UNCA CSCI 431
Exam 3 Fall 2019
3 December 2019 – 3:15 pm to 4:55 pm

You may use your notes, printouts, scratch paper, and your textbook. You may not use any calculators, electronic devices, or help from any other source or person.

Anyone needing a break during the exam must leave their exam with the instructor.

This exam must be turned in before 4:55 PM.

Name:________________________________

There are five equally-weighted questions.

Problems 2 and 5 spread over two pages to allow more room for artwork.

Extra space for long-winded answers of exam problems.

Problem 1: Regular Expressions, Grammars, and Derivations

Problem 1A: Regular expressions

Write a regular expression that represents **all** strings of 0 and 1 with an even number of 1's. You can use either the textbook or **`grep`** syntax in your answer.

Note that 001100 and 11 and ε are **in**, but 000100 and 1 are **out**.

Problem 1B: Context free grammar

Write a context free grammar that represents **all** strings of 0 and 1 with an even number of 1's.

Note that 001100 and 11 and ε are **in**, but 000100 and 1 are **out**.

Problem 1C: String derivation

Illustrate the derivation of the string 0001100 using the grammar you created for Problem 1B.

Problem 2: CFG transformations

Use the following CFG in both subproblems

- The alphabet for the language is **Σ** = {**0**, **1**}.
- The start variable for the grammar is **T**.
- Here are the rules:
    - **T → 0M0**
    - **M → 1T | T1 | ε**

Problem 2A: CFG → CNF

Convert the Context Free Grammar to Chomsky Normal Form.

You should stick with the procedure described in Example 2.10 of the textbook.

Continuing with Problem 2

Use the following CFG in both subproblems

- The alphabet for the language is **Σ** = {**0**, **1**}.
- The start variable for the grammar is **T**.
- Here are the rules:
  - **T → 0M0**
  - **M → 1T | T1 | ε**

Problem 2B: CFG → PDA
Generate a PDA for the Context Free Grammar shown above.

You should stick with the **flower algorithm** presented in Theorem 2.20 of the textbook.

Pumping Lemma (Theorem 1.70)

If A is a regular language, then there is a number p (the pumping length) where, if s is any string in A of length at least p, then s may be divided into three pieces, $s = xyz$, satisfying the following conditions:

- for each $i \geq 0$, $xy^iz \in A$
- $|y| > 0$
- $|xy| \leq p$

Problem 3: Disproving regularity

Show that the language $\{0^i10^i \mid i \geq 0\}$ is **not** regular.

Note that 1 and 001 and 00100 are **in**, but 0000, 00011000 and 101 are **out**.

Problem 4: Turing Machines

In Problem 4 of the last exam, you (hopefully) proved that the following language is **not** context free:

- $\{a^i b^j c^k \mid i \geq j \text{ and } i \geq k\}$

In this exam, show that this language is decidable by designing a Turing machine that decides the language. You may assume that the input alphabet for the language are the symbols a, b and c.

In your design, do not write a super-formal description that satisfies the formalities of Definition 3.3 as shown in Figure 3.10. Instead give a concise “informal” description similar to that shown in Example 3.11 (p 174). You could also use some drawings of Turing Machine configurations to illustrate your solution.

If you run out of room here, continue on the bottom of the page 7.

Problem 5: Counting and reducing

Problem 5a: Enumerating the powers

Show that the set of all integral powers of integers, i.e., i^n for integers i and n, is countable. Consider 0^0 to be 1, just as C, Java, JavaScript and Python do. Also, 3^{-5} would be in this set.

Extra space for long-winded answers of previous problems.

Continuing with Problem 5

Problem 5B: Problem 5.24

Show that the language $5B_{TM}$ defined below is undecidable.

- $5B_{TM}$ ={ <*M*, *n*, *w*> | *n* encodes a prime number
  or *M* is a Turing machine that accepts w}

Hint: The language A_{TM} described below is undecidable.

- A_{TM} = { <*M*, *w*> | *M* is a Turing machine that accepts *w* }

Extra space for long-winded answers of previous problems.