Minimizing DFA's

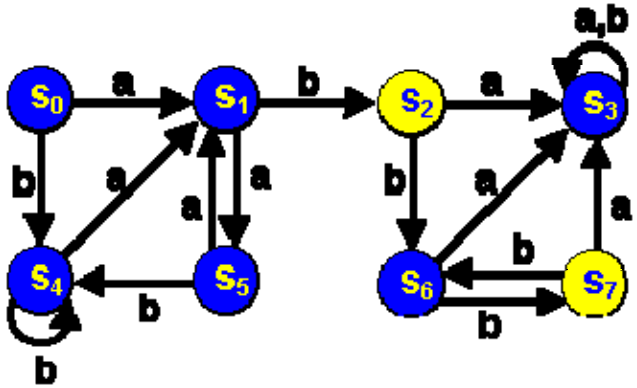
By Partitioning

Minimizing DFA's

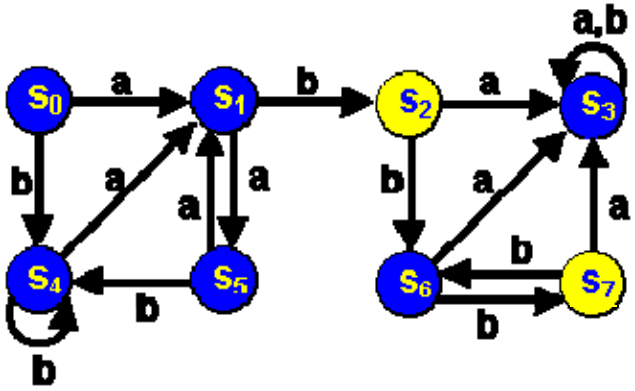
- ***Lots*** of methods
- All involve finding **equivalent states**:
  - States that go to equivalent states under all inputs (sounds recursive)
- We will use the ***Partitioning Method***

Minimizing DFA's by Partitioning

Consider the following dfa (from Forbes Louis at U of KY):



- **Accepting** states are **yellow**
- Non-accepting states are **blue**
- Are any states really the same?



- S_2 and S_7 are really the same:
  - Both Final states
  - Both go to S6 under input *b*
  - Both go to S3 under an *a*
- S_0 and S_5 really the same. Why?
- We say each pair is equivalent

Are there any other equivalent states?
We can merge equivalent states into 1 state

Partitioning Algorithm

- First
  - Divide the set of states into

    Final and
    Non-final states

Partition I

Partition II

	a	***b***
S_0	S_1	S_4
S_1	S_5	S_2
S_3	S_3	S_3
S_4	S_1	S_4
S_5	S_1	S_4
S_6	S_3	S_7
$*S_2$	S_3	S_6
$*S_7$	S_3	S_6

Partitioning Algorithm

- Now
  - See if states in each partition each go to the same partition
- S_1 & S_6 are different from the rest of the states in Partition I (but like each other)
- We will move them to their own partition

	a	*b*
S_0	S_1 I	S_4 I
S_1	S_5 I	S_2 II
S_3	S_3 I	S_3 I
S_4	S_1 I	S_4 I
S_5	S_1 I	S_4 I
S_6	S_3 I	S_7 II
*S_2	S_3 I	S_6 I
*S_7	S_3 I	S_6 I

Partitioning Algorithm

	a	*b*
S_0	S_1	S_4
S_5	S_1	S_4
S_3	S_3	S_3
S_4	S_1	S_4
S_1	S_5	S_2
S_6	S_3	S_7
$*S_2$	S_3	S_6
$*S_7$	S_3	S_6

Partitioning Algorithm

- Now again
  - See if states in each partition each go to the same partition
  - In Partition I, S_3 goes to a different partition from S_0, S_5 **and** S_4
  - **We'll move S3 to its own partition**

	a	***b***
S_0	S_1 III	S_4 I
S_5	S_1 III	S_4 I
S_3	S_3 I	S_3 I
S_4	S_1 III	S_4 I
S_1	S_5 I	S_2 II
S_6	S_3 I	S_7 II
$*S_2$	S_3 I	S_6 III
$*S_7$	S_3 I	S_6 III

Partitioning Algorithm

Note changes in S_6, S_2 and S_7

	a	*b*
S_0	S_1 III	S_4 I
S_5	S_1 III	S_4 I
S_4	S_1 III	S_4 I
S_3	S_3 IV	S_3 IV
S_1	S_5 I	S_2 II
S_6	S_3 IV	S_7 II
$*S_2$	S_3 IV	S_6 III
$*S_7$	S_3 IV	S_6 III

