Week 9: Equivalence of Regular Expressions, DFA, and NFA

CSC 236:Introduction to the Theory of Computation

Summer 2024

Instructor: Lily

Announcement

- Got suggestion for review topics? Discuss on Piazza!
  - Proof of correctness
  - Structural induction
- Final exam schedule is coming out soon:
  - Number of questions: 8 + bonus
  - Q1 + Q2: T/F, multiple choice, short answer
  - Q3, 4, 5, 6, 7, 8: 20% “I don’t know” policy
  - Aid: two-sided A4 sheet
  - Duration: 3 hours
- Tutorials this week: regular expressions practice

Review

- A **language** L over an alphabet Σ is a subset of Σ^*.
- **Regular expressions** describes languages. **Regular languages** are sets of strings which can be *represented* by a regular expression.
- **Finite automatons** are collections of states and transition rules upon those states base on input characters (and ϵ in the case of NFA) in Σ**.**

DFA	NFA
• Each state ρ has exactly one transition for each symbol $a \in \Sigma$ • ϵ-transitions *not* allowed	• ρ has any number of transitions on a symbol $a \in \Sigma$ • ϵ-transitions *are* allowed

Question: DFA vs NFA, which is more powerful?

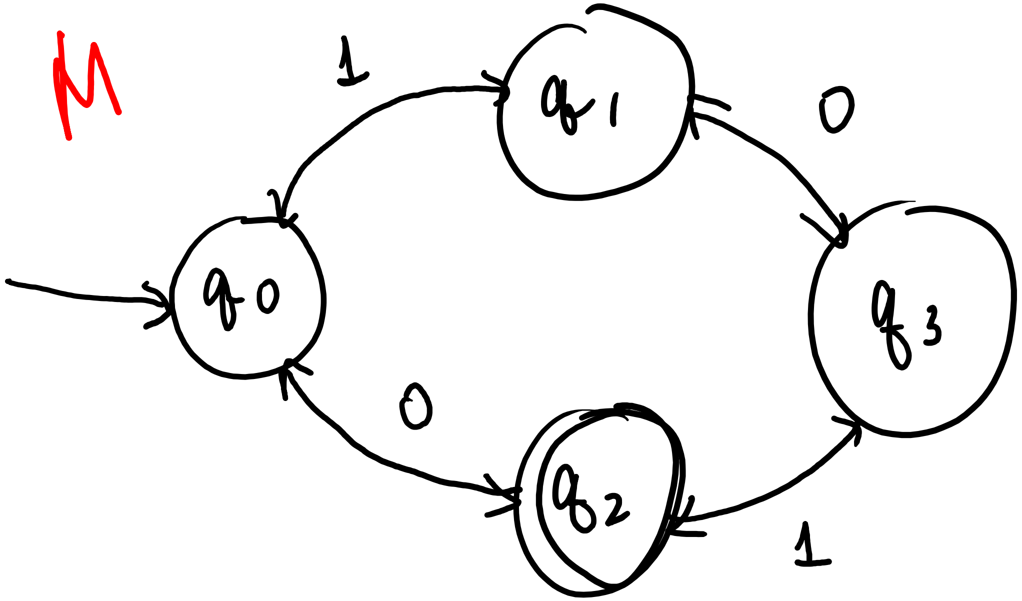
Finite Automaton Correctness

Consider

$$L = \{w \in \{0,1\}^* : w \text{ has odd number 0s and even number 1s}\}$$

Construct DFA M such that $\mathcal{L}(M) = L$.

M



Q: How do we know $\mathcal{L}(M) = L$?

A: construct disjoint & exhaustive set of state invariants, one for each state of $q \in Q$

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

states, chars, transition, start, accept

Proving Correctness of DFA

transition function δ

$\delta^*(q_0, w) = \delta(q_{n-1}, w_{n-1})$

$[w = w_0 \dots w_{n-1}]$ $w_i \in \Sigma$

$q_i = \delta(q_{i-1}, w_{i-1})$ $\forall i \in [n-1]$

For DFA $M = (Q, \Sigma, \delta, s, F)$, construct *disjoint* and *exhaustive* **state invariant** for each state $q \in Q$.

intuitive : $\delta^*(q_0, w)$ is state we are in if start at q_0 and execute δ on w.

