Prep Work 11 - Turing Machines

CS 234

due April 14, before class

0 Introduction

This assignment has 1 part: Turing machines.

This assignment is to be completed individually, but feel free to collaborate according to the course's external collaboration policy (which can be found in the syllabus).

The deliverables consist of one `.pdf` file. The deliverables should be submitted electronically by the deadline. Put any attribution text in the `.pdf` file.

Every file should be named like `FLast_cs234_pX.ext` where `F` is your first initial, `Last` is your last name, `X` is the assignment number, and `ext` is the appropriate file extension. For example, Alan Turing's `.pdf` file should be given the name `ATuring_cs234_p11.pdf`. (Alan Turing was the inventor of the Turing machine, and this work is considered to have started the field of computer science. Turing's advisor, Alonzo Church, also advised Stephe Cole Kleene (of the Kleene star), who in turn advised Robert Constable, who in turn advised me when I was an undergrad.)

1 Turing Machines

Read chapter 13 in the textbook. Then complete the following tasks in your `.pdf` submission. Clearly label your responses with the task number.

1. What does a Turing machine have that a DFA does not?
2. By convention, we don't write all the transitions that a Turing machine could take, even though it is deterministic. What does the Turing machine do if it encounters a transition that we did not write down?
3. Describe the parts of the 7-tuple defining a Turing machine.
4. Look at Figure 13.1's Turing machine. Give the trace for the Turing machine on input 01. (See Examples 13.2-4 for examples of how a trace can be written.)
5. In your own words, why can every regular language be accepted by a Turing machine? (You do not need to prove this, just give intuition.)
6. In your own words, describe how a Turing machine can be used to represent a function.