Partitioning Algorithm

- Now S_6 goes to a different partition on an *a* from S_1
- S_6 gets its own partition.
- We now have 5 partitions
- Note changes in S_2 and S_7

	a	*b*
S_0	S_1 III	S_4 I
S_5	S_1 III	S_4 I
S_4	S_1 III	S_4 I
S_3	S_3 IV	S_3 IV
S_1	S_5 I	S_2 II
S_6	S_3 IV	S_7 II
$*S_2$	S_3 IV	S_6 V
$*S_7$	S_3 IV	S_6 V

Partitioning Algorithm

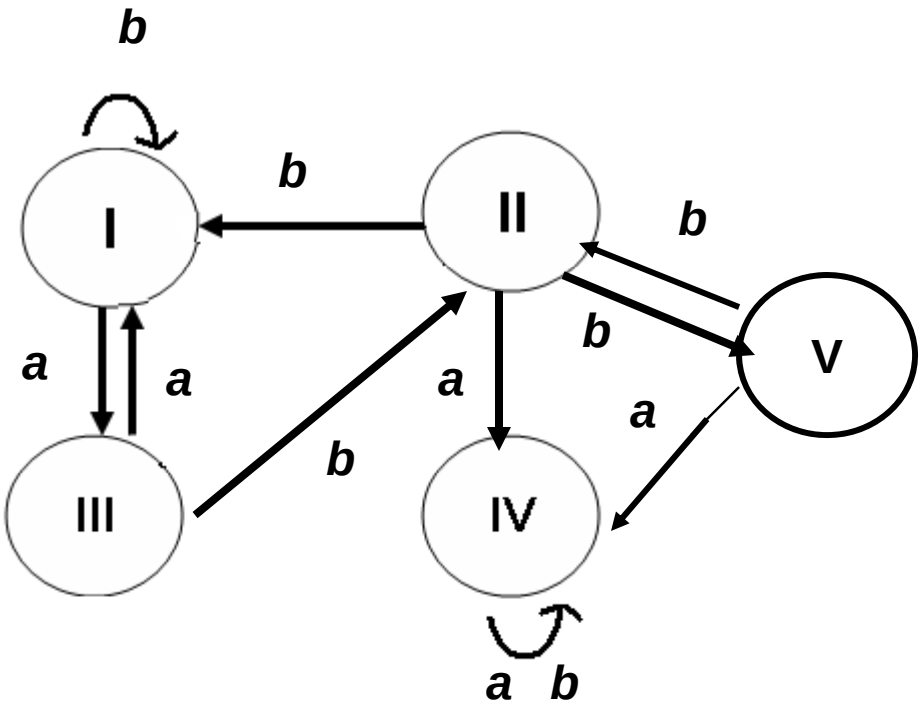
- All states within each of the 5 partitions are identical.
- We might as well call the states I, II III, IV and V.

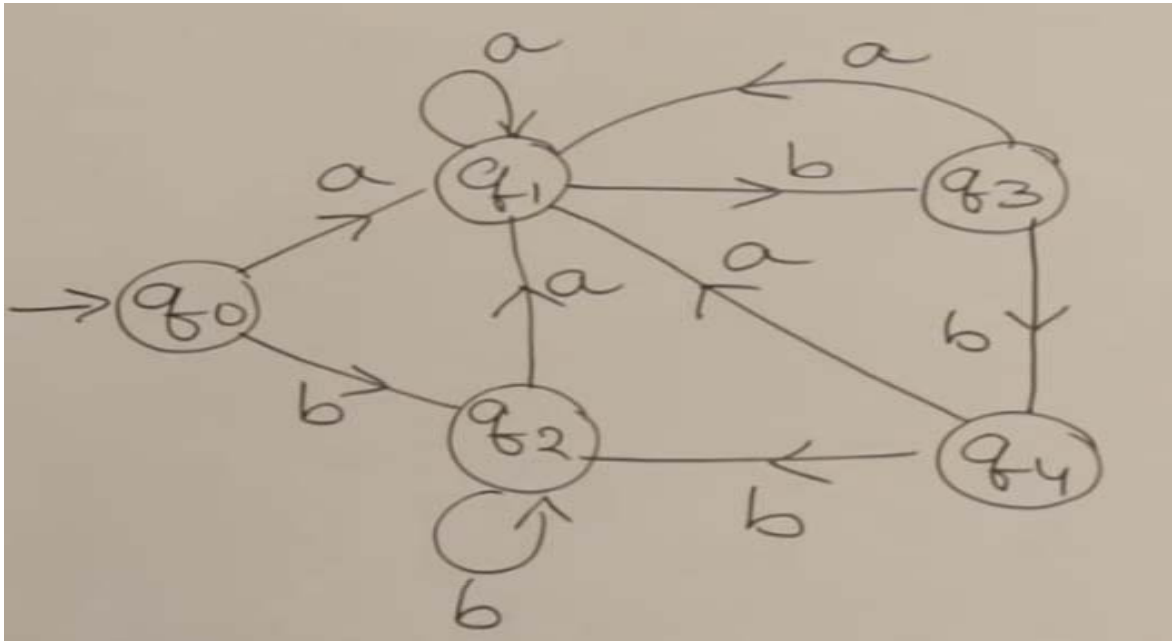
	a	*b*
S_0	S_1 III	S_4 I
S_5	S_1 III	S_4 I
S_4	S_1 III	S_4 I
S_3	S_3 IV	S_3 IV
S_1	S_5 I	S_2 II
S_6	S_3 IV	S_7 II
$*S_2$	S_3 IV	S_6 V
$*S_7$	S_3 IV	S_6 V