structural induction on Σ^*

$\varepsilon \in \Sigma^*$

$a \in \Sigma^*$ for all $a \in \Sigma$

$w \in \Sigma^*$, then $wa \in \Sigma^*$ for all $a \in \Sigma$

$\delta(q_0, w_0) = q_1$

$\delta(q_1, w_1) = q_2$

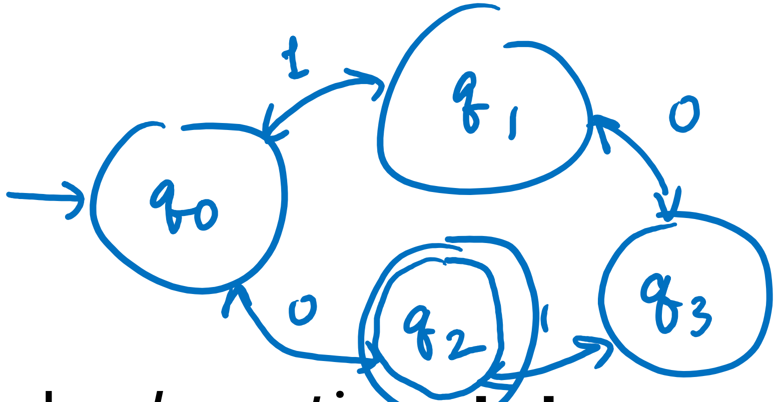
$\vdots$

$\delta(q_{n-1}, w_{n-1}) =$ A state

Induct on: Σ^* set of all strings

$\delta^*(q_0, w)$://

Proving Correctness of DFA

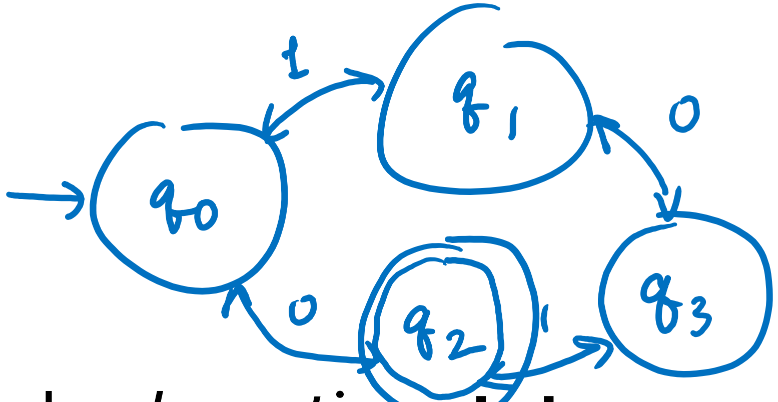


For DFA $M = (Q, \Sigma, \delta, s, F)$, construct *disjoint* and *exhaustive* **state invariant** for each state $q \in Q$.

$$P(w) := \delta^*(q_0, w) = \begin{cases} q_0 & w \text{ has an even \# of 0s and 1s} \\ q_1 & \text{" " " even \# of 0s and an odd \# of 1s} \\ q_2 & \text{" " " odd \# of 0s and an even \# of 1s} \\ q_3 & \text{" " " odd \# of 0s and 1s.} \end{cases}$$

Base Case: ε: $\delta^*(q_0, \varepsilon) = q_0$ satisfies condition of q_0.

Proving Correctness of DFA



For DFA $M = (Q, \Sigma, \delta, s, F)$, construct *disjoint* and *exhaustive* **state invariant** for each state $q \in Q$.

INDUCTIVE STEP, $w \in \Sigma^*$ and $P(w)$ true

Case $a = 0$, string $w0$

$\delta^*(q_0, w)$ START STATE of w IH

i) $= q_0$ w has even # of 0s and 1s.

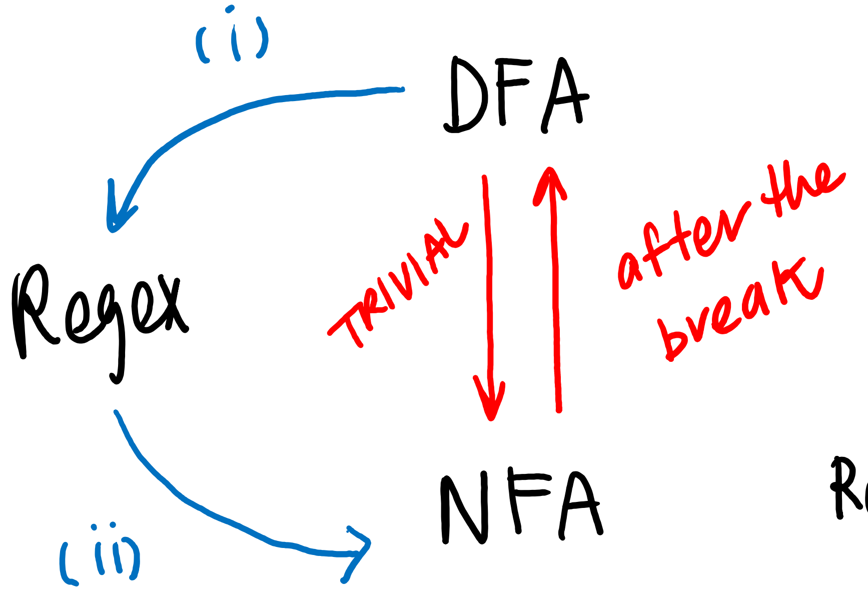
$\Rightarrow$ $w0$ has an even # of 1s and an odd # of zeros.

