CSCE 355
2/14/2024

String homomorphisms. Pumping lemma for reg. lang's

Def: Let Σ, Γ be alphabets. A string homomorphism from Σ to Γ is a map

$$\varphi : \Sigma^* \longrightarrow \Gamma^*$$

that respects concat, i.e., $\forall x, y \in \Sigma^*$

$$\varphi(\underbrace{xy}_{\text{concat in } \Sigma^*}) = \underbrace{\varphi(x)\varphi(y)}_{\text{concat in } \Gamma^*}$$

Prop: $\varphi(\varepsilon) = \varepsilon$.

Proof: $\forall x \in \Sigma^*$, $\varphi(x) = \varphi(\varepsilon x) = \underbrace{\varphi(\varepsilon)}\varphi(x)$

$\therefore \varphi(\varepsilon) = \varepsilon$. //

Prop: φ is completely determined by how it maps strings of length 1.

Proof: $\forall x \in \Sigma^*$, let $x = x_1 \cdots x_n$ $(x_i \in \Sigma)$ Then

$$\varphi(x) = \varphi(x_1 \cdots x_n) = \varphi(x_1 \cdots x_{n-1})\varphi(x_n)$$

$$= \varphi(x_1 \cdots x_{n-2})\varphi(x_{n-1})\varphi(x_n)$$

$$= \cdots = \varphi(x_1)\varphi(x_2)\cdots\varphi(x_n). \quad //$$

Ex: $\Sigma = \{a, b, c\}$, $\Gamma = \{0, 1\}$

$$\varphi(a) = 011$$
$$\varphi(b) = 10$$
$$\varphi(c) = \varepsilon$$

$$\varphi(abacacb) = \underbrace{011}_{\varphi(a)}\underbrace{10}_{\varphi(b)}\underbrace{011}_{\varphi(a)}\underbrace{}_{\varphi(c)}\underbrace{011}_{\varphi(a)}\underbrace{}_{\varphi(c)}\underbrace{10}_{\varphi(b)}$$

Def: $L \subseteq \Sigma^*$. Define the image $\varphi(L)$ of L under φ as

$$\varphi(L) := \{\varphi(x) : x \in L\} \subseteq \Gamma^*$$

Let $M \subseteq \Gamma^*$. The inverse image of M under φ is

$$\varphi^{-1}(M) := \{x \in \Sigma^* : \varphi(x) \in M\}.$$

Prop: Let Σ, Γ be as above, φ a string homom. from Σ to Γ.

1. If $L \subseteq \Sigma^*$ is regular, then $\varphi(L)$ is regular
2. If $M \subseteq \Gamma^*$ is regular, then $\varphi^{-1}(M)$ is regular.

Proof of (1): Given a regex r for L, give rules to produce a regex r' over Γ such that

$$L(r') = \varphi(L(r)):$$

	r	r'
	$\emptyset$	$\emptyset$
$(a \in \Sigma)$	a	$\varphi(a)$ ← makes for a different regex
s,t regexes over Σ	$s+t$	$s'+t'$
	st	$s't'$
	s^*	$(s')^*$

Proof of correctness by induction on the syntax omitted. //

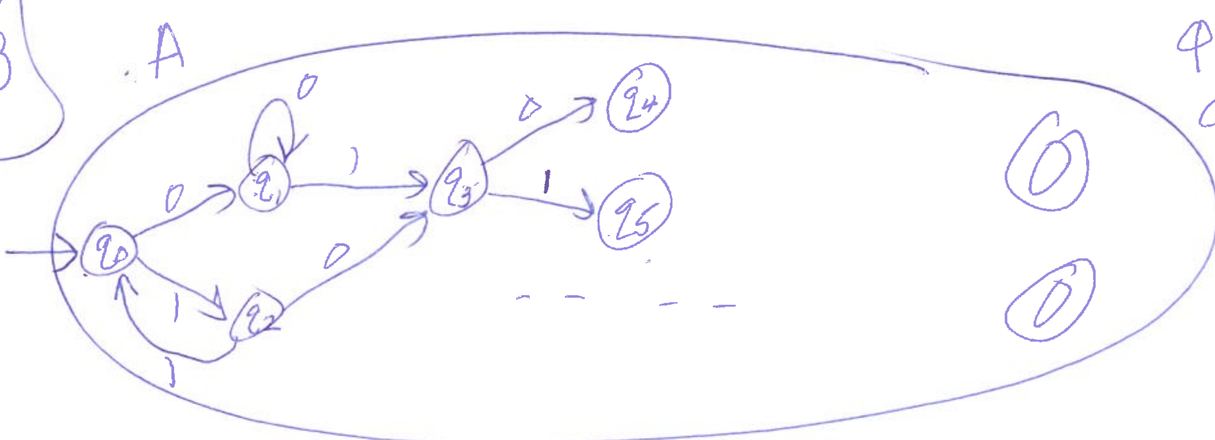
<u>Proof of (2):</u> $M \subseteq \Gamma^*$. Given a DFA A recognizing M, transform into a DFA A' recognizing $\varphi^{-1}(M)$:

If $A = \langle Q, \Gamma, \delta, q_0, F \rangle$, then

$$A' = \langle Q, \Sigma, \delta', q_0, F \rangle$$

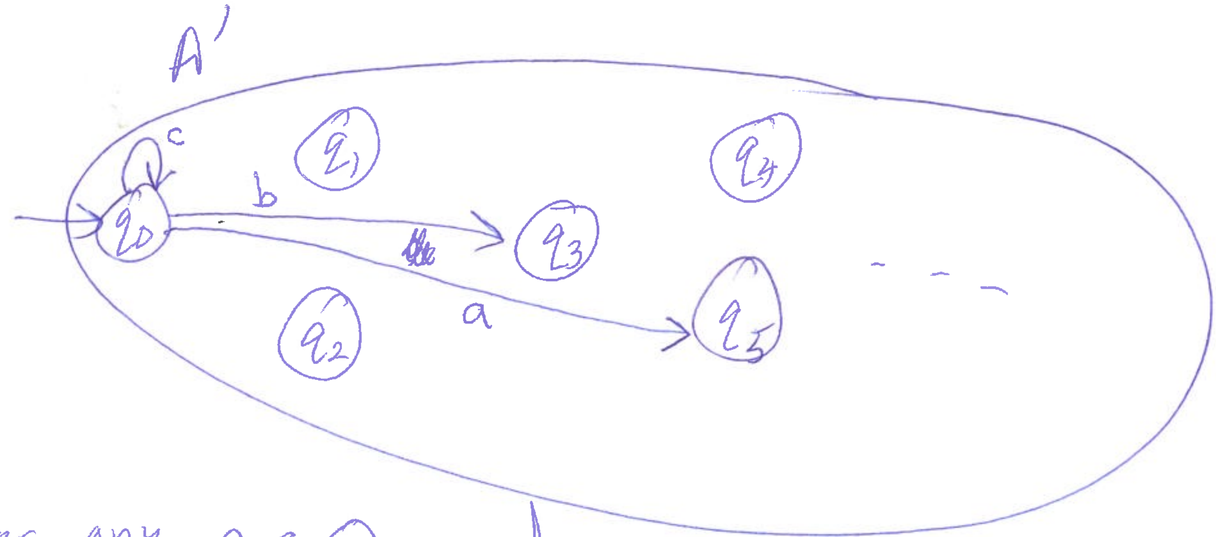
$\Sigma = \{a,b,c\}$
$\Gamma = \{0,1\}$

$\varphi(a) = 011$
$\varphi(b) = 10$
$\varphi(c) = \varepsilon$



Want transitions in A' on symbol $\underline{a}$ to go to the same state that A goes to reading $\varphi(a)$

$\underline{b}$
$\underline{c}$



for any $q \in Q$ and $a \in \Sigma'$,

$\delta'(q, a)$ is the unique end state of A's computation reading $\varphi(a)$ starting in state q.

Any $w \in \Sigma'^*$, A' reading w from q_0 ends in the same state as A reading $\varphi(w)$ from q_0.

$\therefore$ A' accepts w iff A accepts $\varphi(w)$.

$\therefore$ $w \in L(A')$ iff $\varphi(w) \in L(A)$

$\therefore$ $L(A') = \varphi^{-1}(L(A))$

$\therefore$ $\varphi^{-1}(L(A))$ is regular. //

Ex: Let L be a language, let $\quad (L \subseteq \{a, b, c\}^*)$

DoubleA(L) := the set of all strings obtained from strings in L by replacing each a with aa.

DoubleA$(\{\varepsilon, abac, bc\}) = \{\varepsilon, aabaac, bc\}$

<u>Prop</u>: If L is regular, then DoubleA(L) is regular.

<u>Proof</u>: DoubleA(L) $= \varphi(L)$ where $\varphi : \Sigma^* \to \Sigma^*$ is unique str. homom. such that

$$\varphi(a) = aa$$
$$\varphi(b) = b$$
$$\varphi(c) = c$$

$\therefore$ DoubleA(L) is regular by (1) of prev. prop. //

<u>Prop</u>: If L is regular then $\text{DoubleA}^{-1}(L)$ ($= \{w \in \Sigma^* : \text{DoubleA}(w) \in L\}$) is regular.

<u>Proof</u>: By part (2) of the Prop.

<u>Proving a lang. is not regular.</u>

<u>Def</u>: $L \subseteq \Sigma^*$ any language. L is <u>pumpable</u> if

$\exists\, p > 0$ (pos. integer — the pumping length)

$\forall s \in L$ such that $|s| \geq p$,

$\exists\, x, y, z$ strings with

$$xyz = s$$
$$|xy| \leq p$$
$$y \neq \varepsilon,$$
$$\forall i \geq 0, \quad xy^iz \in L.$$

Lemma (Pumping Lemma for Regular Languages):

Every regular language is pumpable.

Ex: Σ^* is pumpable: let $p := 1$

$\forall s \in \Sigma^*$, $|s| \geq 1$, we can let

$x := \varepsilon$
$y :=$ first symbol of s
$z :=$ rest of s.

