Comp 330 Assignment 1

# Question 1

## Question 1.1

Equivalence class R:

* Reflexive:
* Symmetry:
* Transitive:

Preorder:

* Reflexive
* Transitive

**Proof:**

* Reflexive

As R is a preorder its reflexive then ~ is also reflexive

* Symmetric

Then ~ is symmetric

* Transitive

Then ~ is transitive and thus an equivalence class

## Question 1.2

We have

* the set of element equivalent to a with ~ equivalence
* the set of element equivalent to b with ~ equivalence

**Case 1: a is equivalent to b**

Then by definition of equivalent classes

Then for any

Thus it’s well defined in this case

**Case 2: a is not equivalent to b**

Then

Then for any

Thus its well defined

## Question 1.3

But as then are the same set () and then is anti-symmetric

# Question 2

## Question 2.a

* Reflexive

Let

For all then the first condition cannot be competed

However x is a prefix of x then

* Transitive

Let and and

We also have

Let such that either

Let such that either

We know that for all

We know that for all

If then for all but as and then

If then for all but as and then

If then either x = y or y = z and thus

Similarly if is a prefix of or is a prefix of we get

* Anti-symmetric

Let and

Let then

* and
* and

However those two conditions are not possible to satisfy at the same time.

Similarly if x is shorter than y then the conditions won’t be satisfied. Same for y shorter than x

Then and as the first conditions showed for all then

So it’s anti-symmetric

## Question 2.b

It’s not well founded.

Indeed we can have a sequence of a and b named

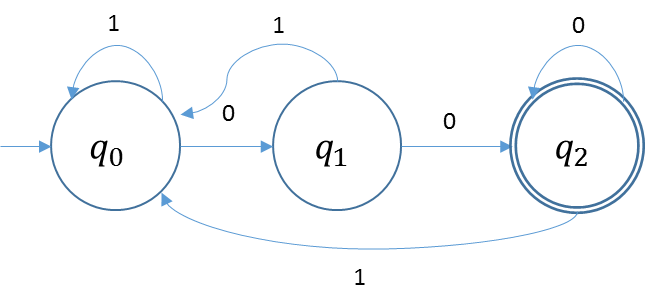
Let be the word compose of

But is smaller than .

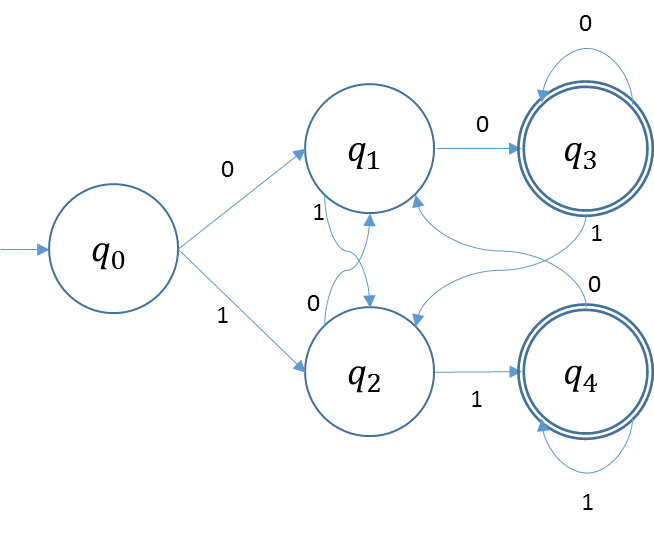
We can repeat this step indefinitely so there is no minimal element

# Question 3

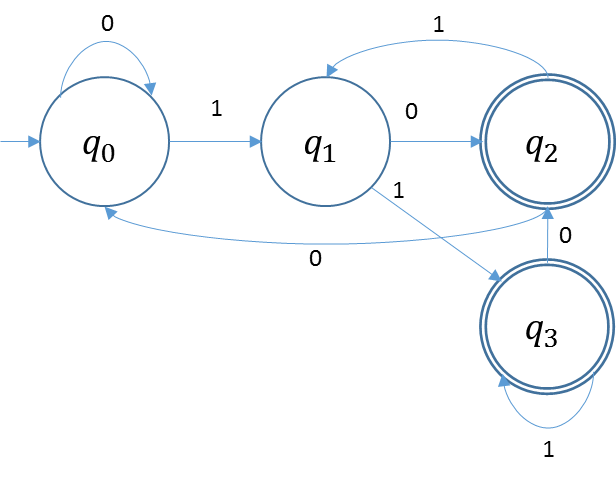
## Question 3.1



## Question 3.2



## Question 3.3



# Question 4

Let be a machine that recognize

Let with

* is a new state that has transistions to all final states in
* all transitions in M are reversed

Let be a machine that recognize

Let with

New transitions have been added that loop on the final state

# Question 5

Let be a machine that recognize

Let be a machine that recognize

* . So keep track of the current state of as well as the state that can reach an accept state of in i move steps where i is the current length of the word we are reading
* we have an move and note that we are 0 steps from an accepted state
* So we accept only if the current state is at x, and x is at i steps from an accept state of
* where that satisfies the previous equation,