CPSC 421/501 Oct 21

Midterm covers up to today,

but only regular, non-regular, Myhill-Nerode } languages (Ch 1 or [Sip])

not Ch 3 & Turing machines

Midterm

	Tu	Th
Oct	19	21
	26	28
Nov	2	4

Assign grade HW 6

due Oct 29

release solutions

Midterm

Homework 7 assigned no earlier than after the midterm

Focus on

$$C_k = \{ w \in \{a,b\}^* \mid \text{the } k^{th} \text{ last symbol is an } a \}$$

$$\{ 0^n 1^n \mid n \in \mathbb{Z}_{\geq 0} \}$$

PALINDROME = fix $|\Sigma| \geq 2$

$$\{ w \in \Sigma^* \mid w = w^{rev} \}$$

non-regular

- §1.4 [Sip]
- today via Myhill-Nerode or "derived"

- easy to program
with a simple
Turing machine

Also

$$\{0^{(n^2)} \mid n \in \mathbb{Z}_{\geq 0}\}$$

$$\left[\{ w \in \{a,b\}^* \mid |w| = n^2 \text{ for some } n \in \mathbb{Z}_{\geq 0} \} \right.$$

§1.4 (S, p)

Myhill-Nerode

$$L' = \{a^0 = \varepsilon, a^3, a^6, a^9, \ldots\}$$

$$L = \{a^3, a^6, a^9, \ldots\}$$

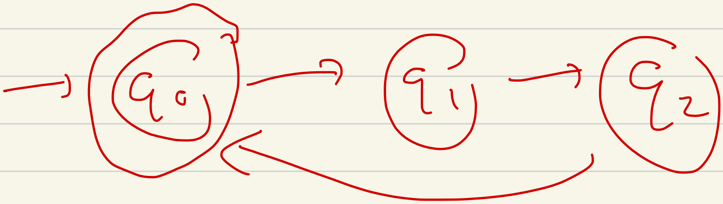
$$\mathrm{AccFut}_{L'}(a) = \mathrm{AccFut}_{L}(a)$$

$$\mathrm{AccFut}_{L'}(a^2) = \mathrm{AccFut}_{L}(a)$$

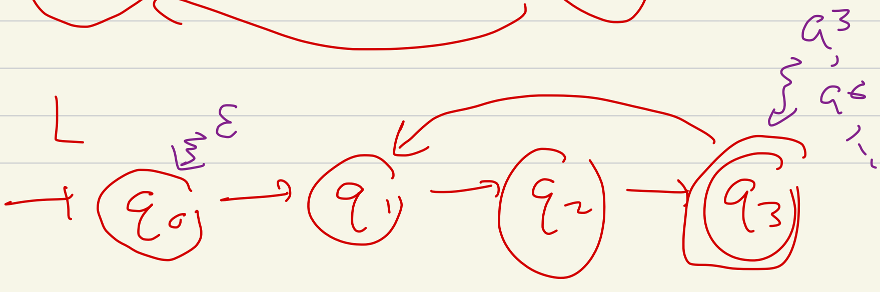
$$\mathrm{AccFut}_{L'}(\varepsilon) = L' = \{a^0, a^3, \ldots\}$$

$$\mathrm{AccFut}_{L'}(a^3) = \{a^0, a^3, \ldots\} = L'$$

DFA for L'



DFA for L



Non-regular examples:

$$L_{\text{zero then ones, but same number}} = \{ 0^n 1^n \mid n \in \mathbb{Z}_{\geq 0} \}$$

$$= \{ \varepsilon, 01, 0011, 000111, 0^4 1^4, 0^5 1^5, \ldots \}$$

$$\neq 0^* 1^*$$

also $\neq (01)^*$

Intuition: $000\_\_01$

?

what have we seen before???

