class 6]

MYHILL-NERODE THEOREM (not in book!)

- Given a language A, what is a minimal DFA for A? So far we assumed that A is given through a DFA itself, but can we give a more abstract construction?

- Can we view a language $A \subseteq \Sigma^*$ as an automaton?

state = string over Σ (history)

$D_A \stackrel{def}{=} (\Sigma^*, \Sigma, \delta, \varepsilon, A)$

$\delta(x,a) \stackrel{def}{=} xa$ then $\hat{\delta}(x,y) = x \cdot y$ and so

$$L(D_A) = \{x \in \Sigma^* \mid \hat{\delta}(\varepsilon, x) \in A\}$$
$$= \{x \in \Sigma^* \mid \varepsilon \cdot x \in A\}$$
$$= A$$

Is D_A a DFA? No - infinite states!

- Idea: construct the quotient automaton $D_A/\approx$

state = equivalence class of histories indistinguishable by experiment! "the relevant part"

$$x \approx y \iff \forall z \in \Sigma.\ (\hat{\delta}(x,z) \in A \iff \hat{\delta}(y,z) \in A) \quad \{\text{Def} \approx\}$$
$$\iff \forall z \in \Sigma.\ (x \cdot z \in A \iff y \cdot z \in A)$$

• Myhill-Nerode equivalence on strings over Σ induced by $A \subseteq \Sigma^*$

DEF Let A be a language over Σ. For all $x, y \in \Sigma^*$

$$\boxed{x \equiv_A y \overset{def}{\iff} \forall z \in \Sigma^*.\ (x \cdot z \in A \iff y \cdot z \in A)}$$

THM Language $A \subseteq \Sigma^*$ is regular iff $\Sigma^*/_{\equiv_A}$ is finite

An automata-independent characterization! Ex below

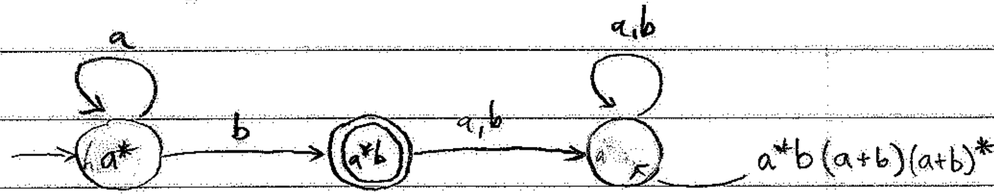
DEF Let A be a language over Σ. The Myhill-Nerode automaton for A is simply $D_A/\approx$, the quotient automaton for D_A: (an abstract construction!)

$$D_A/\approx \;=\; (\Sigma^*/\equiv_A,\ \Sigma,\ \delta',\ [\varepsilon]_{\equiv_A},\ A/\equiv_A)$$

where $\delta'([x]_{\equiv_A}, a) = [xa]_{\equiv_A}$

EX Let $A \overset{def}{=} L(a^*b)$ over $\{a,b\}$. Construct $D_A/\approx$.

Compare $\varepsilon, a, b, aa, ab, ba, bb$, guess equivalence classes, represent by reg. expr., build automaton:



One can prove for any $D = (Q, \Sigma, \delta_D, s, F)$ accepting $A \subseteq \Sigma^*$

$$\hat{\delta}_D(s,x) \approx \hat{\delta}_D(s,y) \iff x \equiv_A y \qquad \text{(work out in detail)}$$

and therefore $D/\approx$ is isomorphic to $D_A/\approx$!

HW3, Problem 1: The same for $L(a^*b^* + b^*a^*)$

• First application: LOWER BOUNDS

THM For all m there is a language L_m such that:

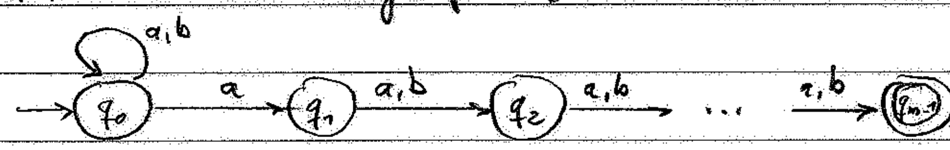
i) there is an NFA with m states accepting L_m

ii) every DFA accepting L_m has at least 2^{m-1} states

Proof: Let $\Sigma = \{a,b\}$. Define

$L_m \overset{def}{=} \{x \in \Sigma^* \mid |x| \geq m-1$ and $(m-1)$ butlast of x is $a\}$

i) An NFA for this language with m states:



ii) We shall show that $\Sigma^*/\equiv_{L_m}$ has at least 2^{m-1} equivalence classes by identifying 2^{m-1} incomparable strings over Σ. Consider Σ^{m-1}, i.e. all strings $x \in \Sigma^*$ of length $m-1$. For all $x, y \in \Sigma^{m-1}$ we have

$$x \neq y \Rightarrow x \not\equiv_{L_m} y$$

since if $x \neq y$ then $x(i) \neq y(i)$ for some $i \leq m-1$, and then every $z \in \Sigma^*$ of length $i-1$ is a distinguishing experiment:

