Lecture 3 : Today ~ the rest of chapter 1.1

Logistics: Checkup #1 due tonight
(attempt 1)

OHs start today => calendar is on the webpage

near Talt
my office (CSB 501 ?

in CSB, go in direction of the lounge.

=> go down the halway & up the stairs

=> an open area with whiteboards

Next lecture : Myhill Nerode => not in the book

See Handout 3 on the 2024 webpage if you want to read ahead our first model of computation!

Last time DFAs:

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

start state

FINITE set of states

alphabet

transition function

set of accept states (can be empty)

Some definitions:

The computation of a DFA M on a string $w = w_1 \cdots w_n$ is a sequence of states $(r_0, r_1, r_2, \ldots, r_n)$ such that : $r_0 = q_0$

$$\delta(r_{i-1}, w_i) = r_i$$

If $r_n \in F$ we say this is an accepting computation

In this case we say "M accepts w"

(else we say "M rejects w")

Note: for a DFA M and a string w, there is a unique computation of M on w

The language recognized by a DFA M is

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

A language A is regular if $\exists$ a DFA that recognizes it

(ie, A is regular $\Leftrightarrow$ $\exists$ a DFA M s.t $L(M) = A$)

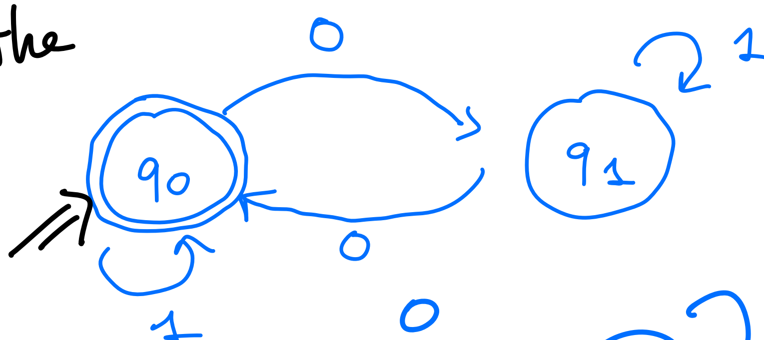
Intuition: you have a finite amount of memory and you can use it process arbitrarily long inputs & determine if they are in L

You can specify a DFA as

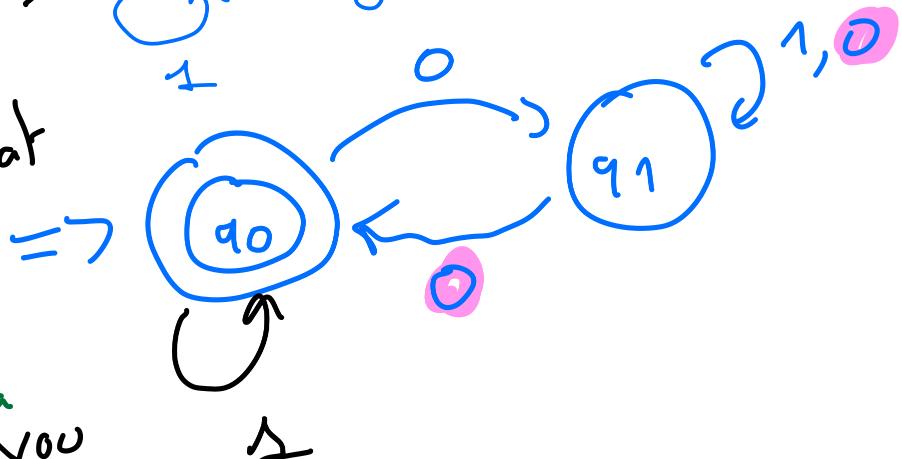
* a 5 tuple $(Q, \Sigma, \delta, q_0, F)$
* a state diagram

Be careful with state diagrams!

(1) make sure the start state is clear



(2) make sure that for all states q and symbol σ (sigma), you have a transition (exactly one)



Σ, σ

Some examples to try yourself:

$L = \{ x \in \{0,1\}^* \mid x \text{ ends in } 101 \}$

↳ in class we did "ends in 111"

$L = \{ x \in \{0,1\}^* \mid x$ starts and ends with a 0 AND x has even length $\}$

