Announcements

We will post announcements on webpage while waitlist is enabled, so you don't have to have access to courseworks

- First checkup out today. Due date:

  First attempt: Tue Sep 15 before midnight

  Final attempt: Tue Sep 22 " "

5% of grade
two worst checkups dropped

First attempt should be individual without seeking help

- Handout with Discrete math review will be added soon.

- office hours start next week - see webpage for calendar

- Tentative dates for in-class quizzes:

  Quiz 1: Tue Oct 6 (or Thur Oct 8)

  Quiz 2: Thur Oct 22

  Quiz 3: Tue Nov 17

  Quiz 4: Tue Dec 8

- should attend in your official section, unless taking with DS/CARDS
- Let us know right away if you foresee a problem

Recap from last time:

Defined alphabet, strings ...

A language L over alphabet Σ is a set of strings

$$L \subseteq \Sigma^*$$

Each language corresponds to a function $f: \Sigma^* \to \{0,1\}$

where $x \in L \iff f(x) = 1$

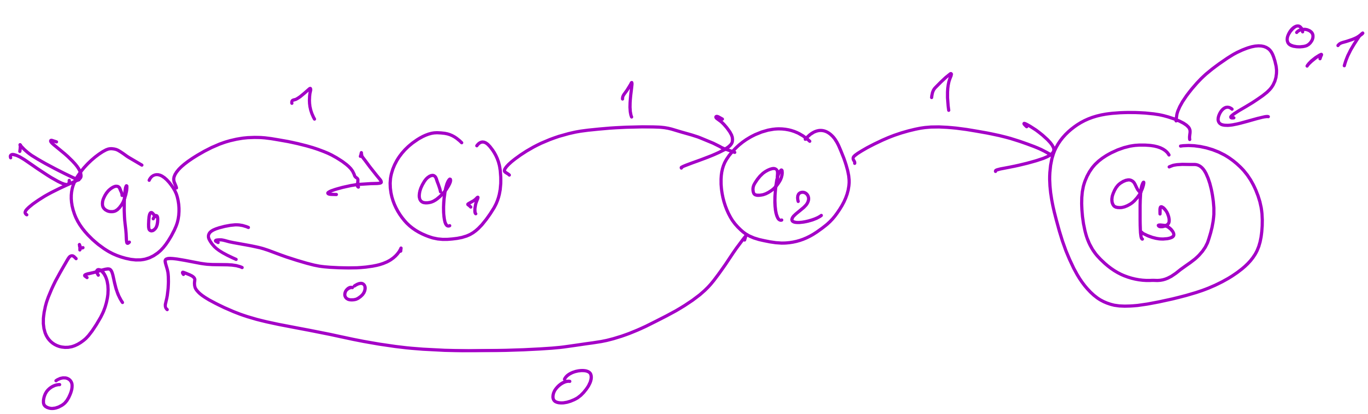
Today: our first model of computation

Deterministic Finite Automaton (DFA)

Informally: DFA is a "computer with very little memory" which can only read its input left-to-right then decide on output (accept/reject)

Example:

M:



- Each DFA has finitely many states (◯)
- For every state and alphabet symbol there's a transition (◯→)
- Optionally some of the states are marked ◎ – "accept states"
- there is one state marked ⇒ (start state)

Computation of above DFA M on 01101:

$$q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_0 \xrightarrow{1} q_1$$

i.e. $(q_0, q_0, q_1, q_2, q_0, q_1)$

Since q_1 is not an accepting state, 01101 is rejected by M

Computation on 01110:

$$q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{0} q_3$$

Since q_3 is accepting, 01110 is accepted

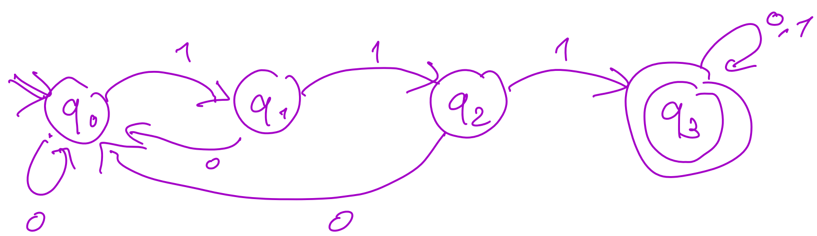
Def: A DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$

where:

