String homomorphisms

CSCE 355 9/22/2022

Def: Let Σ and Γ be alphabets. A (*string*) **homomorphism** from Σ to Γ is a map $\varphi : \Sigma^* \to \Gamma^*$ that preserves concatenation: That is, for any $x, y \in \Sigma^*$

If $\begin{cases} \varphi(x) = v \\ \varphi(y) = w \end{cases}$ then $\varphi(xy) = vw$

Basic fact: If φ is a homomorphism as above, then $\varphi(\varepsilon) = \varepsilon$.

Proof: $\underbrace{\varphi(\varepsilon)}_{\text{length } \ell} = \varphi(\varepsilon\varepsilon) = \underbrace{\varphi(\varepsilon)\varphi(\varepsilon)}_{\text{length } 2\ell}$

Thus $\ell = 2\ell$, $\therefore$ $\ell = 0$, that is, $|\varphi(\varepsilon)| = 0$

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

Basic fact: φ is completely determined by how it maps
uniquely
strings of length 1.

"Proof": $w \in \Sigma^*$ arbitrary. $w = w_1 w_2 \cdots w_n$ $\left(n = |w|, w_i \in \Sigma\right)$

$$\varphi(w) = \varphi(w_1 w_2 \cdots w_n) = \varphi(w_1)\varphi(w_2 \cdots w_n)$$
$$= \varphi(w_1)\varphi(w_2)\varphi(w_3 \cdots w_n) = \cdots = \varphi(w_1)\cdots\varphi(w_n). \quad //$$

Converse: any map from Σ to Γ^* is uniquely extendable to a homom. $\Sigma^* \to \Gamma^*$.

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

$$\left.\begin{aligned} \varphi(a) &= 100 \\ \varphi(b) &= 11 \\ \varphi(c) &= \varepsilon \end{aligned}\right\} \quad \begin{aligned} &\varphi(abacbca) \\ &= 1001110011100 \end{aligned}$$

$\varphi : \Sigma^* \to \Gamma^*$ homom. $\quad L \subseteq \Sigma^*$

Define $\varphi(L) := \{\varphi(w) : w \in L\} \subseteq \Gamma^*$ — homomorphic image of L

For any $L' \subseteq \Gamma^*$, define

$$\varphi^{-1}(L') := \{w \in \Sigma^* : \varphi(w) \in L'\}$$

— inverse homom. image of L'

Thm: Let $\varphi : \Sigma^* \to \Gamma^*$ be a homom. and let $L \subseteq \Sigma^*$. If L is regular, then $\varphi(L)$ is regular.

Pf: By induction on regex syntax: Let r be any regex over Σ. We define r' regex over Γ such that $\varphi(L(r)) = L(r')$.

[Since L is regular, there exists an r such that $L = L(r)$. Then $\varphi(L) = L(r')$ hence regular.]

Table describing r' for any r:

	r	r'
	$\emptyset$	$\emptyset$
$(a \in \Sigma)$	a	$\varphi(a)$
s, t regexes over Σ.	$s+t$	$s'+t'$
	st	$s't'$
	s^*	$(s')^*$

[regex concat of the symbols in $\varphi(a)$]

Ex: $\varphi(a) = 110$ ~~then~~ and $r = a$ then $r' = 110$ (as a regex)

[Skipping proof of correctness] //

Ex: $\varphi(a) = 100$ $\quad r = (ab + bc)^*(a+b)$

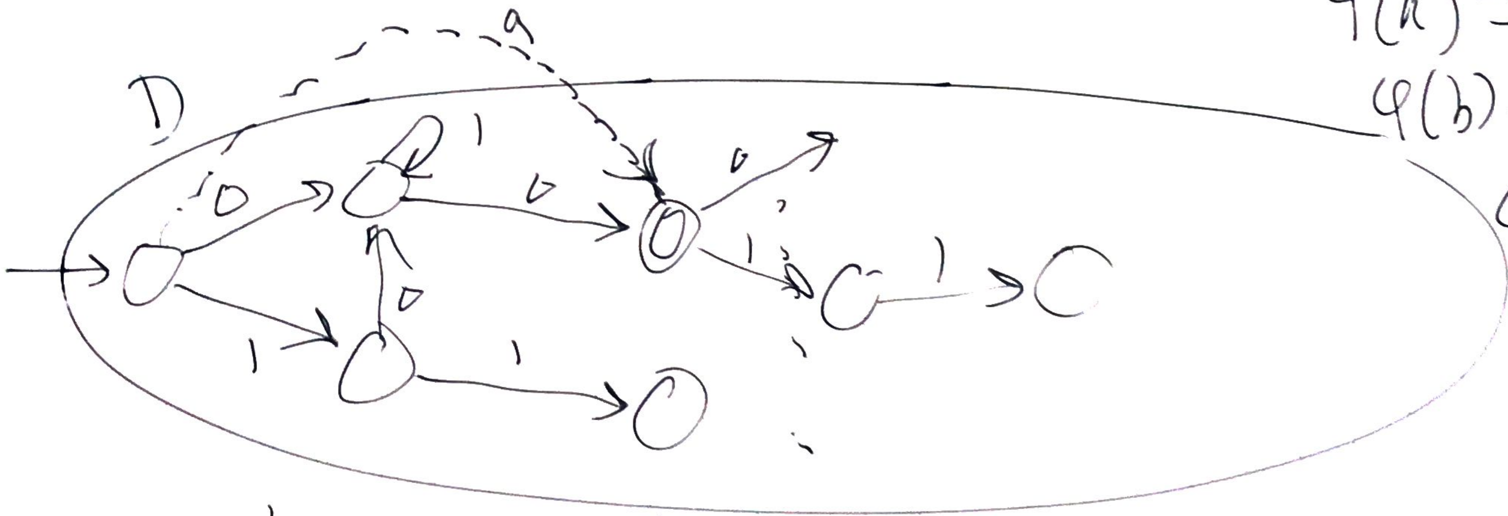
$\varphi(b) = 11$ $\quad r' = (10011 + 11)^*(100 + 11)$

