Midterm Exam 2 Study Guide

CSE 30151 Spring 2022

Exam Date: March 24, 2022

Instructions

- The exam will be worth a total of 90 points, or 15% of your final grade.
- You will have the whole class period of 75 minutes to write your solutions.
- You may consult the following materials during the exam: the Sipser textbook, your notes, and homework solutions. All of these materials must be printed on paper. If you have a digital copy of the textbook, you must print out any sections you want to use beforehand.
- The following materials may *not* be used during the exam: solutions to problems taken from the Internet or other outside sources; any electronic devices (including smart watches, phones, tablets, computers, and other Turing-equivalent machines).
- You may re-use any theorems or proofs provided in class, in the textbook, or in the homework assignments. If you re-use a theorem or proof from the textbook not used in class, please cite the page number. Example: (Sipser p. 42).
- If you think any problem contains typos or is unclear, please ask the instructor for clarification.

Topics

This exam covers HW3 (including non-regular languages, but not DFA minimization), HW4, and HW5 (including PDAs and CFG-to-PDA conversion, but not non-context-free languages). It does *not* cover DFA minimization, deterministic PDAs/CFLs, CFG parsing using the CKY algorithm, proving that a language is not context-free, the pumping lemma for CFLs, or Turing machines.

The exam may include any of the following topics or types of question:

- Prove that a language is not regular. You can use the pumping lemma, known closure properties on regular languages (particularly reversal, intersection, and homomorphisms), or any combination thereof. You may refer to any known non-regular languages discussed in class, the textbook, or homework.
- Given a language, design a CFG that generates it.
- Given a language, design a PDA that recognizes it. Provide a state diagram for your PDA. You do not need to label the states, although you may do so if you wish. Do *not* use shorthand notation for popping or pushing multiple symbols.
- Given a regular expression or DFA, design an equivalent CFG. We discussed regular expression-to-CFG conversion in class. A construction for DFA-to-CFG conversion is given at the top of Sipser p. 107.
- Given a CFG and a string it generates, show its parse tree(s) and derivation(s).

- Given a CFG, show that it is ambiguous, and remove its ambiguity without changing the language it generates.
- Convert a CFG to Chomsky normal form.
- Convert a CFG to an equivalent PDA. Do *not* use shorthand notation for pushing multiple symbols.
- Answer questions about the PDA-to-CFG construction (this is unlikely to be a full PDA-to-CFG conversion, as these conversions tend to be too long and repetitive for exam questions, even for small PDAs).
- Proofs or conceptual questions about non-regular languages and context-free languages.

Practice Problems

Problem numbers for the international edition of the textbook are given in parentheses when they differ from the domestic edition.

1. For each of the following languages, prove that it is not regular.
   (a) $\{(\mathtt{ab})^n\mathtt{c}^n \mid n \geq 0\}$

   Advice: Try proving this using the pumping lemma, then try proving it again using a proof by contradiction with a homomorphism to reduce it to a known non-regular language.

   (b) $\{(\mathtt{ab})^n(\mathtt{ba})^n \mid n \geq 0\}$

   (c) $\{\mathtt{a}^n\mathtt{b}^m \mid n, m \geq 0 \text{ and } ((n \text{ is even and } m \leq n) \text{ or } (n \text{ is odd and } m \geq n))\}$

   (d) $\{w\mathtt{a}^i w^R \mid w \in \{\mathtt{a},\mathtt{b}\}^* \wedge 0 \leq i \leq 2|w|\}$

   (e) Exercise 1.29 in the textbook.

2. For each of the following regular expressions, provide a CFG that recognizes the language it describes.
   (a) $\mathtt{a}(\mathtt{b}^+\mathtt{a})^*$

   (b) $\mathtt{b}^*\mathtt{ab}^* \cup (\mathtt{aba} \cup (\mathtt{cc})^*)^+$

3. For each of the following languages, provide a CFG that generates it and a PDA that recognizes it.
   (a) $\{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i, j, k \geq 0 \wedge j \geq \min(i, k)\}$

   (b) $\{\mathtt{a}^n\mathtt{b}^n\mathtt{a}^m\mathtt{b}^m \mid n, m \geq 0\}$

   (c) $\{\mathtt{a}^i\mathtt{b}^j\mathtt{a}^k\mathtt{b}^\ell \mid i, j, k, \ell \geq 0 \wedge (i = k \vee j = \ell)\}$

   (d) $\{\mathtt{a}^i(\mathtt{bc})^j\mathtt{a}^k \mid i, j, k \geq 0 \wedge k \geq i\}$

   (e) $\{\mathtt{a}^n\mathtt{b}^m \mid n, m \geq 0 \wedge \frac{n}{2} \leq m \leq n\}$

   (f) Exercises 2.4-2.7 and problem 2.47 (intl. 2.58) in the textbook (for 2.7, give the state diagrams).

