CPSC 313 — Fall 2018
Assignment 3 — Non-Context-Free Languages and Turing Machines

Jasmine Roebuck, 30037334

1. Problem 2.31, page 157 (3rd ed) and page 131 (2nd ed) of Sipser. For convenience, the question is reproduced here. Recall that a *palindrome* is a string that reads the same forwards as backwards. Let L be the language of all palindromes over the alphabet $\{0, 1\}$ that contain an equal number of 0's and 1's. For example, 0110 and 010110011010 belong to L, but 0101 and 010 do not. Use the Pumping Lemma for context-free languages to prove that L is not context-free.

   Solution.

   By way of contradiction, assume that L is context-free. Then L satisfies the Pumping Lemma. Let p be the pumping length of L, and consider the string $s = 0^p1^{2p}0^p$. Since s is a palindrome with equal numbers of 1's and 0's, $s \in L$. Since $|s| = 4p \geq p$, by the Pumping Lemma, s can be written as $s = uvwxy$ with binary strings u, v, w, x, y such that $v \neq \varepsilon$ or $x \neq \varepsilon$, $|vwx| \leq p$, and $uv^iwx^iy \in L$ for all $i \geq 0$. In particular, $uwy = uv^0wx^0y \in L$. Hence, $uwy = 0^k1^{2k}0^k$ for some $k \geq 0$.

   Since $|vwx| \leq p$, vx cannot consist of 0's followed by 1's followed by 0's. It can consist of 0's only (at the start or the end of s), 0's followed by 1's, 1's only, or 1's followed by 0's. Thus we have one of the following four cases:

   Case 1. vx consists of 0's only, either at the start of s or the end of s. Then uwy is of the form $uwy = 0^l1^{2p}0^p$ or $uwy = 0^p1^{2p}0^l$ with $l < p$, contradicting $uwy = 0^k1^{2k}0^k$ with $k < p$.

   Case 2. vx consists of 0's followed by 1's. Then uwy is of the form $uwy = 0^l1^m1^p0^p$ with $l, m < p$. This again contradicts that $uwy = 0^k1^{2k}0^k$ with $k < p$.

   Case 3. vx consists of 1's only. Then uwy is of the form $uwy = 0^p1^l0^p$ with $l < 2p$, again contradicting $uwy = 0^k1^{2k}0^k$ with $k < p$.

   Case 4. vx consists of 1's followed by 0's. Then uwy is of the form $uwy = 0^p1^p1^l0^m$ with $l, m < p$. This contradicts that $uwy = 0^k1^{2k}0^k$ with $k < p$.

   In all cases, we derive a contradiction. Therefore the initial assumption that L is context-free must be false. Hence L cannot be context-free.

2. (a) Create a state diagram for a single-tape deterministic Turing machine for the language

   $$L = \{a^ib^ic^i \mid i \geq 0\}.$$

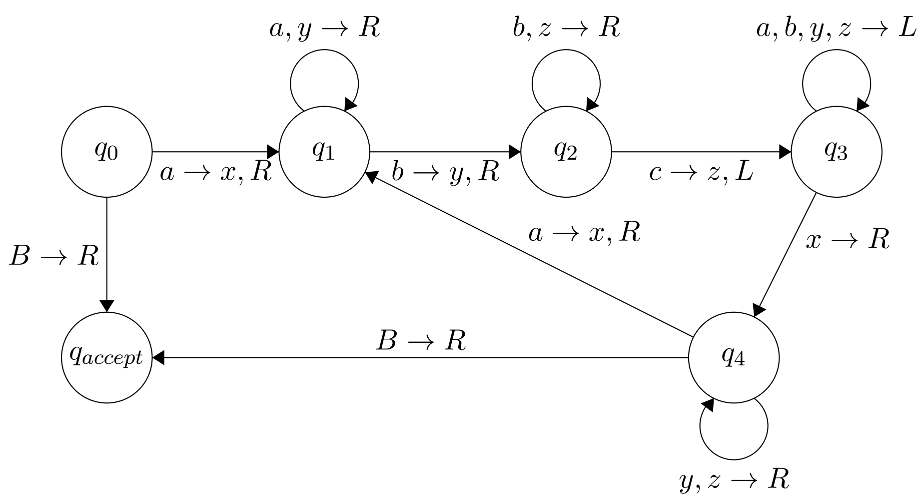
   The input alphabet of your TM should be $\Sigma = \{a, b, c\}$ and the tape alphabet should be $\Gamma = \{a, b, c, x, y, z, B\}$ where B is the blank symbol.

