Discrete Maths Assignment 4
Brandon Dooley (#16327446)

1)

(a) Is $\{(x, \sin(\pi x))\ x \in R\} \cap \{Q \times Z\}$ finite, countably infinite, or uncountably infinite? Justify your answer.

This set is countably infinite as both Q and Z are countably infinite sets, as a result the Cartesian product of these two sets also produces a set that is countably infinite.

Although the set $\{(x, \sin(\pi x))\ x \in R\}$ in itself is uncountably infinite, it's intersection with the set $\{Q \times Z\}$ produces a set that is within the bounds of $\{Q \times Z\}$ which as discussed above is countably infinite.

(b) Is $\{(x, y) \in R \mid xy = 1\}$ finite, countably infinite, or uncountably infinite? Justify your answer.

This set R itself is an uncountably infinite set as a result of the set of all irrational numbers being a subset of R.

A set is countably infinite iff $\exists f : A \rightarrow J$ a bijection, namely $A \sim J$ s.t J is the set of Natural Numbers.

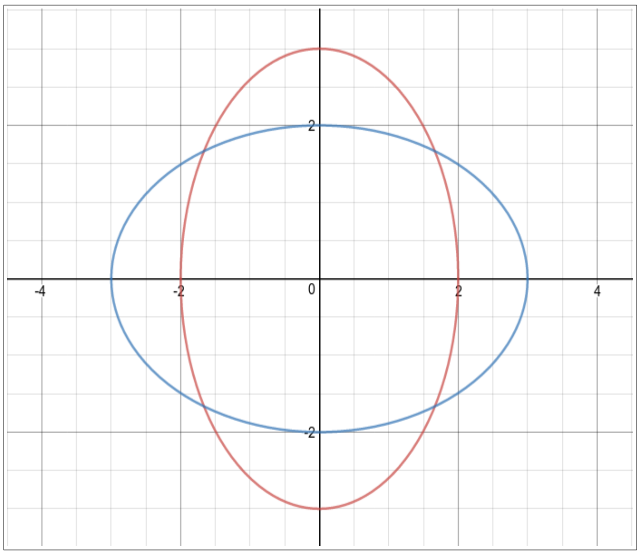
As the bounds of this set are confined to $xy=1$ this implies that all elements of the set follow the form (x, x) since $xy=1$ *iff* $x = y = 1$. This yields in itself an finite set containing only one element, namely the co-ordinate (1,1).

(c) Is $\{(x, y) \in R^2 \mid x^2/4 + y^2/9 = 1\} \cap \{(x, y) \in R^2 \mid x^2/9 + y^2/4 = 1\}$ finite,countably infinite, or uncountably infinite? Justify your answer.

The set R^2 itself is uncountably infinite. Since, as discussed above, the set R is uncountably infinite this yields that the Cartesian product R^2 is also uncountably infinite.

However, the set $\{(x, y) \in R^2 \mid x^2/4 + y^2/9 = 1\}$ *and the set* $\{(x, y) \in R^2 \mid x^2/9+ y^2/4 = 1\}$ are the sets containing the co-ordinates of two ellipses, the first of which being with a vertical major axis and the second of which being with a horizontal major axis.

The intersection of these two sets is namely the set containing the co-ordinates of the intersections of the two ellipses which is a finite set containing four elements. This can be illustrated in the following graph.



(c) Let A = {0, 1}. Is (0 ∗ ◦ 1 ∗) ∩ {A ∗ ◦ 11 ◦ A ∗ } finite, countably infinite,or uncountably infinite? Justify your answer.

(0* ◦ *1*)* describes a word containing an infinite amount of 0's concatenated with an infinite amount of 1's e.g 0001111. This set itself is countably infinite as it is a language given under the regular expression (0* ◦ *1*)* and the set of all regular languages over a finite alphabet is countably infinite..

{A*} is the set of all words over A. Since A is a finite alphabet, the set {A*} is a countably infinite set.

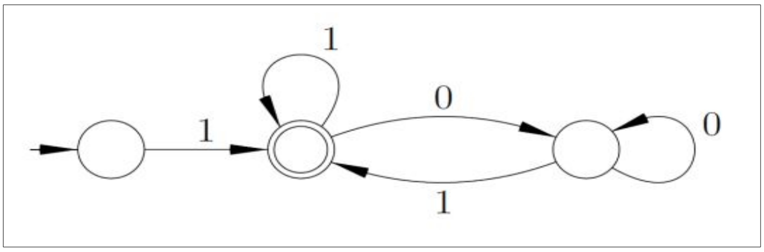
{A ∗ ◦ 11 ◦ A ∗ } is also a countably infinite set, similar to what was discussed above it is a language defined under a regular expression and a regular language over a finite alphabet is countably infinite.

