Timothee Guerin
260447866

Comp 330 Assignment 1

Question 1

- ### Question 1.1

Equivalence class R:

- Reflexive: xRx
- Symmetry: $xRy \iff yRx$
- Transitive: $xRy \,\&\, yRz \Rightarrow xRz$

Preorder:

- Reflexive
- Transitive

Proof:

- Reflexive

$$a \sim a \Leftrightarrow a\ R\ a\ \&\ a\ R\ a$$

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

- Symmetric

$$a\ R\ b\ \&\ b\ R\ a\ \Leftrightarrow a \sim b$$

$$b\ R\ a\ \&\ a\ R\ b\ \Leftrightarrow b \sim a$$

$$a\ R\ b\ \&\ b\ R\ a\ \Leftrightarrow a \sim b$$

Then ~ is symmetric

- Transitive

$$a\ R\ b\ \&\ b\ R\ c \rightarrow a\ R\ c$$

$$c\ R\ b\ \&\ b\ R\ a \rightarrow c\ R\ a$$

$$a\ R\ b\ \&\ b\ R\ a\ \&\ b\ R\ c\ \&\ c\ R\ b \rightarrow a\ R\ c\ \&\ c\ R\ a$$

$$a \sim b\ \&\ b \sim c \rightarrow a \sim c$$

Then ~ is transitive and thus an equivalence class

Type equation here.

- ### Question 1.2

We have

- $[a]$ the set of element equivalent to a with ~ equivalence
- $[b]$ the set of element equivalent to b with ~ equivalence

Case 1: a is equivalent to b

Then by definition of equivalent classes $[a] = [b]$

Then for any $x \in [a] and\ y \in [b], x \sim y \rightarrow xRy$

Thus it's well defined in this case

Case 2: a is not equivalent to b

Then $[a] \neq \ [b]$

Then for any $x \in [a] and\ y \in [b], not\ x \sim y \rightarrow not\ xRy$

Thus its well defined

- ### Question 1.3

$$[a] \leq [b] \, \& \, [b] \leq [a]$$

$$aRb \, \& \, bRa$$

$$a \sim b$$

But as $a \sim b$ then $[a]\ and\ [b]$ are the same set ($[a] = [b]$) and then $\leq$ is anti-symmetric

Question 2

- ### Question 2.a
- Reflexive

Let $x = x_1 x_2 \dots x_n$

For all $i, x_i = x_i$ then the first condition cannot be competed

However x is a prefix of x then $x \leq x$

- Transitive

Let $x = x_1 x_2 \dots x_n$ and $y = y_1 y_2 \dots y_m$ and $z = z_1 z_2 \dots z_s$
We also have $x \leq y\ and\ y \leq z$
Let i such that either $x_i = a\ and\ y_i = b$
Let j such that either $y_j = a\ and\ z_j = b$
We know that for all $k < i\ ,\ x_k = y_k$
We know that for all $k < j\ ,\ y_k = z_k$

If $i < j$ then for all $k < i, x_k = z_k$ but as $x_i = a \text{ and } y_i = b \text{ and } y_i = z_i \text{ then } z_i = b$ and then $x \leq z$
If $i > j$ then for all $k < j, x_k = z_k$ but as $y_j = a \text{ and } z_j = b \text{ and } x_j = y_j \text{ then } x_j = a$ and then $x \leq z$
If $i = j$ then either x = y or y = z and thus $x \leq z$

Similarly if x is a prefix of y or y is a prefix of z we get $x \leq z$

- Anti-symmetric

Let $x = x_1 x_2 \ldots x_n$ and $y = y_1 y_2 \ldots y_m$

Let $x \leq y \text{ and } y \leq x$ then

- $\exists i \ s.t \ x_i = a \text{ and } y_i = b$ and $\forall \ k < i, x_k = y_k$
- $\exists j \ s.t \ x_j = b \text{ and } y_j = a$ and $\forall \ k < j, x_k = y_k$

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

Similarly if x is shorter than y then the $y \leq x$ conditions won't be satisfied. Same for y shorter than x

Then $m = n$ and as the first conditions showed for all $k, x_k = y_k$ then $x = y$

So it's anti-symmetric

- ### Question 2.b

It's not well founded.

