MA2C03: ASSIGNMENT 4
DUE BY FRIDAY, APRIL 21
SOLUTION SET

BRIAN TYRRELL

Q1

(a) Let $X = \{(\frac{m}{2}, \frac{n}{3}) \in \mathbb{R}^2 \ : \ m, n \in \mathbb{Z}\}$. There is a one-to-one correspondence between X and $\mathbb{Z} \times \mathbb{Z}$:

$$f : \mathbb{Z} \times \mathbb{Z} \to X \qquad \text{by} \qquad (m, n) \mapsto \left(\frac{m}{2}, \frac{n}{3}\right),$$

$$f^{-1} : X \to \mathbb{Z} \times \mathbb{Z} \qquad \text{by} \qquad (x, y) \mapsto (2x, 3y).$$

Therefore $\#X = \#\mathbb{Z} \times \mathbb{Z} = \#\mathbb{Z} = \#\mathbb{N}$ by previous theorems, making X countably infinite.

(b) Let $Y = \{(x, y) \in \mathbb{R}^2 \ : \ y = x^2\} = \{(x, x^2) \in \mathbb{R}^2\}$. There is a one-to-one correspondence between X and $\mathbb{R}$:

$$f : X \to \mathbb{R} \qquad \text{by} \qquad (x, x^2) \mapsto x,$$

$$f^{-1} : \mathbb{R} \to X \qquad \text{by} \qquad x \mapsto (x, x^2).$$

Therefore $\#X = \#\mathbb{R}$ so X is uncountably infinite.

(c) $L_q = \{(q, y) \in \mathbb{R}^2\}$ is in one-to-one correspondence with $\mathbb{R}$, via $(q, y) \in L_q \leftrightarrow y \in \mathbb{R}$, meaning L_q is uncountably infinite. Therefore

$$\#L_q \leq \# \bigcup_{p \in \mathbb{Q}} L_p \Rightarrow \bigcup_{p \in \mathbb{Q}} L_p \text{ is uncountably infinite.}$$

(d) Recall $\mathcal{P}(\mathbb{N})$ is uncountably infinite. Note $\mathcal{P}(\mathbb{N})$ can be injectively embedded into $\mathcal{P}(J_n) \times \mathcal{P}(\mathbb{N})$ via:

$$f : \mathcal{P}(\mathbb{N}) \hookrightarrow \mathcal{P}(J_n) \times \mathcal{P}(\mathbb{N}) \qquad \text{by} \qquad A \mapsto (\emptyset, A),$$

So $\#\mathcal{P}(\mathbb{N}) \leq \#(\mathcal{P}(J_n) \times \mathcal{P}(\mathbb{N}))$ meaning $\mathcal{P}(J_n) \times \mathcal{P}(\mathbb{N})$ is uncountably infinite.

Remarks: Try to give a precise bijection in your answers. You get marks for correctly stating whether the set is finite, countably infinite or uncountably infinite, however the easiest way to get your *justification* marks is to be precise as to how the given set is in bijection to $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ or $\mathbb{R}$. 'It feels like it is' type answers get no marks.

Q2

(a) Here is the algorithm for recognising $L = \{0^m 1^{2m} : m \in \mathbb{N}\}$.

(1) If there is a blank in the first cell, ACCEPT. If there is anything else, apart from 0, then REJECT.
(2) If 0 is in the first cell, delete it, then move right to the first 1.
(3) If there is no first 1, REJECT. Otherwise change 1 to x.
(4) Move to the leftmost non blank symbol. If 0, go to step 2. If 1, REJECT. If x, go to step 5. If y, go to step 6.
(5) Delete x, move right to the nearest 1. If none, REJECT. Otherwise change it to y and go to step 4.
(6) Move right to the rightmost non blank character. If anything but y is found, REJECT. Otherwise, ACCEPT.

Here is how the following strings are treated:
- ϵ is accepted immediately.
- $01 \to \sqcup 1 \to \sqcup x \to \sqcup\sqcup \to$ REJECT.
- $011 \to \sqcup 11 \to \sqcup x1 \to \sqcup\sqcup 1 \to \sqcup\sqcup y \to$ ACCEPT.
- $010 \to \sqcup 10 \to \sqcup x0 \to \sqcup\sqcup 0 \to$ REJECT.

(b) The transition diagram for

$$T = (\{i, s_1, s_2, s_3, s_4, s_5, s_{\text{acc}}, s_{\text{rej}}\}, \{0,1\}, \{0,1,x,y\}, t, i, s_{\text{acc}}, s_{\text{rej}})$$

is at the end of the solution set, along with an example accepted string.

Remarks: For (b), large, clear, well labelled diagrams are a necessity. If it cannot be read, it cannot be awarded marks!

Q3

Note that $L = \{3m : m \in \mathbb{N}\} = \{m \equiv 0 \mod 3 : m \in \mathbb{N}\}$.

We can tell if a number is divisible by three by just examining its digits. We will use modular arithmetic heavily - recall the Michaelmas lectures and tutorials!

