1. Complete the exercise of tracing through the configuration sequence for the given Turing machine. **Answer:** Completed on Udacity.

2. Program a Turing machine the shifts its input one square to the right and places a \$ sign at the beginning of the tape. **Answer:** Completed on Udacity.

3. Program a Turing machine that tests if the input string has an equal number of zeros and ones. **Answer:** Completed on Udacity

4. Program a two-tape Turing machine to perform substring search. **Answer:** Completed on Udacity

5. Another alternative Turing machine model has a single, one-way infinite tape, but two read-write heads. The transition function has the form $\delta : Q \times \Gamma^2 \to Q \times \Gamma^2 \times \{L, R, S\}^2$, the same as a multi-tape machine. Describe how you would program such a machine to decide the language $L = \{ww \mid w \in \{0, 1\}^*\}$.

   Hint: You can use the S (stay put) movement to achieve the effect of having one head move faster than the other.

   (Writing down the transition function is not required, but you should go into enough detail to convince the reader that you could do so if asked.)

   Answer:

   This procedure finds the middle of an input string by advancing H2 twice as far as H1 for every move, then rewinding the heads to the start of each half of the string to compare characters. It handles empty strings, and strings with odd and even numbers of characters.

   - **State 1**: Start state
     * If blank, accept
     * Otherwise move to state 2

   - **State 2**: Shift string to the right by one
     * Insert special character, "#", at beginning
     * Shift string to the right by one
     * Move to state 3

– **State 3:** Rewind heads

  ∗ Move both heads back to first character (not #)
  ∗ Move to state 4

– **State 4:** Find middle

  Comment: this is the step where H2 moves twice for every step of H1.

  ∗ Move both H1 and H2 R
  ∗ Leave H1 stationary and move H2 R
  ∗ Check if H2 reads a blank
      · If H2 reads a blank, move to state 5
      · If H2 does not read a blank, repeat both previous steps

– **State 5**: Check if string length is odd

  ∗ Move H2 L
  ∗ If H2 reads blank, Fail. (*Comment:* string length is odd)
  ∗ If H2 reads non-blank, move to state 6. (*Comment:* string length is even)

– **State 6**: Rewind

  Comment: positions H1 over first character of w and H2 over the first character of 2nd half of string.

  ∗ Move H1 and H2 L simultaneously
  ∗ When H1 reads #, move both H1 and H2 R
  ∗ Move to state 7

– **State 7**: Match characters

  ∗ Compare characters read by H1 and H2
  ∗ If they match, move both R and repeat
  ∗ If they don't match: Fail
  ∗ When H2 reads a blank: Accept

6. Suppose we have a one-tape Turing machine M whose head instead of having to move just left or right in each computation step, can move left or right or stay put. We called these "stay-put machines" in the lesson. Argue that it is possible to create a new Turing machine M' (no "stay put", just moves left and right) that recognizes the same language as M by changing only the transition function δ, keeping the same Q and Γ.

Answer:

In this argument we are showing that: $\delta_M : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$ can be represented by: $\delta_{M'} : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$.

- Start by taking every stay-put transition from M, $\delta_M(q_i, b) = (q_j, c, S)$, where q_i, q_j are states in Q and b, c are characters in Γ.
- Replace each stay-put transition with $|\Gamma| + 1$ transitions:
  * $\delta_{M'}(q_i, b) = (q_j, c, R)$, (*comment:* states and characters remain the same as in $\delta_M(q_i, b)$)
  * and $\delta_{M'}(q_i, x) = (q_j, x, L)$, where x represents any character in Γ (*comment:* we want to move L without changing the tape).
- M' can now represent M without a stay-put move, S

7a. Suppose that we constrained the standard 1-tape Turing machine to only be able to write to each square on the tape two times. Prove that this model can decide every language that a standard Turing machine can.

Hint: Use lots of tape and do a lot of copying.

2-write answer:

Call this new machine a "2-write" machine. The following procedure will show that the 2-write machine can simulate a standard Turing machine (TM) and can, as a result, decide every language that a standard TM can.

The procedure involves copying the tape for each move of the machine: writing to a clear space off the right of the original string and crossing off characters from the original string as they are copied. There are two special situations around the head position that require a replacement rather than a copy. For each transition of the machine:

1. Mark the head position
2. Write a separator to the right of the original string.
3. Case 1: Right move

    (a) Copy symbols up to the cell before the mark, crossing each off on the original string as it is copied.

    (b) Instead of copying the cell with the mark, replace it in the new string with the "write" specified by the TM. Cross it off the original string.

    (c) Write the next symbol, marking it as the new head position. Cross it off the original string.

    (d) Continue copying the rest of the input, crossing each symbol off as it is copied.

4. Case 2: Left move

    (a) Copy symbols up to two cells before the mark, crossing each off on the original string as it is copied.

    (b) Copy the cell before the mark, adding the mark for the head position to the symbol. Cross it off the original string.

    (c) Replace the symbol from the original string with the mark with the "write" specified by the TM. Cross it off the original string.

    (d) Continue copying the rest of the input, crossing each symbol off as it is copied.

The preceding procedure shows a simulation of a standard TM with only two writes per cell: once when the input is written to the tape and the second when it is crossed off. Consequently the 2-write machine can simulate a standard Turing machine (TM) and can, as a result, decide every language that a standard TM can.

7b. Bonus (hard): Describe how you would modify your machine so that it only writes to each square once.

1-write answer:

To build a single-write machine, start with the 2-write machine. Modify it as follows:

– For every cell being copied, replace it with two cells, one blank and the other with the copied character.

– Now, instead of crossing through cells as they are copied, place a cross in the blank cell to indicate the associated symbol has been "used".

– Proceed as before.

8. Consider a Turing machine with a two-dimensional tape where the head can move up and down as well as right and left. Assume that the paper is infinite in the right and downward directions. Give a formal definition of this machine

Answer:[1]

A Turing machine is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where Q, Σ, Γ are all finite sets and

1. Q is the set of states,

2. Σ is the input alphabet not containing the blank symbol $\sqcup$,

3. Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,

4. $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, U, D\}$ is the transition function (*comment:* where allowed moves are L-left, R-right, U-up, D-down).

5. $q_0 \in Q$ is the start state,

6. $q_{accept} \in Q$ is the accept state, and

7. $q_{reject} \in Q$ is the reject state, where $q_{reject} \neq q_{accept}$.

---

[1]Definition modfied from Sipser, Michael, *Introduction to the Theory of Computation, 3e*, p. 168