NAME: _____

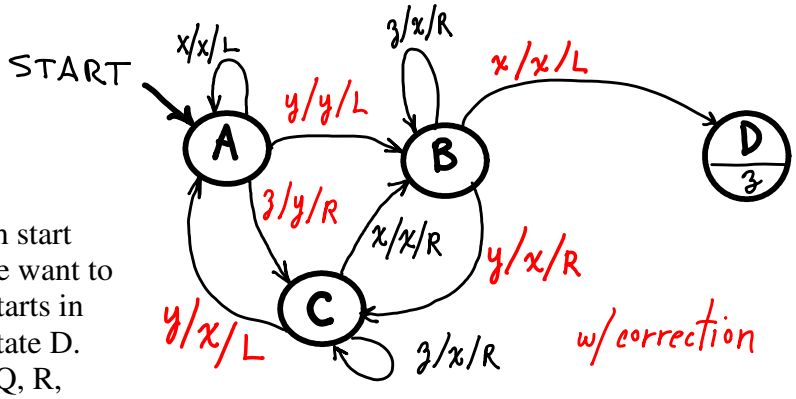
At right is a unary-encoded description of a TM, M1. Each row is a rule, and as input to a UTM the description would be laid out on the tape left-to-right, starting with the first row. The start state is encoded as "1". A halting state has the symbol "H" in the next-state field of every rule for that state.

The symbol set for the UTM, U, includes the set of characters { **<** , **>** , **1** , * , **L** , **R** , **H** }, as well as some others. Exactly one character appears in each of U's tape cells. E.g., the unary code, 111, occupies three cells. So, spaces in this table do not indicate empty cells: they are just there to make the table more readable.

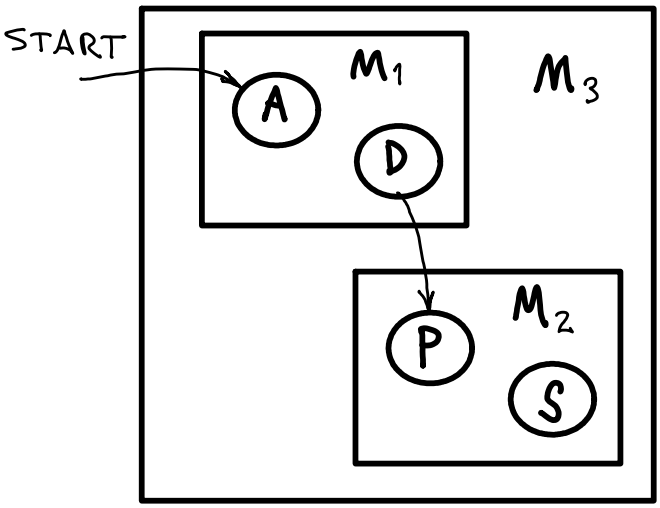
Q. Fill in the missing labels on the state-transition arcs in M1's diagram at right. Assume state D does a final "z" output, moves L, and then M1 halts.

current state		input		output		move		next state	
<	1	*	1	*	1	*	L	*	1	>
<	1	*	11	*	11	*	L	*	11	>
<	1	*	111	*	11	*	R	*	111	>
<	11	*	1	*	1	*	L	*	1111	>
<	11	*	11	*	1	*	R	*	111	>
<	11	*	111	*	1	*	R	*	11	>
<	111	*	1	*	1	*	R	*	11	>
<	111	*	11	*	1	*	L	*	1	>
<	111	*	111	*	1	*	R	*	111	>
<	1111	*	1	*	111	*	L	*	H	>
<	1111	*	11	*	111	*	L	*	H	>
<	1111	*	111	*	111	*	L	*	H	>

Assume the following encodings for M1:
states: 1 = A, 11 = B, 111 = C, 1111 = D;
symbols: 1 = x, 11 = y, 111 = z.