Note that for all $n \in \mathbb{N}, n \geq 1$, $10^n \equiv 1 \mod 3$ (also written $10^n \equiv_3 1$).

Consider 279. Is this divisible by 3?

$$279 = 200 + 70 + 9 = 2 \cdot 100 + 7 \cdot 10 + 9 \equiv_3 2 \cdot 1 + 7 \cdot 1 + 9.$$

To determine if 279 is divisible by 3, we just need to determine the conjugacy classes of its digits:

$$2 \equiv_3 2, 7 \equiv_3 1, 9 \equiv_3 0 \quad \to \quad 2 + 7 + 9 \equiv_3 2 + 1 + 0 \equiv_3 3 \equiv_3 0.$$

Therefore 279 is indeed divisible by 3. We see to determine this, we need only know the conjugacy classes of the digits $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$. Our Turing machine will reflect that.

Let $T = (\{s_0, s_1, s_2, s_{\text{rej}}, s_{\text{acc}}\}, A, A, t, s_0, s_{\text{rej}}, s_{\text{acc}}\}$. The states s_0, s_1, s_2 correspond to "congruent to 0, 1, 2 mod 3". Note that we begin in state s_0.

Here is the algorithm:

(1) If there is a blank in the first cell, REJECT. Otherwise move to step 2.

(2) If in state s_0, remain in s_0 if the digit is $0, 3, 6, 9$, then move right and go to step 3. If the digit is $1, 4$ or 7, move to state s_1, move right, and go to step 3. If the digit is $2, 5$ or 8, move to state s_2, move right, and go to step 3.

If in state s_1, remain in s_1 if the digit is $0, 3, 6, 9$, then move right and go to step 3. If the digit is $1, 4$ or 7, move to state s_2, move right, and go to step 3. If the digit is $2, 5$ or 8, move to state s_0, move right, and go to step 3.

If in state s_2, remain in s_2 if the digit is $0, 3, 6, 9$, then move right and go to step 3. If the digit is $1, 4$ or 7, move to state s_0, move right, and go to step 3. If the digit is $2, 5$ or 8, move to state s_1, move right, and go to step 3.

(3) Suppose there is a blank cell here. If in s_0, ACCEPT, otherwise, REJECT.

If the cell is not blank, move right and go to step 4.

(4) Move left and go to step 2. Do not change the contents of the current cell.[1]

Note that this is indeed a **decider** as there are no loops.

Here is how the following strings are treated:

[1] If we allow our Turing machine the option to 'stay in place' after reading the contents of a cell, this step is not needed.

- Begin in s_0. Read 0, remain in s_0. ACCEPT.
- Begin in s_0. Read 1, move to s_1. REJECT.
- Begin in s_0. Read 5, move to s_2. REJECT.
- Begin in s_0. Read 9, remain in s_0. ACCEPT.
- For the purposes of our example, suppose the input is 279. The configurations are as follows:

$$\epsilon s_0 279 \to 2 s_2 79 \to 27 s_0 9 \to 279 s_0 \to \text{ACCEPT}.$$

This answer is technically correct, but maybe difficult to think up of in an exam. In the revision session we talked about an algorithm for a two tape Turing machine. Here it is:

(1) The input is on the first tape (T1), and 0 is on the second tape (T2) initially.
(2) If the number on T2 is equal to the number on T1, ACCEPT. If the number on T2 is bigger than the number on T1, REJECT. If the number on T2 is smaller than the number on T1, add 3 to the number on T2 and write the result on T2.
(3) Go to step 2.

During the revision session we ran into the issue essentially of how a Turing machine 'understands' what the number on a tape is - how a sequence of cells like $\|5\|0\|1\|$ becomes the number 501 in the machine. According to Professor Nicoara, "I wouldn't think this level of detail would be required for the description of the Turing machine (as you can see by the level of detail in the notes whenever Turing machines were constructed in the last unit) or it could be treated as a subroutine." A long story short: the above algorithm from the revision session is an acceptable answer.

If one wanted to be more precise, one could write an algorithm that explains how to compare strings of digits on two tapes (e.g. comparing $\|5\|0\|9\|$ and $\|1\|2\|$), and another algorithm explaining how addition by 3 works (that is, how $\|5\|0\|9\| \to \|5\|1\|2\|$) - this was attempted during the revision session with mixed reviews.

Remarks: There's a joke that applies to the answers for this question: "A woman sent her programmer husband to the store. She told him 'buy a loaf of bread, and if they have eggs, get a dozen'. He returns home with 12 loaves of bread."
The point being if you're not precise in your phrasing and grammar when writing algorithms, mistakes like the above can occur. Some solutions say:

"If there is a blank one slot to the right, do X and accept if Y, otherwise reject".
This makes the step sound like "if there is a blank one slot to the right do X and accept if Y. Otherwise reject." What was probably intended was "if there is a blank one slot to the right, do X. Accept if Y, otherwise reject."