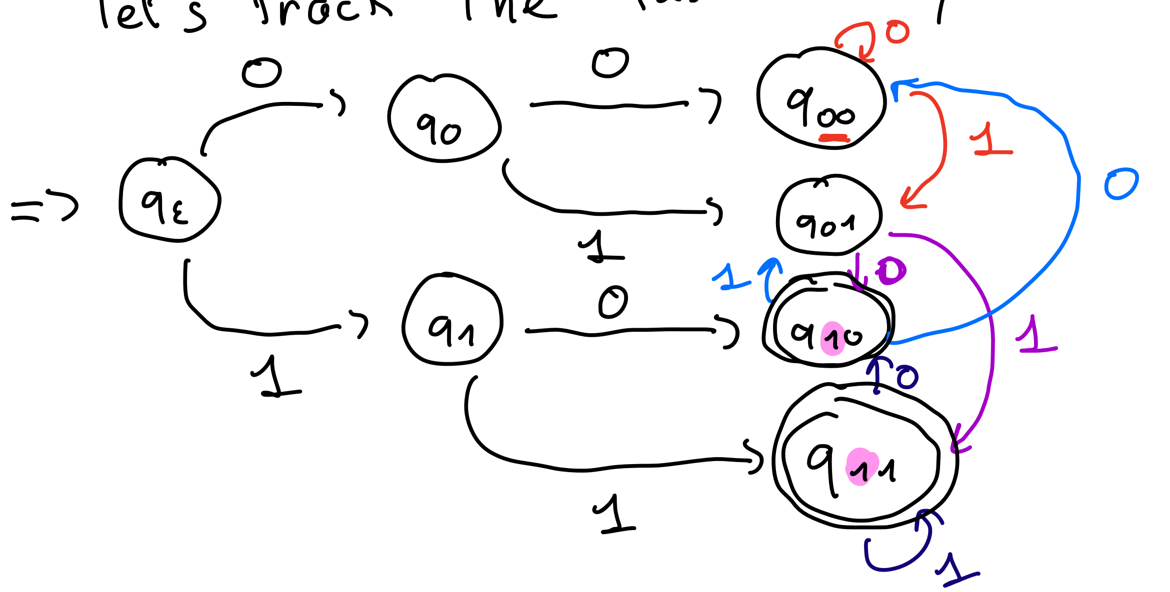
$L = \{ x \in \{0,1\}^* \mid |x|$ is NOT divisible by $3 \}$

Book: 1.1 – 1.6 are pretty good

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

$001 \notin L \qquad 110011 \in L$

let's track the last 2 symbols

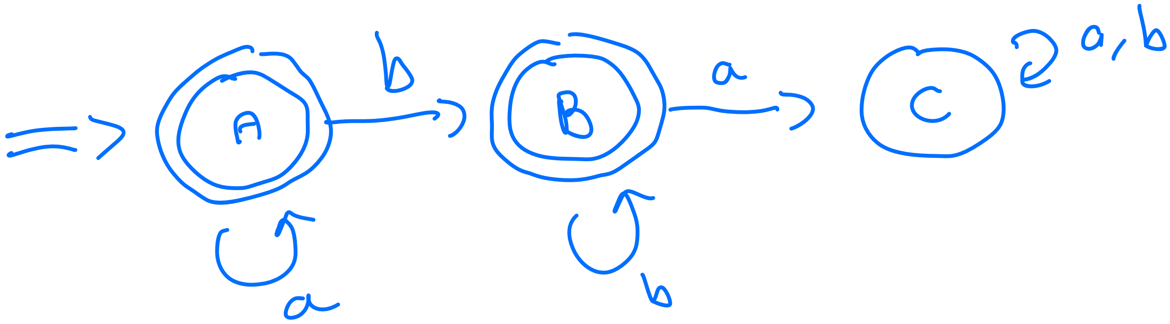


$L = \{ a^i b^j \mid i, j \geq 0 \}$ (so $\Sigma = \{a, b\}$)

$a^i = \underbrace{a\,a \ldots a}_{i \text{ times}}$, $a^0 = \varepsilon$

$a^i b^j = \underbrace{a \ldots a}_{\geq 0} \underbrace{b \ldots b}_{\geq 0}$

$bba \notin L$ $\qquad$ $\underbrace{bbb}_{a^0 b^3} \in L$ $\qquad$ $a^0 b^0 = \varepsilon \in L$



while at A: string so far is of the form a^i, $i \geq 0$

while at B: the string so far is $a^i \underline{b} b^j$ $\quad i, j \geq 0$

you took $A \xrightarrow{b} B$

c is for bad orderings => you've seen "ba"