$w0$ goes to q_2 by DFA

q_0	w has an even # of 0s and 1s
q_1	" " " even # of 0s and an odd # of 1s
q_2	" " " odd # of 0s and an even # of 1s
q_3	" " " odd # of 0s and 1s.

Equivalence

Want to show: Regular expression, DFA, and NFAs have the same expressive power.



(i) and (ii) on Assignment 6.

Regular Language:
Language accepted by finite automaton (DFA or NFA)

Application

$\mathcal{P}$(Regular languages) → Regular Languages → strings

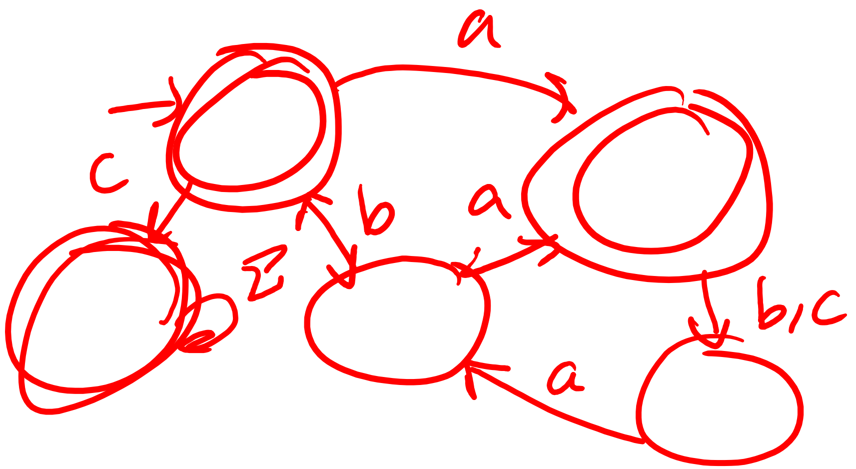
contains $\{\varepsilon\}$ contains.

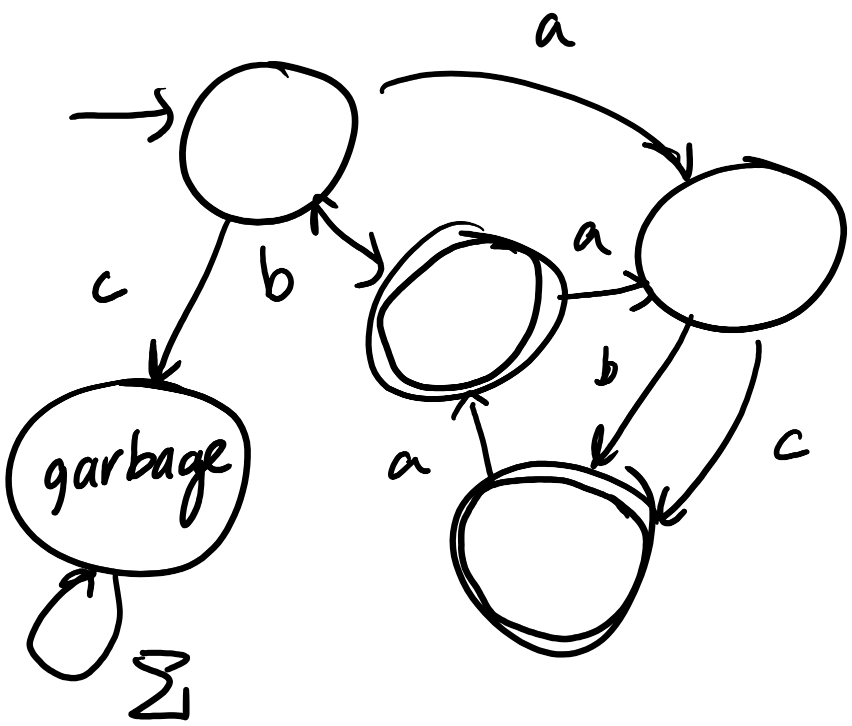
Suppose L is a regular language, is the *complement* of L, denoted $\bar{L}$, regular? $\bar{L} = \{w \in \Sigma^*: w \notin L\}$.

