# Exercise 2

1. The set FM contains all final states of the DFA M. It is possible to build a DFA M’ such that M’ accepts all strings that are rejected by M by doing the following:

* FM’ = QM - FM

In other words, the accepted states of M’ must be all those states of M that are not accepted. For all the other aspects, M and M’ must behave in the same way.

1. Let A be a regular set. If A is a regular set then there must exist a FINITE AUTOMATA N such that

Without loss of generality and for simplicity we are going to assume that N is a DFA.

If A=L(N) then where s is the INITIAL STATE of N and F is the set of the ACCEPT STATES of N.



In other words, for every string in A, this string leads N to an ACCEPT STATE starting from the INITIAL STATE s.

Let Σ denote the INPUT ALPHABET of N.

It follows that Σ\*\A is the set of all strings that lead N starting from its initial state s to a NON-ACCEPT STATE and Σ\*\A is equivalent to the complement of A (**~**A)

In other words, .



Let Q be the set of the states of N.

Thus, the NONACCEPT STATES are Q/F.

If we build a DFA M such that it has the same characteristics of N but its set of ACCEPT STATES is Q/F



then **~**A = L(M) because we have said before that and so every x of leads to an accept state of M.

Thus, we have proved that give that A is a regular set then **~**A is a REGULAR SET as well.

# Exercise 3

1. The set Σ\* is regular.

Proof:

Let M be a DFA such that M =(Q, Σ, δ, s, F) where

* + Q = {0}
  + δ (0,a) = 0 for all a that belong to Σ
  + s = 0
  + F = {0}

It turns out that L(M) = Σ\*

The set ∅ is a regular language as all DFAs that have no accepted states have the empty set as their accepted language.

The set {ε} is a regular language as there exists a DFA N such that L(N) = {ε} which does not accept any inputs and its starting state is also an accepted state.

The other proofs are in Lecture 2…

1. If L1 and L2 are REGULAR then L1L2 is regular as well

Proof:

Let N be the DFA such that L(N) = L1 and M be the DFA such that L(M) = L2.

Let O be the DFA such that O = (QO, Σ, δ, s, F) where:

* QO = QN ∪ (QM - sM)
* δO (q,a) = δN(q,a) if q belongs to QN – FN

δM(q,a) if q belongs to QM

δM(sM,a) if q belongs to FN

* s = sN
* F = FM

Explanation:

Since there exists a DFA N such that L(N) = L1 and a DFA M such that L(M) = L2 then it is possible to create a DFA O such that L(O) = L1L2. O consists of merging N and M together such that the initial state of O is the initial state of N as well. From the initial state O will follow the transition function of N. Once, an accepted state of N is reached then the DFA M is attached to this final state of N such that each final state of N becomes the initial state of M. From the final states of N (which are initial states of M) the DFA O follows the transition function of M.

The language accepted by O is such that L(O) = L1L2

1. If L is regular then L\* is regular as well.

Proof:

Given that L is regular then there exists a DFA N such that L(N)=L.

Let M be a ε-transition NDFA such that it behaves exactly in the same way as N only that the accepted states of M (which are accepted states of N as well) are linked to the starting state of M (which is the starting state of N as well) through ε-transitions.

It follows that L(M) = L\*.

More formally:

N = (QN,Σ,δN,sN,FN)

then

N = (QM,Σ,δM,sM,FM)

where:

* QM = QN
* δM = δN ∪ (f,ε,sM) such that f belongs to FM. In other words, all accepted states have ε-transitions with the starting state of M
* sM= sN
* FM = FN