Comp 330 - Lecture 8 - September 26th

Italian expression: Avere le mani in pasta

→ To be in a sticky/problematic situation

DFA Minimization

Today: Algo which minimizes DFA

=> Avoid waste of comp. resources/ memory

Thursday: Unique minimal DFA

=> Min Algo + Uniqueness min DFA

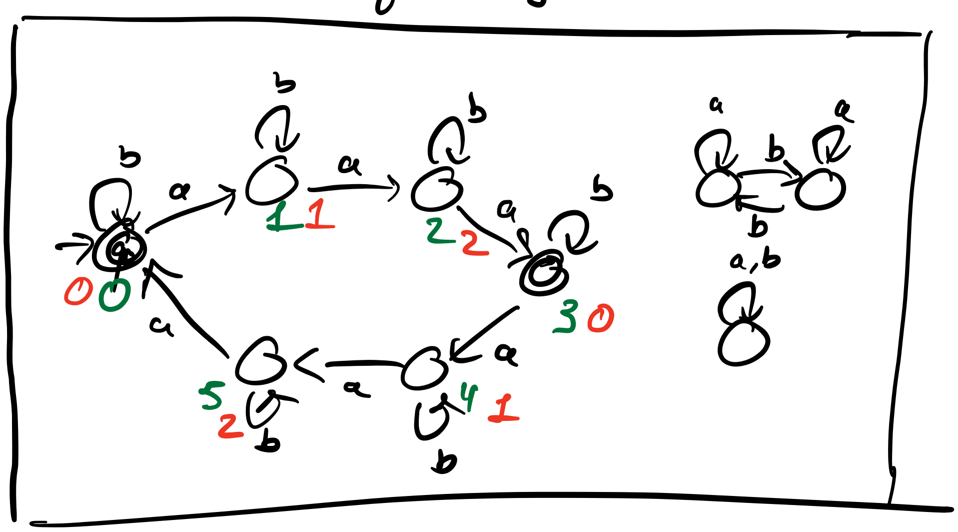
=> "Given reg. exp. r_1, r_2, is $L(r_1) = L(r_2)$?"

<u>Ex</u> Consider the following not minimal DFA

M

no mod 6

no mod 3



Why is M not minimal?

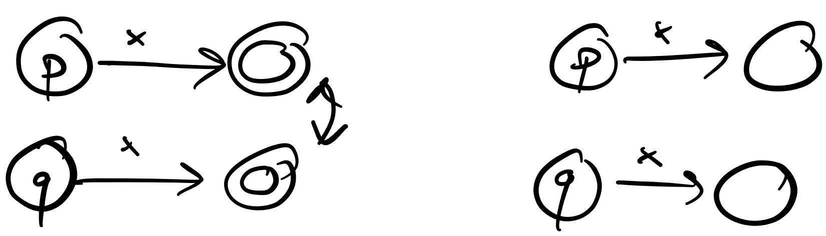
1. Unreachable states => Easy fix.
   Run DFS starting at q_0, drop the unmarked states. [OR unvisited]
2. Redundancy in the state transition diagram.
   M is keeping track of $n_a \bmod 6$
   Accept strings s.t. $n_a \bmod 6 = 0$ or 3
   $\iff n_a \bmod 3 = 0$
   Min alg : Collapse the states which serve the same purpose/ function =>
   Equivalent states.

Equivalent states in DFA

Def. Given DFA $M = (Q, \Sigma, \delta, s_0, F)$, $p, q \in Q$, p & q are equivalent, $p \approx q$, if