Because L is regular

it is accepted by some DFA M.

DFA which accepts $\bar{L}$:





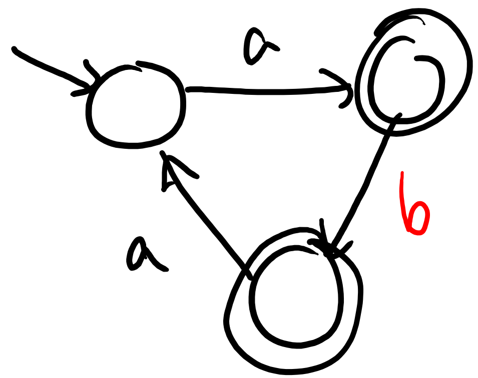
Application

Suppose L is a regular language, is the *reversal* of L, denoted $\mathrm{Rev}(L)$, regular? $\mathrm{Rev}(L) = \{(w)^R \in \Sigma^* : w \in L\}$.

Since L regular language,

DFA which accepts L

M

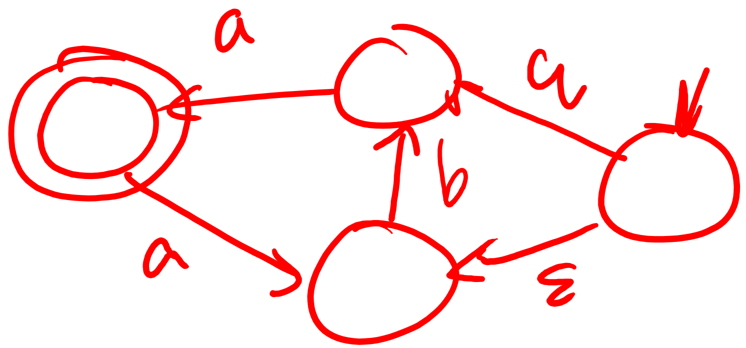


if $w = w_0 \cdots w_{n-1}$ then

$(w)^R = w_{n-1} \cdots w_0$

NFA

M' which accepts $\mathrm{Rev}(L)$.



Now you try!

1. Let L be a language on the alphabet $\Sigma = \{0, \dots, n-1\}$ of size n. Let $s_1, \dots., s_{n-1}$ be string. Let $L(s_0, \dots, s_{n-1})$ is the *replacement language*, where every instance of i is replaced with s_i. Is $L(s_0, \dots, s_{n-1})$ regular?
2. Consider the language
$$L_1 = \{w \in \{0,1\}^*: \text{second last letter of } w \text{ is } 0\}.$$

Construct a finite automaton for L_1 and prove it is correct.

3. Consider the language we saw in the previous lecture
$$L_2 = \{0^n 1^m: m, n \geq 0, m + n \text{ is odd}\}.$$

Show that the finite automaton for L_2 is correct.

1. Let L be a language on the alphabet $\Sigma = \{0, \dots, n-1\}$ of size n. Let $s_0, \dots, s_{n-1}$ be strings. Let $L(s_0, \dots, s_{n-1})$ is the *replacement language*, where every instance of i is replaced with s_i. Is $L(s_0, \dots, s_{n-1})$ regular?

