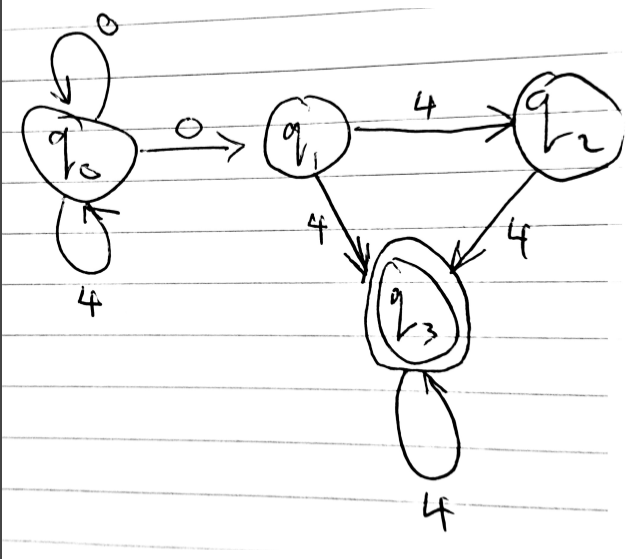
Coursework 1:

Question 1 :

A = ({0,4}, {(q0, q1, q2, q3)}, {(q0, 0, q0), (q0, 4, q0), (q0, 0, q1), (q1, 4, q2), (q1, 4, q3), (q2, 4, q3), (q3, 4, q3)}, q0, {q3}).

1. Draw the automaton:



1. Give three words included in the language and three words not included in the language.

Included: Not included:

0404 4444

0004 0000

4044 4440

1. Describe in English the language A.

The language begins with any number of 0’s or 4’s, followed by a ‘0’ and at least one ‘4’.

1. Give a regular expression for the language accepted by A

R = (0 + 4)\*0(4+44)4\*

1. Explain what makes A non-deterministic.

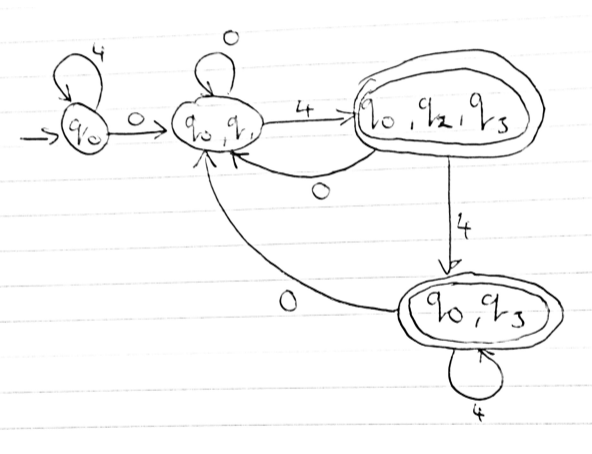
A is non-deterministic due to the state q1 having two transitions from it with the same input, but different destinations.

1. Give as a transition graph, a deterministic FSA accepting A.

We can use a subset construction to find the DFA.

|  |  |  |
| --- | --- | --- |
| δ | 0 | 4 |
| q0 | {q0, q1} | {q0} |
| q1 | { } | {q2, q3} |
| q2 | { } | {q3} |
| q3 | { } | {q3} |

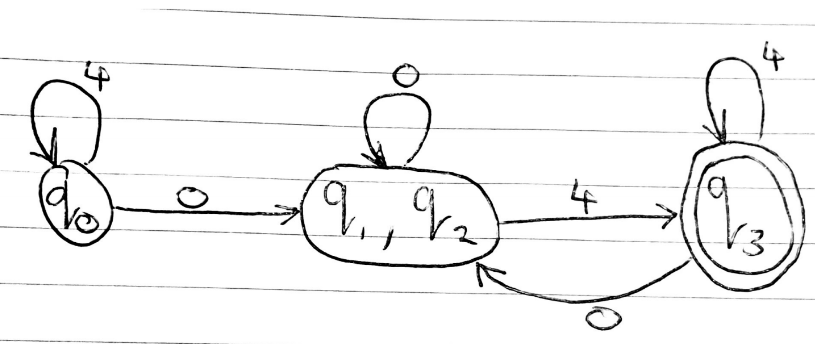
|  |  |  |
| --- | --- | --- |
| δ’ | 0 | 4 |
| q0 | {q0, q1} | {q0} |
| q0, q1 | {q0, q1} | {q0, q2, q3} |
| q0, q2, q3 | {q0, q1} | {q0, q3} |
| q0, q3 | {q0, q1} | {q0, q3} |