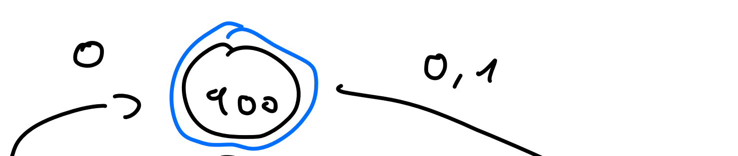
Our first theorem: Let $L \subseteq \Sigma^*$ be finite, then L is regular

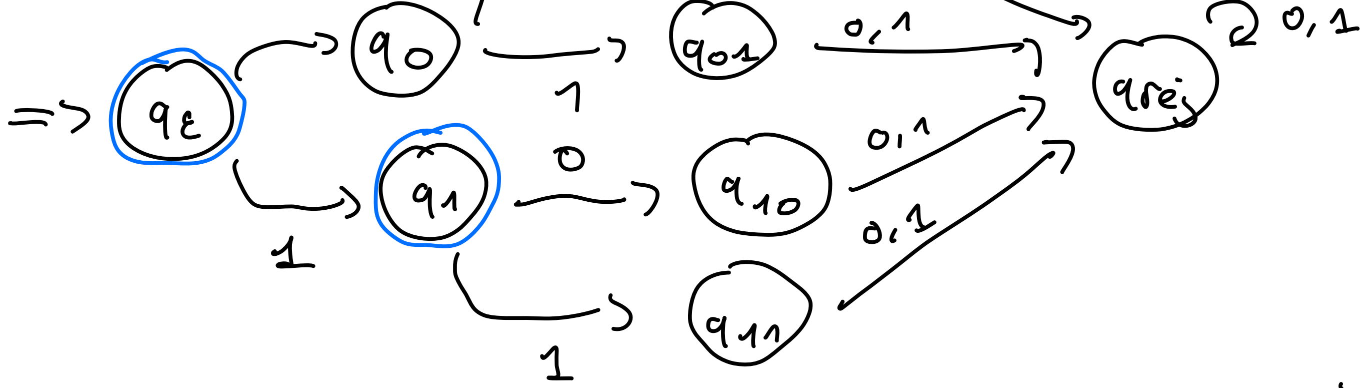
proof: Let L be a finite language. There is a number m such that $\forall w \in L$, $|w| \leq m$

We build a DFA where:

* we have a state q_w for all strings w with $|w| \leq m$
* an extra state for strings that have length $> m$

$L = \{ \varepsilon, 00, 1 \}$





$\delta(q_w, b) = q_{w.b}$ if $|w| < m$ else $\delta(q_w, b) = q_{rej}$

The accept states of the DFA are q_w where $w \in L$.

The converse "if L is regular, then L is finite"

False: ex: $\{w \mid |w| \text{ is } 0 \bmod 3\}$
we saw this is regular

Closure properties:

For languages L_1, L_2 over Σ:

complement: $\overline{L_1} = \Sigma^* \setminus L_1 = \{w \in \Sigma^* \mid w \notin L_1\}$

union: $L_1 \cup L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ OR } w \in L_2\}$

intersection: $L_1 \cap L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ AND } w \in L_2\}$

Theorem: the class of regular languages is closed under: complement, union, intersection

If L is regular $\Rightarrow \overline{L}$ is a regular language

If L_1, L_2 are regular $\Rightarrow L_1 \cup L_2$ and $L_1 \cap L_2$ are regular

Proof: complement. Let L be a regular language, we want to show $\overline{L}$ is regular.

Since L is regular there exists a DFA $D = (Q, \Sigma, q_0, \delta, F)$ with $L(D) = L$. Idea: flip the accept & reject state

$$D' = (Q, \Sigma, q_0, \delta, \underbrace{Q \setminus F}_{})$$

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

q is accept state of D' $\Leftrightarrow$ q is a reject state of D

Need to prove: $L(D') = \overline{L}$

$x \in \overline{L} \Leftrightarrow x \notin L \Leftrightarrow$ D does not accept x

$\Leftrightarrow$ the computation of D on x ends in $q \notin F$

The start state and transition functions of D and D' are the same : the computation of D on x is the same as D' on x

$\Leftrightarrow$ the computation of D' on x ends in $q \in \underbrace{Q \setminus F}_{\text{accept states of } D'}$

$\Leftrightarrow$ D' accepts x

so $L(D') = \{x \mid x \notin L\} = \overline{L}$ $\square$

Intersection: Let L_1, L_2 be regular languages over Σ. Want to show $L_1 \cap L_2$ is regular.

