Class # 4

Date	@Jan 15, 2021 10:30 AM
Hours	1
Type (Test/Normal)	Normal Class
Unit	Unit 1

Topic/Chapter: Unit 1

The Actual notes:

- *Examples - Constructing a DFA from an NDFA*
  - **Numero uno**

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

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

$\underline{Q' = 2^Q}$, $q_0' = [q_0]$ $\quad \delta'([q_1, q_2, \ldots, q_i], a) = [p_1, \ldots, p_j]$

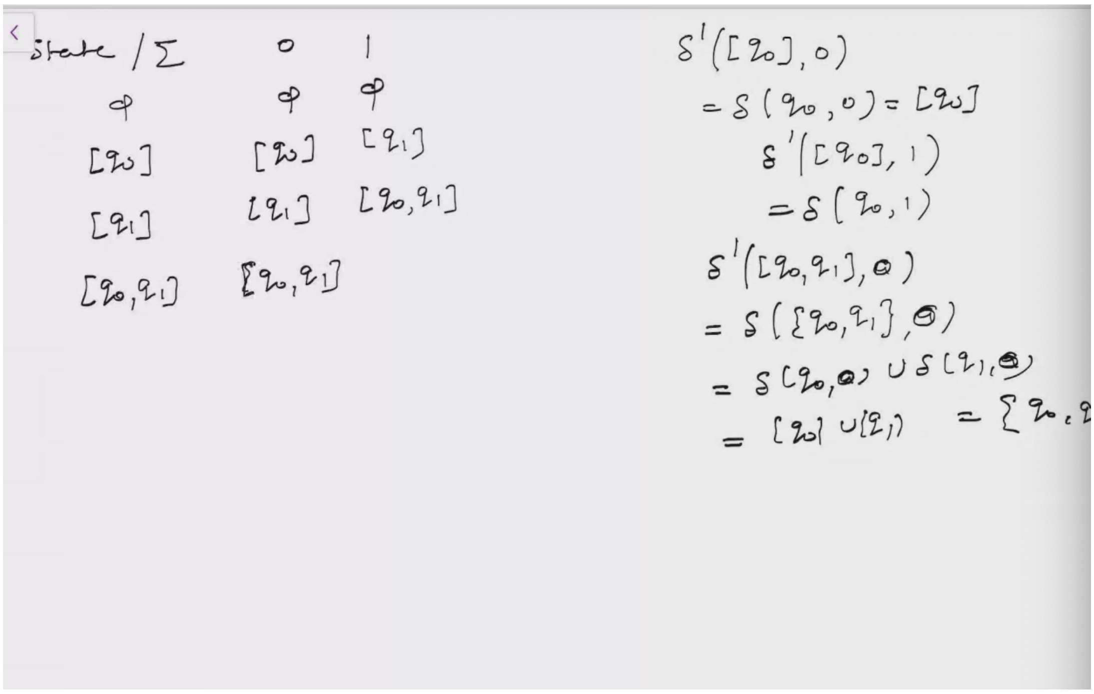
$[q_i, q_j \ldots q_k]$ $\quad$ iff $\underline{\delta(\{q_1, q_2, \ldots q_i\}, a)} = \{p_1, \ldots, p_j\}$