1. Give as a transition graph, a minimal deterministic FSA accepting the same language A.

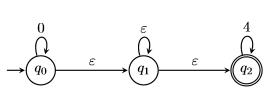
Marking table construction:

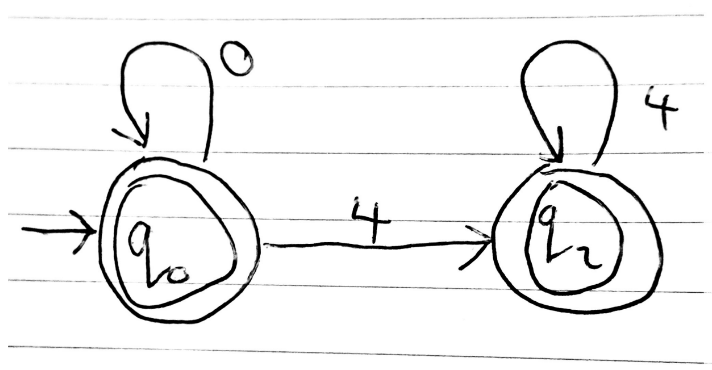
|  |  |  |  |
| --- | --- | --- | --- |
| q1 | X |  |  |
| q2 | X |  |  |
| q3 | X | X | X |
|  | q0 | q1 | q2 |



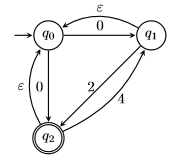
The FSA on the left can use just empty transitions to get to the final state. This means it can produce an empty word. The word produced can have any number of 0’s (including none). The word produced does not necessarily need to include a 4, but if it does it moves to another final state. Just like the previous automaton, if the word contains a 4, it cannot be followed by a 0.

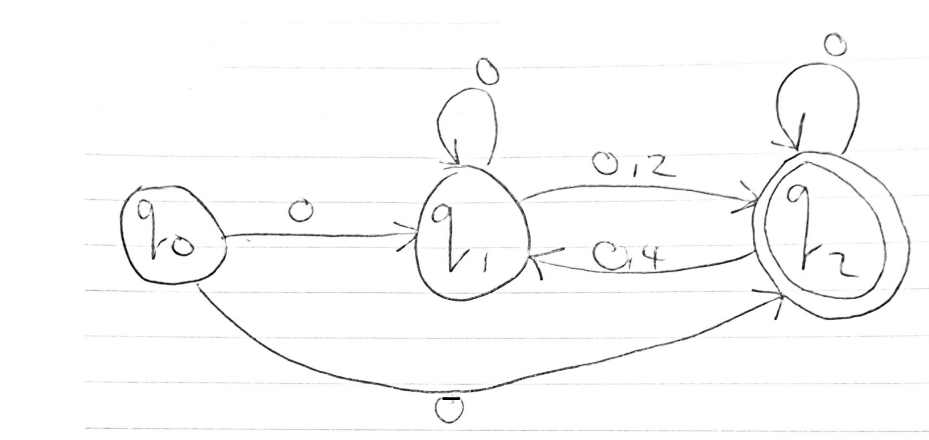
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| δ | 0 | 4 | ε | ε’ |
| q0 | {q0} | { } | {q1} | {q0, q1, q2} |
| q1 | { } | { } | {q1, q2} | {q1, q2} |
| q2 | { } | {q2} | { } | {} |





The FSA on the right can use empty transitions to return to the starting state. We can remove the need for these by introducing a 0 loop to the final state. We also need to make a 0 transition from q1 to q2.





