Homework 2

1- State the Turing Machines in the tabular format discussed in class and lecture notes that implement the following RATM statements:

(a)add =c ;

I assumed that c is a binary coded integer.

	Condition	Turing Machine	
>A	--------	$R_\#^0\ L^0\ R_\#^c\ L^c\ B$	
B	$\sigma^c = 0$	$L^0\ L^c\ B$	
	$\sigma^c = 1$ and $\sigma^0 = 0$	$1^0\ L^0\ L^c\ B$	
	$\sigma^c = 1$ and $\sigma^0 = 1$	$0^0\ L^0\ L^c\ C$	carry state
	$\sigma^c = 1$ and $\sigma^0 = \#$	$1^0\ L^0\ L^c\ B$	
	$\sigma^c = \#$	$R^0\ h$	
C	$\sigma^c = 0$ and $\sigma^0 = 0$	$1^0\ L^0\ L^c\ B$	
	$\sigma^c = 0$ and $\sigma^0 = 1$	$0^0\ L^0\ L^c\ C$	
	$\sigma^c = 0$ and $\sigma^0 = \#$	$1^0\ L^0\ L^c\ B$	End carry state
	$\sigma^c = 1$ and $\sigma^0 = 0$	$0^0\ L^0\ L^c\ C$	
	$\sigma^c = 1$ and $\sigma^0 = 1$	$1^0\ L^0\ L^c\ C$	
	$\sigma^c = 1$ and $\sigma^0 = \#$	$0^0\ L^0\ L^c\ C$	

(b) jpos s ;

If R0 > 0 then K:= s

We need to check if R0> 0 then if it is true the program state should go to K_s state else it should continue with K_{j+1}

I assumed that R0 does not contain any ineffectual leading zeros.

	Condition	Turing Machine	
$>K_j$	--------	$L_\#^0\ R^0\ A$	
A	$\sigma^0 = 0$	$L_\#^0\ K_{j+1}$	
	$\sigma^0 = 1$	$L_\#^0\ K_s$	

(c) sub j

2- Assuming a 2 tape TM that multiplies the binary coded positive integers in tapes 1 and 2 and writes the result in tape 1 is available and is named as MULT ; construct in tabular format a multitape , nondeterministic TM that decides whether a given binary coded integer is a prime number making use of the TM MULT.

Solution:

We can create this Turing Machine by using the MULT TM in the following way:

Put the given binary coded integer into tape 3.

Erase tape 1.

If the given binary coded integer is smaller than 2 then halt with h_{no}

Else select two numbers in a nondeterministic way between 1 and the given number and put them into tape 1 and tape 2.

Run MULT TM

If the result in tape 1 is equal to the given binary coded integer, then halt with h_{yes} else halt with h_{no}.

If at least one of the computation trees halts with a h_{yes} this means the binary coded integer is not a prime number. Else it is a prime number.

We can construct this machine like this with using 3 tapes:

The machine starts with this configuration: (s, ◊ $\underline{\#}$ w, ◊ $\underline{\#}$, ◊ $\underline{\#}$)

	Condition	Turing Machine	Comments
>	-----	$R^1\ R^3$ A	
A	$\sigma^1 = 0$	$0^3\ R^1\ R^3$ A	
	$\sigma^1 = 1$	$1^3\ R^1\ R^3$ A	
	$\sigma^1 = \#$	$L^1\ L^3_{\#}$ B	Erase tape 1
B	$\sigma^1 \neq \#$	$\#^1\ L^1$ B	
	$\sigma^1 = \#$	R^3 C	Check w > 1
C	$\sigma^3 = 0$	h_{no}	
	$\sigma^3 = 1$	R^3 D	
D	$\sigma^3 = \#$	h_{no}	
	$\sigma^3 \neq \#$	$L^3_{\#}\ R^1\ R^2$ E	w is > 1
E	-----	$1^1\ R^1\ 1^2\ R^2$ F	Put at least one digit and eliminate possible zeros
	-----	$1^1\ R^1\ 0^2\ R^2$ F	
	-----	$0^1\ R^1\ 1^2\ R^2$ F	
F	-----	$1^1\ R^1\ 1^2\ R^2$ F	
	-----	$1^1\ R^1\ 0^2\ R^2$ F	
	-----	$0^1\ R^1\ 1^2\ R^2$ F	
	-----	$0^1\ R^1\ 0^2\ R^2$ F	
	-----	$L^2_{\#}\ L^1_{\#}\ R^1\ R^2$ G	
G	$\sigma^1 = 0$	$\#^1\ R^1$ H	Remove leading zeros at tape 1
	$\sigma^1 = 1$	$L^1_{\#}\ R^1$ I	Remove leading zeros at tape 2
H	$\sigma^1 \neq \#$	$L^1\ \sigma^1$ H	Remove leading zeros at tape 1 cont.
	$\sigma^1 = \#$	$L^1_{\#}\ R^1$ I	
I	$\sigma^2 = 0$	$\#^2\ R^2$ J	Remove leading zeros at tape 2

