Any questions?

... remember that we were discussing closure properties for regular languages...

L reg $\Rightarrow L^6$ reg?

$\hookrightarrow$ regex R s.t. $\mathcal{L}(R) = L$

$\mathcal{L}(RRRRRR) = L^6$

Uh, oh! Must be "midterm week" because it seems there are fewer people than usual... or maybe everyone is just running a little late tonight/today/this morning... :)

Example 2.26: Is $L_0 = \{a^n b^n : n \in \mathbb{N}\}$ regular?

Explore

- Q: Is $L_0 = \mathcal{L}(a^* b^*)$? ✗ a, aaab, bb, ...

  $L_0 \subset \mathcal{L}(a^* b^*)$

  $\mathcal{L}((ab)^*)$ ✗ ababab

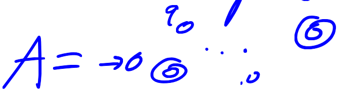
- conjecture: L_0 is <u>not</u> regular.
- Proof? show <u>no</u> DFA recognizes L_0, i.e., for all DFA A, A makes a mistake on some input: $\boxed{\text{A rejects some } w \in L_0 \text{ or A accepts some } w \notin L_0}$

$\equiv$ (A accepts all $w \in L_0$) $\Rightarrow$ (A accepts some $w \notin L_0$)

Example 2.26 *(cont'd)* $A = (Q, \{a,b\}, \delta, s, F)$

Proof

- Let A be a DFA. Assume A accepts all $w \in L_0$
- WTP: A accepts some $w' \notin L_0$.
- let $p = |Q|$ (# states of A)
- consider input $w = a^{p+1}b^{p+1}$. $w \in L_0 \Rightarrow A$ accepts w



- while processing w, A must loop during the processing of a^{p+1}

$\rightarrow$ for some state $q \in Q$, $i \in \mathbb{Z}^+$, $\delta^*(q, a^i) = q$

- when A is in state q it cannot track how many times the "loop" was used: $\delta^*(q, a^{2i}) = q$

$\Rightarrow A$ accepts $a^{p+1+i}b^{p+1} \notin L_0$

Observations on Regular Languages

In general

- If $L \subseteq \Sigma^*$ is regular, then $L = \mathcal{L}(A)$ for some DFA A

A accepts all $w \in L$
A rejects all $w \notin L$

A is fixed

- let p be the number of states of A
- let $w \in L$ be a string with $|w| > p$

· is there always such a string?
· no: e.g., $L = \{a, bb\}$

- What happens when A processes w?

A must reuse some states — path through A contains a loop
$\Rightarrow$ part of w processed by the loop can be repeated any number of times without changing the outcome (accept/reject)

The Pumping Lemma

Theorem 2.27: Pumping Lemma

- *For all regular languages $L \subseteq \Sigma^*$,*
- *there exists a positive integer $p \in \mathbb{Z}^+$ such that*
- *for every string $w \in L$ with $|w| \geqslant p$*
- *there are sub-strings $x, y, z \in \Sigma^*$ such that*
  - $w = xyz$,
  - $|xy| \leqslant p$,
  - $|y| \geqslant 1$, *and*
  - $xy^iz \in L$ *for all* $i \in \mathbb{N}$.

intuition:
x = part of w before loop
y = part of w in the loop
z = part of w after loop

Discussion

- Preceding slide contains high-level proof idea
- Imposes a condition on the structure of all "long enough" strings in *L*
- What if *L* does not contain such strings?

If L is finite, lemma is true when p is any value > length of longest string in L

How does this help?

Pumping Lemma: applies to **all** regular languages

- For all regular languages $L \subseteq \Sigma^*$,
- there exists $p \in \mathbb{Z}^+$ such that
- for every string $w \in L$ with $|w| \geqslant p$
- there are $x, y, z \in \Sigma^*$ such that
  - $w = xyz$ and
  - $|xy| \leqslant p$ and
  - $|y| \geqslant 1$ and
- $xy^iz \in L$ for all $i \in \mathbb{N}$.

Application: proving language L non-regular (by contradiction)

- Assume $L \subseteq \Sigma^*$ is regular
- let $p \in \mathbb{Z}^+$ (arbitrary)
- choose $w \in L$ with $|w| \geqslant p$
- let $x, y, z \in \Sigma^*$ (arbitrary) such that
  - $w = xyz$ and
  - $|xy| \leqslant p$ and
  - $|y| \geqslant 1$, then
- choose $i \in \mathbb{N}$ such that $xy^iz \notin L$.

key — requires insight

Example 2.28: $L_0 = \{0^n 1^n : n \in \mathbb{N}\}$

Proof L_0 <u>not</u> reg.

- Assume L_0 reg.
- Let $p \in \mathbb{Z}^+$
- choose $w = \underline{0^p 1^p}$
- let $x, y, z \in \{0,1\}^*$ satisfy

$$\left.\begin{array}{l} w = xyz \\ |xy| \leq p \\ |y| \geq 1 \end{array}\right\} \Rightarrow y = 0^k \text{ for some } k \geq 1$$

$$w = \underbrace{00\cdots0}_{x}\underbrace{0\cdots0}_{y}\underbrace{0\cdots0\,11\cdots111}_{z}$$

with $\overbrace{00\cdots00\cdots00\cdots0}^{p}\overbrace{11\cdots111}^{p}$

- $\Rightarrow xy^2z = xyyz = 0^{p+k}1^p \notin L_0$

Example 2.29: $L_1 = \{xx : x \in \{a, b, c\}^*\}$

Proof