$\in \Sigma$

EXAMPLE

$\Sigma = \{0, 1, 2\}$ $\quad L = \{0^i 1^j 2^k : i, j, k \geq 0\}$

ORIGINAL STRING

0 1 2

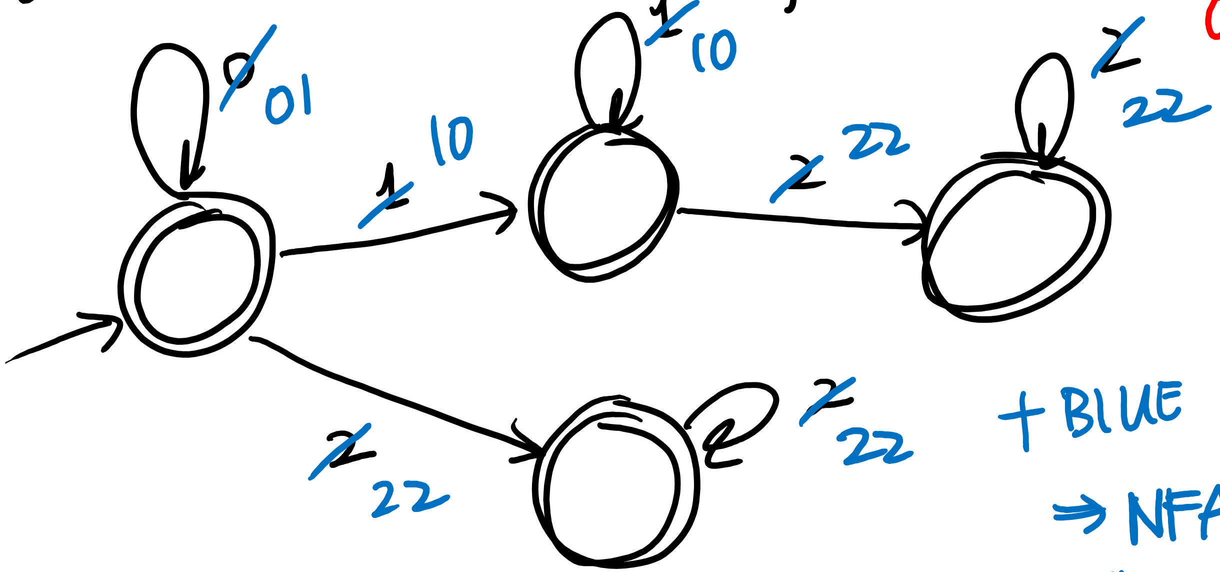
$\boxed{L(01, 10, 22)} = \{(01)^i (10)^j (22)^k : i, j, k \geq 0\}$

NEW STRING

01 10 22

THIS IS REGULAR!

since L is regular, construct DFA M which accepts L



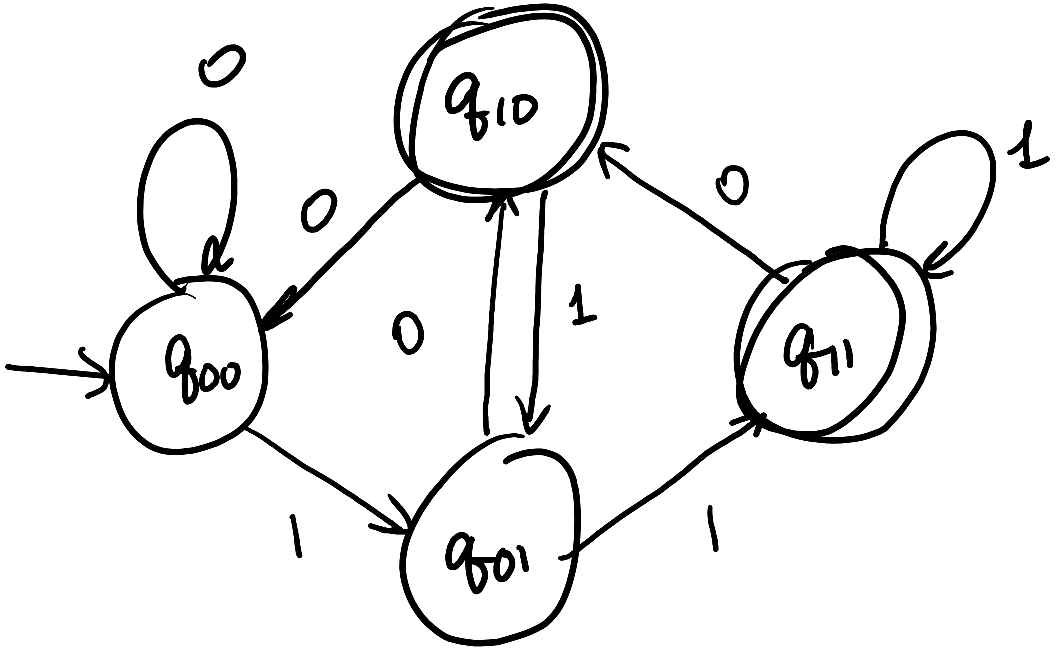
+ BLUE CHANGES

→ NFA which accepts Rep language

2. Consider the language

$$L_1 = \{w \in \{0,1\}^* : \text{second last letter of } w \text{ is } \overset{1}{\cancel{0}}\}.$$

Construct a finite automaton for L_1 and prove it is correct.



$$p(w) := \delta^*(q_0, w)$$

q_{00}	last two chars of w is 00 or is 0 or is ε
q_{10}	- - - - - - - - 10
q_{01}	- - - - - - - - 01
q_{11}	- - - - - - - - 11 or is 1

2. Consider the language

$$L_1 = \{w \in \{0,1\}^* : \text{second last letter of } w \text{ is } \cancel{0}\ 1\}.$$

Construct a finite automaton for L_1 and prove it is correct.

