

Announcements:

- Waitlist ~~~~~ 1: 0 ppl
2: ~28 ppl
3: ~6 ppl
 - Checkup #1 ~~~~~ last attempt: due next Tue night
grading based only on last attempt (checkup 2 and on: you must submit a first attempt on time).
we drop 2 worst checkups
 - HW 1
 - Handout 1
 - Checkup #2
- } soon (will email)
- How to succeed in this class? Try these 10 simple tricks...
 - recording → usually none.
 - lecture notes (in addition to these)
(for M/N thm, since we're deviating from textbook).
- See tips on class webpage under learning objectives.

Recap

DFA

$D = (Q, \Sigma, \delta, q_0, F)$

states alphabet start state set of accepting states
transition

$$L(D) = \{w \mid D \text{ accepts } w\}$$

$L \subseteq \Sigma^*$ is regular if $\exists$ DFA D s.t. $L(D) = L$

Thm: Every finite lang is regular

Thm: The class of regular lang is closed
under complement, intersection, union

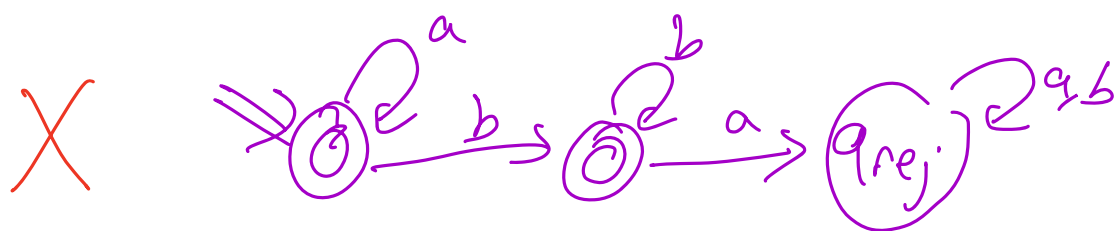
Saw examples of regular languages.

Another example:

$$L = \{ a^n b^n \mid n \geq 0 \} = \{ \epsilon, ab, aabb, aaabbb, \dots \}$$

i.e. $w \in L \Leftrightarrow w$ consists of a sequence of (0 or more) a's followed by the same number of b's

Attempts to show regular: $\rightarrow$ [I'm including one of the attempts suggested in section 1 and one suggested in section 2]



Does not work! (valid DFA but not for L)

Counter example: b is accepted but $b \notin L$

(in fact, this DFA recognizes $\{ a^i b^j \mid i, j \geq 0 \}$ - this is a different language! doesn't enforce equal number)

Another attempt:

X Express $L = \{ a^n \mid n \geq 0 \} \cap \{ b^n \mid n \geq 0 \}$
show that this L_1 and this L_2 are regular

Using closure property from last time

$\Rightarrow L_1 \cap L_2$ is also regular

Does not work!

While it's true that L_1, L_2 are regular, ^{→ can you show?}
and therefore, by closure, $L_1 \cap L_2$ is regular,
it is not true that $L_1 \cap L_2 = L$

$$\{a^n \mid n \geq 0\} \cap \{b^n \mid n \geq 0\} = \text{not } L \\ = \{\epsilon, a, aa, \dots\} \cap \{\epsilon, b, bb, \dots\} = \{\epsilon\}$$

(no string other than ϵ is both of the form $a \dots a$
and of the form $b \dots b$)

(Note: when writing $\{a^n \mid n \geq 0\} \cap \{b^n \mid n \geq 0\}$

not same n ! var is bound only inside $\{\}$.

It would be the same to write

$$\{a^i \mid i \geq 0\} \cap \{b^j \mid j \geq 0\}$$

It always mean $\{\epsilon, a, aa, \dots\} \cap \{\epsilon, b, bb, \dots\}$

Turns out there is no DFA for L
— not regular $\{a^n b^n \mid n \geq 0\}$

intuition: with fixed # of states, we
don't have enough memory to process
arbitrarily long strings — seems like we
need to count how many a's we've seen
so we can later compare to # b's.

But, how do we know how much memory
we need? maybe there's some clever algorithm?

