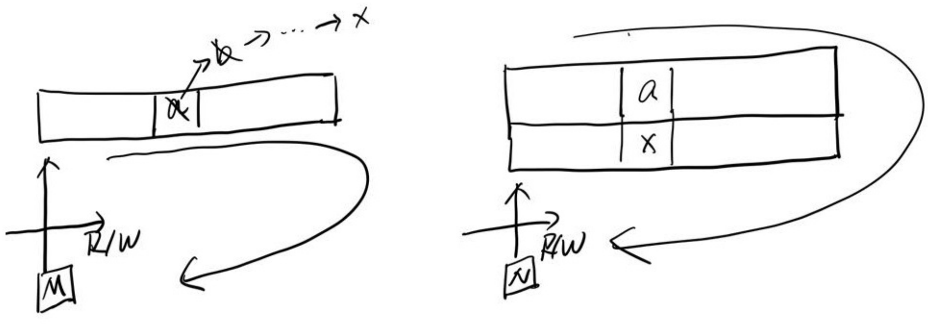
Home Work 2

Rusu Wu
11694764

1.(1) The smallest class of languages that 10-change Turing machines can accept is regular language. To prove, We can defined a FA N to simulate a nondeterministic Turing M. The input of M is w, and the input of N is ww_1. w_1 is the symbol library that would be written in to w when w has changes. Such that when M reads a symbol and changes it to a new symble, N change its current state and read the symbol from w_1. In this case, 10-change Turing machines can be simulated by FA .

(2) Defined a deterministic Turing machine M_1 and a 10-change-per-cell Turing machine M_2. Initially, we assume that their input are all w. If all the cells of M_2 change less than 10 times, M_2 read w as what M_1 did. If one of the cell changes 10 times, then M_2 copy all the w from the start to the frist Blank and defines it as w'. Such that the input of M_2 can be seen as ww'. Next time change happens, M_2 only can change the cells from the latest copy of input w'. If there are forward more than 10-times changes happen in the latest input string, M_2 would copy the latest sring from its start to the first Blank. Such that the new input can be seen as $ww'w''w'''...$, and therefore, the 10-change-per-cell Turing machine M_2 can simulate the deterministic Turing machine M_1 with the copy input w^n.

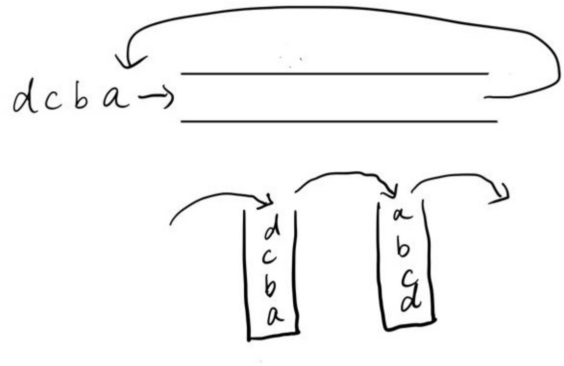
2. The smallest class of languages that 1-turn Turing machines can accept is regular language. Since Turing machine can read and write, we can use a 2-way FA that called N to simulate the 1-turn Turing machine M. If there is not change from the initial to the turning, the input of M is exactly the input of N. If there are changes of symbol in the state p before move to next state q in M, N will write the final symbol as the input of the turning-back reading. This means that if there are several changes within one cell in M, such as a→ b→...→ x, N only needs to write the final change x into the related input state. After TM M finishes the first turn and starts the back turn execution, the input of the second way of the FA N is exactly the same as M. Therefore, 1-turn Turing machines can be simulated by 2-way FA, and such that the smallest class of languages that 1-turn Turing machines can accept is regular language.



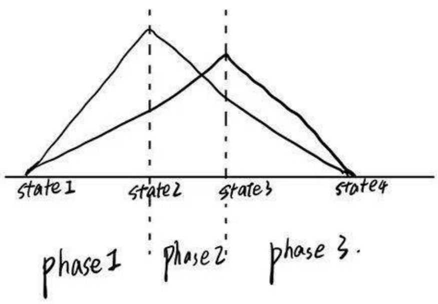
3. We can construct a standard model of TM M and a jump-TM M'. When there are R (right) and S (stay)instructions, it is obvious that M' can simulate M. When there is a L(left)instruction of M occuring, M' will mark the previous symbol it read or wrote on the tape. Then M' excutes the J(jump) instruction, and the tape head jumps to the left end of the tape begining to search the marked symbol. Once the tape find the marked symbol, it will remove the remarked symbol and do read or write. In this case, M' simulate the L(left) instructions of M. Such that TM can be simulated by a jump-TM.

