Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F29FB - Foundations 2
Type of assessment:	**Individual**
Coursework Title:	Assignment
Student Name:	Lucca Anthony Marcondes Browning
Student ID Number:	H00369673

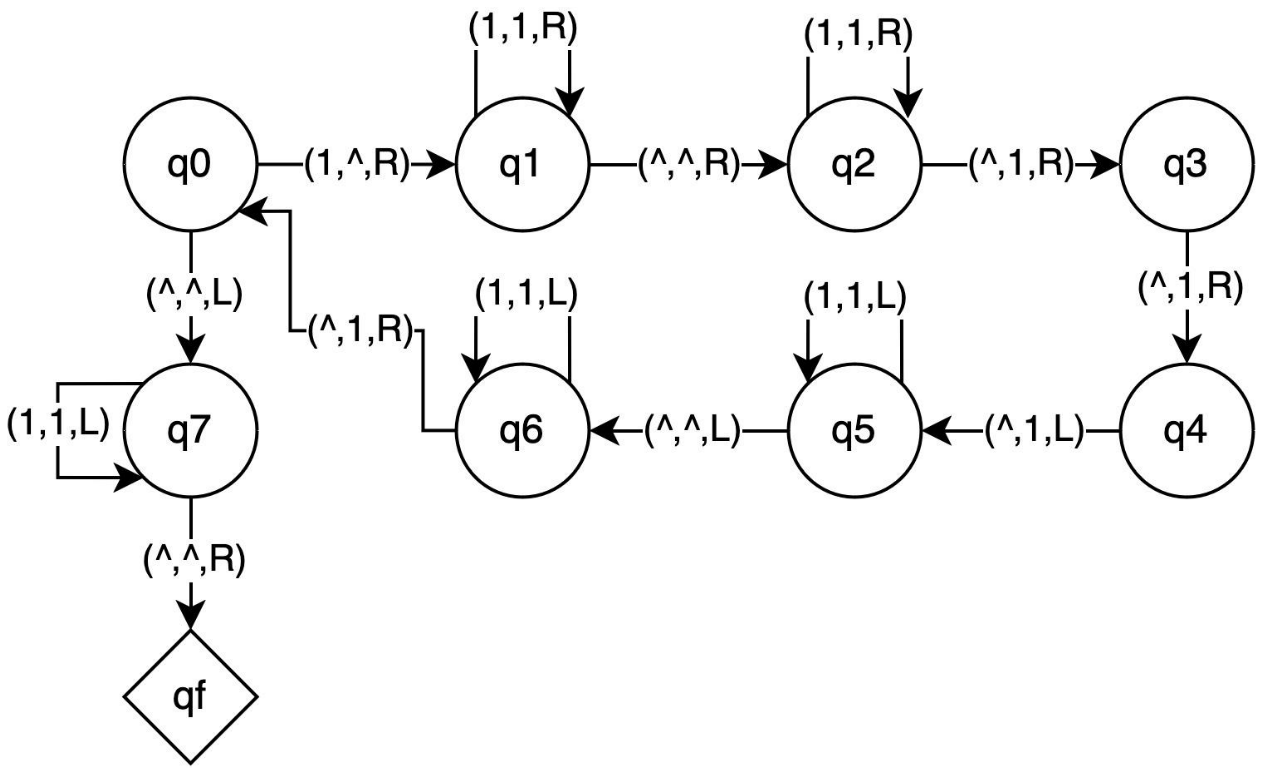
Student Signature: Lucca Anthony Marcondes Browning

Date: 19/02/24

Please note that this one is the second, revised version. The first version has a mistake. Thank you.

Foundations II Assignment

1. Give the graph of the Turing machine which takes its input in unary representation (e.g., 111 represents the number 3) and returns the original input followed by a blank space, followed by the original input multiplied by 3. Note all the conditions below.
   a. Your graph should be logically as elegant as possible. Remember that you should not use handwritten text, and so, in addition to the logical elegance of your graph, it should be drawn using the drawing software of your choice.
   b. You must return back to the start of the input number before you stop. So, your final output tape must have the pointer at the leftmost digit of the input number.



2. Give the formal definition of your proposed graph as per the course's formal definition of a Turing machine. That is, you should identify all the states including the start state, you should also identify all the symbols and the action table.