Question 2:

1. There is still an alphabet of 2, there are now 7 states (q0 to q6), there are more transitions, the initial state is q0 and the final state is q6.

i)

A.alphabet\_size=2;

A.n\_states=7;

A.delta=new[][] {(0,1,0), (0,-1,1), (1,0,2), (1,-1,6), (1,-1,2), (3,1,5), (4,0,5), (4,-1,6), (5,0,5), (5,1,5), (6,-1,6), (6,1,5)};

A.initial\_state=0;

A.final\_states=new int[]{6};

ii)

A.alphabet\_size=3;

A.n\_states=6;

A.delta=new[][] { 0,2,0),(0,-1,1),(1,-1,2),(2,0,3), (3,1,0),(0,0,4),(4,-1,5),(5,2,0)

A.initial\_state=0;

A.final\_states=new int[]{0, 5};

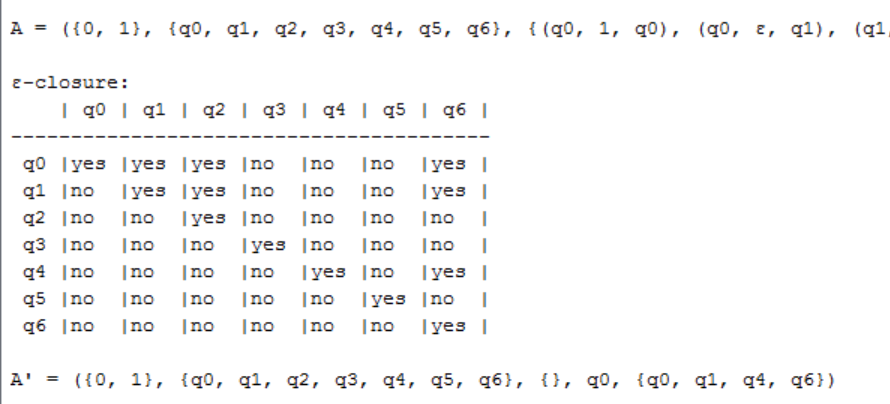
* 1. FSA i):

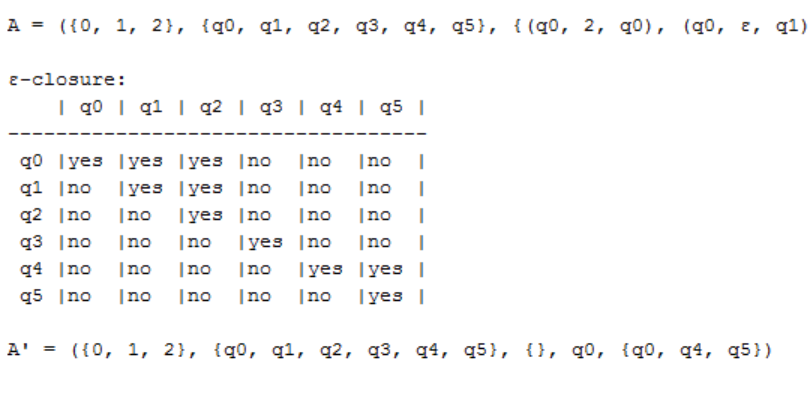
|  |  |  |  |
| --- | --- | --- | --- |
|  | q0 | q1 | q2 |
| q0 | Yes | Yes | Yes |
| q1 | No | Yes | Yes |
| q2 | No | No | Yes |

FSA ii):

|  |  |  |  |
| --- | --- | --- | --- |
|  | q0 | q1 | q2 |
| q0 | Yes | No | No |
| q1 | Yes | Yes | No |
| q2 | Yes | No | Yes |

* 1. FSA i:



FSA ii)

* 1. Matrix i) q0 can reach the most states by empty transitions.

Matrix ii) q0 can reach the most states by empty transitions.

1. n