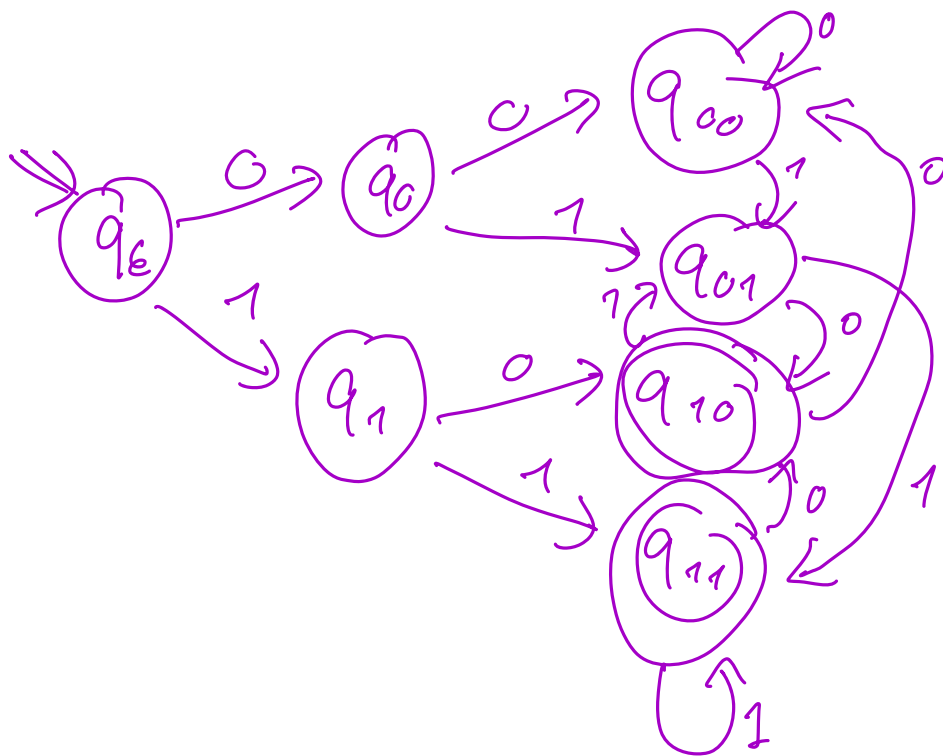
Let's go back to examples we've seen
in previous lectures:

these will be our
running examples for
today.

(from lecture 2:)

$$L = \{w \in \{0,1\}^* \mid |w| \geq 2, \text{ and second to last char in } w \text{ is } 1\}$$

We constructed this DFA for L :



→ does this mean we "need" 7 states?

No! in fact there is a smaller DFA recognizing L !

Try to construct it yourself!

(hint: combine states q_ϵ, q_0, q_{00} into one)

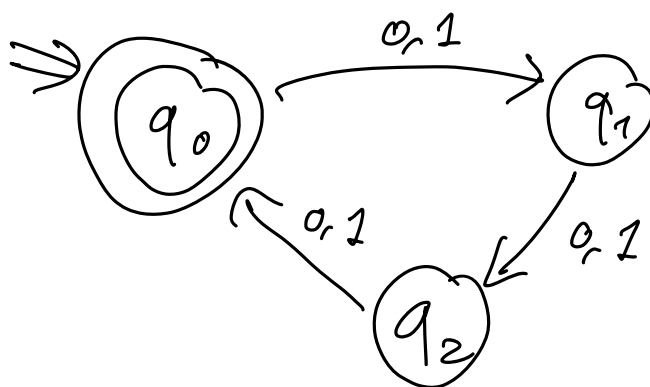
(you can also see solution in William's lecture notes)

(from lecture 1):

$$L = \{ x \in \{0,1\}^* \mid |x| \text{ is divisible by } 3 \}$$

this is the length of x

DFA for L :



is there a smaller DFA?

(say with 2 states?)

Today we'll see a theorem^{MIN} that tells you what is the minimum # states you need in a DFA for L , or if no finite # of states is sufficient and L is non-regular.