States = {q0, q1, q2, q3, q4, q5, q6, q7, qf}

Symbols = {Λ,1}

Action Table (Function M):

M_{mult3}(q0, 1) = (q1, Λ, R)	M_{mult3}(q0, Λ) = (q7, Λ, L)
M_{mult3}(q1, 1) = (q1, 1, R)	M_{mult3}(q1, Λ) = (q2, Λ, R)
M_{mult3} (q2, 1) = (q2, 1, R)	M_{mult3}(q2, Λ) = (q3, 1, R)
M_{mult3}(q3, Λ) = (q4, 1, R)	
M_{mult3}(q4, Λ) = (q5, 1, L)	
M_{mult3}(q5, 1) = (q5, 1, L)	M_{mult3}(q5, Λ) = (q6, Λ, L)
M_{mult3}(q6, 1) = (q6, 1, L)	M_{mult3}(q6, Λ) = (q0, 1, R)
M_{mult3}(q7, 1) = (q7, 1, L)	M_{mult3}(q7, Λ) = (qf, Λ, R)

Start State = q0

Tape Contents (of example input tape):

- $t_{mult3}(0) = t_{mult3}(1) = t_{mult3}(2) = t_{mult3}(3) = 1$
- $t_{mult3}(i) = \{\Lambda \mid i \in \mathbb{Z}$ and $i \leq -1$ and $i \geq 4\}$

Starting Turing Machine Configuration (of example input tape):

- $C_{mult3} = (M_{mult3}, t_{mult3}, 0, q0)$.

3. Give the logic of your graph (i.e., clearly explain how it works and say what each part of the graph actually does).

The first thing my Turing machine will do in state q0 is check whether there is still items to be copied or not. If the current location of the head is on a “1”, then there is still a value to be copied. It will replace that value with a ʌ (essentially copying it), and then move to state q1. If the current location of the head is on a ʌ, then there are no more values to be copied. It will go left and move to state q7. The reason why this works is because every time after the machine has finished copying a “1” in the original string, and then placing it three times to the right of the input “string”, it will return to the spot in which it took the “1” from and move one place to the right. If this place does not have a “1”, it will actually be the gap between “n” and “3n”, meaning there will be no more numbers in “n” to copy over to “3n”.

Moving on, state q1 is where the machine skips over any future “1”s in the “n” part of the tape so that it can copy that “1” that it is holding to the “3n” part of the tape. It will come back to these “1”s after placing the current “1” it has three times. This means that the head will keep going right on the tape and stay in state q1 until it reaches a ʌ. When the tape reaches a ʌ, it will switch to state q2, and move one to the right. This is so that the tape can keep the gap between “n” and “3n”.

When state q2 is reached, the machine knows that it needs to skip over any “1”s that have already been placed in the “3n” part of the tape in order to deposit the “1” it is holding. Therefore, as long as the head keeps reading “1”s, it will continue to move right. When it reaches the end of the “string” of “1”s it’s already placed, it will reach a symbol ʌ. That’s when it knows to place its first “1” and move to the right.

However, how does the machine know it’s already placed its first “1” out of the three it needs to copy and not two of them or all of them? The way I decided to clear this confusion was with state q3, which knows that one of the “1”s is already placed. This state will only ever exist after the first “1” has been placed and the head is on a ʌ. After reading that it’s on a ʌ, the head will place the second “1”, move one to the right, and the machine will change to state q4.

I used the same solution so that the machine knows it’s placed the second “1” out of the three with state q4. This state only ever exist after the second “1” has been placed and the head is on a ʌ. It will place the third “1” in the head’s current location, and move to the left because it doesn’t have anything else to place to the right. The machine will switch to state q5.

In state q5, the machine is beginning to return to the spot in which it took the original “1” from in order to replace the ʌ it left in the “1”’s place. The same way state q2 had to skip over all of the already placed “1”s in the “3n” part of the tape to place the “1” it was holding, state q5 is skipping over all of the “1”s (leftward) to return the “1” to the spot it stole it from. It will keep moving left without changing anything until it reaches the ʌ between “n” and “3n”. When it reaches the ʌ, it will move one to the left (skipping over the ʌ and leaving it unchanged) and move to state q6.