   The idea for your design should be as follows. Successively, scanning left to right and "rewinding" after each pass, replace each a in the input string by an x, each b by a y, and each c by a z. The input string shoud be accepted only if after the last pass, you are left with a string on the tape that consists of x's, y's, and z's only.

Solution.

The following Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ accepts the language stated above, where:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_{accept}, q_{reject}\}$,
- $\Sigma = \{a, b, c\}$,
- $\Gamma = \{a, b, c, x, y, z, B\}$, where B is the blank symbol,
- $q_0 \in Q$ is the start state,
- $q_{accept} \in Q$ is the accept state,
- $q_{reject} \in Q$ is the reject state, and
- δ is the transition function shown in the state diagram below: (Note that the reject state has been excluded from the diagram, and any missing transitions should be assumed to write nothing to the tape, move the tape head to the right, and transition into the reject state. If a transition does not indicate what to write to the tape, assume M leaves the tape unchanged.)



Explanation of Design

- M starts in q_0. If a blank is read, transition to q_{accept} and accept. If a b or c is read, reject. If an a is read, write an x to the tape, move the tape head right, and transition to q_1.
- Continually read a's and y's (though at this point there are no y's). If a b is read, mark it with a y, move the tape head right and transition to q_2. If any other symbol is read instead, reject.
- Continually read b's and z's. If a c is read, mark it with a z, move the tape head back left, and transition to q_3. If any other symbol is read instead, reject.
- Rewind the tape, reading a's, b's, y's, and z's. When an x is read, stop rewinding and move the tape head to the right. Transition to q_4.
- If an a is read, mark is with an x, move the tape head right, and transition back to q_1 to repeat the marking process. If instead the tape only reads y's and z's with no remaining b's or c's, transition to q_{accept} and accept.

(b) Write down the sequence of configurations that your Turing machine of part (a) goes through on each of the following input strings:

- $w = \varepsilon$ (the empty string),
- $w = a$,
- $w = ab$,
- $w = ba$,
- $w = abc$,
- $w = aabbcc$.

Solution.

- $w = \varepsilon$ (the empty string),
  - $q_0 B$
  - $B q_{accept} B$
- $w = a$,
  - $q_0 a$
  - $x q_1 B$
  - $x B q_{reject} B$
- $w = ab$,
  - $q_0 ab$
  - $x q_1 b$
  - $x y q_2 B$
  - $x y B q_{reject} B$
- $w = ba$,
  - $q_0 ba$
  - $b q_{reject} a$
- $w = abc$,
  - $q_0 abc$
  - $x q_1 bc$
  - $x y q_2 c$
  - $x q_3 y z$
  - $q_3 x y z$
  - $x q_4 y z$
  - $x y q_4 z$
  - $x y z q_4 B$
  - $x y z B q_{accept} B$
- $w = aabbcc$.
  - $q_0 aabbcc$
  - $x q_1 abbcc$
  - $x a q_1 bbcc$
  - $x a y q_2 bcc$
  - $x a y b q_2 cc$
  - $x a y q_3 b z c$
  - $x a q_3 y b z c$
  - $x q_3 a y b z c$
  - $q_3 x a y b z c$

- xq_4aybzc
- xxq_1ybzc
- $xxyq_1bzc$
- $xxyyq_2zc$
- $xxyyzq_2c$
- $xxyyq_3zz$
- $xxyq_3yzz$
- xxq_3yyzz
- xq_3xyyzz
- xxq_4yyzz
- $xxyq_4yzz$
- $xxyyq_4zz$
- $xxyyzq_4z$
- $xxyyzzq_4B$
- $xxyyzzBq_{accept}B$

3. Create a 4-tape deterministic Turing machine for the same language as in Question 2. The input alphabet of your TM should be $\Sigma = \{a, b, c\}$ and the tape alphabet should be $\Gamma = \{a, b, c, B, \#\}$ where B is the blank symbol.

   The idea for your design should be as follows.