Q4

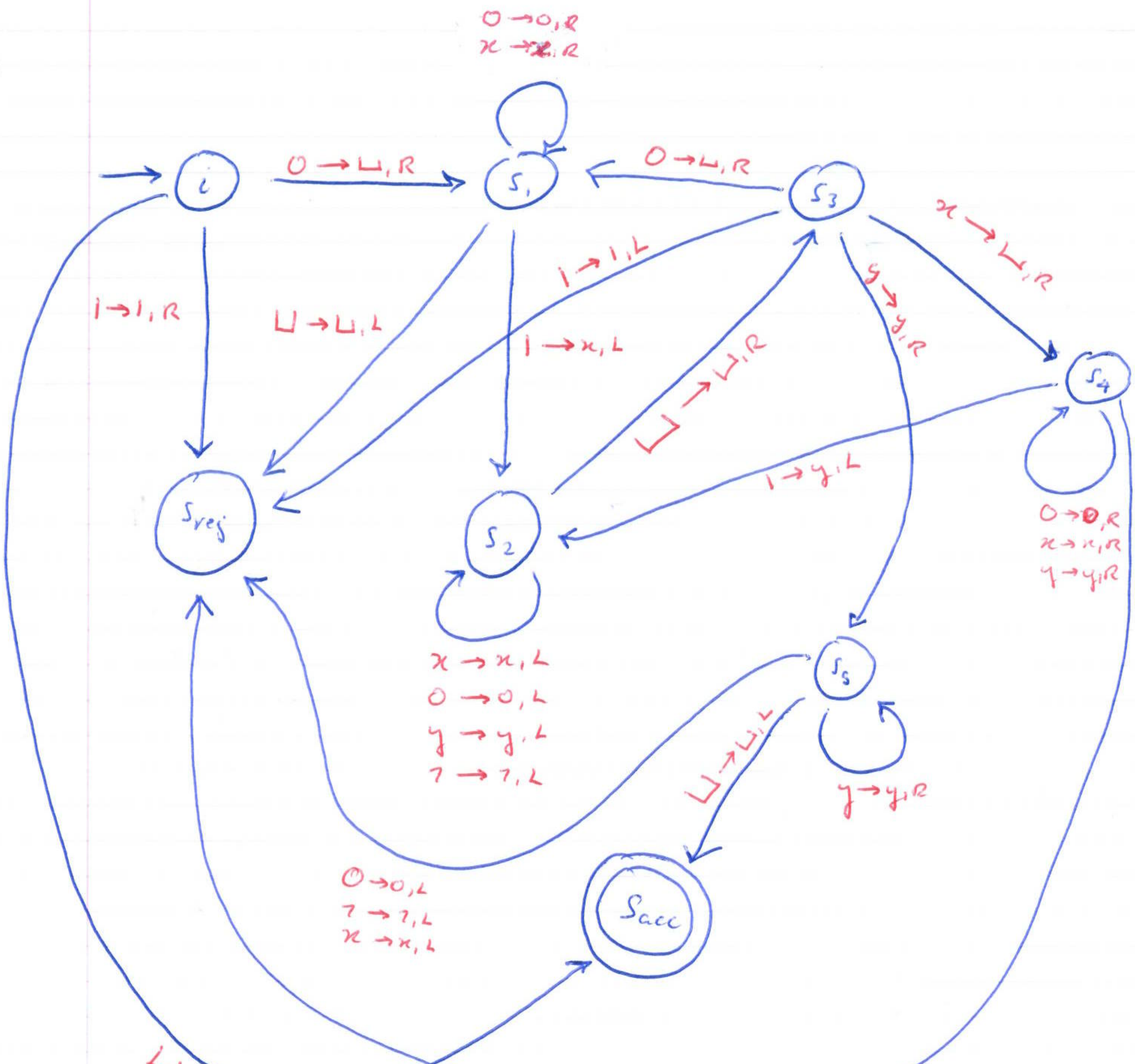
Let $\mathfrak{L} = \{\text{all Turing-recognisable languages}\}$. Consider $L_1, L_2 \in \mathfrak{L}$. As these are Turing-recognisable there exist Turing machines T_1, T_2 such that $L_1 = L(T_1), L_2 = L(T_2)$. $L_1 \cap L_2$ would be the set of those strings accepted by both T_1 and T_2. The Turing machine $T_{1,2}$ recognising $L_1 \cap L_2$ follows from this fact.

Here is the algorithm for $T_{1,2}$:

(1) Consider the string x. If T_1 accepts x, go to step 2. Otherwise REJECT.
(2) If T_2 accepts x, ACCEPT. Otherwise, REJECT.

If x is accepted by $T_{1,2}$, then $x \in L_1 \cap L_2$. If $x \in L_1 \cap L_2$, then by design x is accepted by $T_{1,2}$.

Therefore $L_1 \cap L_2 \in \mathfrak{L}$ as required.



E.g

001111	i
⊔01111	s_1
0x111	s_2
0x111	s_3
⊔x111	s_1
xx11	s_2
xx11	s_3
⊔x11	s_4
xy1	s_2
xy1	s_3
⊔y1	s_4
yy	s_2
yy	s_3
	s_{acc} !