4. For each of the following CFGs, convert it to an equivalent PDA, and convert it to Chomsky normal form.

(a) $S \to S\texttt{+}T \mid T$
$T \to U\texttt{(}S\texttt{)} \mid U$
$U \to \texttt{a} \mid \texttt{(}S\texttt{)}$

Show the parse tree of the string `a(a+a)` according to the original grammar, and show the sequence of stacks after each transition in an accepting path of your PDA.

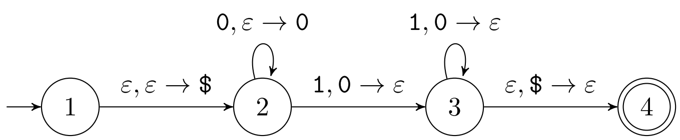
Show the parse tree of the string `(a(a+a))(a)` according to the original grammar.

(b) $S \to T\texttt{a}T\texttt{b}$
$T \to U \mid \varepsilon$
$U \to U\texttt{c} \mid \texttt{c}$

(c) CFG to PDA conversion: Exercises 2.11-2.12 in the textbook.

(d) Conversion to Chomsky normal form: Exercise 2.14 in the textbook.

5. Consider the following PDA for the language $\{\texttt{0}^n\texttt{1}^n \mid n \geq 1\}$.



(a) According to the conversion procedure in Lemma 2.27, what would the start variable of the equivalent CFG be?

(b) List all of the rules that would be created due to point 1 on p. 122.

(c) How many rules would be created due to points 2 and 3?

(d) Of all the rules created by points 2 and 3, only one needs to be added to the rules from part (b) to make the CFG complete. Which one is it?

6. Ambiguity.

(a) Show that the following CFG is ambiguous. Then, provide a CFG that generates the same language and is not ambiguous. (Note: this problem is marked as "hard" in the textbook and is more difficult than what would appear on an exam.)

$S \to \texttt{0}S\texttt{1} \mid \texttt{1}S\texttt{0} \mid SS \mid \varepsilon$

(b) Problems 2.28 (intl. 2.40) and 2.46 (intl. 2.18) in the textbook.

7. Proofs and conceptual questions.

(a) In English, the first letter of the first word of a sentence is always capitalized. However, when we designed CFGs for fragments of English, the grammars always generated strings of words that were entirely in lower case. It would be useful to be able to transform those grammars into a form that always capitalizes the first word in the sentence. To do this, let us define an operation on languages called CAP that capitalizes the first symbol in every string in a language.

Let us treat each word as a single symbol, and let the capitalized version of word/symbol a be denoted $\textsc{Cap}(a)$. For example, $\texttt{boy} \in \Sigma$, $\texttt{Boy} \in \Sigma$, and $\textsc{Cap}(\texttt{boy}) = \texttt{Boy}$. Given a string $w = w_1 w_2 \cdots w_n$, let $\textsc{Cap}(w)$ represent the string w with its first symbol capitalized (if

w is empty, then $\textsc{Cap}(w) = \varepsilon$). Put more formally,

$$\begin{aligned}\textsc{Cap}(w_1 w_2 \cdots w_n) &= \textsc{Cap}(w_1) w_2 \cdots w_n \quad \text{if } n \geq 1 \\ \textsc{Cap}(\varepsilon) &= \varepsilon\end{aligned}$$

Finally, if L is a language, let

$$\textsc{Cap}(L) = \{\textsc{Cap}(w) \mid w \in L\}$$

Prove that CFLs are closed under Cap. (Hint: You can use the fact every CFL can be expressed as a CFG in Chomsky normal form to simplify your proof.)

(b) Exercises 2.18a (intl. 2.30), 2.15, and problem 2.35 (intl. 2.47) in the textbook.

Changelog

- **Mar 21:** Added international edition problem numbers; added note about example 6a.
- **Mar 21:** Typo fix in 3e.
- **Mar 22:** Removed exercise 2.2 and added 7a.