As a result the intersection of (0* ◦ *1*)* and *{A ∗ ◦ 11 ◦ A ∗ }* yields a set that is also countably infinite since the intersection of two countably infinite sets is itself a countably infinite set.

2)

(a) Let L be the language consisting of the binary representations of all odd natural numbers. Write down the algorithm of a Turing machine that recognizes L. Process the following strings according to your algorithm: empty string, 0, 11, and 100.

Binary representations of odd natural numbers can be described under the following expression $1\cup1(0\cup1)*1$. An example of a finite state acceptor that accepts binary representations of odd natural numbers can be found below[1]. This representation does not account for leading zeros.



This can be described in the form of a Turing Machine M which performs the following algorithm:

1. If anything other than 1 is in the first cell REJECT.
2. Move to the end of the string (the next _ character), then move left.
3. If 1 is in the current cell ACCEPT. Else REJECT.

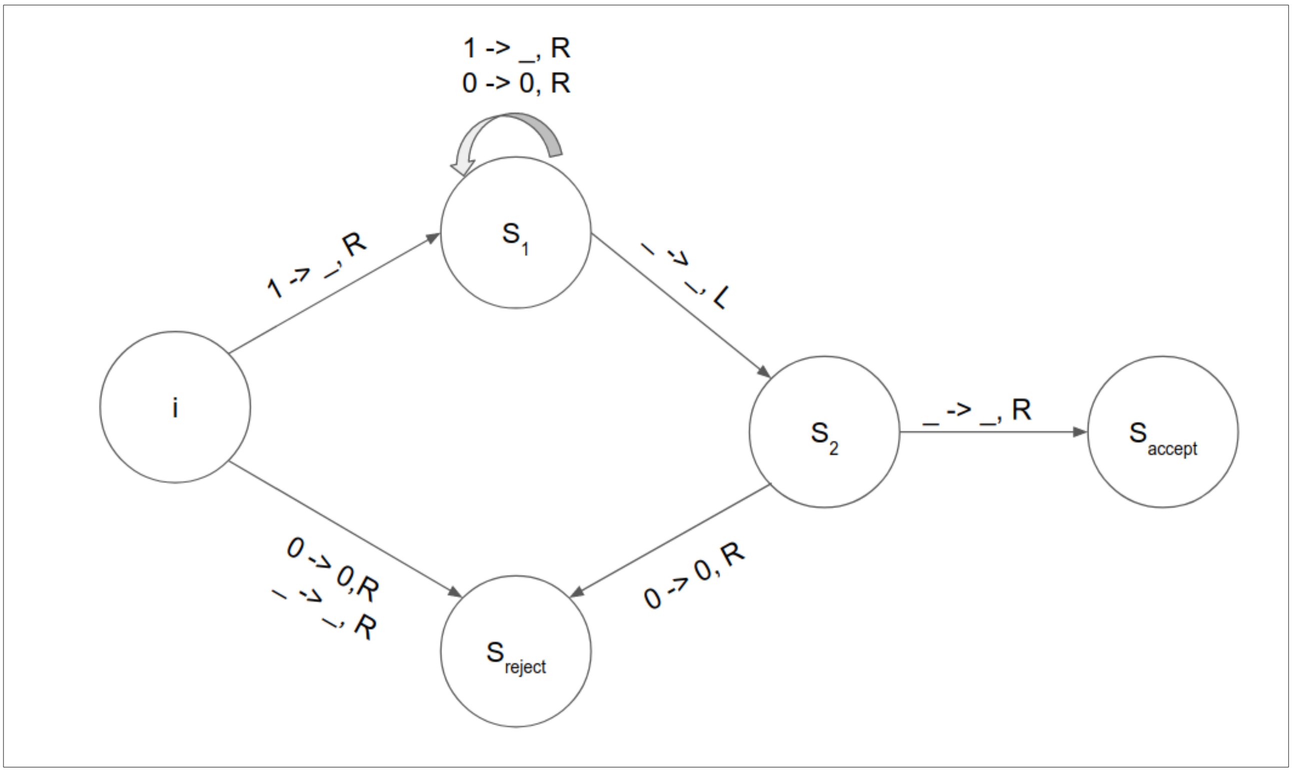
Processing Empty Word	Processing '0'	Processing '11'	Processing '100'
[_] REJECT - Step 1	[0] REJECT - Step 1	[1][1][_] Step 1	[1][0][0][_] Step 1
		[1][1][_] Step 2	[1][0][0][_] Step 2
		[1][1][_] Step 2	[1][0][0][_] Step 2
		[1][1][_] ACCEPT Step 3	[1][0][0][_] REJECT Step 3

1 https://cs.stackexchange.com/questions/10983/regular-expression-for-odd-binary-numbers-without-leading-zeros?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

(b) Write down the transition diagram of the Turing machine from part (a) carefully labelling the initial state, the accept state, the reject state, and all the transitions specified in your algorithm.