$$\delta(q_1, a) \cup \delta(q_2, a) \cup \ldots \cup \delta(q_i, a)$$

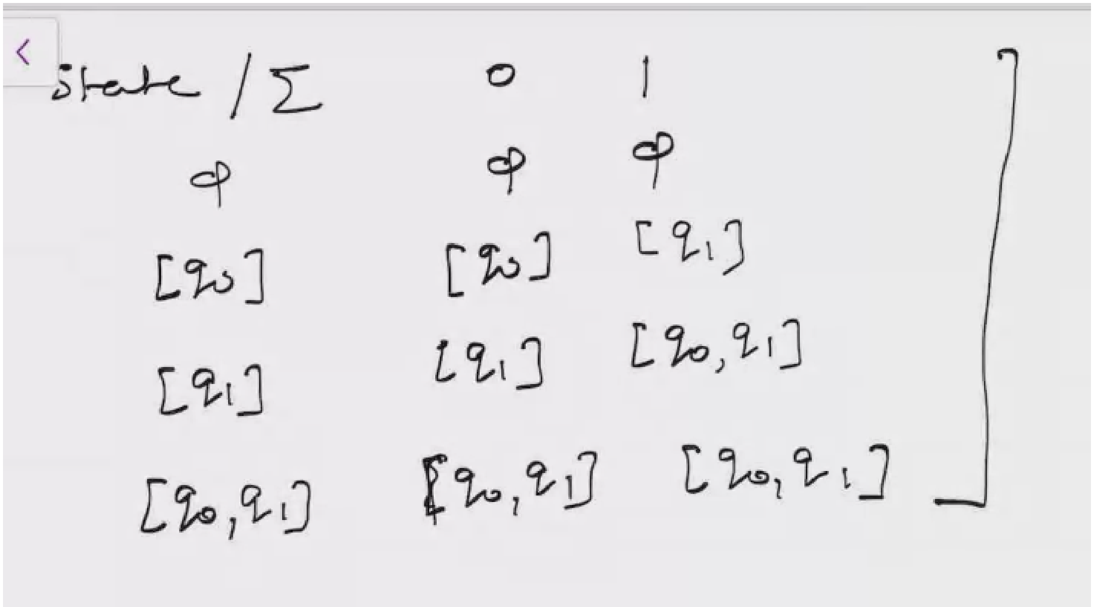
$\underline{Ex}$ $\quad M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_0\})$

State/Σ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_1	q_0, q_1

Only use square basket for DFA, not curly while listing the tuple in reference to the provided NDFA tuple



corresponding DFA that accepts the same set of states:



▼ **Numero dōs**

$M = (\underline{\{q_0, q_1, q_2, q_3\}}, \{a,b\}, \delta, q_0, \{q_3\})$

State/Σ	a	b
$\rightarrow q_0$	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	–	q_2

$\omega \in T(M)$

$\omega \in T(M')$

$\delta(q_0', \omega) = - \; - \rightarrow \in F'.$

$M' = 2^Q$, $\Sigma = \{a,b\}$, $[q_0] = q_0'$

$F = \{[q_3], [q_0,q_3], [q_1,q_3], [q_2,q_3], [q_0,q_1,q_3], [q_0,q_2,q_3], [q_1,q_2,q_3], [q_0,q_1,q_2,q_3]\}.$

$2^4 = 16$ states.

$$\begin{array}{ccc} [q_0] & \longrightarrow & [q_0,q_1] \\ & \swarrow b & \downarrow a \\ \underline{[q_0,q_1]} & & [q_0,q_1,q_2] \\ & & \downarrow ? \end{array}$$

$\delta'([q_0,q_1], a)$

$= \delta(\{q_0,q_1\}, a)$

$= \delta(q_0,a) \cup \delta(q_1,a)$

State/Σ	a	b
$[q_0]$	$[q_0, q_1]$ ✓	$[q_0]$ ✓
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$ ✓
✓ $[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$ ✓
✓ $[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$

