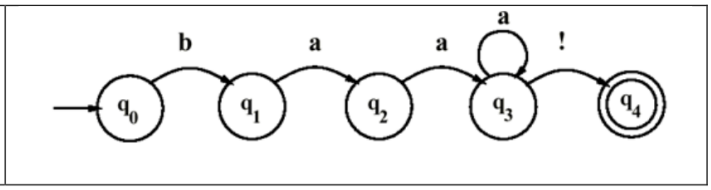
A Deterministic Finite State Automaton (DFA), often referred to as simply a Finite State Automaton may be defined as follows:

- The set of states: Q
- A finite alphabet: Σ
- A start state
- A set of final states
- A transition function that maps Q × Σ to Q.



The sheep language as defined in the notes may be represented in the State Diagram above right:

A program is provided (on the student share) to implement this DFA using a text files where each line represents a tuple from the DFA. *sheep.txt* thus looks like this:

```
5          ———— Number of States
4          ———— List of final States
ab!        ———— Alphabet
0 b 1      ┐
1 a 2      │
2 a 3      │ Set of Transition functions
3 a 3      │
3 ! 4      ┘
```

You are asked to run the program, *DFATest.java,* to open the *Sheep.txt* file and test the entry that it obeys the morphological rules of the sheep language. Do the same to test the phone number

Create a deterministic finite state automaton (DFA) to represent standard mobile phone numbers with the following structure: (08X)-XXX XXXX where the space, hyphen and brackets are included as part of the alphabet of symbols. The mobile providers should be (087), (086) and (085)

The next digit can only be either be 8 for Dublin or 7 for outside Dublin. So, for example the following numbers are acceptable:

(085)-976 1234 (087)-789 2345 (086) 876 2345

The following numbers are not acceptable: (089)-976 1234 or (087)-9761234 *(no space)*

Your answer should include the following:

(i) A formal definition of the DFA
(ii) A state diagram
(iii) A state transition table

The state diagram

is given below. At present the program does not account for the space, nor does it allow you to use symbols in the form of words. You are required to:

(i) Amend the program to allow the space to be entered in the program