$$\forall x \in \Sigma^* \quad \delta^*(p, x) \in F \iff \delta^*(q, x) \in F$$

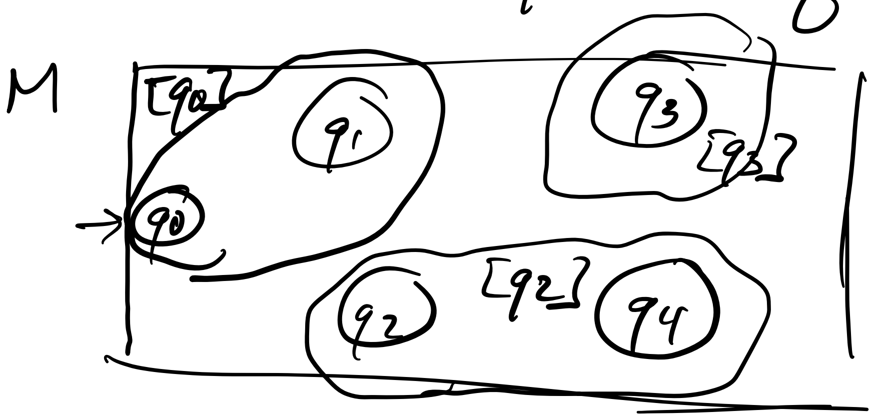
$x \in \Sigma^*$, p, q. $p \approx q$



p & q are equivalent if $\forall x \in \Sigma^*$, the destination states have the same type (◎/○)

<u>Fact</u> (About $\simeq$) Given DFA M, $\simeq$ is an equivalence relation, $p \in Q$, $[p]$ is an equivalence class of $\simeq$.

<u>Goal of min alg</u>: Make states in a min DFA M' the eq. classes of $\simeq$.



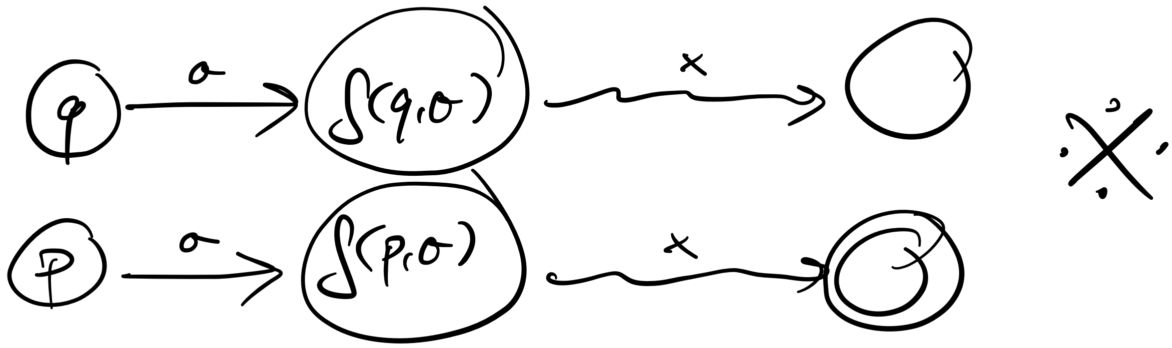
$Q = \{q_0, \ldots, q_4\}$

$\simeq$

$p, q \in Q$

<u>Lemma A</u> Given DFA M, $p \simeq q$
$\Rightarrow \forall \sigma \in \Sigma$, $\delta(p,\sigma) \simeq \delta(q,\sigma)$.

<u>Pf</u> $p, q \in Q$, $p \simeq q$, $\sigma \in \Sigma$, suppose
$\delta(q,\sigma) \not\simeq \delta(p,\sigma)$



Implication: $p, q \in [p]$ Then $\forall a \in \Sigma$
$\delta(q,a), \delta(p,a) \in [\delta(p,a)]$

Explicit construction of M' (Quotient construction)

Given DFA $M = (Q, \Sigma, \delta, s_0, F)$, we define a new DFA $M' = (Q', \Sigma, \delta', s_0', F')$

$$Q' := \{ [q]_{\approx} : q \in Q \}$$

$$s_0' := [s_0]_{\approx}$$

$$F' := \{ [f]_{\approx} : f \in F \}$$

$$\delta'([q], a) = [\delta(q,a)]_{\approx}$$ → Well-defined thanks to Lemma A

→ 1. Prove $L(M') = L(M)$
2. M' is minimal

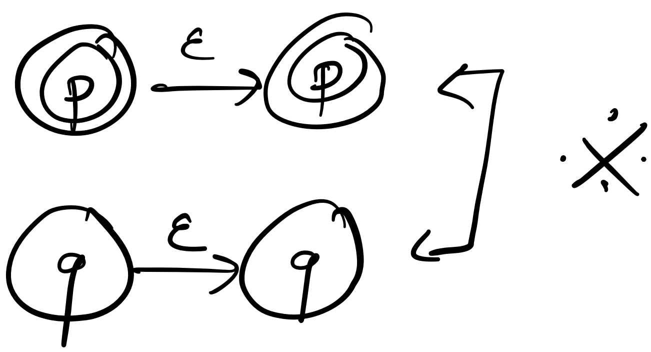
Lemma B DFA M, M', $p \in F$, $\forall q \in Q$.

$$p \simeq q \Rightarrow q \in F$$

↳ [Eq classes of M' which contain accept states of M only contain accept states of M].

