Lecture 18

February 29, 2024 10:28 AM

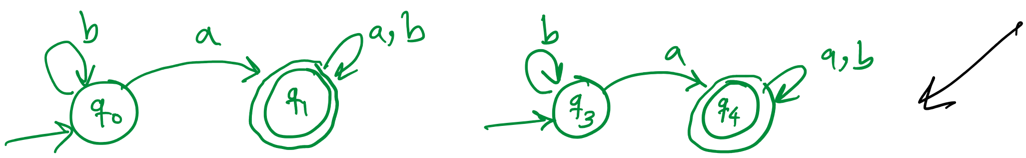
More on Pumping Lemma

* There are languages that are not regular but you cannot directly use pumping lemma to show it
    * Exercise 43 in the book.

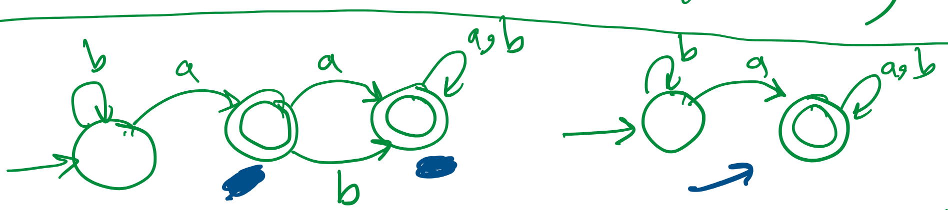
* There is a more involved vesion of pumping lemma (Exercise 44) that "characterizes" regular sets.

* we skip the section on ultimate periodic sequences.

Equivalence of DFAs (or DFA minimization)



Isomorphic DFAs (they the same up to renaming the states)



These DFAs represent the same set. (the right DFA is "simpler")

Checking Isomorphism:

Checking Isomorphism:

* start with the start state in both DFAs and call both states q_0.

* Continue labeling the other states that are adjacent to the already labeled states (and check consistency)

DFA Minimization

maybe turn them into regular expressions?

$$(\underbrace{a^* b}_{b} + \underbrace{b^* a}_{a})^* \stackrel{?}{=} (ba + ab + aa + bb)^* (a + b + \varepsilon)$$

but again, how to check if the expression are equivalent?

The idea is that we can remove "redundant" states iteratively and it turns out that we will always end up in the same "minimal" DFA (up to isomorphism)

We say states p and q are equivalent ($q \approx p$) if the following holds:

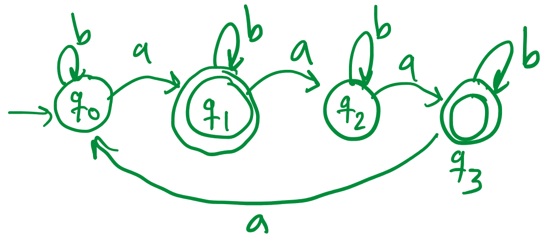
equivalent ($q \approx p$) if the following holds:

* For every string x,

$$\hat{\delta}(p, x) \in F \Longleftrightarrow \hat{\delta}(q, x) \in F$$

But how to figure out equivalent states?

DFA Minimization Algorithm



1. create a table of pairs of states and initialize it to 0.

[0 "means" the pair of states are equivalent]

	q_0	q_1	q_2	q_3
q_0	0	~~0~~ 1	0	~~0~~ 1
q_1		0	~~0~~ 1	0
q_2			0	~~0~~ 1
q_3				0

2. mark (p, q) if $p \in F$ but $q \notin F$.

3. Repeat until no other updates can be made:
   if $\exists (p, q), a \in \Sigma$ such that $(\delta(p, a), \delta(q, a))$ is already marked then mark (p, q).

---

entries in the

At the end, 0 entries in the table represent equivalent states.