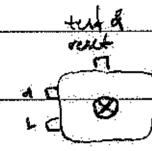
$$x \cdot z \in L_m \Leftrightarrow y \cdot z \notin L_m$$

and hence $x \not\equiv_{L_m} y$.

• Second application: <u>REGULAR</u> <u>INFERENCE</u>

Models in design and verification : pre-hoc and post-hoc
Post-hoc models : automatic extraction : white-box and black-box
aka "model mining". Under-approximations good for finding errors!

Regular inference : given a black box automaton, learn its reg. language by constructing a minimal DFA for it.



<u>Ex</u> (see next page)

Standard setup for regular inference : (see slides)

Observations $\overset{Obs^+ \quad Obs^-}{Obs} \subseteq \Sigma^*$ $\quad \overset{obs}{x} = \overset{staterepr.}{u} \cdot \overset{experiment}{z}$

State representatives $StateRep = Pref(Obs)$ (Obs should be prefix-closed)

Experiments $Exp = Post(Obs)$ For state reps $x, y \in StateRep$

$$\boxed{x \equiv_{Obs} y \iff \forall z \in Exp : xz, yz \in Obs . (xz \in Obs^+ \Leftrightarrow yz \in Obs^+)}$$

Hypothesis DFA

$$D_{Obs} \stackrel{def}{=} (StateRep/\equiv_{Obs}, \Sigma, \delta, [\varepsilon]_{\equiv_{Obs}}, Obs^+/\equiv_{Obs})$$
where $\delta([x]_{\equiv_{Obs}}, a) \stackrel{def}{=} [xa]_{\equiv_{Obs}}$ (defined only if $xa \in Obs$)

but Obs has to be "complete" to allow forming a DFA

Angluin : every state has a "short representative" x such that $xa \in StateRep$ for every $a \in \Sigma$

notice that $x \equiv_{Obs} y \Rightarrow xa \equiv_{Obs} ya$!

<u>Task</u> : from Obs, construct the minimal DFA consistent with it

6.5

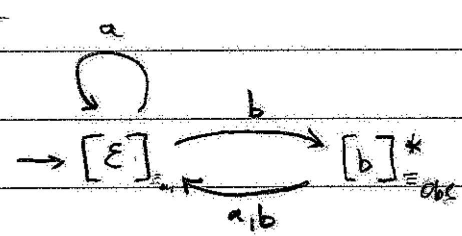
Ex $Obs = \{(\varepsilon,-), (a,-), (b,+),$
$(aa,-), (ab,+), (ba,-), (bb,-)\}$ $\quad (A = L(a^*b))$

prefix- and postfix-closed, hence:
StateRep = Exp = Obs

Equivalence classes $StateRep/\equiv_{Obs}$:
$\{\varepsilon, a, aa, ba, bb\}$
$\{b, ab\}$

StateRep \ Exp	ε	a	b	aa	ab	ba	bb	bab
ε	−	−	+	−	+	−	−	−
a	−	−	+					
b	+	−	−		−			
aa	−							
ab	+							
ba	−		−					
bb	−							
bab	−							

D_{Obs}:



counter-example: (bab, −)

we add (bab, −) to Obs, complete and make consistent

Adaptive Model Checking

- Regular inference and "black-box" verification

  We want to know whether $L(A_{sys}) \subseteq L(A_{spec})$, A_{sys} is a black box.

  We experiment with the system to build hypothesis D_{obs}.

  We check $L(D_{obs}) \subseteq L(A_{spec}) \Leftrightarrow L(D_{obs} \times \overline{A_{spec}}) = \emptyset$.

  - "yes": then we don't know, but we can try to refine D_{obs}.
  - "no": then we also obtain from the reachability check a string x such that $x \in L(D_{obs})$ but $x \notin L(A_{spec})$.

    Check $x \in L(A)$ x witness for

    - "yes": we found a real bug : $L(A) \not\subseteq L(A_{spec})$
    - "no": spurious counter-example, use to build a finer hypothesis : add observation $(x,-)$