Let $D_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ be a DFA for L_1

Let $D_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be a DFA for L_2

Idea: run D_1 and D_2 in parallel and accept if and only if they both accept

We let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA where:

- $Q = Q_1 \times Q_2 = \{\underbrace{(r_1, r_2)}_{\text{pair of states one from } D_1 \text{, one from } D_2} \mid r_1 \in Q_1, r_2 \in Q_2\}$
- $q_0 = (q_1, q_2)$
- $F = F_1 \times F_2 = \{(r_1, r_2) \mid r_1 \in F_1, r_2 \in F_2\}$

- $\forall (r_1, r_2) \in Q$, $a \in \Sigma$

$$\delta(\underbrace{(r_1, r_2)}_{\text{a state of } D}, a) = (\underbrace{\delta_1(r_1, a)}_{\text{apply transition from } D_1 \text{ on } r_1}, \delta_2(r_2, a))$$

apply transition from D_2 on r_2

Why does that work: let $w = w_1 \dots w_n \in \Sigma^*$

q_1

Let $(r_0, \dots, r_n)$ be the computation of D_1 on w

$\delta_1(r_0, w_1)$

Let $(r_0', \dots, r_n')$ be the computation of D_2 on w

q_2 $\delta_2(r_0', w_1)$

What's the computation of D on w?

$(r_0, r_0') = (q_1, q_2)$

$$\delta((r_0, r_0'), w_1) = (\delta_1(r_0, w_1), \delta_2(r_0', w_1))$$
$$= (r_1, r_1')$$

So computation is $(r_0, r_0'), (r_1, r_1'), \dots, (r_n, r_n')$

So $w \in L_1 \cap L_2 \iff D_1$ accepts w and D_2 accepts w

$\iff r_n \in F_1, \ r_n' \in F_2$

$\iff (r_n, r_n') \in F_1 \times F_2$

$\iff (r_n, r_n') \in F$

$\iff D$ accepts w

Union: proof 1: L_1, L_2 are regular,

want to show $L_1 \cup L_2$ is regular

in the intersection proof

$\Rightarrow$ replace F by $\{(r_1, r_2) \mid r_1 \in F_1$ or $r_2 \in F_2\}$

before was AND

The rest of the DFA stays the same

Other proof use De Morgan's law

Let L_1, L_2 be regular.

We have $L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$

we know L_1, L_2 are regular so

complement

$\overline{L_1}, \overline{L_2}$ are regular

intersection

so $\overline{L_1} \cap \overline{L_2}$ is regular

complement

so $\overline{\overline{L_1} \cap \overline{L_2}}$ is regular

$L = \{ x \in \{0,1\}^* \mid x \text{ ends in } 101 \}$

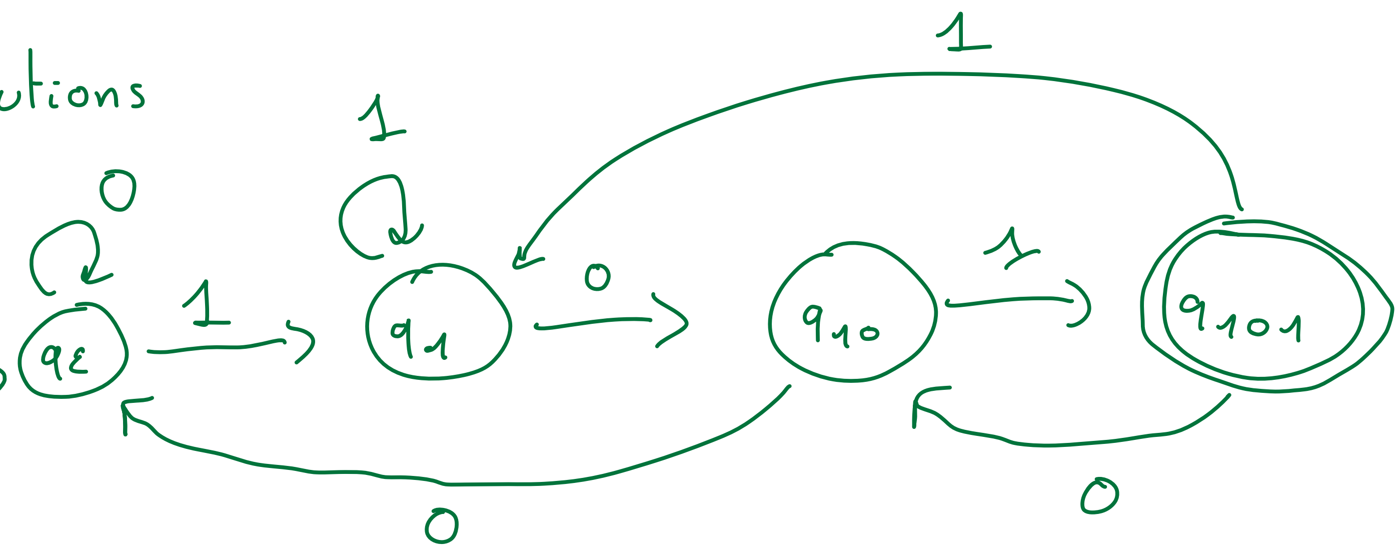
↳ in class we did "ends in 111"