State q6 is where the TM is skipping over all of the other “1”s (leftward) it will pick up later to place down the “1” it is currently using, the same way state q1 was skipping over them to go right. The tape head will keep moving left, and leaving the “1”s it’s reading unchanged until it reaches the Λ from which it stole the “1” it is holding from. It will place this “1” back down, replacing the current Λ, and move one to the right in order to read the next “1” it need to copy. Then it will switch back to state q0 and continue the cycle again.

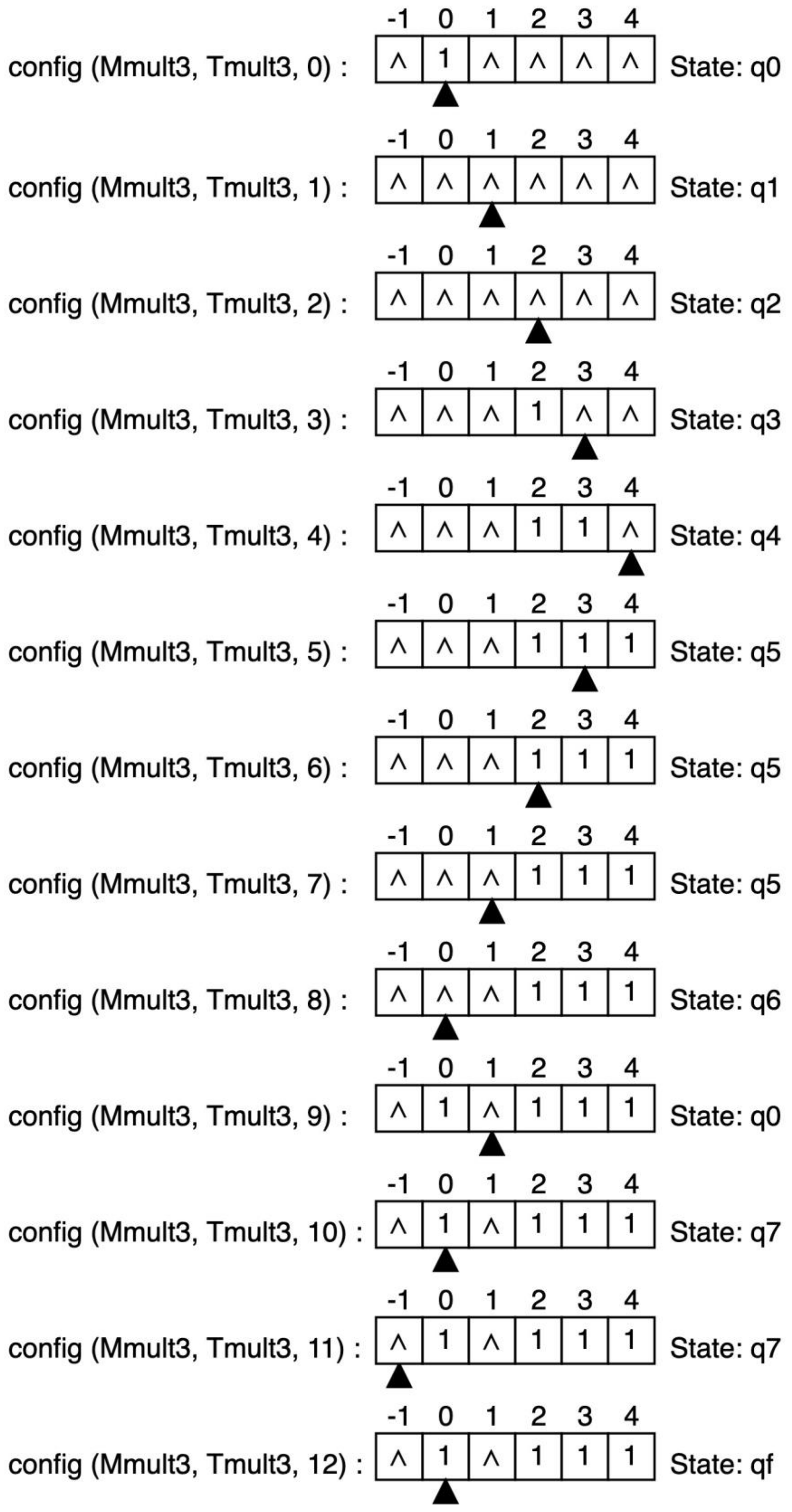
Let’s say there wasn’t another “1” for the machine to copy when it was in state q0. This means three things. First, the machine was in the Λ between “n” and “3n”. Second, the machine has just gone left and ended up at the tail end of the input string “n”. Third, the machine needs to return to the start of the input string “n” according to the specification of the coursework. Thus, the machine will remain in state q7 while moving to the left and reading “1”s without changing them until it hits the Λ in location –1, which is the head location on the tape right before the input string. In that situation, the machine will move one right, and end up in the beginning of the input string “n” and move to state qf, the final state. The Turing machine will now no longer have an action from the action table it can do, and it will halt. The program has finished running.