M1 and M2 are shown below as box diagrams with start and halting states. S is the halting state for M2. We want to build a new machine, M3, using M1 and M2. M3 starts in state A, and transitions to P instead of halting in state D. M2 uses M1's symbol set and also has 4 states: P, Q, R, and S. So, to code states for M3, we code P as 11111, Q as 111111, and so forth.



Q. To create a description of M3 we append a description of M2 to M1's description, modifying rules as needed. How many rules in the description of M3? Explain.

M1's description has 12 rows: for each of 4 states there are 3 symbols. M2 has 4 states and uses the same 3 symbols; so, combining both we need 24 rules, or 24 rows.

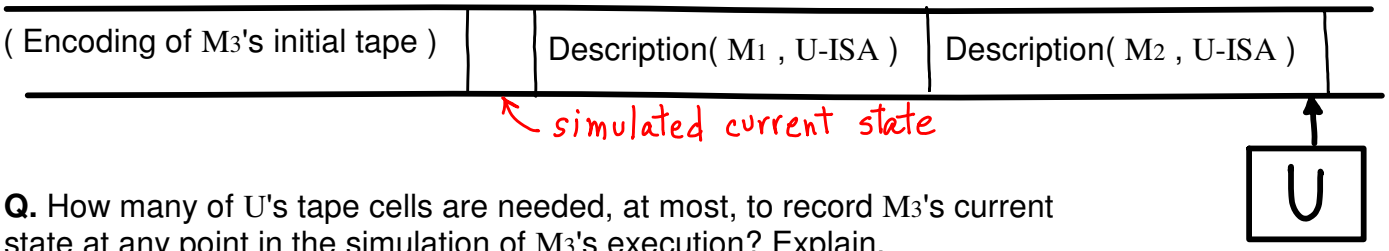
Q. Show any rules that need to be modified to form the description of M3 as (1) labeled state transition arcs between two of M3's states, and (2) as unary-encoded rules from M3's description.

The only rules that need to be changed are the rules for state D: the next-state field needs to be P (11111):

<1111 * 1 * 111 * L * 11111> <1111 * 11 * 111 * L * 11111>

<1111 * 111 * 111 * L * 11111>

Below is shown U's initial tape with U about to simulate M3. The descriptions of M1 and M2 have been modified appropriately as indicated above, and input for M3, suitably encoded, is on the left. In between is an area for U to keep track of M3's current state. "U-ISA" is the encoding scheme used above.



Q. How many of U's tape cells are needed, at most, to record M3's current state at any point in the simulation of M3's execution? Explain.

The state w/ the longest code is state S, whose code is 8 ones (1111 1111). Each 1 is in a separate cell on U's tape. So, we will need at most 8 cells to record M_3's current state.

Q. How many of U's tape cells are needed, at least, to encode the initial state of M3's tape? Assume the input consists of 5 repeats of the string "xyz". Recall that U needs to have symbols separating M3's individual simulated tape cells, and that U needs an indicator to remember M3's R/W head location. Explain your answer. Show any needed diagram.

Each of M_3's tape cells contain a single symbol (x, y, or z). On U, a simulated cell contains one of the codes (1, 11, 111) for each M_3 symbol. Each 1 is in a separate cell on U's tape. "xyz" → (1 11 111) requires 6 cells on U's tape. 5 repeats gives us 30 cells. Plus a seperater symbol for each simulated M_3 cell: 15 M_3 cells → 15+1 seperators → 46 U cells.

Q. Suppose M1 were instead a machine that reads no input, and writes out a description of M2 encoded as above. That is, it starts up, writes out Description(M2 , U-ISA), and halts. If we intend to have U simulate M1 and then when it halts, begin simulating M2, what symbol set would M1 have to use? Explain. Suggest very briefly how M1 might work.

If M_1 produces a table like the one above, it will need to use symbols from that set {<, >, 1, L, R, H}. But if U simulates M_1, all its symbols will be encoded in unary. It looks like U needs to function differently somehow.