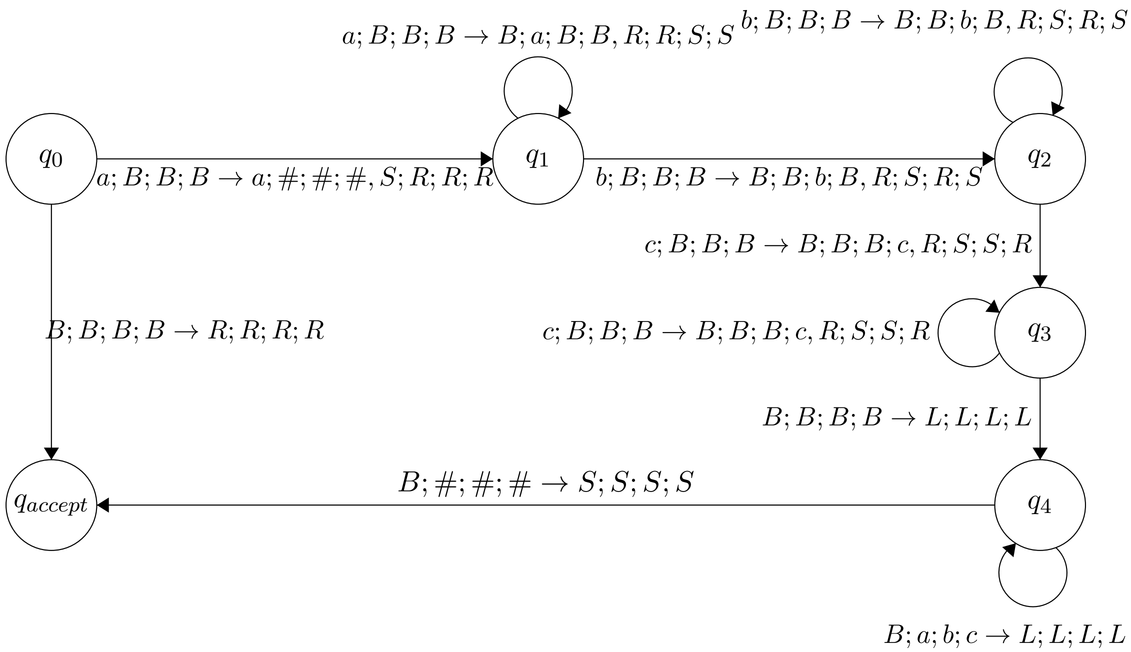
   - Write all the a's of the input string to tape 2, then all the b's to tape 3, and finally, all the c's to tape 4;
   - Scan tapes 2, 3, and 4 simultaneously to to compare the number of input symbols on each tape.

   Be careful that your don't find yourself in an infinite loop when "rewinding" your tapes. For example, scanning left on tape 2 whenever the tape head reads an a will result in an infinite loop unless the beginning of the input string is marked.

   Solution.

   The following Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ accepts the language stated above, where:

   - $Q = \{q_0, q_1, q_2, q_3, q_4, q_{accept}, q_{reject}\}$,
   - $\Sigma = \{a, b, c\}$,
   - $\Gamma = \{a, b, c, B, \#\}$, where B is the blank symbol,
   - $q_0 \in Q$ is the start state,
   - $q_{accept} \in Q$ is the accept state,
   - $q_{reject} \in Q$ is the reject state, and
   - δ is the transition function shown in the state diagram below: (Note that the reject state has been excluded from the diagram, and any missing transitions should be assumed to write nothing to the tapes, move the tape heads to the right, and transition into the reject state. If a transition does not indicate what to write to the tapes, assume M leaves the tapes unchanged. This TM design also allows the tape heads to stay in one location rather than having to move left or right only.)



Explanation of Design

- M starts in q_0. Tape 1 contains the input string, and tapes 2, 3, and 4 contain only blanks. If a blank is the first symbol read on tape 1, transition to q_{accept} and accept. If an a is read on tape 1, write '#' to tapes 2, 3, and 4, move their tape heads to the right while leaving tape 1 and tape head 1 alone. Transition to q_1. (If any symbol besides a blank or a is read on tape 1, reject).
- Continually read a's from tape 1, copy them to tape 2, and overwrite the a's on tape 1 with blanks. Do not move the tape heads on tapes 3 or 4. If a c or a blank is read on tape 1, reject. When we encounter the first b, copy it to tape 3, move tape heads 1 and 2 to the right, and transition to q_2.
- Continually read b's from tape 1, copy them to tape 3, and overwrite the b's on tape 1 with blanks. Do not move the tape heads on tapes 2 or 4. If an a or a blank is read on tape 1, reject. When we encounter the first c, copy it to tape 4, move tape heads 1 and 3 to the right, and transition to q_3.
- Continually read c's from tape 1, copy them to tape 4, and overwrite the c's on tape 1 with blanks. Do not move the tape heads on tapes 2 or 3. If an a or b is read on tape 1, reject. When we encounter a blank on tape 1, move all tape heads to the left and transition to q_4.
- Rewind all tapes, reading blanks on tape 1, a's on tape 2, b's on tape 3, and c's on tape 4. Continue reading and moving tape heads left until we encounter a '#' on tape 2, 3, or 4. If we encounter a '#' on one or two tapes only, reject. If the '#' is read from tape 2, 3, and 4 at the same time, transition to q_{accept} and accept.