4. Use the TM simulator at https://math.hws.edu/eck/js/turing-machine/TM.html to upload the Json file for your TM on canvas in the location marked assignment.

I ended up using this website that Luka suggested in the labs. Here is a link to my Turing machine. I will also upload the link to canvas.

http://morphett.info/turing/turing.html?ac9222992f1be33840d5af6808a98360

5. Test your machine code on the numbers below and produce the following:
   a. Input 1.



b. Input 111.

```
                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 0) :    ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q0
                                   ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 1) :    ∧  ∧  1  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q1
                                      ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 2) :    ∧  ∧  1  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q1
                                         ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 3) :    ∧  ∧  1  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q1
                                            ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 4) :    ∧  ∧  1  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q2
                                               ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 5) :    ∧  ∧  1  1  ∧  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q3
                                                  ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 6) :    ∧  ∧  1  1  ∧  1  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q4
                                                     ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 7) :    ∧  ∧  1  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q5
                                                  ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 8) :    ∧  ∧  1  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q5
                                               ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 9) :    ∧  ∧  1  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q5
                                            ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 10) :   ∧  ∧  1  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q6
                                         ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 11) :   ∧  ∧  1  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q6
                                      ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 12) :   ∧  ∧  1  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q6
                                   ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 13) :   ∧  1  1  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q0
                                      ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 14) :   ∧  1  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q1
                                         ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 15) :   ∧  1  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q1
                                            ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 16) :   ∧  1  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q2
                                               ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 17) :   ∧  1  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q2
                                                  ▲

                               -1  0  1  2  3  4  5  6  7  8  9 10 11 12
config (Mmult3, Tmult3, 18) :   ∧  1  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q2
                                                     ▲
```

config (Mmult3, Tmult3, 19) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	∧	∧	∧	∧	∧	∧
								▲					

State: q2

config (Mmult3, Tmult3, 20) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	∧	∧	∧	∧	∧
									▲				

State: q3

config (Mmult3, Tmult3, 21) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	∧	∧	∧	∧
										▲			

State: q4

config (Mmult3, Tmult3, 22) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	1	∧	∧	∧
									▲				

State: q5

config (Mmult3, Tmult3, 23) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	1	∧	∧	∧
								▲					

State: q5

config (Mmult3, Tmult3, 24) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	1	∧	∧	∧
							▲						

State: q5

config (Mmult3, Tmult3, 25) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	1	∧	∧	∧
						▲							

State: q5

config (Mmult3, Tmult3, 26) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	1	∧	∧	∧
					▲								

State: q5

config (Mmult3, Tmult3, 27) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	1	∧	∧	∧
				▲									

State: q5

config (Mmult3, Tmult3, 28) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	1	∧	∧	∧
			▲										

State: q6

config (Mmult3, Tmult3, 29) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	∧	1	∧	1	1	1	1	1	1	∧	∧	∧
		▲											

State: q6

config (Mmult3, Tmult3, 30) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	1	∧	1	1	1	1	1	1	∧	∧	∧
			▲										

State: q0

config (Mmult3, Tmult3, 31) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	∧	∧	∧
				▲									

State: q1

config (Mmult3, Tmult3, 32) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	∧	∧	∧
					▲								

State: q2

config (Mmult3, Tmult3, 33) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	∧	∧	∧
						▲							

State: q2

config (Mmult3, Tmult3, 34) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	∧	∧	∧
							▲						

State: q2

config (Mmult3, Tmult3, 35) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	∧	∧	∧
								▲					

State: q2

config (Mmult3, Tmult3, 36) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	∧	∧	∧
									▲				

State: q2

config (Mmult3, Tmult3, 37) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	∧	∧	∧
										▲			

State: q2

-1 0 1 2 3 4 5 6 7 8 9 10 11 12

config (Mmult3, Tmult3, 38) :