- Q is a finite set of states
- Σ is an alphabet
- $\delta : Q \times \Sigma \to Q$

  transition function: takes (state, letter) output next state
- $q_0 \in Q$ called "start state"
- $F \subseteq Q$ the set of accepting states

Example above is:

M: 

$$M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$$

where δ:

	0	1
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_0	q_3
q_3	q_3	q

Def: for a DFA $M = (Q, \Sigma, \delta, q_0, F)$ and $w \in \Sigma^*$ where $|w| = n$, $w = w_1 \dots w_n$ (each $w_i \in \Sigma$)

the computation of M on w is a sequence of states $(r_0, \dots, r_n)$ s.t.

- $r_0 = q_0$ → (start with start state)
- $\forall i \in \{1, \dots, n\}$ $\delta(r_{i-1}, w_i) = r_i$

(every transition is correct)

A computation $(r_0, \dots, r_n)$ is accepting if $r_n \in F$.

M accepts w iff the computation of M on w is accepting (ends in an accepting state).

(if M does not accept w, we say M rejects w)

Note:
- For any DFA M and string $w \in \Sigma^*$ there's one computation of M on w
- The computation of a DFA M on ε (empty string) is (q_0). M accepts ε iff $q_0 \in F$

→ [this came up in section 2, havent explicitely mentioned it in section 1.]

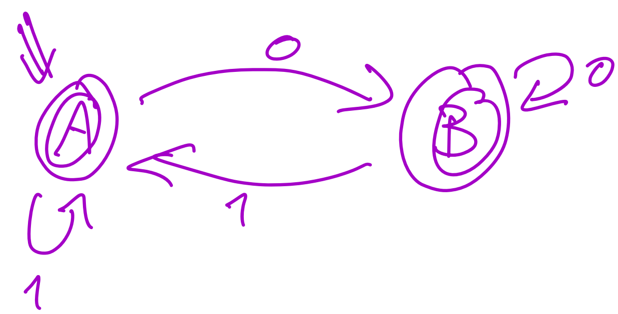
Def: for a DFA M, we define the language recognized by M, as

$$L(M) := \{ w \in \Sigma^* \mid M \text{ accepts } w \}$$

Def: a language L is regular if there exists a DFA M s.t. $L(M) = L$

Examples:

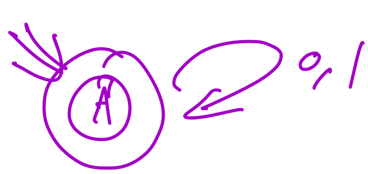
- M:



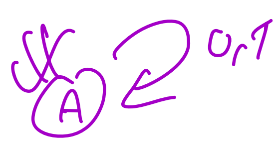
$\Sigma = \{0,1\}$

$L(M) = \Sigma^*$

A simpler DFA for same lang Σ^*:

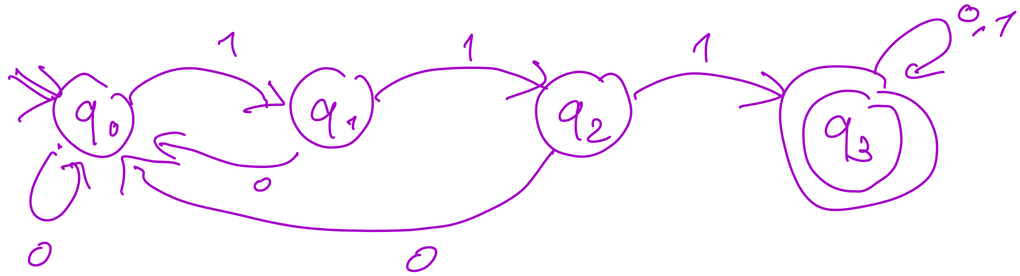


- DFA for $\emptyset$: eg



• previous example:

M:



What's $L(M)$?

$$\boxed{L(M) = \{w \in \{0,1\}^* \mid w \text{ contains three 1's in a row}\}}$$

why?