Then $s = xyz$
$|xy| = 1 \leq p = 1$
$y \neq \varepsilon$.
$\forall i \geq 0 \;\; xy^iz \in \Sigma^*$

$\left[\; y^i = \underbrace{yy\cdots y}_{i \text{ times}} \right.$

Ex: Every finite language is pumpable.

Let L be a finite language.

Set $p := 1 + \max\limits_{s \in L} |s|$.

Then $\forall s \in L$ with $|s| \geq p$

$\cdots\,\Big]$ true "vacuously" because there are no such strings.

Proof sketch:

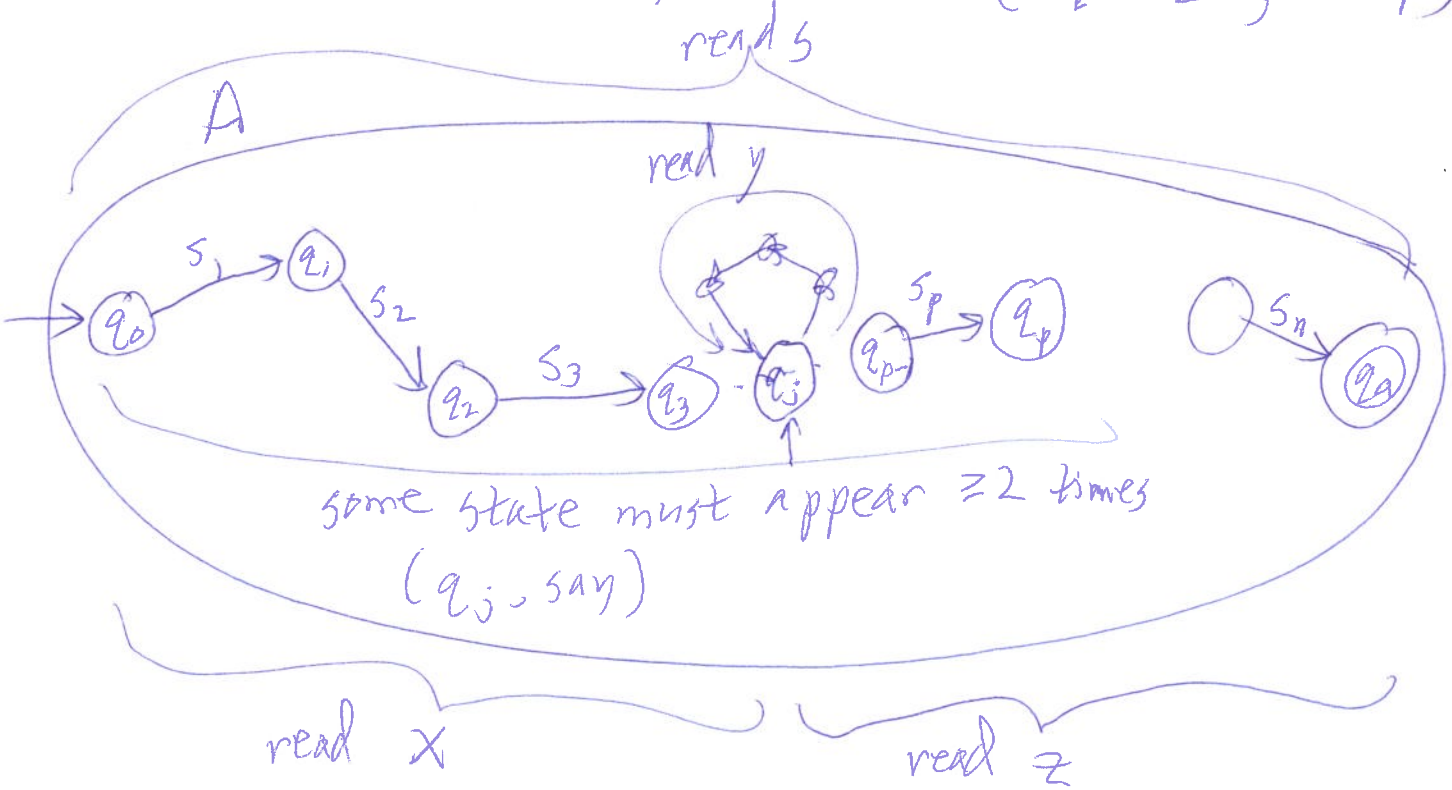
Let $L \subseteq \Sigma^*$ be regular and let

A be a DFA recognizing L.

Let p be the number of states of A.

Consider any string $s \in L$ with $|s| \geq p$.

let $s = \cancel{\ } s_1 s_2 \cdots s_n$ $(s_i \in \Sigma, n \geq p)$



∴ Can read x followed by any # of y's (incl. 0) followed by z to get to the same accepting state as with reading s.

∴ $xy^iz \in L$ (accepted by A) $\forall i \geq 0$. ∎

To show a lang. not regular, we show it is not pumpable, which suffices by the Pumping Lemma