$\varphi(c) = \varepsilon$

Thm: $\varphi : \Sigma^* \to \Gamma^*$ homom. $L \subseteq \Gamma^*$ arbitrary.
If L is regular, then $\varphi^{-1}(L)$ is regular.

Proof idea: Consider a DFA D recognizing L.
Build a DFA D' recognizing $\varphi^{-1}(L)$

$\varphi(a) = 100$
$\varphi(b) = 11$
$\varphi(c) = \varepsilon$



$w = abc$
$\varphi(w) = 10011$

~~##~~ Reading w, go to the same state in D as if reading $\varphi(w)$.

Formally: Let $D = \langle Q, \Gamma, \delta, q_0, F \rangle$.

Then $D' := \langle Q, \Sigma, \delta', q_0, F \rangle$

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

$$\delta'(q, a) := \hat{\delta}(q, \varphi(a))$$

<u>Proof</u> by string induction that $L(D') = \varphi^{-1}(L(D)) = \varphi^{-1}(L)$.

(skipped)

<u>Uses of regexes</u>: - text search $*$.doc   <u>ex</u>.

- token recognition in prog. lang.

int constants

fp "

identifiers

Shorthands: $\varepsilon := \emptyset^*$ (matches ε and nothing else)

""

R, S regexes

$R|S := R+S$

$R+ := RR^*$ (1 or more R's)

$R? := R+\varepsilon$

$R|""$ (optional R)

0 or 1 occurrences of R

Character classes:

$[abc] := (a+b+c)$

$[a-z] := (a+b+c+\cdots+z)$

Identifiers: in C, C++, Java: `[_A-Za-z][_A-Za-z0-9]*`

Int constants (unsigned): `[0-9]+`

FP constants: ≥ 1 digits followed by "." followed by ≥ 1 digits followed by optional exponent (letter E followed by optionally signed int const)

$$[0-9]+"."[0-9]+([eE][+-]?[0-9]+)?$$

Def: Let $L \subseteq \Sigma^*$ be language. Say that L is **pumpable** if

there exists $p > 0$ (the pumping length) such that

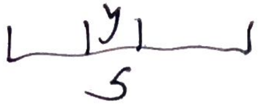
for every string $s \in L$ such that $|s| \geq p$

there exists x, y, z such that

- $s = xyz$
- $|xy| \leq p$
- $y \neq \varepsilon$

such that for any $i \geq 0$, $xy^iz \in L$.

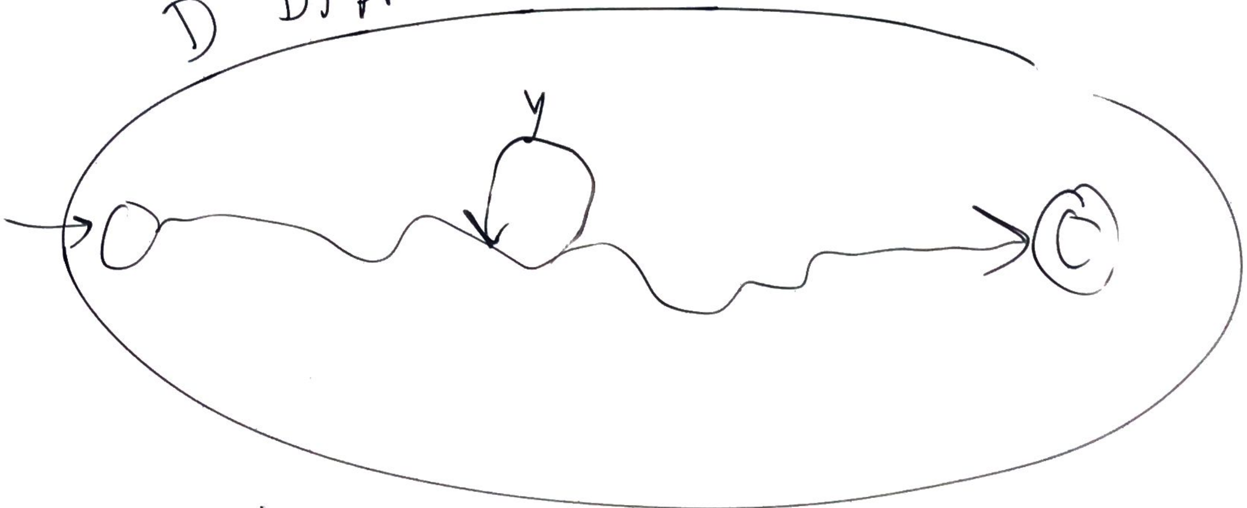
every sufficiently long string in L can be "pumped"



Lemma (Pumping Lemma): Every regular language is pumpable.

Ex:

D DFA for L



let $p :=$ # of states of D