Pf

$p \in F$, $q \in Q$, $p \simeq q$, $q \notin F$



Implication: $p \in F \Leftrightarrow [p] \in F'$

Lemma C M, M', $\forall w \in \Sigma^*$,

$$\delta'^*([p], w) = [\delta^*(p, w)]$$

Pf By ind on $|w|$.

As constructed above.

Thm M, M' DFA $L(M) = L(M')$

Pf $x \in \Sigma^*$ $\quad x \in L(M) \iff \delta^*(s_0, x) \in F$

(Lemma B) $\iff [\delta^*(s_0,x)] \in F'$

(Lemma C) $\iff \delta'^*([s_0], x) \in F'$

$\iff x \in L(M')$.

<u>Claim</u> M' is a minimal DFA

<u>Proof</u> Suppose $|Q'| = n$

$$\begin{array}{cccc} [q_1] & [q_2] & \cdots & [q_n] \\ \uparrow x_1 & \uparrow x_2 & & \uparrow x_n \\ [s_0] & [s_0] & & [s_0] \end{array}$$

Assume that M & $\therefore$ M' have no unreachable states.

$[q_1] \neq [q_2] \Rightarrow q_1 \not\equiv q_2 \Rightarrow \exists \boxed{x} \in \Sigma^*$ s.t.

$\underline{\delta^*(q_1, x) \in F}$ &

$\underline{\delta^*(q_2, x) \notin F}$

(or vice-versa)

$$\delta^*(\underbrace{\delta^*(s_0, x_1)}_{= q_1}, \boxed{x}) = \delta^*(s_0, x_1 x) \in F$$

$\Rightarrow x_1 x \in L(M) = L(M')$

$$\delta^*(\underbrace{\delta^*(s_0, x_2)}_{q_2}, \boxed{x}) = \delta^*(s_0, x_2 x)$$

$$\Rightarrow x_2 x \notin L(M) = L(M')$$

Consider some arbitrary DFA M'' s.t. $L(M'') = L(M')$, label the states of M''

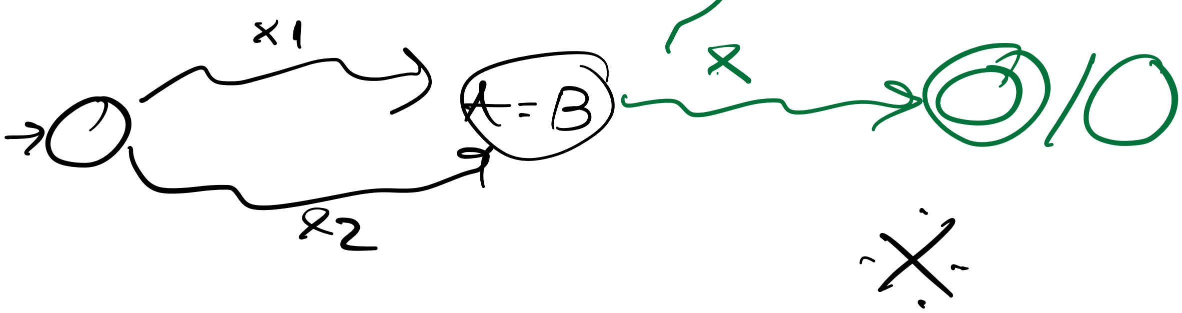
$x_1 \rightsquigarrow A$

$x_2 \rightsquigarrow B$

$\vdots$

$x_n \rightsquigarrow N$

Is it possible that $A = B$? No!



Who cares?

The way the min alg works:

① Merge all states of M together

② Keep splitting machine by finding states which are <u>distinguishable</u> until can't split any further.

<u>Def</u> DFA $M = (Q, \Sigma, \delta, s_0, F)$, $p, q \in Q$,
p & q are <u>distinguishable</u>, $p \bowtie q$, if
$p \not\approx q$ i.e.

