Formal Languages — CSE 4083 & CSE 5210

Instructor: William David Shoaff Email: [wds@cs.fit.edu](mailto:wds@cs.fit.edu) Phone: (321) 474-7823 Term: Spring 2022, Published: Wednesday, March 2, 2022; Due: Saturday

March 5, 2022 (11:59 P.M.

Answer the questions in the space provided.

Your Name, Class, & Major:

You are not to consult other people when answering these questions.

You may consult sources (books, web pages, journals, *etc*, but cite your sources and don’t simply copy and paste answers.

# 1 Deterministic Finite Automata

1. (10 points) Consider a DFA M = (*Q,* Σ*, δ, s, f* ) with States *Q* = *{s, q*1*, q*2*, f}*, where

*s* is the *start* and *f* is the *final* state; Alphabet Σ = *{*0*,* 1*}*; and transition function *δ*.

Construct a state transition table for *δ* (or you can draw a state transition diagram) that recognizes regular expressions that are binary strings and multiples of 3, for example, the strings

0*,* 11*,* 110*,* 1001*,* 1100*, . . .*

would be accepted strings, but

1*,* 10*,* 100*,* 101*, . . .*

would not be accepted.

(Hint: Think, if *n* = 3*k* is a multiple of 3, then the next multiple of 3 is 3*k* + 3. this could be accomplished by a transition from the current state to a next state by scanning 3 ones.)

# Nondeterministic Finite Automata

1. (15 points) Explain how any NFA (with *λ* (or *c*) transitions) can be converted into a DFA that accepts the same language as that accepted by the NFA. That is, the expressive power of NFAs and DFAs are equivalent. This is known as the *Rabin-Scott Theorem*.

# Regular Expressions in the Programming World

1. (10 points) Consider a programming language that has identifiers that start with a low- ercase ASCII letter

*A* = *{a...z}*

followed by a string of 1 or more digits

*D* = *{*0*...*9*}*

or 1 or more lowercase ASCII letters. Show how to write this specification as a regular expression.

# Closure Properties of Languages

Answer these True (T) or False (F) questions. Give a brief explanation of your answer (for example, explain how to construct a machine that implements the property.)

1. (5 points) Regular languages are closed under *intersection*.
2. (5 points) Regular languages are closed under *Kleene-star*.

# Decision Properties of Languages

1. (5 points) What does it mean to say that a “yes” or “no” question is *undecidable*?
2. (5 points) Answer True (T) or False (F): It is decidable whether or not the language of a DFA is empty or non-empty. Give an explanation of your answer.
3. (5 points) It is decidable whether or not the language of a DFA is finite or infinite.
4. (5 points) It is undecidable whether or not the a string *s* is accepted by a DFA. Answer True (T) or False (F): Give an explanation of your answer.
5. (5 points) Answer True (T) or False (F): It is decidable whether or not two regular languages *L*1 and *L*2 are equal.

Give an explanation of your answer.

# Equivalence Relations

1. (10 points) On the set N of natural numbers define an equivalence relation *n m* if and only if

*≡*

*n* mod 3 = *m* mod 3

(Hint: Recall any natural number *n* can be written as *n* = 3*q* + *r n* with quotient *q* and remainder *r*. And *n* mod 3 = *kr* : *k* N The set of all natural numbers that have a remainder of *r* when divided by 3.

*{ ∈ }*

The slick way of saying this is: *n m* if and only if they both have the same reminder when divided by 3.

*≡*

Prove that *≡* is an equivalence relation on the set of natural numbers.

# The Pumping Lemma for Regular Languages

1. (10 points) DFAs can’t count to an arbitrary natural number! Use the pumping lemma for regular languages to show that language

{

EQ = *w ∈ {a, b}∗* : *w* = *aibi*

is not regular. Here the number of *a*’s in the prefix of *w* equals the number of *b*’s in the suffix of *w*.

# Context Free Languages

1. (5 points) Consider the CFG G defined by the productions:

*S → aS|Sb|a|b*

Prove by induction that no string in L(G) has *ba* as a sub-string. Hint: To show this do induction on the length of the strings.

1. (5 points) Give simple English language descriptions for the strings generated by the productions following four grammars (*G* = (*V, T, P, S*):
   1. *G*1 *→ S|aS|a*
   2. *G*2 : *S*!*aSa|aa|a*
   3. *G*3 : *S|SaS|a*

|  |  |  |
| --- | --- | --- |
| Question | Points | Score |
| [1](#_bookmark0) | 10 |  |
| [2](#_bookmark1) | 15 |  |
| [3](#_bookmark2) | 10 |  |
| [4](#_bookmark3) | 5 |  |
| [5](#_bookmark4) | 5 |  |
| [6](#_bookmark5) | 5 |  |
| [7](#_bookmark6) | 5 |  |
| [8](#_bookmark7) | 5 |  |
| [9](#_bookmark8) | 5 |  |
| [10](#_bookmark9) | 5 |  |
| [11](#_bookmark10) | 10 |  |
| [12](#_bookmark11) | 10 |  |
| [13](#_bookmark12) | 5 |  |
| [14](#_bookmark13) | 5 |  |
| Total: | 100 |  |