- any string with 111 is accepted because from any state, 111 takes you to q_3, and once you're in q_3 you stay in q_3
- any string without 111 substring is rejected because from any state other than q_3, 0 takes you to q_0, and from q_0 you cannot get to q_3 without 111.

In general, to show $L(M) = L$ need to show

- every string in L is accepted by M
- every string not in L is rejected by M

or equivalently:

- every string that is accepted by M is in L

We've seen so far that Σ^*, $\emptyset$, $\{w \mid w \text{ has } 111 \text{ as a substring}\}$ are <u>regular</u> languages.

Are all languages regular??

(no... we'll see later)

think about how you would prove that some language is not regular?

• $L = \{ s \in \{0,1\}^* \mid s \text{ has an even \# of 0's} \}$

this is an example from last time!

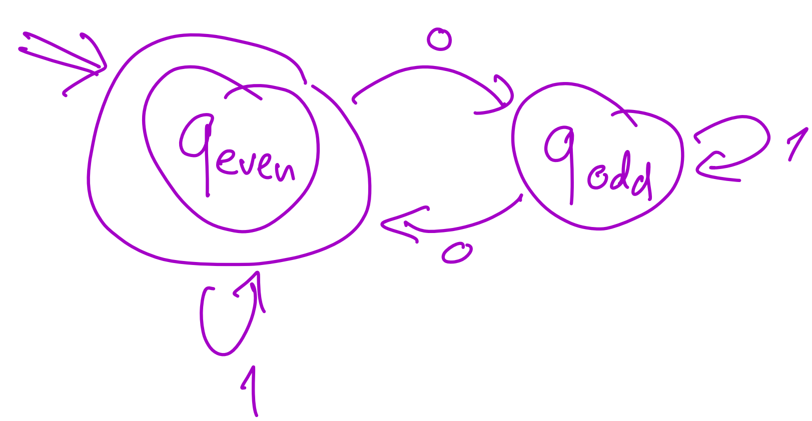
we saw an algorithm for it:

Algorithm: On input s:
- go over s, count how many 0's it has
- if this number is 0 mod 2 output yes else output no

Seems hard (impossible?) to implement this alg. in a DFA, because how could you count arbitrarily high with a finite # of state?

But, there are other algorithms for same L!

M:



Corresponds to following algorithm:
- keep the parity of the # of 0's seen so far (even/odd) everytime you see 0, flip.

For this DFA M, $L(M) = \underbrace{\{s \mid s \text{ has an even \# of 0}\}}_{L}$

So this is also a regular language.

More examples of regular languages

(can you show it?)

1. $\{x \in \{0,1\}^* \mid x \text{ ends in } 111\}$
2. $\{x \in \{0,1\}^* \mid x \text{ ends in } 101\}$
3. $\{x \in \{0,1\}^* \mid |x| \text{ is divisible by } 3\}$
4. $\{x \in \{0,1\}^* \mid x \text{ contains no 0's}\}$
5. $\{x \in \{0,1\}^* \mid x \text{ starts and ends with 0 and has even length}\}$

You should try on your own but solutions are on the next page (added after class).

Challenge: is the following regular?

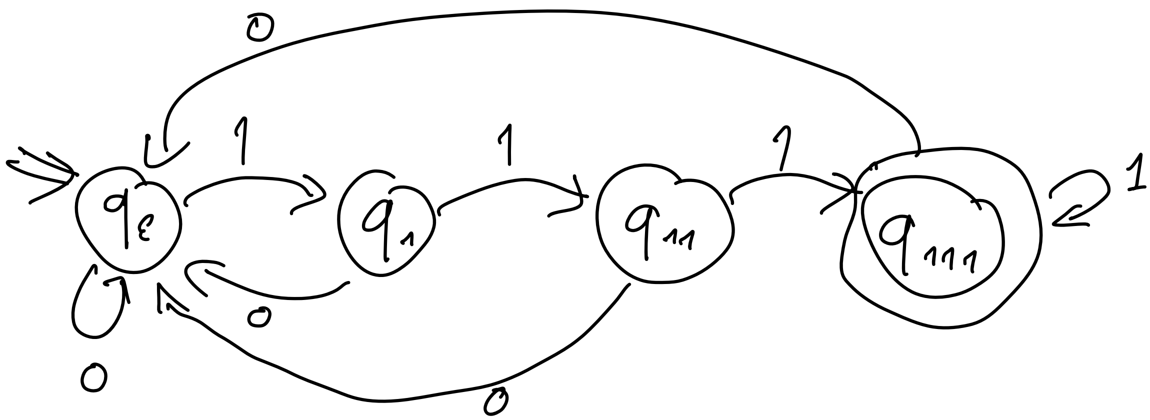
$$L = \{ w \in \{0,1\}^* \mid w = x \circ y \quad \begin{array}{l} x \text{ has even \# of 1's} \\ y \text{ has even \# of 0's} \end{array} \}$$

