Formal Languages – CSE 4083/CSE5010 – Final Exam

William David Shoaff wds@cs.fit.edu

Spring 2022 The Final Examination will be a take-home exam Posted on Tuesday, May 3, and due by Friday, May 6 by 10:00 A.M.

1 Final Information

1.1 Resources

Most questions and answers can be found in the Linz textbook, the slides, and Doctor Stansifer's (excellent) Web page for this course.

Send me an email (wds@cs.fit.edu) should you have question, concerns, or doubts.

2 Deterministic Finite Automata

1. (5 points) Give a precise (mathematical) definition of a DFA machine M and how its parts work together to solve problems.

3 Nondeterministic Finite Automata

2. (10 points) Give a precise (mathematical) definition of a NFA machine M and outline how the NFA can be converted into a DFA.

4 Regular Expressions

3. (10 points) Describe how regular expressions are use in programming. For example, to define numbers, identifiers, keywords, *etc*.

5 Closure Properties of language classes and operations

4. (5 points) Some operations are closed for certain language classes. List classes that are closed with respect to (common formal language operations)

6 Decision Problems about language classes

5. (5 points) Some "yes/no" questions can be answered correctly by some algorithm. Other questions place constrains on parameters to filter the solutions and perhaps select a best answer depending on some criterion. Other questions have no known computable answer.

7 Pushdown Automata & Context-Free Languages

6. (5 points) Give a precise (mathematical) definition of a PDA and how its parts work together to solve problems.

8 Production Rule for Context-free Grammars

7. (5 points) What is *ambiguity* with respect to parsing a string w for membership in a language L generated by a context-free grammar G? Why does it master? Can it be mitigated?

8. (5 points) Chomsky Normal Form: Most useful grammars can be translated into a *normal form* making answers to questions about the language more transparent.

9 Pumping Lemmas

The *Pumping lemma for regular languages* is easy to visualize. (if a FSM for a regular language has fewer states than it's input string length, then by the pigeonhole principle, there must be a loop in the machine that can pump a sub-string that can be pumped.)

9. (5 points) How is this pumping lemma used (what is its proof technique)? Describe the structure of its pre-conditions, and give an example of it's use.

   The *Pumping lemma for Context-Free languages* is more difficult for me to visualize.

   (if a PDA for a context-free language has accepts a sufficiently long string then there is an early sub-string and a later sub-string each of which may be pumped.)

   If w is a efficiently long string in a CFL then its parse tree can be pruned into 5 parts: A prefix tree, a sub-tree that can be repeated, a middle tree, another repeatable sub-tree, and a boundary tree.

10. (5 points) How is this pumping lemma used (proof technique)? That is, describe the structure of its pre-condition, and give an example of it's use.

11. (5 points) When is a Context Free Grammar in Chomsky Normal Form?

10 Transition Functions and Relations

12. (5 points) What are precise (mathematical) definitions of *function* and *relation*?

13. (5 points) Why is the (major) difference between function and relation interesting as related to classifying machines as *deterministic* or *non-deterministic*?

14. (10 points) In the study of formal languages state-transition diagrams are often used to visualize changes in a machine's configuration as it acts on input. To visualize a machine's configuration think of its parts: Commonly a finite set of states, a finite input alphabet Σ, Perhaps storage devices (a stack, input/output tapes). The states are pictured as named circles sometimes decorated with symbols to denote special states, e.g. start and final states. Changes in configuration are denoted by labeled edges and perhaps changes in storage.

    1. Describe how edges are labeled and their meaning for finite state machines.
    2. Describe how edges are labeled and their meaning for pushdown automata.
    3. Describe how edges are labeled and their meaning for Turing machines.

11 Turing Machines

15. (5 points) Give a precise (mathematical) definition of a Turing machine (TM) and how its parts work together to solve problems. Perhaps, not surprisingly (maybe surprisingly ?), this most powerful machine envisioned is also the most simple known. (Other methods of computation have been developed, but each have been shown to be *equivalent* in *expressible* power as Turing machines). What does does this statement mean?

16. (5 points) Describe a problem that cannot be solved by a TM. What does *Undecidability* mean to you *philosophically*?

17. (5 points) Is there a pumping lemma for Turing machines? if so, what is it? if not, why not?

Question	Points	Score
1	5	
2	10	
3	10	
4	5	
5	5	
6	5	
7	5	
8	5	
9	5	
10	5	
11	5	
12	5	
13	5	
14	10	
15	5	
16	5	
17	5	
Total:	100	