	$\sigma^2 = 1$	$L^2_\# R^2 K$	Check numbers 1 > 1
J	$\sigma^2 \neq \#$	$L^2 \sigma^2 J$	Remove leading zeros at tape 2 cont.
	$\sigma^2 = \#$	$L^2_\# R^2 K$	
K	$\sigma^1 = \#$	h_{no}	Number 1 is 1
	$\sigma^1 \neq \#$	$L^1 M$	Check number 2 > 1
M	$\sigma^2 = \#$	h_{no}	Number 2 is 1
	$\sigma^2 \neq \#$	L^2 MULT N	Start multiplication
N	$\sigma^1 \sigma^2 = 00 \vee 11$	$L^1 L^2 N$	Compare tape 1 and tape 3
	$\sigma^1 \sigma^2 = 01 \vee 10 \vee 0\# \vee 1\# \#0 \vee \#1$	h_{no}	
	$\sigma^1 \sigma^2 = \#\#$	h_{yes}	t1 * t2 is equal to w

3- Problems from the main textbook (note the word accepts means semidecides in our class terminology): 4.5.1, 4.5.2.

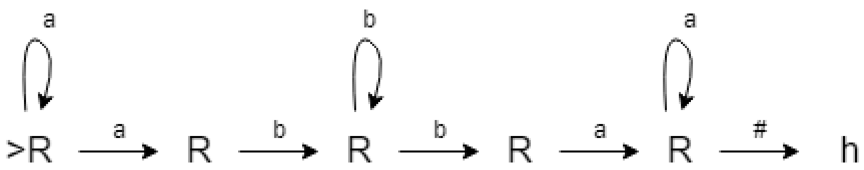
5.1. Give (in abbreviated notation) nondeterministic Turing machines that accept these languages.

(a) $a^*abb^*baa^*$

(b) $\{ww^Ruu^R : w, u \in \{a, b\}^*\}$

Solution:

a)



b)

	Condition	**Turing Machine**	**Comments**
> M	-----	R M	Split like w w^R # u u^R
	-----	Right-Shift-TM $L_\#$ R T	
T	σ = #	$R_{\{a,b\}}$ C	w w^R is found
	σ = a	# $R_\#$ L A	Remember a at state A
	σ = b	# $R_\#$ L B	Remember b at state B
A	σ = a	# $L_\#$ R T	
	σ ≠ a	I	Go to infinity
I	-----	R I	Infinity state
B	σ = b	# $L_\#$ R T	
	σ ≠ b	I	Go to infinity
C	σ = #	h	u u^R is found
	σ = a	# $R_\#$ L D	Remember a at state D
	σ = b	# $R_\#$ L E	Remember b at state E
D	σ = a	# $L_\#$ R C	
	σ ≠ a	I	Go to infinity
E	σ = b	# $L_\#$ R C	
	σ ≠ b	I	Go to infinity

4.5.2. Let $M = (K, \Sigma, \delta, s, \{h\})$ be the following nondeterministic Turing machine:

$$\begin{aligned} K &= \{q_0, q_1, h\}, \\ \Sigma &= \{a, \triangleright, \sqcup\}, \\ s &= q_0, \\ \Delta &= \{(q_0, \sqcup, q_1, a), (q_0, \sqcup, q_1, \sqcup), (q_1, \sqcup, q_1, \sqcup), (q_1, a, q_0, \rightarrow), (q_1, a, h, \rightarrow)\} \end{aligned}$$

Describe all possible computations of five steps or less by M starting from the configuration $(q_0, \triangleright\underline{\sqcup})$. Explain in words what M does when started from this configuration. What is the number r (in the proof of Theorem 4.5.1) for this machine?