$L = \{ x \in \{0,1\}^* \mid x \text{ starts and ends with a } 0$ AND $x \text{ has even length} \}$

$L = \{ x \in \{0,1\}^* \mid |x| \text{ is NOT divisible by } 3 \}$

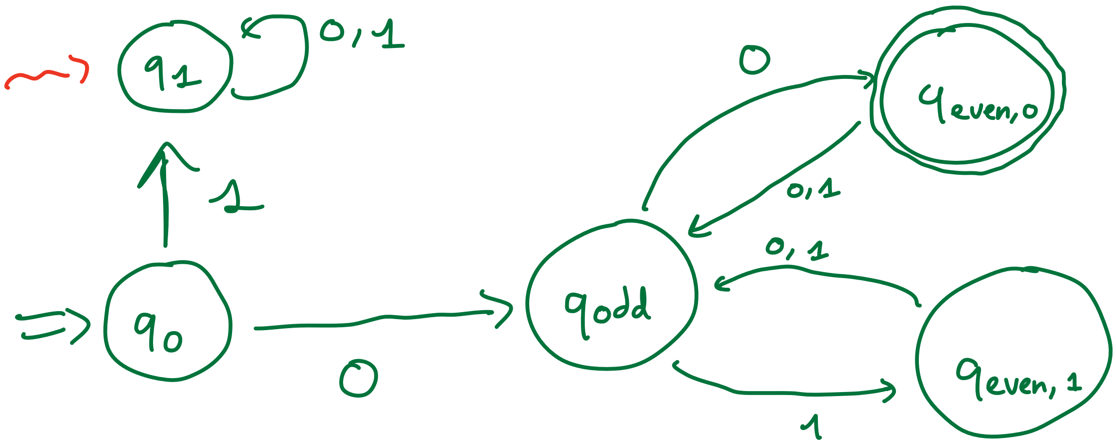
Solutions

(1)

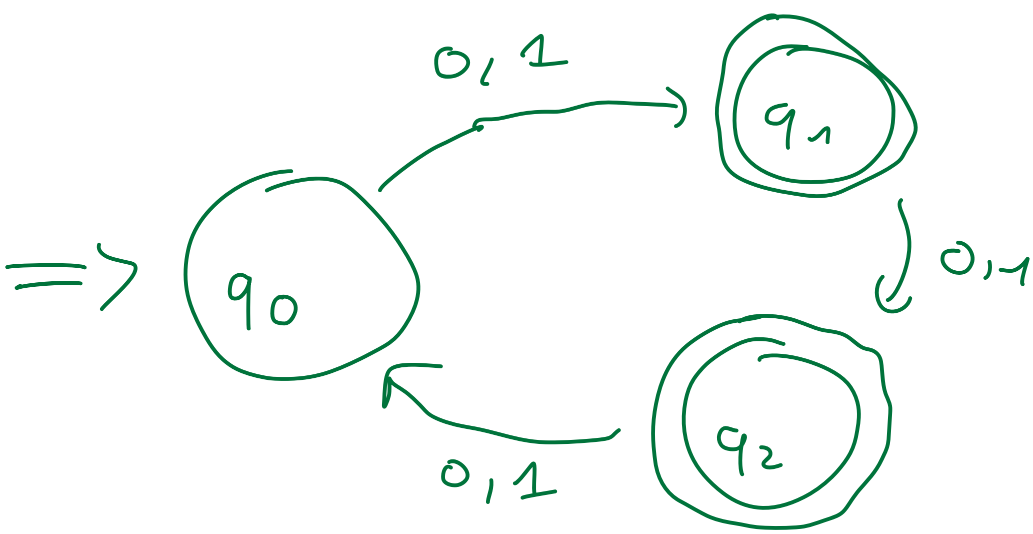


Idea: track how much the last 3 symbols match 101 start with 1

(2)



(3)



look at the DFA for

$L = \{ w \in \{0,1\}^* \mid w \text{ has length divisible by } 3 \}$

lecture 2

we simply "switched" accept & reject state