Def: Let $L \subseteq \Sigma^*$ be a language, $x, y \in \Sigma^*$

we say x and y are distinguishable by L

if $\exists z \in \Sigma^*$ s.t. exactly one of xz, yz is in L

$x \circ z$ (concatenation)

In this case we call z

a distinguishing extension for x, y (wrt L)

If x, y are not distinguishable by L , we say

they are indistinguishable by L , denoted $x \sim_L y$

i.e. 

$$\forall z \in \Sigma^* \quad xz \in L \iff yz \in L$$

Ex: $L = \{w \in \{0,1\}^* \mid |w| \geq 2, \text{ and second to last char in } w \text{ is } 1\}$

Are the following pairs distinguishable or not?

• take $x = \epsilon$, $y = 1$: distinguishable!

eg choose $z = 0$. $xz = 0 \notin L$

but $yz = 10 \in L$.

• take $x = \epsilon$, $y = 0$: indistinguishable!

[to show indistinguishability not enough to choose one z ,
need to show for all z]

In fact, for any $x = 0^i$ $y = 0^j$
 $x \not\sim_L y$ because including the case $x = \epsilon, y = 0$

$$\forall z \in \{0,1\}^* \quad xz = 0^i z \in L$$

$\Leftrightarrow$ the second to last char
in z is 1

$$\Leftrightarrow 0^j z \in L \Leftrightarrow yz \in L$$

Lemma 1: Let $L \subseteq \Sigma^*$ be a regular language,
and let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L .

If x, y are distinguishable by L , then
the computations of D on x and on y
must end in different states.

proof sketch: If x, y are distinguishable

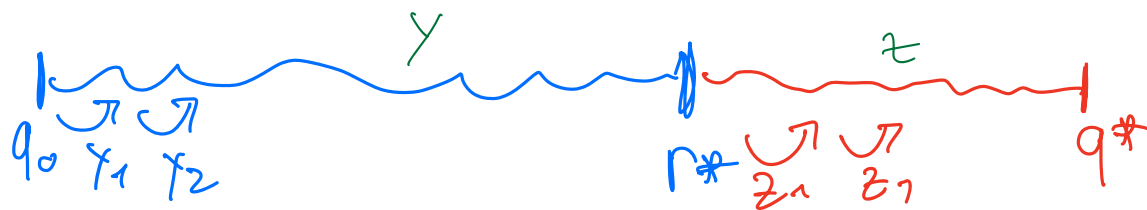
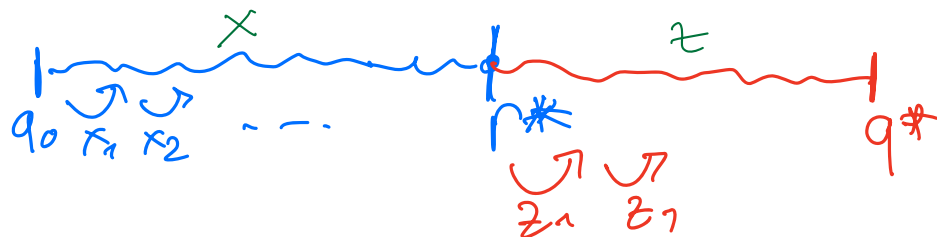
let z be a distinguishing extension for x, y .

Suppose towards contradiction that the computations of D on x and on y end in same state, r^* .

↙
 $(r_0, \dots, r_{|x|})$
 $\parallel$
 q_0 $\parallel$ r^*

↘
 $(r'_0, \dots, r'_{|y|})$
 $\parallel$
 q_0 $\parallel$ r^*

Consider the computations of D on xz and yz :



both reach the same r^* before processing z , so

they must end in same state, call it q^* .

But since exactly one of xz, yz is in L

(other one isn't), q^* must be accepting } contradiction
 q^* must be rejecting



Lemma 2: Let $L \subseteq \Sigma^*$ be a language.