∧	1	1	∧	∧	1	1	1	1	1	1	∧	∧	∧
											▲		

State: q2

config (Mmult3, Tmult3, 39) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	∧	∧
												▲	

State: q3

config (Mmult3, Tmult3, 40) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	∧
													▲

State: q4

config (Mmult3, Tmult3, 41):

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
												▲	

State: q5

config (Mmult3, Tmult3, 42) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
											▲		

State: q5

config (Mmult3, Tmult3, 43) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
										▲			

State: q5

config (Mmult3, Tmult3, 44) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
									▲				

State: q5

config (Mmult3, Tmult3, 45) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
								▲					

State: q5

config (Mmult3, Tmult3, 46) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
							▲						

State: q5

config (Mmult3, Tmult3, 47) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
						▲							

State: q5

config (Mmult3, Tmult3, 48) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
					▲								

State: q5

config (Mmult3, Tmult3, 49) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
				▲									

State: q5

config (Mmult3, Tmult3, 50) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	∧	∧	1	1	1	1	1	1	1	1	1
			▲										

State: q6

config (Mmult3, Tmult3, 51) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	1	∧	1	1	1	1	1	1	1	1	1
				▲									

State: q0

config (Mmult3, Tmult3, 52) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	1	∧	1	1	1	1	1	1	1	1	1
			▲										

State: q7

config (Mmult3, Tmult3, 53) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	1	∧	1	1	1	1	1	1	1	1	1
		▲											

State: q7

config (Mmult3, Tmult3, 54) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	1	∧	1	1	1	1	1	1	1	1	1
	▲												

State: q7

config (Mmult3, Tmult3, 55) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	1	∧	1	1	1	1	1	1	1	1	1
▲													

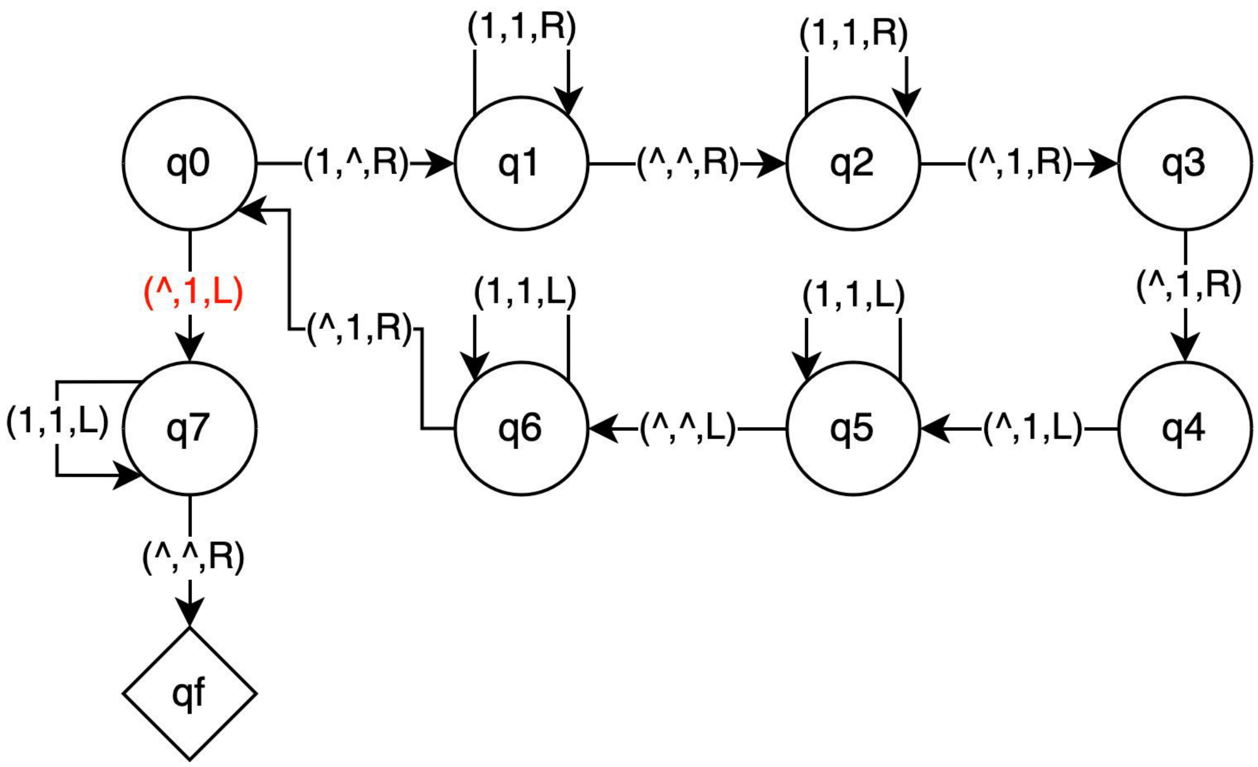
State: q7

config (Mmult3, Tmult3, 56) :