Minimisation of finite automata

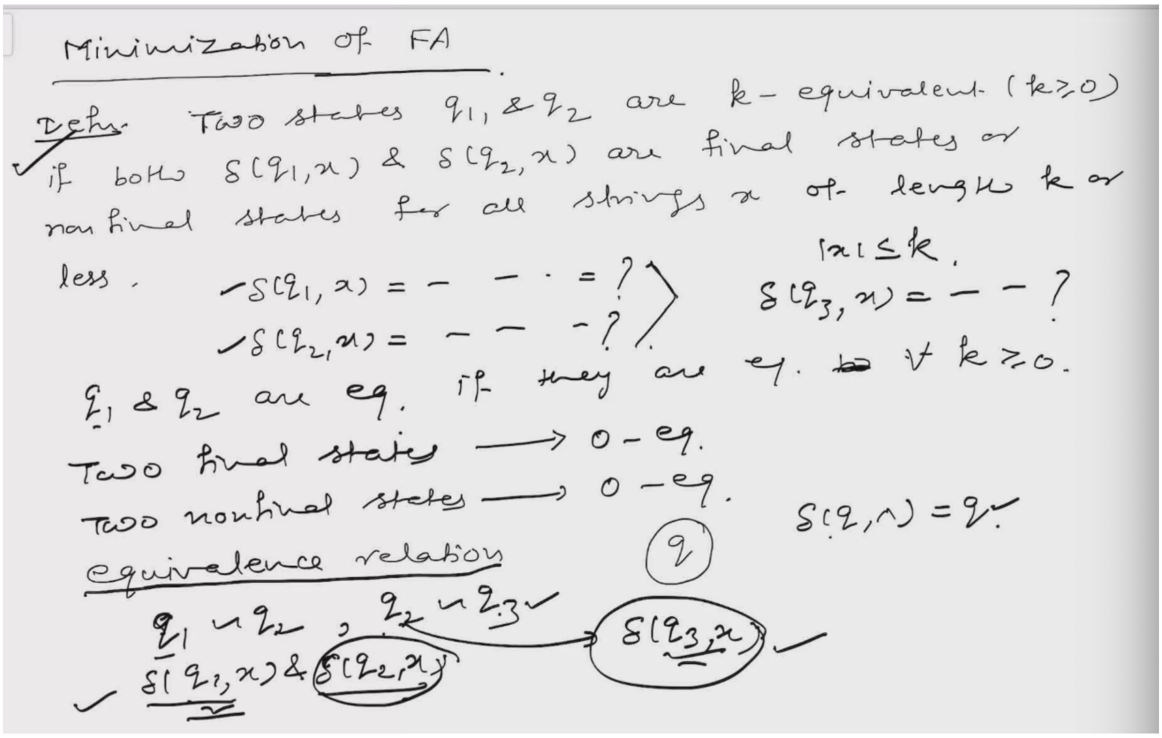
Definition: 2 states q1 and q2 are **k-equivalent (k ≥ 0)** if both delta (q1,x) and delta (q2,x) are final states or non-final states for all strings x of length ≤ k

2 states q1 and q2 are **equivalent** if they are equivalent for all k ≥ 0

2 final states → **0-equivalent**

2 non-final states → **0-equivalent**

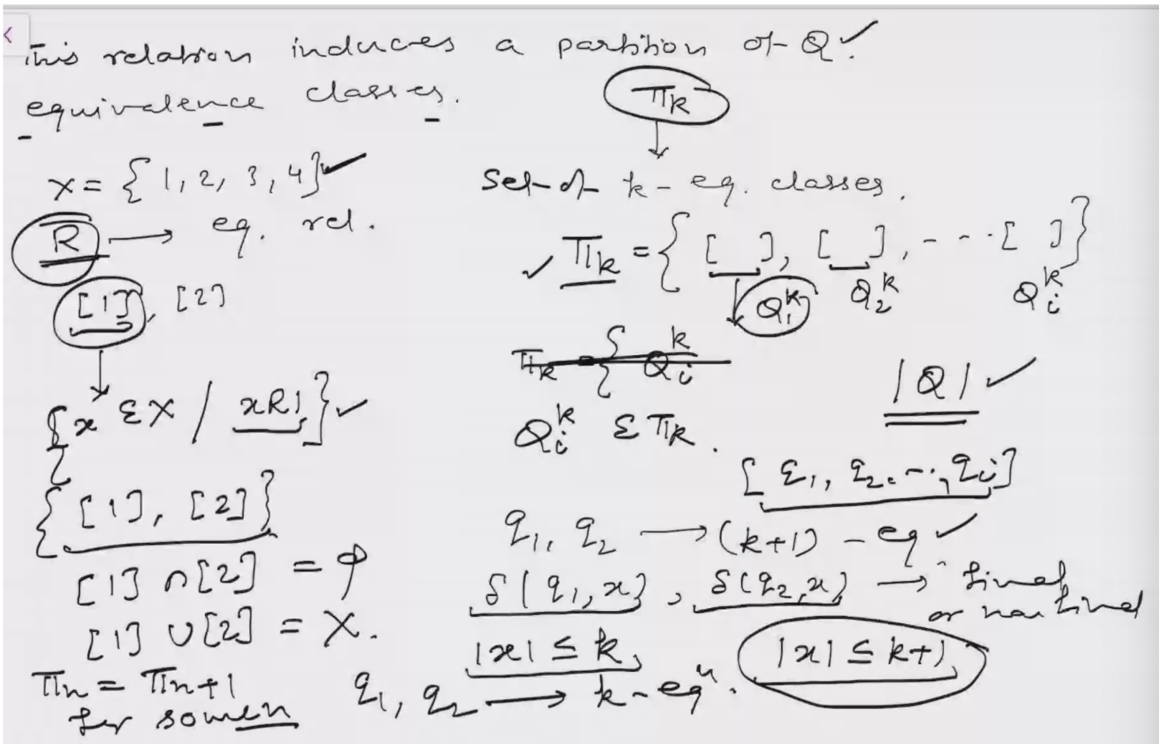
This type of relation on a set of states is an **equivalence relation.**



An equivalence relation induces a partition(*intersection is phi, and union is the whole set*) on the set Q , called **equivalence classes.**

Suppose this partition is πk → the elements of πk are k-equivalence classes - and any 2 elements in any of these classes are k-equivalent.