4. There are two unary stacks N_1, N_2 achieveing the two counters. We construct a FIFO queue Q with n cells. We can use Q to simulate N_1 and N_2.

Also, there is a string $w = abcd$. If Q is empty, N_1 and N_2s' hight is equal to 0.
When a symbol of w queues into Q, the symbol also push into N_1. Counter N_1 's height become $x = x+1$, and the recording value of Q called $z = z+1$. After all symbols input into Q and N_1 ,they will be squeezed out from Q one by one. As a result, counter N_1 's height $x = x-1$.If all the symble is pushed out from N_1, then $x = 0, z = 0$.
Then the output symbol w turn back to the initial state of Q and run again. And also N_2 read the output w' of N_1. One thing we should know is that w' is the reverse of w. In second round of w, we called the recording value of Q called z'.
Similary to the first counter, when a symbol of w queues into Q, the symbol of w' also push into N_2. Counter N_2 's height become$y = y+1$, and the recording value of Q called $z' = z'+1$. After all symbols input into Q and N_2 ,they will be squeezed out from Q one by one. As a result, counter N_2 's hight $y = y-1$.If all the symble is pushed out from N_2, then $x = 0, z = 0$.In this case, the output is exactly the same as w.
Such that one (FIFO) queue can simulate two counters by reading the input two times.



5. There are 3 cases, including (1)x decreasing first, (2)y decreasing first and (3)both x and y decreasing at the same time.To explain, I assume that y decreases first, and the trend of x and y can be shown as the figure below.



We can devide the excution to 3 phases as the figure. Phase 1 only excutes instructions $x' := x+1$ and $y' = y+1$; Phase 2 only excutes instructions $x' := x+1$ and $y' = y-1$; Phase 3 only excutes instructions $x' := x-1$ and $y' = y-1$; The count of $x' := x+1$, $y' = y+1$,$x' := x-1$, $y' = y-1$ are defined as $C_{x++}, C_{x--}, C_{y++}, C_{y--}$. There are four test states that are defined in the excution State 1, State 2, State 3 and State 4.
The Values of x and y at each State are defined as $X_n n = 1,2,3,4$. $Y_n n = 1,2,3,4$. At each state, the Values should satisfy:

State 1:$X_1 = 0, Y_1 = 0$
State 2 and State 3: $X_n \geq 0, Y_n \geq 0 (n = 2,3)$
State 4: $X_4 \geq 0, Y_n \geq 0$

Since initial value x=0 and y=0 can be simulated in Finite memory, and X_n, Y_n can be counted by the caculation of $C_{x++}, C_{x--} and C_{y++}, C_{y--}$. Hence, if G is terminating, then in this case, it need to satisfy following requirements:

Phase 1:$C_{x++} > 0; C_{y++} > 0$
Phase 2:$C_{x++} > 0; C_{y--} > 0;; C_{y++} \geq C_{y--}$

Phase 3:$C_{x--} \geq 0; C_{y--} \geq 0; C_{x++} \geq C_{x--}; C_{y++} \geq C_{y--}$
Check that if ①the walk only has two kinds of excuting instructions in each Phases and ②satisfy the requirements of all the Phases or not. If it say yes to ①and② we can say YES to the walk.

6. In order to seek the value of B, we can simulate G with a FA M. If we want to test whether $x == 0$? or not, we can check the empty condition of the tape. $x++$ represents the instruction R(right), and L(left) is $x--$. Then we simulating G with M, By marking the last symbol of the tape that M reaches, we can determin the length as the value of B. Then we can construct another FA with a B cell tape, and to run G with the tape. Such that we can answer whether G is terminating or not.

7. Since one computer's power is limit, exchanging computation results via communication between computers can be one of the major trend. Thus we need distribution systems to achieve this goal. Mabe we can define hierachy levels of computing power based on the scale of distribution systems.