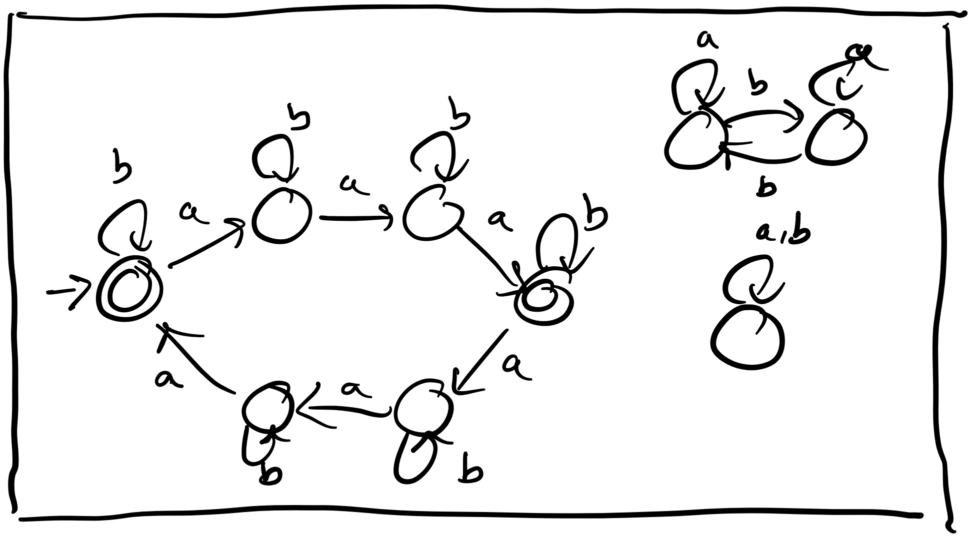
$\exists\, x \in \Sigma^* .\ \delta^*(p, x) \in F$ & $\delta^*(q, x) \notin F$
[or vice-versa]

distinguishing string

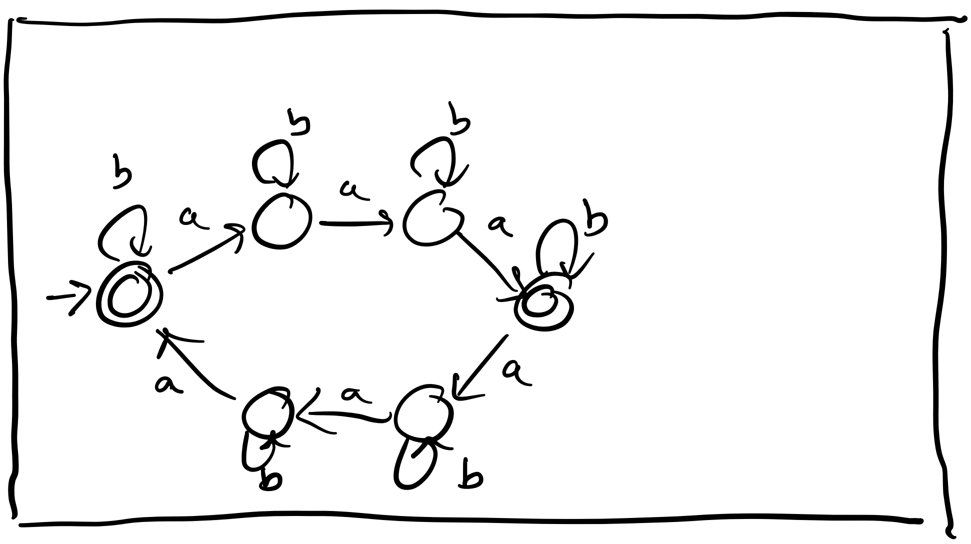
<u>Fact</u> if $\exists a \in \Sigma$ s.t. $\delta(p, a) \bowtie \delta(q, a)$
then $p \bowtie q$.