Let's show that as ω varies over $\{0,1\}^*$,

$\text{AccFut}_{L_{\text{zero then ones}}}(\omega)$ has infinitely many possible values

$$\text{AccFut}_L(\varepsilon) = L = \{\varepsilon, 01, 0^2 1^2, \ldots\}$$

$$\text{AccFut}_L(0) = \{1, 011, 0^2 1^3, 0^3 1^4, \ldots\}$$

$$\text{AccFut}_L(00) = \{1^2, 01^3, 0^2 1^4, \ldots\}$$

$$\vdots$$

$$\text{AccFut}_L(0^k) = \{1^k, 01^{k+1}, 0^2 1^{k+2}, \ldots\}$$

$$\vdots$$

We need to argue that there are infinitely many possible sets...

Break 10:11 – 10:16

We know

$$\text{AccFut}_L(0^k) = \{1^k, 01^{k+1}, \ldots\}$$

claim that $k \neq k'$

$$\text{AccFut}_L(0^k) \neq \text{AccFut}_L(0^{k'})$$

=

Induction: Assume

$$\text{AccFut}_L(\varepsilon), \ldots, \text{AccFut}_L(0^k)$$

are all distinct, now we want to prove that

$$Accfut_L(\varepsilon), Accfut_L(0), \ldots,$$

$$Accfut_L(0^k), Accfut_L(0^{k+1})$$

are distinct

why: $Accfut_L(0^{k+1})$ contains 1^{k+1}

but $Accfut_L(0^0), \ldots, Accfut_L(0^k)$ don't.

~~Pf.~~ If $k \neq k'$

then $1^k \in Accfut_L(0^k)$

but $1^k \notin Accfut_L(0^{(k')})$

Or:

The shortest word in

$\text{AccFut}_L(0^k)$ is of length k

(namely 1^k)

A little trickier...

$C_4 = \{ w \in \{a,b\}^* \mid$ the 4th last letter of w is "a" $\}$

$C_2 =$ ... 2nd ...

C_2 describe $\{a,b\}^* a \{a,b\}$

$$\text{Accfut}_{C_2}(\varepsilon) = C_2 = \{ w \in \{a,b\}^* \mid \text{blah blah blah} \}$$

$$= \Sigma^* a \Sigma$$

$$\text{Accfut}_{C_2}(a) = \{a,b\} \cup C$$

$a \ \underline{\overset{a}{b}}$

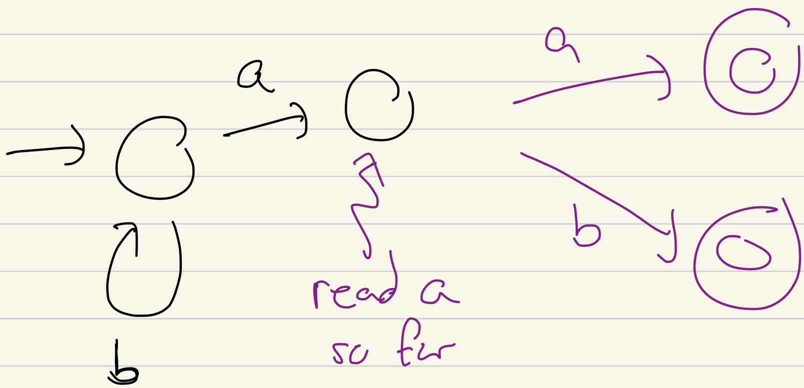
$a \ \_\ \_\ \_\ \ldots \ \underbrace{\_\ \underline{\overset{a}{b}}}_{\text{last 2}}$

$$\text{Accfut}_{C_2}(b) = C_2$$

Let's try a few more...

