COMS 3003A
HW 4

Dmitry Shkatov

8 March, 2024

Reading:
Sipser. Chapter 3.

For debugging Turing machines, use this simulator.

Bonus questions are not compulsory.

Warm-up:

(a) Sipser, Exercise 3.1.

(b) Sipser, Exercise 3.2.

Exercises:

(1) Sipser, Exercise 3.4.

(2) Sipser, Exercise 3.5.

(3) Sipser, Exercise 3.6.

(4) Sipser, Exercise 3.7.

(5) Sipser, Exercise 3.8.

(6) Sipser, Exercise 3.11.

(7) Show that TMs with a two-way infinite tape and TMs with a one-way infinite tape have the same computing power.

Simulation of multi-tape machines by single-tape machines:

In this section, we will show that single-tape TMs can simulate multiple-tape TMs. Sipser sketches out the argument in Theorem 3.13, but we want to see a bit more detail to make sure that Sipser is not misleading us.

1. Show that the following instructions can be implemented on a single-tape TM:

$$\begin{aligned} q_i s_k &\rightarrow q_j s_l R^2; \\ q_i s_k &\rightarrow q_j s_l L^2, \end{aligned}$$

where R^2 stands for 'move to the right twice' and L^2 stands for 'move to the left twice'.

2. Generalise the previous problem: implement instructions involving movements R^n and L^n, for every $n \geqslant 2$.

3. Construct a Turing machine that takes input w and writes all the symbols of w on its tape two spaces apart (e.g., abc becomes $a\sqcup\sqcup b\sqcup\sqcup c$).

4. Generalise the previous problem to n spaces apart, with $n \geqslant 1$.

5. Suppose we have a Turing machine that solves a decision problem encoded in an alphabet containing k symbols, with $k > 2$. Show that we can construct a machine that can solve the same problem encoded in binary.

6. Generalise the previous problem: go from an alphabet with k symbols, where $k > 2$, to an alphabet with m symbols, where $2 \leqslant m < k$.

7. Now you should be able to fill in the details of the proof of Sipser's Theorem 3.13: given a multi-tape Turing machine that solves a problem, construct a single-tape machine that solves the same problem. Start with a particular case, say with 3-tape machines, and then generalise to any number of tapes.

8. Bonus: If the original mupti-tape machine solves the problem it has been designed to solve in time $t(n)$, where n is the length of the input string, what time does the single-tape machine that you constructed take to solve the problem?

Bonus:

- We have seen that instructions of the form

  $$q_i s_k \rightarrow q_j s_l S,$$

  where S means 'stay in place', can be simulated on the Turing machines whose original instruction set includes only movement to the left and to the right. Suppose that we now modified the form of instructions so that our machines can only move to the right and stay in place, but cannot move to the left, i.e., every instruction either has the form

  $$q_i s_k \rightarrow q_j s_l S,$$

  or has the form

  $$q_i s_k \rightarrow q_j s_l R.$$

  Show that there exist decision problems that can be solved by Turing machines that can move to the right and the left, but cannot be solved by the proposed modification to Turing machines.