	Condition	TM
>M	---	Q0
Q0	σ = #	a Q1
	σ = #	# Q1
Q1	σ = #	# Q1
	σ = a	R Q0
	σ = a	R h

This machine uses an a, a blip, and a blank in its alphabet. We can enumerate all possible configurations starting from (q0, ◊ $\underline{\#}$)

1 -> 4, 5

2 -> 3

3 -> 3

4 -> 1, 2

5 -> h

1 4 1 4 1 4

1 4 1 4 1 5

1 4 1 4 2 3

1 4 1 5 h

1 4 2 3 3 3

1 5 h

1: (q0, ◊ $\underline{\#}$)

2: (q0, ◊ $\underline{\#}$)

3: (q1, ◊ $\underline{\#}$)

4: (q1, ◊ $\underline{a}$)

5: (q1, ◊ $\underline{a}$)

1 4 1 4 1 4

(q0, ◊ $\underline{\#}$) $\vdash_M$ (q1, ◊ $\underline{a}$) $\vdash_M$ (q0, ◊ a $\underline{\#}$) $\vdash_M$ (q1, ◊ a $\underline{a}$) $\vdash_M$ (q0, ◊ a a $\underline{\#}$) $\vdash_M$ (q1, ◊ a a $\underline{a}$)

Or

1 4 1 4 1 5

(q0, ◊ $\underline{\#}$) $\vdash_M$ (q1, ◊ $\underline{a}$) $\vdash_M$ (q0, ◊ a $\underline{\#}$) $\vdash_M$ (q1, ◊ a $\underline{a}$) $\vdash_M$ (q0, ◊ a a $\underline{\#}$) $\vdash_M$ (q1, ◊ a a $\underline{\#}$)

Or

1 4 1 4 2 3

(q0, ◊ $\underline{\#}$) $\vdash_M$ (q1, ◊ $\underline{a}$) $\vdash_M$ (q0, ◊ a $\underline{\#}$) $\vdash_M$ (q1, ◊ a $\underline{a}$) $\vdash_M$ (q0, ◊ a a $\underline{\#}$) $\vdash_M$ (q1, ◊ a a $\underline{\#}$)

Or

1 4 1 5 h

(q0, ◊ $\underline{\#}$) $\vdash_M$ (q1, ◊ $\underline{a}$) $\vdash_M$ (q0, ◊ a $\underline{\#}$) $\vdash_M$ (q1, ◊ a $\underline{a}$) $\vdash_M$ (h, ◊ a a $\underline{\#}$)

Or

1 4 2 3 3 3

(q0, ◊ $\underline{\#}$) $\vdash_M$ (q1, ◊ $\underline{a}$) $\vdash_M$ (q0, ◊ a $\underline{\#}$) $\vdash_M$ (q1, ◊ a $\underline{\#}$) $\vdash_M$ (q1, ◊ a $\underline{\#}$) $\vdash_M$ (q1, ◊ a $\underline{\#}$)

Or

1 5 h

(q0, ◊ $\underline{\#}$) $\vdash_M$ (q1, ◊ $\underline{a}$) $\vdash_M$ (h, ◊ a $\underline{\#}$)

As we can see, there are two possible ways that M can start from the configuration (q0, ◊) It can either write the symbol a or the blank symbol #, and then move to the right while writing additional symbols until it either halts in state h or enters an infinite loop. There are 6 types of computation which two halt in state h and four take five steps.

To determine the number r for this machine, we need to find the maximum number of quadruples that can be applicable in any given circumstance. Since M is a nondeterministic Turing machine, it can have multiple applicable quadruples in some circumstances.

Based on the given transition function, there are two different circumstances under which two quadruples are applicable: when M is in state q0 and it is scanning a blank symbol #, or when M is in state q1 and it is scanning the input symbol a. There are no circumstances under which three or more quadruples are applicable. Therefore, r = 2 for this machine.

4- Let M_1 and M_2 be single tape DTMs that decide the languages L1 and L2 respectively. Construct using a tabular format a 2-tape NDTM M that decides the language L1.L2. (Assume : (i) that both M1 and M2 decide L1 and L2 leaving the tape contents clean, namely with (hYES ,◊ #) or (hNO ,◊ #) final configurations ; (ii) if X and Y are DTMs that decide languages then X.Y is defined as the sequential composition where control passes from X to Y if X reaches the state hYES .)

This 2-tape NDTM should first decide the language L1 with M1 on tape 1. If it does not accept NDTM should hold with hno. Else it should decide the language L2 with M2. If it is accepted, then the NDTM should hold with hyes else with hno

We need to map hyes of M1 to state B and hno of M1 to state C and hyes of M2 to hyes of NDTM and hno of NDTM to hno of NDTM.

Machine starts with this configuration (s, ◊ $\underline{\#}$ L1, ◊ $\underline{\#}$ L2)

	Condition	**TM**
M>	-----	A
A	-----	M_1^1
B	-----	M_2^2
C	-----	hno

5- If M is a TM that decides the language L where the assumptions of question 4 hold, construct using a tabular format a 2-tape NDTM M' that decides the language L* .

This machine should allow the empty string also. Thus we need to add a state that checks empty string and accepts it.