$\mathrm{AccFut}_L(\varepsilon) = \mathrm{AccFut}_L(b)$

Think about a DFA

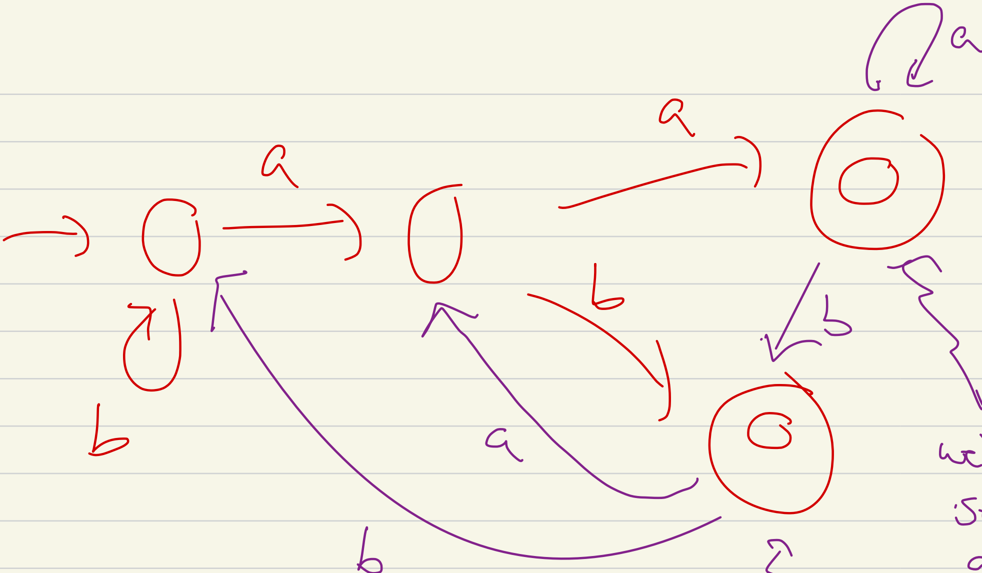


Can ab and aa take to same state?

$\mathrm{AccFut}_L(aa) = \{\varepsilon, a, b, \ldots\}$

$\mathrm{AccFut}_L(ab) = \{\varepsilon, \quad \}$

can't accept after b



we've
seen
aa

we've
seen
ab

what
is this?

aba

abb

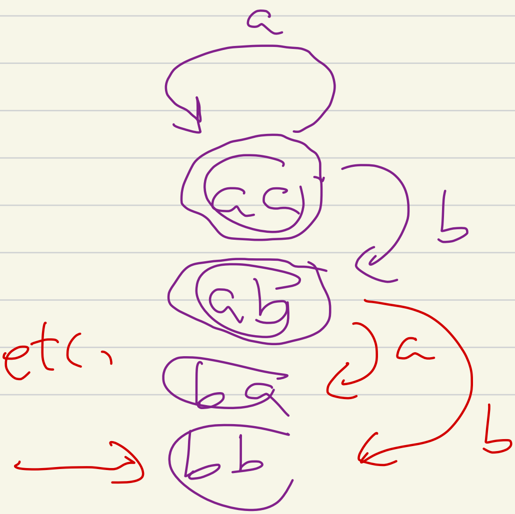
$C_{10} = \_\_\_\_$

Look more closely at

C_2, min # states = 4

think of

$\underbrace{\_\ \_\ \_\ \_\ \_\ \_\ \_\ \_}\ \underbrace{\_\ \_}_{\text{last 2 symbols}}$



- - - - aa a

— — ab b

- - ab { a
ab b

the words ε, a, b

$$\varepsilon \longleftrightarrow bb$$

$$\mathrm{Accfut}_L(\varepsilon) \stackrel{?}{=} \mathrm{Accfut}_L(bb)$$

so

bb initial state

C_{10} !

$$\left.\begin{array}{l} aa\ldots a \\ aa\ldots ab \\ \vdots \\ \underbrace{bb\ldots b}_{10 \text{ symbols}} \end{array}\right\}$$

claim: need 2^{10} states, and this will suffice

C_3:

aaa

aab

aba aba ......

abb

baa

bab

bba bba ......

bbb