Suppose $\exists$ a set of k strings $S = \{x_1, \dots, x_k\}$

st. any pair of strings x_i, x_j ($i \neq j$)

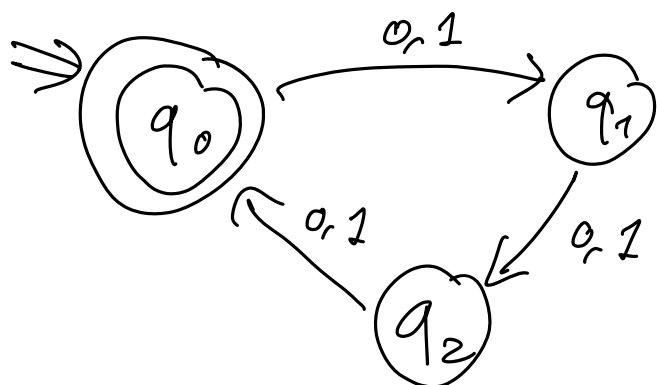
is distinguishable by L ,

then there does not exist a DFA
for L with $< k$ states

proof: follows from lemma 1
and pigeonhole principle. $\square$

Going back to our example:

$\{x \in \{0,1\}^* \mid |x| \text{ is divisible by } 3\}$



is there a smaller DFA? (say with 2 states?)

No! Here is a set S of 3 strings that are all pairwise distinguishable by L .

By lemma 2, can't have a 2-state DFA.

$$S = \{\epsilon, 0, 00\}$$

$\rightarrow z = \epsilon$ also works

For $x = \epsilon, y = 0$, take $z = 00$ ($xz = 00 \notin L$
 $yz = 000 \in L$)
 $x = \epsilon, y = 00$, take $z = 0$ ($xz = 0 \notin L$ $yz = 000 \in L$)
 $x = 0, y = 00$, take $z = \epsilon$ ($xz = 00 \notin L$
 $yz = 000 \in L$)

[many other z 's work too]

Claim: For any L , the indistinguishability relation $\approx$ is an equivalence relation

$\hookrightarrow$ i.e.; reflexive, symmetric, transitive.

proof: exercise.

Thus, $\approx$ partitions the set of strings into

equivalence classes $[x] = \{y \in \Sigma^* \mid x \approx y\}$

$\rightarrow$ If $x \approx y$ then $[x] = [y]$ is the same equivalence class.
These classes are disjoint and cover all of Σ^*

Myhill - Nerode thm: Let $L \subseteq \Sigma^*$ be a language.

L is regular if and only if

the number of equivalence classes for $\sim_L$ is finite.

Moreover, if L is regular, then the number of states in the smallest DFA recognizing L is the number of equivalence classes of L .

proof: next time.

Using MN thm, if you can determine what the equivalence classes of $\sim_L$ are, then you can tell whether or not L is regular, and if it is, you can construct the minimal DFA for L .

Our focus: use MN to prove a lang L is non-regular

To do this, you just need to prove that $\sim_L$ has infinitely many equivalence classes.

How? Come up with a set S s.t.:

- S has infinitely many strings.

- All strings in S are pairwise distinguishable:

$\forall x, y \in S (x \neq y), x$ and y are distinguishable by L .

→ "recipe" for proving non-regularity.

Going back to example we started with:

Claim:

$L = \{ a^n b^n \mid n \geq 0 \} = \{ \epsilon, ab, aabb, aaabbb, \dots \}$

is not regular

proof:

Let $S = \{ a^i \mid i \geq 0 \} = \{ \epsilon, a, aa, aaa, \dots \}$

Now:

- S has infinitely many strings

— For any two strings $x = a^i, y = a^j \in S$ where $i \neq j$, we show that x, y are distinguishable by L :

Let $z = b^i$.

Then $xz = a^i b^i \in L$

$yz = a^j b^i \notin L$ since $i \neq j$

$\Rightarrow \approx_L$ has infinitely many equivalence classes

$\Rightarrow$ By MN thm, L is not regular $\square$

More examples to think about
(prove both non-regular):

— $\{w \in \{a, b\}^* \mid w \text{ is a palindrome}\}$

$w = w^R$
i.e. if we reverse
the order of
symbols in w ,
we still get w

— $\{w \in \{a\}^* \mid w = a^{2^i} \quad i \geq 0\}$

We'll show next time

