Midterm Exam 2 Study Guide

CSE 30151 Spring 2018

Exam date: 2018/04/12

- The exam has six questions, worth 10 points each, for a total of 60 points (10% of your grade).

- You have the whole class period of 75 minutes to write your solutions.

- You may use your textbook and paper notes, but computers, smartphones, and tablets are **not** allowed.

- You may use the textbook, lectures, and lecture notes for this course without citation. However, you may **not** copy or quote from any other materials in your notes that you are not the author of.

- On the first page, please write your name and NetID, but please don't write any solutions. On the second page and following (front and back), please write your solutions, but please don't write your name.

The exam covers HW3–6: non-regular languages, CFGs and PDAs, non-context-free languages, and Turing machines. There will be no undecidability proofs on this exam. The six questions will be of the following types. (Exercise/problem numbers are from the 3rd US edition of Sipser; if the 3rd international edition has a different number, it is indicated by "intl.")

1. Prove that a language is nonregular. You can use the pumping lemma, any results proved in the book or in class, or any combination thereof. Like HW3 2b, 3b; Sipser 1.29ac, 1.46b (intl. 1.51b).

2. Write a CFG *and* a PDA that recognizes a language. For the PDA, a formal description (equations or state diagram) is required. Like HW4 1ab, 3c; Sipser 2.4ad, 2.6ac, 2.7ac.

3. Given some operation on languages, prove that CFLs are closed or not closed under this operation, or that this operation turns regular languages into CFLs or vice versa. Like HW5 2ab (but not as hard), 3ab; Sipser 2.38 (intl. 2.50; but not as hard).

4. Prove that a language is not context-free. You can use the pumping lemma, any results proved in the book or in class, or any combination thereof. Like HW5 1abc; Sipser 2.30bc (intl. 2.42bc).

5. Write a Turing machine that decides a language. A formal description (equations or state diagram) is required. Like HW6 1a; Sipser 3.8a.

6. Prove that a kind of machine is equivalent to Turing machines. Only implementation descriptions are needed, and we'll provide a template walking you through the proof (see below). Like HW6 2; Sipser 3.10 (intl. 3.17).

Example template for HW6 2 and its converse   A Turing machine with a doubly infinite tape is like a TM as defined in the book, 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, as usual, but there are infinitely many blanks to the left. Show that TMs with doubly infinite tapes are equivalent to TMs.

(a) Given a standard TM $S = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, construct an implementation-level description of a doubly-infinite TM D:

- If S's tape is $t_1 t_2 t_3 \cdots$, where $t_i \in \Gamma$, and $h \geq 1$ is the head position, what would the corresponding configuration of D look like?
  Solution: The tape would be $\cdots \texttt{\#} t_1 t_2 t_3 \cdots$, where the \texttt{\#} is at position 0 and the head is at position h.

- How should D simulate reading symbol a?
  Solution: Just read a.

- How should D simulate writing symbol b?
  Solution: Just write b.

- How should D simulate moving to the left?
  Solution: Move to the left. Then if the head is on \texttt{\#}, move back to the right.

- How should D simulate moving to the right?
  Solution: Just move to the right.

(b) Given a doubly-infinite TM $D = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, construct an implementation-level description of a standard TM S:

- If D's tape is $\cdots t_{-2} t_{-1} t_0 t_1 t_2 \cdots$, and h is the head position, what would the corresponding configuration of S look like?
  Solution: The tape would be $\texttt{\#} t_\ell \cdots t_r \cdots$, where ℓ is the leftmost visited square in D's tape and r is the rightmost visited square in D's tape. The head would be at position $h - \ell + 2$, that is, it points to t_h.

- How should S simulate reading symbol a?
  Solution: Just read a.

- How should D simulate writing symbol b?
  Solution: Just write b.

- How should S simulate moving to the left?
  Solution: Move to the left. Then if the head is on \texttt{\#}, insert a blank symbol immediately after \texttt{\#}, shifting all other symbols to the right, and leave the head on the new blank cell.

- How should S simulate moving to the right?
  Solution: Just move to the right.