Solutions to examples:

→ (there are other good solutions too)

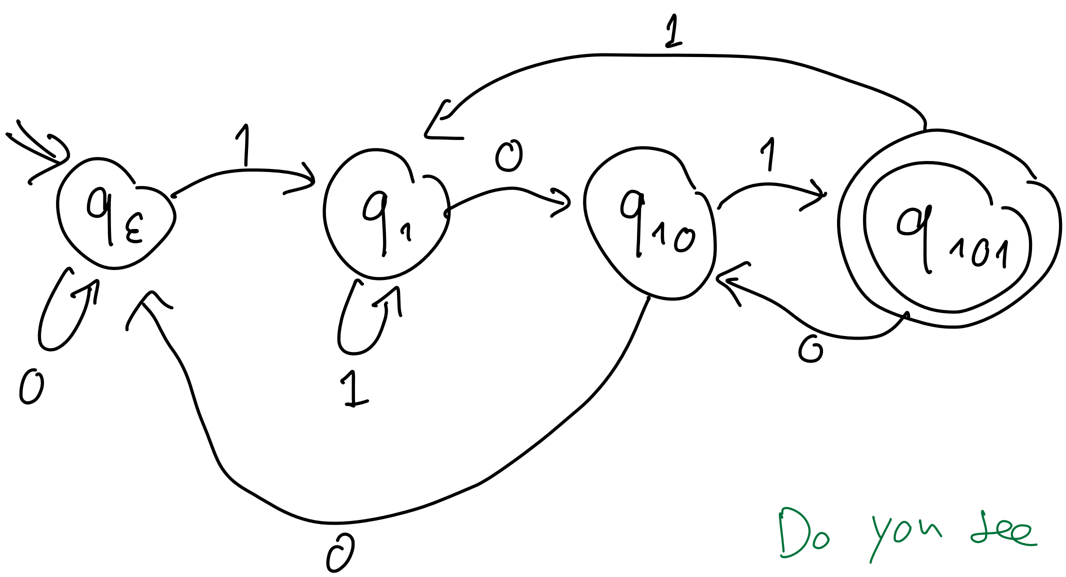
1. $\{x \in \{0,1\}^* \mid x \text{ ends in } 111\}$

Idea: use states to keep track of the longest prefix of 111 we've seen so far (if there's a 0, reset)



2. $\{x \in \{0,1\}^* \mid x \text{ ends in } 101\}$

Same idea! different transitions for 101



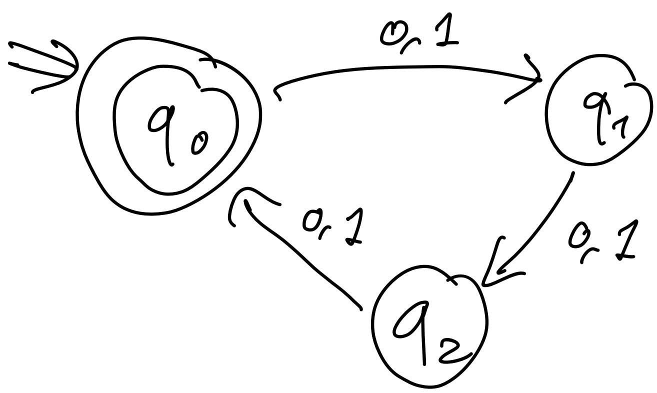
Do you see why?

(maybe easier to try yourself!)