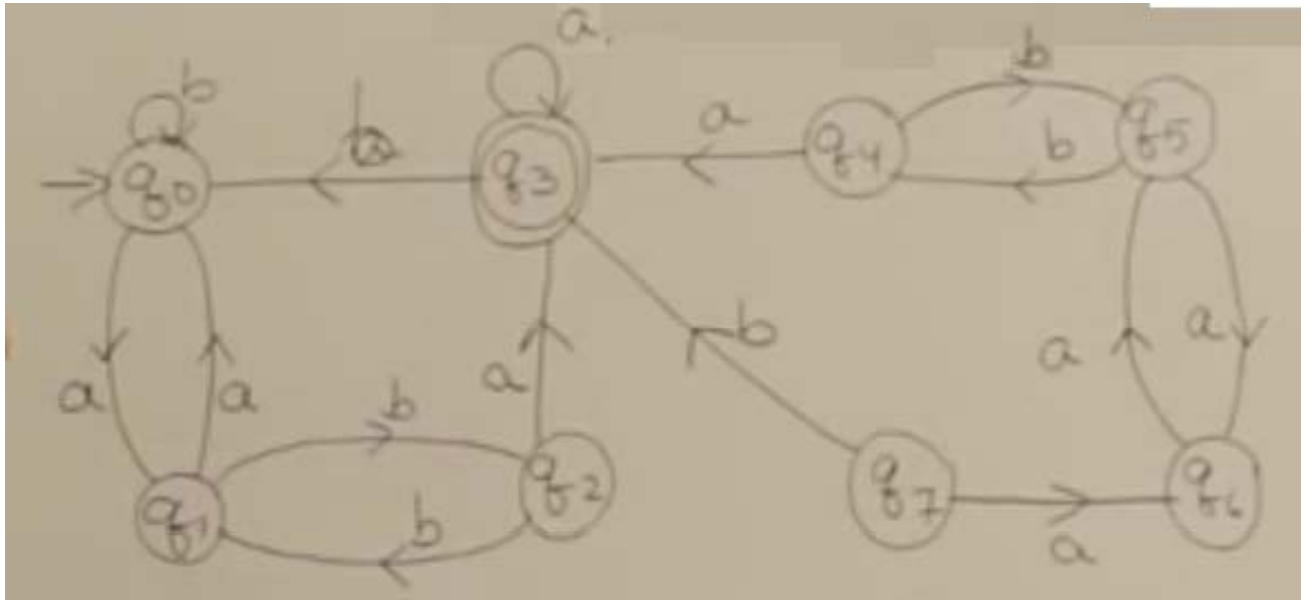
Partitioning Algorithm

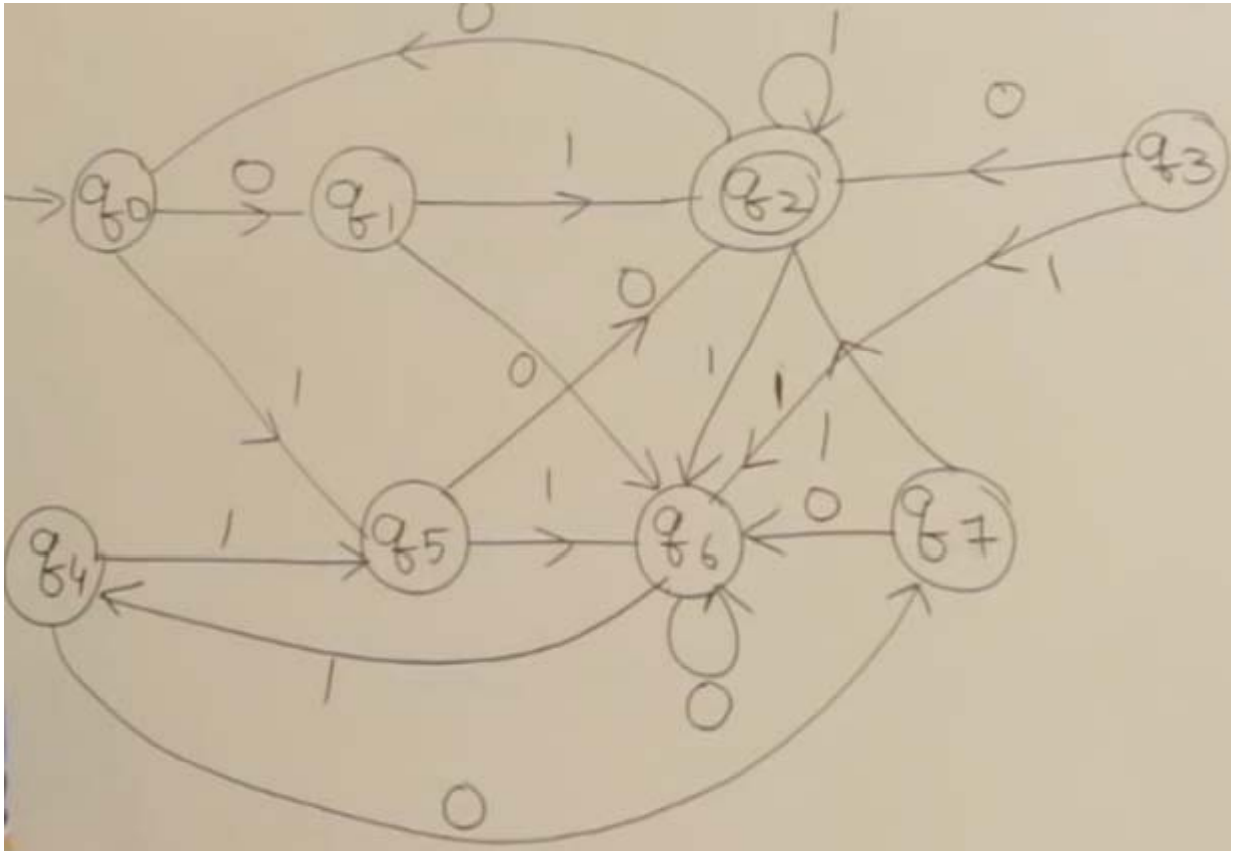
Here they are:

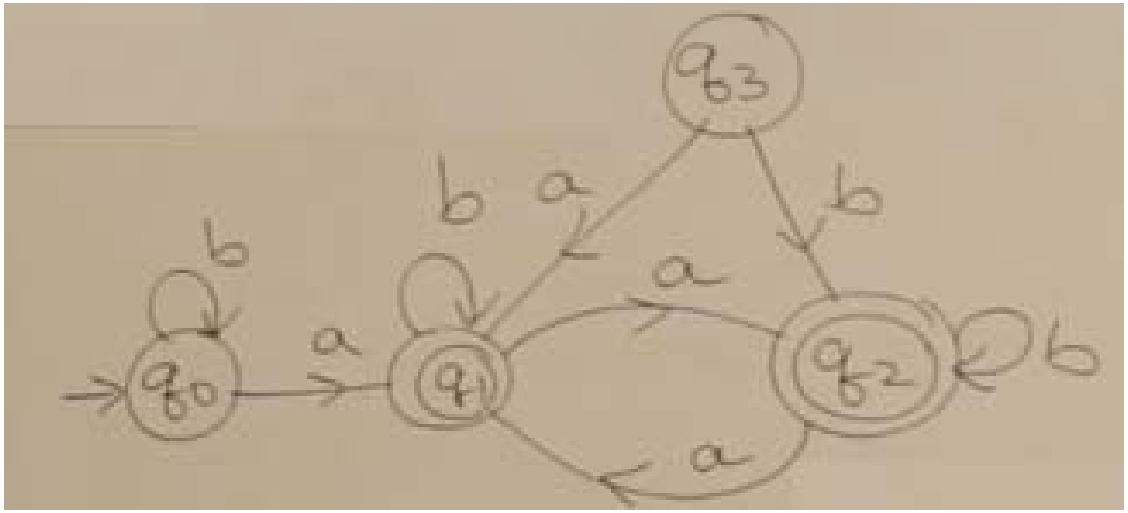
	a	*b*
I	III	I
*II	IV	V
III	I	II
IV	IV	IV
V	IV	II


b
I
b
II
b
b
V
a
a
a
a
b
III
IV
a
b


a
a
a
b
a
a
b
b
b
b


a
b
b
a
b
b
a
a
b
a
a
b
a
a
b
a


q0
q1
q2
q3
q4
q5
q6
q7
0
1


q0
q1
q2
q3
a
b