The Turing Machine above can be described under the 7-tuple (S, A, Ã, t, i, S_{accept}, S_{reject}):

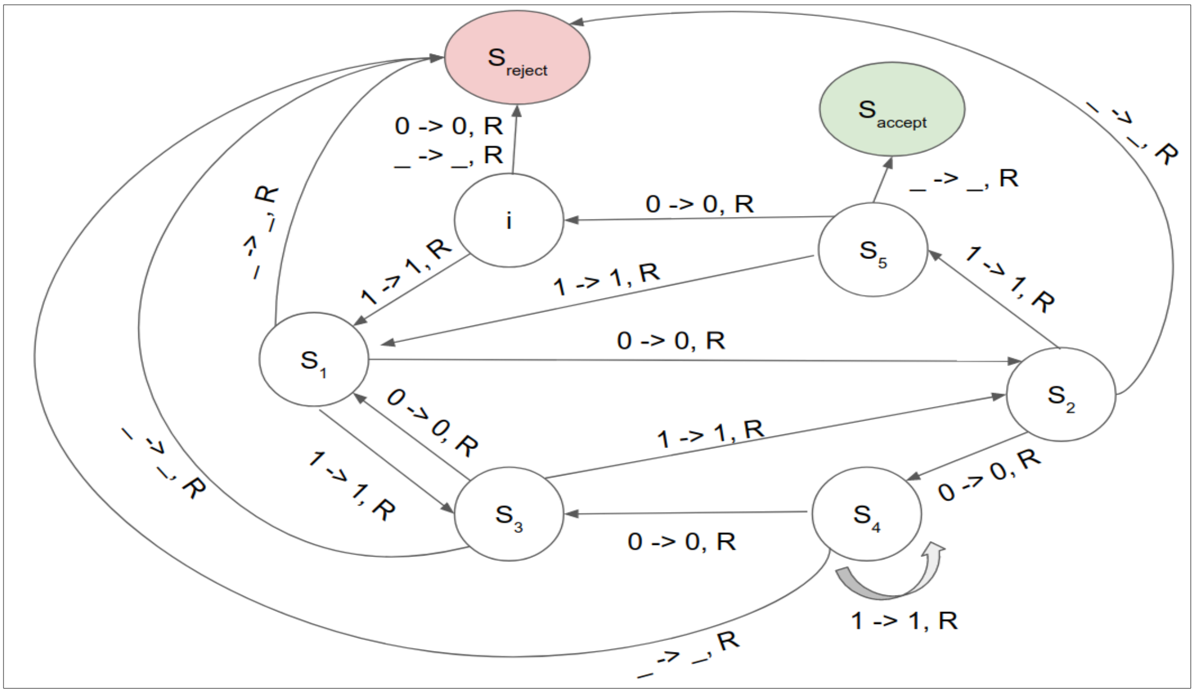
S ={$S_{1,}$ S_2}
A = {0, 1}
Ã = { _ }
t = t : S x Ã → S x Ã x {L, R}
I = S_i
S_{accept} = S_{accept}
S_{reject} = S_{reject}



(b) Write down the algorithm of an enumerator that prints out EXACTLY ONCE every odd natural number divisible by 5. (Hint: Order the words on the input tape.)

-The Turing Machine as described above needs to be somewhat changed as more states needed to be added. The new and updated Turing Machine M is described under the following model:

$S = \{S_1, S_2, S_3, S_4, S_5\}$
$A = \{0, 1\}$
$\tilde{A} = \{ \_ \}$
$t = t : S \times \tilde{A} \rightarrow S \times \tilde{A} \times \{L, R\}$
$I = S_i$
$S_{accept} = S_{accept}$
$S_{reject} = S_{reject}$



<u>Algorithm of Enumerator:</u>

Let L be the language L = $1 \cup 1(0 \cup 1)^*1$. Let M be the above Turing Machine that recognizes all words under L that are binary representations of odd numbers divisible by 5. We construct an enumerator E that outputs these words.

Let A be the alphabet of L{0, 1}. A* has an enumeration as a sequence A* = $\{w_1, w_2, \ldots\}$.

1. Repeat the following for k = 1, 2, 3, ...
2. Run M, passing it the input word w_k
3. If any computations accept, print out the corresponding w_k.

Every string accepted by M (every binary representation of every odd natural number) will appear on the list of E.