<u>Min algorithm</u>

Ex DFA M

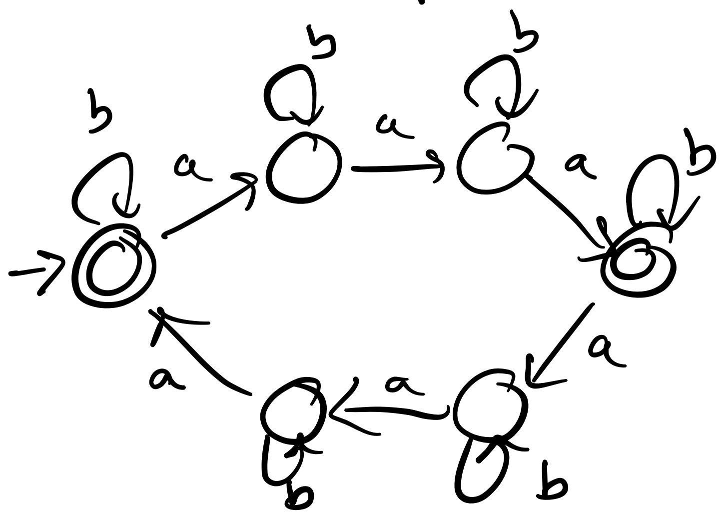


Step 1 : Remove unreachable states.



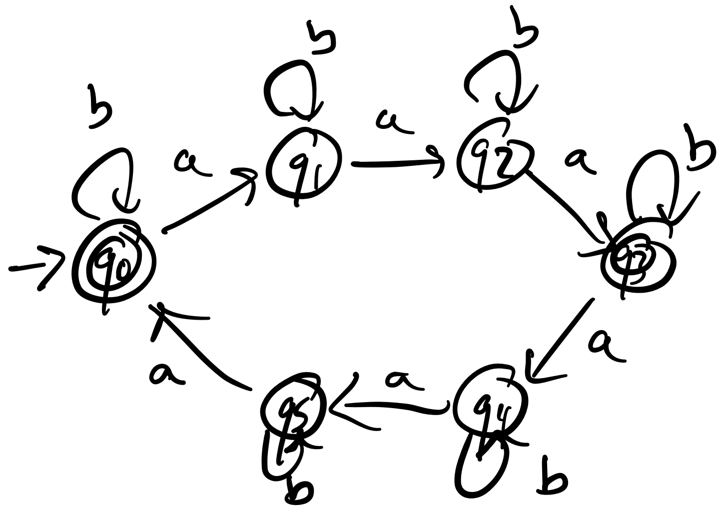
Step 2: Make a 2-D array $|Q| \times |Q|$ [Label states]

$A[p,q] = 0 \iff p \bowtie q$



	q_0	q_1	q_2	q_3	q_4	q_5
q_0						
q_1						
q_2						
q_3						
q_4						
q_5						

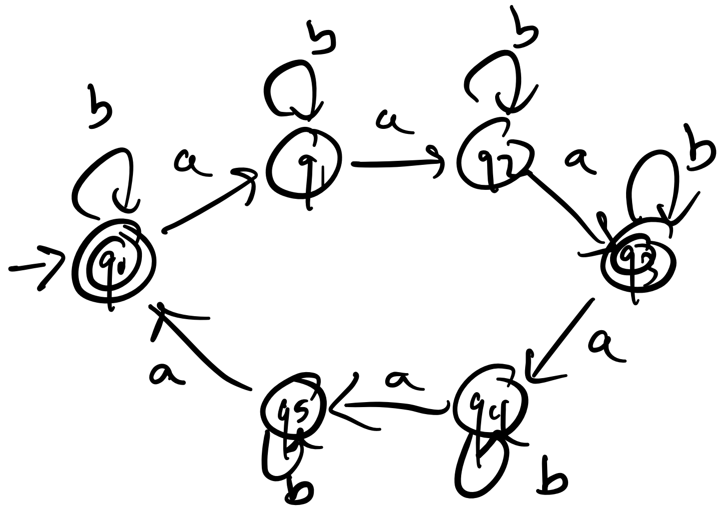
Step3: For every pair of states p, q, $p \in F$, $q \notin F$, $A[p,q] = 0$



	q_0	q_1	q_2	q_3	q_4	q_5
q_0						
q_1	0					
q_2	0					
q_3		0	0			
q_4	0			0		
q_5	0			0		