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

this is the length of x

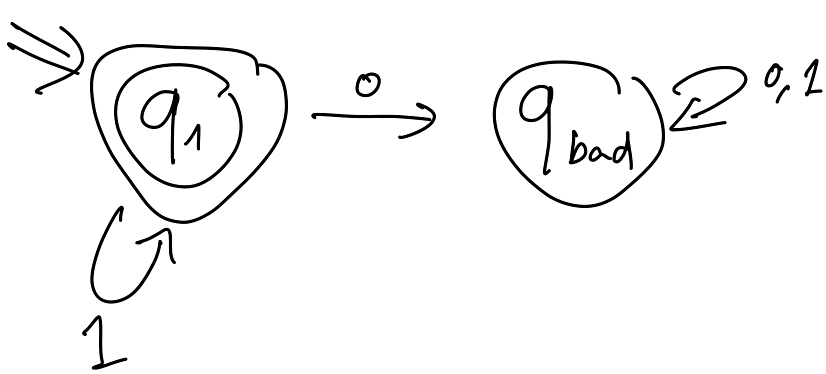
Idea: state will keep track of (length so far) mod 3



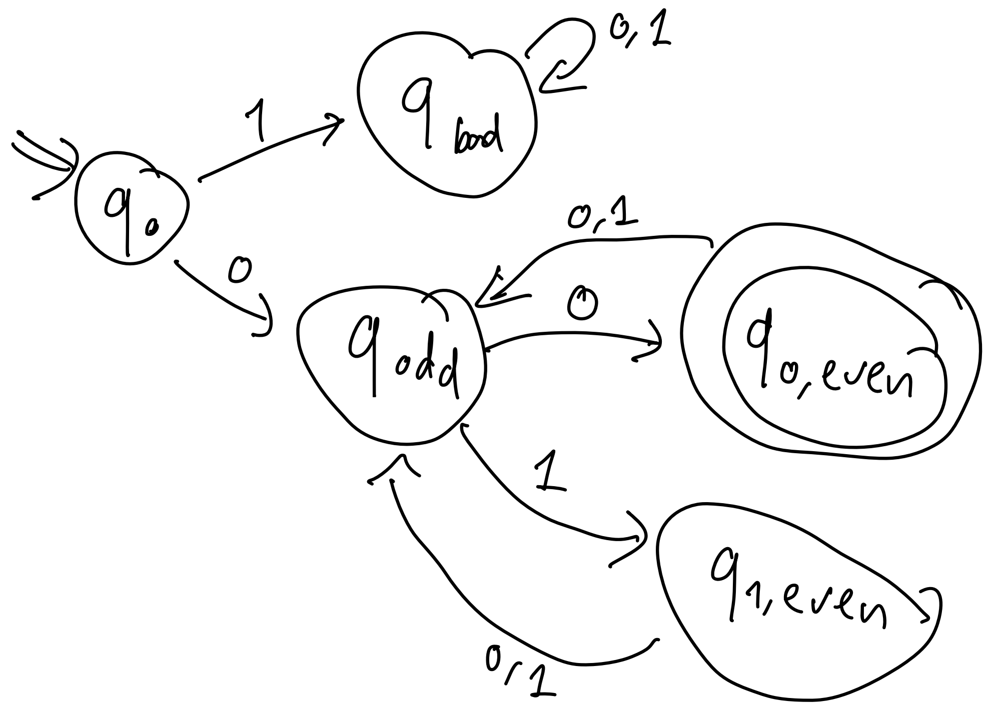
test yourself: Can you show a DFA for

$\{x \mid |x| \text{ is } \underline{\text{not}} \text{ divisible by } 3\}$?

4. $\{x \in \{0,1\}^* \mid x \text{ contains no } 0\text{'s}\}$



5. $\{x \in \{0,1\}^* \mid x \text{ starts and ends with } 0 \text{ and has even length}\}$



Explanation:

- all strings that start with 1 go to q_{bad}
- all strings that start with 0 and have odd length, go to q_{odd}.
- strings that start with 0 and have even length:
  - if end in 1, go to $q_{1,even}$
  - if end in 0, go to $q_{0,even}$

the only accepting state.