Homework 6: Turing Machines

CSE 30151 Spring 2022

Due: Thursday, March 31 at 11:59pm

Instructions

- Use this document to create a PDF file containing your solutions. Do this either by (1) printing this document, writing your solutions on it, and scanning[1] your work into a PDF; or (2) writing your solutions on this PDF digitally. Either way, please ensure that your work is clearly legible.
- If you need extra blank pages, feel free to insert them as needed. The amount of blank space provided beneath a problem does not necessarily indicate the expected length of your solution.
- You have the option to submit your solutions all at once or in parts; late penalties will only be applied to problems that are late. Individual problems cannot be submitted for grading in this way more than once.
- If you plan to submit some parts of your assignment late, before the deadline, upload a single PDF containing the problems you have completed so far. Do not include solutions to problems you want graded later for late credit. After the deadline, if you want to submit additional problems, add them to your original PDF and upload it again.[2]
- Submit your PDF file in Canvas under Assignments > Homework 6: Turing Machines. You may re-submit your work any number of times before the due date.

[1] For tips on scanning your work using your mobile device, see `https://help.gradescope.com/article/0chl25eed3-student-scan-mobile-device`.

[2] For tips on concatenating your old and new PDFs together, see `https://help.gradescope.com/article/tp9kl4yx4q-student-troubleshooting-submissions`.

In all problems below, "implementation description" refers to a plain English description of a Turing machine's operation on its input, making specific reference to the behavior of the tape head, but not including details about states and transitions (it is defined on pages 170-171 and 184-185 of the textbook). See Examples 3.7, 3.11, 3.12, and 3.14 in the textbook for examples of the expected format.

1. For each of the following languages, provide a state diagram for a Turing machine that decides it, and a brief description of how your machine works. The reject state may be implicit.

   (a) $L_1 = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$ **(3 points)**

(b) $L_2 = \{w \in \{\texttt{0}, \texttt{1}\}^* \mid w \text{ is an integer } k \geq 0 \text{ encoded in binary without leading } \texttt{0}\text{'s}\}$

Note that the string `0` is in the language and represents $k = 0$; it does not count as a "leading `0`."

Additionally, when the Turing machine accepts, it must have the binary encoding of $k + 1$ on the tape (also without leading `0`'s). For example,

- If the input is `0`, then the machine should accept with `1` on the tape.
- If the input is `1`, then the machine should accept with `10` on the tape.
- If the input is `10`, then the machine should accept with `11` on the tape.
- If the input is `01`, then the machine should reject, and it doesn't matter what's on the tape.

(4 points)

2. For each of the following languages, provide an implementation description for a Turing machine that decides it. You may use any of the Turing machine extensions discussed in class or in the textbook.

   (a) $L_3 = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid c_{\mathtt{0}}(w) \neq 2c_{\mathtt{1}}(w)\}$ **(3 points)**

(b) $L_4 = \{\mathtt{a}^n \mid n \text{ is a Fibonacci number}\}$ **(4 points)**

3. A Turing machine with a doubly infinite tape is like a standard TM as defined in the textbook, but with a tape that extends infinitely in both directions (not just to the right). Initially, the head is at the first symbol of the input string, but there are infinitely many blanks to the left. Show how, given a TM with a doubly infinite tape, to construct an equivalent standard TM. You may give an implementation description in the style of Proof 3.13, and you may use any results proved in class or in the textbook. (This question is the same as Problem 3.11 in the textbook.) **(4 points)**

4. This question pertains to showing that the classes of decidable languages and Turing-recognizable languages are closed under concatenation (Problems 3.15b and 3.16b in the textbook).

Suppose we have two Turing-recognizable languages L_1 and L_2, and let M_1 and M_2 be Turing machines that recognize them, respectively.

Consider the following description of a Turing machine M' that attempts to recognize the language L_1L_2.

M' = "On input w:
1. Let $x = \varepsilon$ and $y = w$.
2. Run M_1 on input string x. If M_1 accepts, continue to stage 3. If M_1 rejects, go to stage 4.
3. Run M_2 on input string y. If M_2 accepts, *accept*. If M_2 rejects, continue to stage 4.
4. If $y = \varepsilon$, *reject*. Otherwise, remove the first symbol from y and append it to x, and go to stage 2."

(a) Explain why this construction succeeds in showing that decidable languages are closed under concatenation but fails to show that Turing-recognizable languages are closed under concatenation. **(3 points)**

(b) Provide a construction for M' that does work for Turing-recognizable languages. **(3 points)**

Changelog

- **Mar 29:** Clarification in 1b about leading `0`'s in the encoding of $k+1$.
- **Apr 14:** Added remarks to the solution for 2b.