Indeed we can have a sequence of a and b named $w = aa \ldots ab$

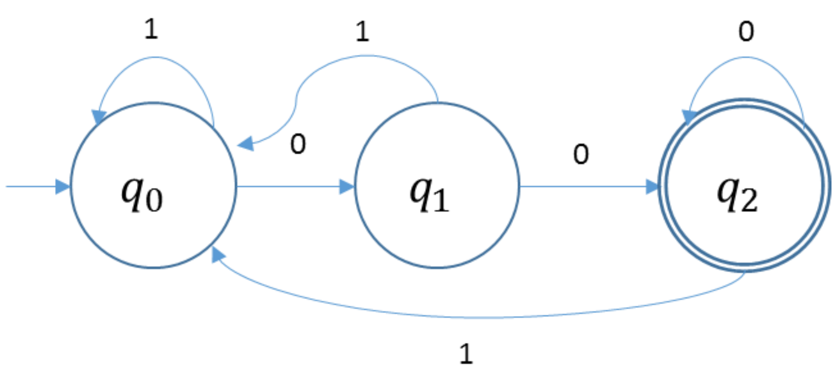
Let w' be the word compose of $a \text{ and } w: w' = aaa \ldots ab$

But w' is smaller than w.

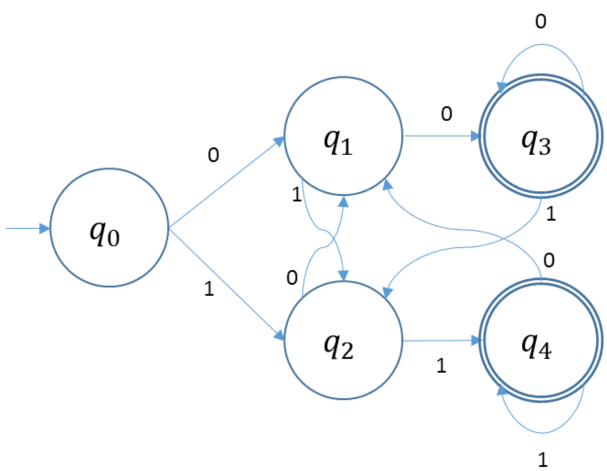
We can repeat this step indefinitely so there is no minimal element

Question 3

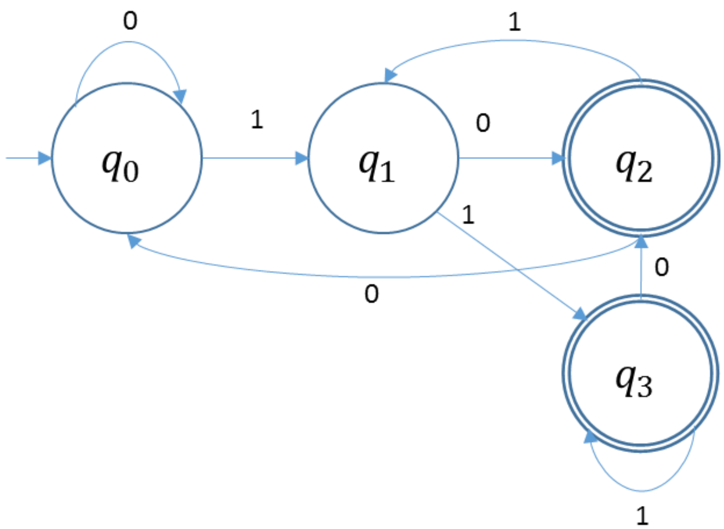
- ### Question 3.1



- ### Question 3.2



- Question 3.3



Question 4

- $Rev(L)$

Let $M(S, s_0, \delta_1, F)$ be a machine that recognize L

Let $N(T, t_0, \delta_2, F_0)$ with

- $T = S \cup \{t_0\}$
- t_0 is a new state that has ϵ transistions to all final states in F
- $F_0 = s_0$
- $\delta_1(x, a)$ all transitions in M are reversed

- $Init(L)$

Let $M(S, s_0, \delta_1, F)$ be a machine that recognize L

Let $N(T, t_0, \delta_2, F_0)$ with

- $T = S$
- $t_0 = s_0$
- $F_0 = F$
- $\delta_1(x, a) = \delta_2(x, a)$

New transitions have been added that loop on the final states

Question 5

Let $M(S, s_0, \delta_1, F)$ be a machine that recognize L

Let $N(T, t_0, \delta_2, F_0)$ be a machine that recognize $lefthalf(L)$

- $T = S * S \cup \{t_1\}$. So $S * S$ keep track of the current state of M as well as the state that can reach an accept state of M in i move steps where i is the current length of the word we are reading
- t_0 we have an ϵ move s_0 and note that we are 0 steps from an accepted state
- $F_0 = \{(x, x) |\ x \in S\}$ So we accept only if the current state is at x, and x is at i steps from an accept state of M
- $(\delta_1(x, a), z) = (\delta_2(x, y), a)$ where $\forall z$ that satisfies the previous equation, $\exists\ c \in \Sigma\ with\ \delta_2(z, c) = y$