-1	0	1	2	3	4	5	6	7	8	9	10	11	12
∧	1	1	1	∧	1	1	1	1	1	1	1	1	1
	▲												

State: qf

6. Make a very minimal change (as minimal as possible) to your graph to return instead the Turing machine that takes its input n in unary representation (e.g., 111 represents the number 3) and returns 4n + 1 (also in unary representation).
    a. Give the full new graph of the new Turing machine.
        i. Make sure that the change from your original graph to the new one is as minimal as possible. For example, one possible change of the original graph to the new graph is simply to change
            1. (certain-old-symbol, certain-new-symbol, certain-direction) to
            2. (certain-old-symbol, another-certain-new-symbol, certain-direction).
        ii. Here, the change is very minimal. Just one symbol is changed in an existing action. Everything else is the same.
        iii. Of course depending on your graph, the change may be bigger than this.



b. Explain clearly why the change you make will transform the graph from the function (n, 3n) to 4n + 1.

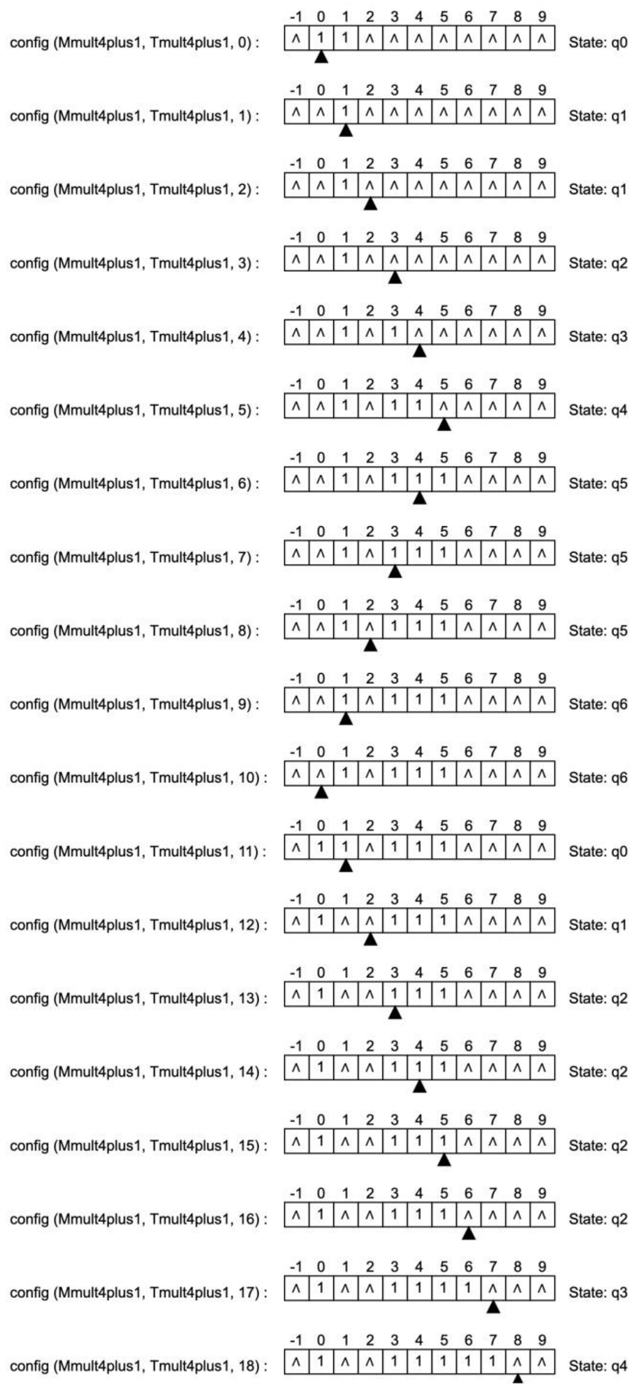
Before explaining why the change transformed the graph, I will explain how, in general, one would be able to transform the graph. By turning the Λ between the input “n” and the product “3n” into a “1”, we can make the function 4n + 1.