Base case : ε, 0, 1 statisfy state invariants.

Inductive STEP : suppose $|w| \geq 1$. and $P(w)$ is TRUE

$\delta^*(q_{00}, w)$ IH GIVES then

case $w1$

$q_{00} \to$ last char 0 or 00 — $w1$ ends up in q_{01} and last two char of $w1$ is 01.

q_{01}

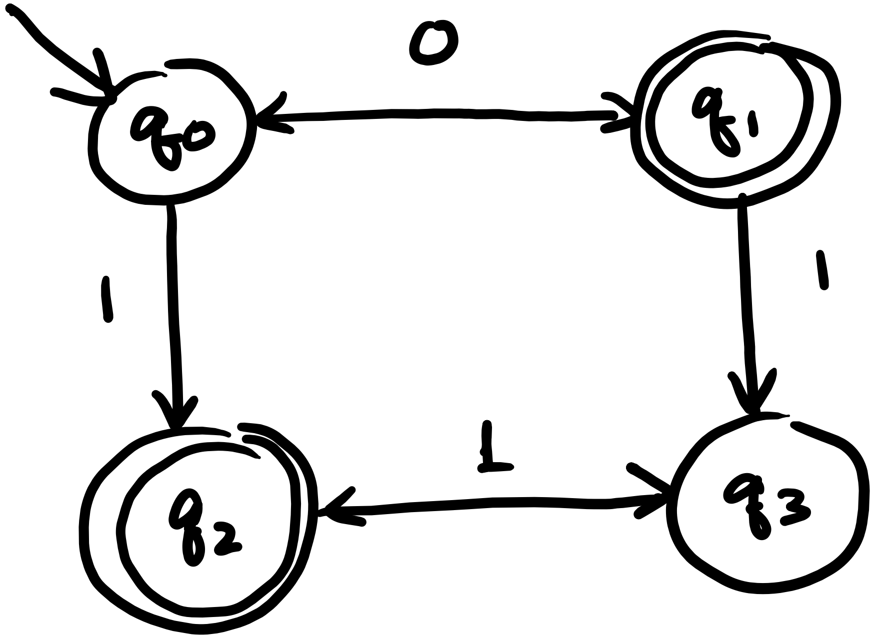
q_{10}

$q_{11} \longrightarrow$ last two chars of w is 11. — $w1$ end in q_{11}. ok since last two chars still 11.

3. Consider the language we saw in the previous lecture

$$L_2 = \{0^n 1^m : m, n \geq 0, m + n \text{ is odd}\}.$$

Show that the finite automaton for L_2 is correct.



$$p(w) := \delta^*(q_0, w) = \begin{cases} q_0 & w = 0^k \text{ for } k \text{ even} \\ q_1 & w = 0^k \text{ for } k \text{ odd} \\ q_2 & w = 0^k 1^\ell \quad k+\ell \text{ odd}, \ \ell \geq 1 \\ q_3 & w = 0^k 1^\ell \quad k+\ell \text{ even}, \ \ell \geq 1 \end{cases}$$

3. Consider the language we saw in the previous lecture

$$L_2 = \{0^n 1^m : m, n \geq 0, m+n \text{ is odd}\}.$$

Show that the finite automaton for L_2 is correct.

proof by structural induction on Σ^*

Base Case : ε has the form 0^k for k even (description of start state q_0)

inductive step : suppose $w \in \Sigma^*$ where $P(w)$ is true.

Consider $w0$ and $w1$:

$w0$: $\delta^*(q_0, w)$ IH gives $\delta^*(q_0, w0)$

q_0 $w = 0^k$ where k is even — q_1 where $w0 = 0^{k+1}$, $k+1$ odd

q_1 $w = 0^k$ where k is odd — q_0 where $w0 = 0^{k+1}$ where $k+1$ is even

q_2 $w = 0^k 1^l$ where $k+l$ is odd and $l \geq 1$ — garbage since $0^k 1^l 0$ is not in the language

q_3 $w = 0^k 1^l$ where $k+l$ is even and $l \geq 1$ — (same as previous)

$w1$ is similar...