Cardinality of πk ≤ cardinality of Q



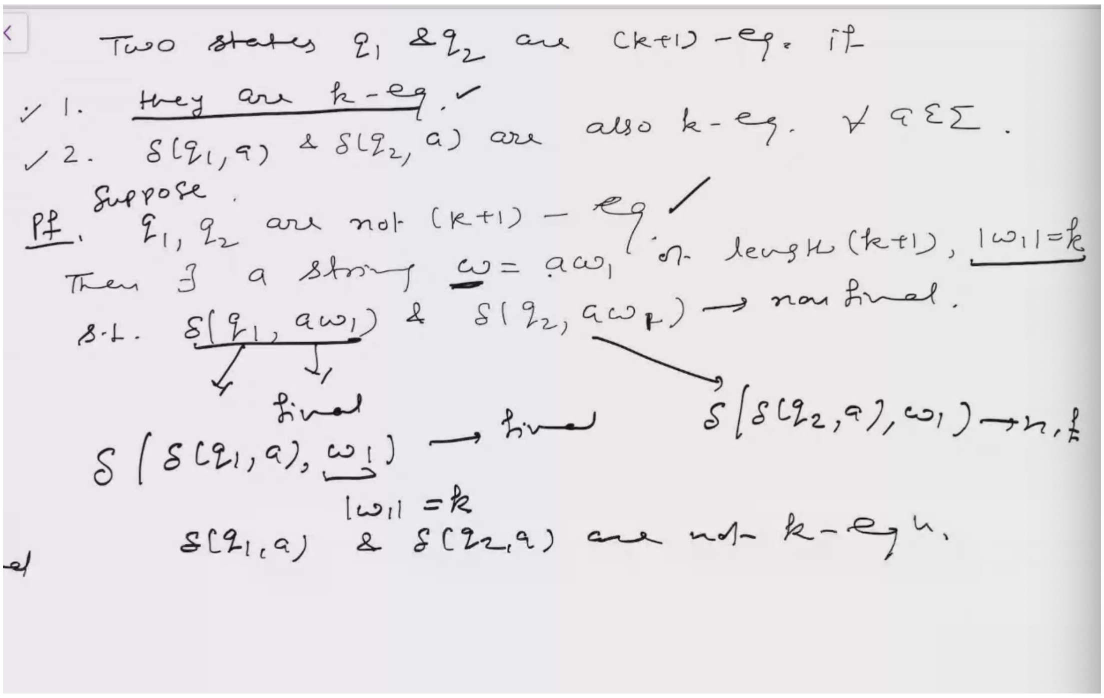
> if 2 states are (k+1) equivalent → they are also k-equivalent; but vice versa is not true

▼ ***Condition under which the vice-versa ^ is true***

The states q1, and q2 are k+1 equivalent if,

1. They are k-equivalent, and
2. delta(q1,a) & delta(q2,a) are also k equivalent, for all a belongs to sigma

Proof:


Two states q_1 & q_2 are $(k+1)$-eq. if
✓ 1. they are k-eq. ✓
✓ 2. $\delta(q_1,a)$ & $\delta(q_2,a)$ are also k-eq. $\forall a \in \Sigma$.
Pf. Suppose q_1, q_2 are not $(k+1)$-eq.
Then $\exists$ a string $w = aw_1$ of length $(k+1)$, $|w_1| = k$
s.t. $\delta(q_1, aw_1)$ & $\delta(q_2, aw_1) \rightarrow$ non final.
final
$\delta(\delta(q_1,a), w_1) \rightarrow$ final
$\delta[\delta(q_2,a), w_1) \rightarrow$ n.f
$|w_1| = k$
$\delta(q_1,a)$ & $\delta(q_2,a)$ are not k-eqv.