From analyzing the graph, I realized that the best possible time to change the gap (Λ) into a “1” would be after everything was copied and multiplied to the right side of the gap. Therefore, when state q0 transitions to state q7, instead of leaving the gap between “n” and “3n” as a Λ, as I mentioned I did in Question 4, before moving to the left, I changed it to a “1”.

7. Upload the Json file of your new TM (the one that represents 4n + 1) and test it on Input 11 giving all the tape configuration sequences as above.

I ended up using this website that Luka suggested in the labs (as I did for Q4). Here is a link to my Turing machine. I will also upload the link to canvas.

http://morphett.info/turing/turing.html?cb4f48770670dab42804a00fa3941b42



```
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 0) :    ∧  1  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q0
                                             ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 1) :    ∧  ∧  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q1
                                                ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 2) :    ∧  ∧  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q1
                                                   ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 3) :    ∧  ∧  1  ∧  ∧  ∧  ∧  ∧  ∧  ∧  ∧   State: q2
                                                      ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 4) :    ∧  ∧  1  ∧  1  ∧  ∧  ∧  ∧  ∧  ∧   State: q3
                                                         ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 5) :    ∧  ∧  1  ∧  1  1  ∧  ∧  ∧  ∧  ∧   State: q4
                                                            ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 6) :    ∧  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q5
                                                         ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 7) :    ∧  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q5
                                                      ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 8) :    ∧  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q5
                                                   ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 9) :    ∧  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q6
                                                ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 10) :   ∧  ∧  1  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q6
                                             ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 11) :   ∧  1  1  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q0
                                                ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 12) :   ∧  1  ∧  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q1
                                                   ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 13) :   ∧  1  ∧  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q2
                                                      ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 14) :   ∧  1  ∧  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q2
                                                         ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 15) :   ∧  1  ∧  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q2
                                                            ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 16) :   ∧  1  ∧  ∧  1  1  1  ∧  ∧  ∧  ∧   State: q2
                                                               ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 17) :   ∧  1  ∧  ∧  1  1  1  1  ∧  ∧  ∧   State: q3
                                                                  ▲
                                         -1  0  1  2  3  4  5  6  7  8  9
config (Mmult4plus1, Tmult4plus1, 18) :   ∧  1  ∧  ∧  1  1  1  1  1  ∧  ∧   State: q4
                                                                     ▲
```

config (Mmult4plus1, Tmult4plus1, 19) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	∧	∧	1	1	1	1	1	1	∧
								▲		

State: q5

config (Mmult4plus1, Tmult4plus1, 20) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	∧	∧	1	1	1	1	1	1	∧
							▲			

State: q5

config (Mmult4plus1, Tmult4plus1, 21) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	∧	∧	1	1	1	1	1	1	∧
						▲				

State: q5

config (Mmult4plus1, Tmult4plus1, 22) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	∧	∧	1	1	1	1	1	1	∧
					▲					

State: q5

config (Mmult4plus1, Tmult4plus1, 23) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	∧	∧	1	1	1	1	1	1	∧
				▲						

State: q5

config (Mmult4plus1, Tmult4plus1, 24) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	∧	∧	1	1	1	1	1	1	∧
			▲							

State: q5

config (Mmult4plus1, Tmult4plus1, 25) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	∧	∧	1	1	1	1	1	1	∧
		▲								

State: q6

config (Mmult4plus1, Tmult4plus1, 26) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	1	∧	1	1	1	1	1	1	∧
			▲							

State: q0

config (Mmult4plus1, Tmult4plus1, 27) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	1	1	1	1	1	1	1	1	∧
		▲								

State: q7

config (Mmult4plus1, Tmult4plus1, 28) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	1	1	1	1	1	1	1	1	∧
	▲									

State: q7

config (Mmult4plus1, Tmult4plus1, 29) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	1	1	1	1	1	1	1	1	∧
▲										

State: q7

config (Mmult4plus1, Tmult4plus1, 30) :

-1	0	1	2	3	4	5	6	7	8	9
∧	1	1	1	1	1	1	1	1	1	∧
	▲									

State: qf