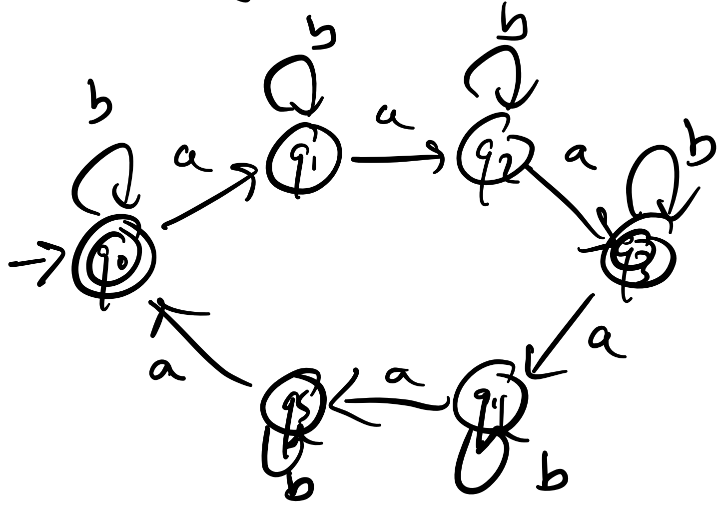
Repeat until A does not change

Step 4: For every pair p, q if $\exists \sigma \in \Sigma$ s.t. $A[\delta(p,\sigma), \delta(q,\sigma)] = 0$ then $A[p,q] = 0$



	q_0	q_1	q_2	q_3	q_4	q_5
q_0						
q_1	0					
q_2	0	0				
q_3		0	0			
q_4	0		0	0		
q_5	0	0		0	0	

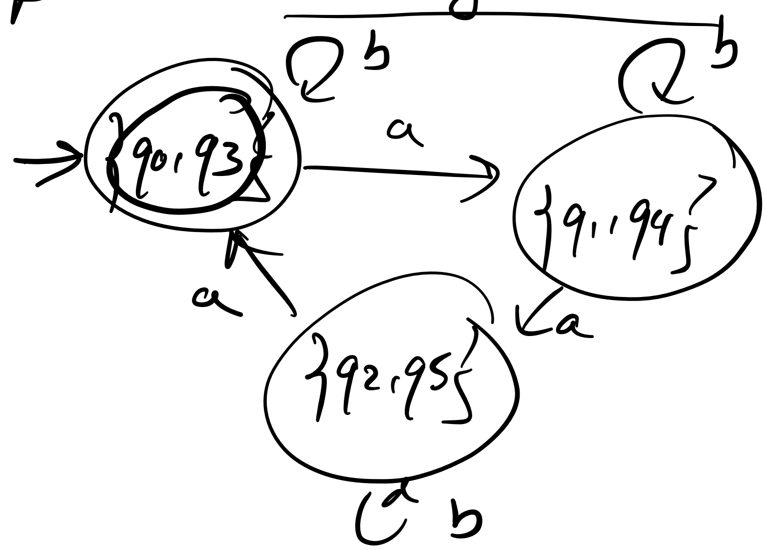
Step 5: Continue until no change.
Once no change mark all other entries with a 1



	q0	q1	q2	q3	q4	q5
q0						
q1	0					
q2	0	0				
q3	1	0	0			
q4	0	1	0	0		
q5	0	0	1	0	0	

q6

Step 6: Building machine



Lemma Given min. algo & input DFA M, Then $\forall\, p, q \in Q$

$$A[p,q] = 1 \iff p \equiv q$$

$\Rightarrow$ DFA $M = (Q, \Sigma, \delta, s, F)$. Suppose $p, q \in Q$, $A[p,q] = 1$, $p \not\equiv q$. Call that a bad pair.

$\therefore$ $p \not\equiv q \Rightarrow \exists$ string x which distinguishes p & q.

Collect all bad pairs & all distinguishing strings. Pick shortest string x. p, q, $p \not\equiv q$

$x = a_1 a_2 \ldots a_n$ $a_i \in \Sigma$

($x \neq \varepsilon$ b/c then $A[0,0] = 1$] which alg takes care of. $\leftarrow n > 0$

Consider $\delta(p, a_1) = p'$ $\delta(q, a_1) = q'$
It must be that $p' \not\approx q'$ b/c $a_2 \dots a_n$ is a dist. string. $\Rightarrow$ p', q' is not a bad pair $\therefore$ alg labels $A[p', q'] = 0$
The alg will mark $A[p, q] = 0$ in the next step. So p & q were not a bad pair to begin with.

$\Leftarrow$ Almost by def.

<u>Corollary</u> The output of min alg is <u>a</u> min DFA

<u>Pf</u> Lemma + Construction in Step 6