COMS 3003A
HW 3

Dmitry Shkatov

26 February, 2024

Reading:
Sipser. Chapter 3.

For debugging Turing machines, use this simulator.

Warm-up questions:

(a) Design a TM chat checks if a binary string contains a 0.

(b) Design a TM that checks if a binary string ends with a 0.

(c) Design a TM that checks if a binary string ends with two consecutive 0's.

(d) Design a TM that checks if a binary string begins and ends with the same symbol.

(e) Design a TM that checks if a binary string begins and ends with the same symbol and has length at least 3.

1. A language over alphabet Σ is a subset of Σ^*. Show that there exists a one-to-one correspondence between languages over Σ and decision problems over Σ.

2. Design a Turing machine that appends 0 to the input string. (Keep in mind the convention that at the end of the computation, the TM should scan the first letter of the output.)

3. Design a Turing machine that increments by 1 a natural number represented in binary.

4. Design a Turing machine that transforms a binary string w into the string ww.

5. Design a Turing machine that inserts a single blank between every pair of adjacent symbols of the binary input string (e.g., 0110 gets transformed into 0␣1␣1␣0).

6. According to the definition of TMs in Sipser, a head of a TM can move left and right, but cannot stay in place, i.e., it cannot be reading the same tape cell in two successive configurations. We will, however, want to design TMs that stay in place while changing configurations. Show that this feature can be implemented on TMs whose instruction set includes only left and right movement; in other words, show that every instruction of the form $q_i s_k q_j s_l S$, where S stands for 'stay in place', can be simulated with a sequence of instructions of a TM with only left and right movements. (Thus, we can think of instructions that instruct the TM to stay in place as part of the library of the TM programming language.)