York University
CSE 2001 Fall 2017 – Assignment 1 of 4
Instructor: Jeff Edmonds

1. (50 marks) **Turing Machine:** Write all the transition rules for a Turing Machine that solves the palindrome problem. A palindrome is a string that is the same if you reverse it. For example, 9235329 is an odd length palindrome and 923329 is an even length one. Your TM given a string of characters from $\{0, 1, ..., 9\}$ will answer with either yes or no. The input will have a blank b at the beginning and end, i.e. $b9235329b$. The head will be at the first character, which is a c.

   (a) Writing down TM descriptions is hard. Instead, start by writing pseudo code using variables and loops. The only allowed actions are to read and write to the tape where the head is and to move the head to the left and right. I am also a big believer in *loop invariants*. This is a clear picture of what the tape looks like at the beginning of each iteration.
   Hint: The loop invariant should be that either the TM has already halted because it has already discovered that the string is not to a palindrome or for some $i \geq 0$, the first i characters and the last i characters have been correctly matched and blanked out. What remains is to check that the remaining string is a palindrome. For example, with $i = 2$ the tape would contain $bbb353bbb$. The head will be at the first remaining character, which is stored in c.

   (b) Translate this code into Turing Machine Transitions

   $$\delta(q_{\langle line=?,x=?\rangle} = \left\langle q_{\langle next\ state\rangle}, char\ to\ write,\ direction\ move\ head\right\rangle$$

   . Recall the states are named with the line number and the value of each variable.

   (c) How many states does your TM have?

- Answer:

  (a) Pseudo Code:

```
0: Put head on first character.
   c = getCharAtHead()
   loop                         Loop Invariant: See above.
1:     if( c =blank )           No remaining characters
           halt with YES is a palindrome
       else
           Remember character in x.
           Blank out character and move right
2:     if( c =blank )           Only was only one char left.
           Halt with YES is a palindrome
3:     Move right till to next blank and then left
4:     If character does not match x, halt with NO is not palindrome
       If does match, blank out, move left, and forget x.
5:     Move left till to next blank and then right
   end loop
```

  (b) The first thing to know is the form of the transition function is $\delta(q, c) = \langle q', c', direction\rangle$, i.e. when in current state is q and the character on the tape being seen by the head is c, then transition to state q', write c' onto the tape, and move the head in the direction indicated.
  The second thing to know is that the names of the states should indicate which line the pseudo code is in and what values are being saved on pooh bear's black board.

  Turing Machine Transitions
  Start State: $q_{\langle line=1\rangle}$
  $\forall x' \in \{0, 1, ..., 9\}$ and $\forall c \in \{0, 1, ..., 9\}$
  $\delta(q_{\langle line=1\rangle}, b) = \left\langle q_{\langle haltYES\rangle}, b, stay\right\rangle$

$\delta(q_{\langle line=1\rangle}, c) = \left\langle q_{\langle line=2, x=c\rangle}, b, right\right\rangle$

$\delta(q_{\langle line=2, x=x'\rangle}, b) = \left\langle q_{\langle haltYES\rangle}, b, stay\right\rangle$
$\delta(q_{\langle line=2, x=x'\rangle}, c) = \left\langle q_{\langle line=3, x=x'\rangle}, c, stay\right\rangle$

$\delta(q_{\langle line=3, x=x'\rangle}, c) = \left\langle q_{\langle line=3, x=x'\rangle}, c, right\right\rangle$
$\delta(q_{\langle line=3, x=x'\rangle}, b) = \left\langle q_{\langle line=4, x=x'\rangle}, b, left\right\rangle$

$\delta(q_{\langle line=4, x=x'\rangle}, c) = \left\langle q_{\langle haltNO\rangle}, b, stay\right\rangle$ if $c \neq x'$
$\delta(q_{\langle line=4, x=x'\rangle}, x') = \left\langle q_{\langle line=5\rangle}, b, left\right\rangle$
$\delta(q_{\langle line=4, x=x'\rangle}, b) = \left\langle q_{\langle haltPANIC\rangle}, b, stay\right\rangle$

$\delta(q_{\langle line=5\rangle}, c) = \left\langle q_{\langle line=5\rangle}, c, left\right\rangle$
$\delta(q_{\langle line=5\rangle}, b) = \left\langle q_{\langle line=1\rangle}, b, right\right\rangle$

One common mistake (that does not arrise here) is:
$\delta(q_{\langle line=1, x=x'\rangle}, c) = \left\langle q_{\langle line=2, x=c\rangle}, x, right\right\rangle$
Note how x is being written. This is very bad because x is nothing in the TM other than part of the name of a state. It is not actually a variable that can be written.

(c) One for each of lines 1 and 5 and ten for each of lines 2, 3, and 4, for a total of 32.

2. (10 marks) What is a *Loop Invariant* and how is it useful in the designing of a Turing Machine?

- Answer: It is a static picture of what must be true every time the computation is at the top of the loop. A TM only as a finite number of states. Hence it must eventually cycle back to the same state. A loop invariant helps to design the algorithm to line up what is true at the end of one such cycle and the beginning of the next.

3. (10 marks) Does a TM have lines of code that it executes and a pooh bear small black board it can write on? Explain these concepts.

- Answer: No. A TM only has a tape with a head and always is in one of a finite set of states. To give structure and meaning to a TM, Jeff imagines that it has lines of code and a black board, but really this is just a renaming of the states.

4. (10 marks) Explain in your own words what it means and why it is true or why it is not.
Use the game taught in class to prove either it or its negation.
When appropriate sketch in a few sentences what the TM does and how many states it has.
$\forall$ integers k, $\exists$ TM M, $\forall x, y \leq k$, M multiplies $x \times y$ in one time step,
i.e. it can read in $\langle x, y\rangle$ and then simply know the answer.

- Answer: It is true. Any finite language can be computed by a TM.
Proof: Let k an arbitrary integer.
Let M be the TM that has $O(k^2)$ states that reads in any $x', y' \leq k$ and goes into state $q_{\langle line=1, x=x', y=y';\rangle}$. Then in one step with table look-up, it can know the product $x' \times y'$.

5. (10 marks) Explain in your own words what it means and why it is true or why it is not.
Use the game taught in class to prove either it or its negation.
When appropriate sketch in a few sentences what the TM does and how many states it has.
$\exists$ TM M, $\forall x, y$, M multiplies $x \times y$ in one time step,
i.e. it can read in $\langle x, y\rangle$ and then simply know the answer.

- Answer: No this is not true. It cannot remember arbitrary integers in its states. To do so, the number of states would needed grows as a function of x and y.

Negation: $\forall$ TM M, $\exists x, y$, M fails to multipliers $x \times y$ in one time step.
Proof: Let M be an arbitrary TM.
Let k be the number of states that it has.
From some of these states in may know how to multiply some values in one step.
But there are infinitely many integers. Let x and y be inputs for which this TM does not know how to multiply in one time step.