cap L \neq \emptyset$
- if $y \in [x]$ and $y \in L$ then $x \in L$. (Otherwise Λ distinguish x and y and then x not in $[x]$.)

$$\text{Thus } [x] \cap L \neq \emptyset \Rightarrow x \in L$$



- We conclude that M_L accepts L .
- To prove that M_L has the fewest states we need to remind the following theorem:

Theorem. Let $L \subseteq \Sigma^*$ and for some $n \in \mathbb{N}$ there are n strings pairwise distinguishable (resp. L). Then every FA recognizing L must have at least n states.

- Then we just choose one string from each equivalence class



Corollary (Myhill-Nerode Theorem)

- L is regular language iff the set of equivalence classes of I_L is finite

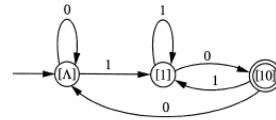


Example

- $L = \{x \in \{0,1\}^* \mid x \text{ ends with } 10\}$
- $\{0,1\}^* \{10\}$
- Strings $\{\Lambda\}$, $\{1\}$ and $\{10\}$ are distinguishable with respect to L
- Classes $[\Lambda]$, $[1]$ and $[10]$ are distinct
- $\forall y \in \Sigma^*$ is in one of these three classes



■



Examples of nonregular languages

- $L = \{0^n 1^n \mid n > 0\}$
- Classes $[1] = \{\text{class of all not prefixes}\}$
- For $j \neq i$, $[0^i] \neq [0^j]$
- $\{0^{j+k} 1^j \mid j > 0\}$ is an equivalence class for every $k > 0$.



Example

- Language in L if the number of '(' parentheses is equal to the number of ')' parentheses



Minimal FA

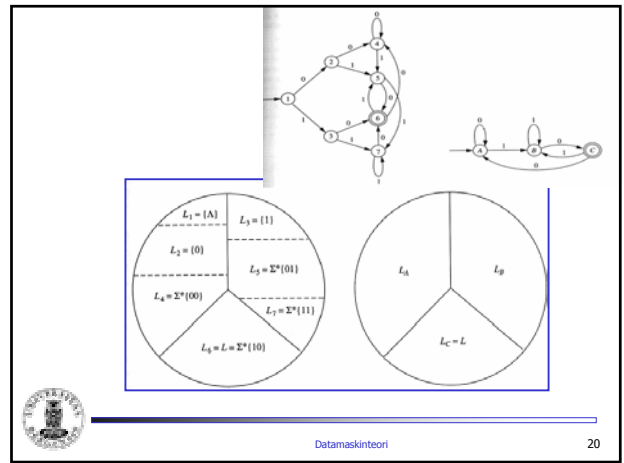
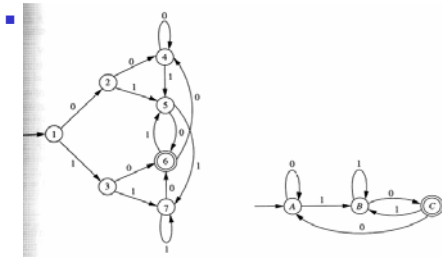
- Question: If we have a FA, how can we 'reduce' it to a 'smaller' one?



- $M = (Q, \Sigma, q_0, A, \delta)$
- $L_q = \{x \in \Sigma^* \mid \delta^*(q_0, x) = q\}$
- Look at L_q and L_L . If the partitions are the same, M is minimal. If not, do something!
- Trivial simplification: Remove all states q with $L_q = \emptyset$



Example: $\{0,1\}^* \{10\}$



Definition

- For pair (p,q) of states of M we write $p \equiv q$ if L_p and L_q are subsets of the same equivalence class



Lemma 5.2 Suppose $p, q \in Q$, and x and y are strings with $x \in L_p$ and $y \in L_q$ (in other words, $\delta^*(q_0, x) = p$ and $\delta^*(q_0, y) = q$). Then these three statements are all equivalent:

- $p \equiv q$.
- $L/x = L/y$ (i.e., $xI_L y$, or x and y are indistinguishable with respect to L).
- For any $z \in \Sigma^*$, $\delta^*(p, z) \in A \Leftrightarrow \delta^*(q, z) \in A$ (i.e., $\delta^*(p, z)$ and $\delta^*(q, z)$ are either both in A or both not in A).



Proof

- $2 \Leftrightarrow 3$: use

$$\delta^*(p, z) = \delta^*(\delta^*(q_0, x), z) = \delta^*(q_0, xz)$$

$$\delta^*(q, z) = \delta^*(\delta^*(q_0, y), z) = \delta^*(q_0, yz)$$

- $2 \Leftrightarrow 1$: also trivial



How to identify those pairs (p,q) with $p \not\equiv q$?

By Lemma, $p \not\equiv q$ means that for some z exactly one of the states $\delta^*(p, z)$ and $\delta^*(q, z)$ is in A

