CS340: Theory of Computation — Sem I 2019-20

Lecture Notes 7: DFA Minimization

Raghunath Tewari — IIT Kanpur

Given a DFA $D = (Q, \Sigma, \delta, q_0, F)$ we define a equivalence relation on the states of the DFA. For any two states $p, q \in Q$, we say that $p \approx p$ if for all string $x \in \Sigma^*$, $(\delta(p, x) \in F \iff \delta(q, x) \in F)$.

Exercise 1. Verify that $\approx$ is an equivalence relation.

Let $[p] = \{q \mid q \approx p\}$ be the equivalence class of all states equivalent to p. We define a *quotient DFA* $D_\approx$ based on the DFA D as $D_\approx = (Q', \Sigma, \delta', q_0', F')$, where,

$$\begin{aligned} Q' &= \{[p] \mid p \in Q\} && \text{(i.e. the set of equivalence classes)} \\ \delta'([p], a) &= [\delta(p, a)] \\ q_0' &= [q_0] \\ F' &= \{[f] \mid f \in F\} \end{aligned}$$

Exercise 2. Show that the definition of δ' is well defined. In other words, if $[p] = [q]$, then $[\delta(p, a)] = [\delta(q, a)]$ for all $a \in \Sigma$.

We will now show that $D_\approx$ and D accept the same language.

Lemma 1. *For all* $x \in \Sigma^*$, $\delta'([p], x) = [\delta(p, x)]$.

Proof. We will use induction on $|x|$.

Base Case If $x = \epsilon$, then

$$\begin{aligned} \delta'([p], \epsilon) &= [p] \\ &= [\delta(p, \epsilon)]. \end{aligned}$$

Induction Step Let $x = ya$ and assume that $\delta'([p], y) = [\delta(p, y)]$. Now

$$\begin{aligned} \delta'([p], ya) &= \delta'(\delta'([p], y), a) \\ &= \delta'([\delta(p, y)], a) \\ &= [\delta(\delta(p, y), a)] \\ &= [\delta(p, ya)]. \end{aligned}$$

□

Theorem 2. $L(D_\approx) = L(D)$.

Proof. For all $x \in \Sigma^*$,

$$\begin{aligned} \delta'(q_0', x) \in F' &\iff \delta'([q_0], x) \in F' \\ &\iff [\delta(q_0, x)] \in F' && \text{(by Lemma 1)} \\ &\iff \delta(q_0, x) \in F. \end{aligned}$$

□

Exercise 3. Can you collapse the quotient DFA any further? What happens if you try to do so?

1 DFA Minimization Algorithm

Remark. A state is said to be unreachable if on no input the DFA ever traverses that state.

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA that does not have any unreachable states. The algorithm to minimize the DFA is as follows:

1. Create a table of pairs $\{p, q\}$, where $p, q \in Q$. All entries of the table are initially unmarked.

2. Mark the pair $\{p, q\}$ if $p \in F$ and $q \notin F$, or vice versa.

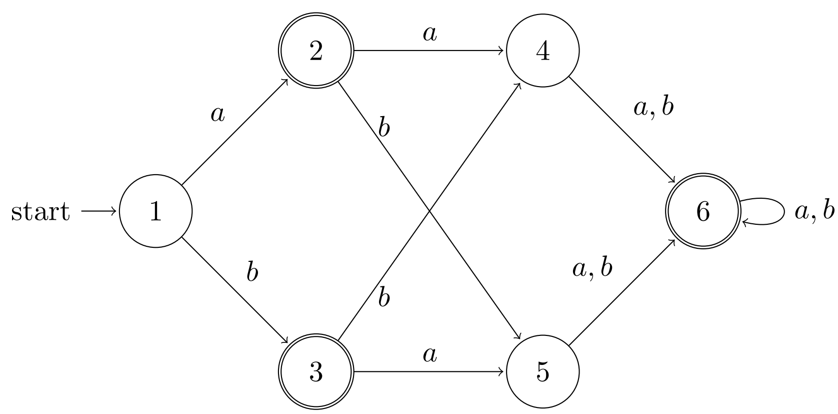
3. Repeat the following until you make an entire pass of the table and no new pair gets marked:

    - If $\{p, q\}$ is unmarked and there exists a symbol $a \in \Sigma$ such that $\{\delta(p, a), \delta(q, a)\}$ is marked, then mark pair $\{p, q\}$.

4. After completion, $p \approx q$ if and only if $\{p, q\}$ is not marked.

1.1 An Example

Consider the following DFA



We want to minimize the above DFA. We first create a table of pairs.

	1	**2**	**3**	**4**	**5**
2	×				
3	×				
4		×	×		
5		×	×		
6	×			×	×

After Step 2

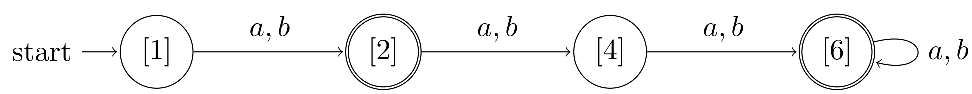
	1	**2**	**3**	**4**	**5**
2	×				
3	×				
4		×	×		
5		×	×		
6	×	×	×	×	×

After 1st iteration of Step 3

	1	**2**	**3**	**4**	**5**
2	×				
3	×				
4	×	×	×		
5	×	×	×		
6	×	×	×	×	×

After 2nd iteration of Step 3

No more pairs can get marked any further. Hence the algorithm terminates. From the final table we have that $2 \approx 3$ and $4 \approx 5$. Hence the minimized DFA will have the following form.



Exercise 4. Minimize the following DFA.