CSE 30151 Theory of Computing Fall 2017: Homework 2

Version 2: Aug. 31, 2017

Instructions

- Unless otherwise specified, all problems from "the book" are from Version 3. When a problem in the International Edition is different from Version 3, the problem will be listed as V3:x.yy/IE:x.zz, where x.zz is the equivalent number. When Version 2 has a different number, it will be listed as V3:x.yy/V2:x.zz. If either IE or V2 does not have a matching number, the problem text will be duplicated.
- You can prepare your solutions however you like (handwriting, LaTeX, etc.), but you must submit them in PDF. You can scan written solutions in the library or using a smartphone (with a scanner app like CamScanner). It is up to you to ensure that submissions are legible.
- Please give every PDF file a unique filename.
  - If you're making a complete submission (all problems), name your PDF file `netid-hw2.pdf`, where `netid` is replaced with your `NetID`.
  - If you're submitting some problems now and other problems later, name your file `netid-hw2-123.pdf`, where `123` is replaced with just the problems you are submitting now.
  - If you use the same filename twice, only the most recent version will be graded.
  - The time of submission is the time the most recent file was uploaded.
- If you use LaTeX and want to draw something like a state diagram, consider using the `tikz` package. A reference document is on the website under "Assignments".
- Submit your PDF file in Sakai. Don't forget to click the Submit (or Resubmit) button!

Practice Problems

These problems are from the book, and most have solutions listed for them. They are listed here for you to practice on as needed and any answers you generate **should not** be submitted. You are free to discuss these with others, but you are not allowed to post solutions to any public forum.

1. 1.1: DFA definition
2. 1.2: DFA formal description
3. 1.4a,b,d: construction of a DFA from intersection of smaller ones
4. 1.5a,b: construction of a DFA from complement of another DFA

Book Exercises

These problems are found in the text book and are to be answered and submitted by each student. You are to solve them yourself. Use of solution manuals from any source or shared solutions is a violation of the ND Honor Code. You are also not allowed to show your solutions to another student.

1. (5 points) 1.3: draw a state diagram
2. (5 points) 1.4c: construction of a DFA from intersection of smaller ones
3. (5 points) 1.5g: construction of a DFA from complement of another DFA
4. (10 points) 1.6c,l: developing DFA state diagrams from languages

Non-book Problems

The following problems are not found in the text book. You are to solve them yourself. Use of any resource you used other than the text book or class notes must be cited. You are also not allowed to show your solutions to another student.

5. (15 points) Consider the DFA defined by $Q = \{A, B, C\}$, $\Sigma = \{0, 1\}$, start state of A, $F = \{A, B\}$, and δ as follows:

	0	1
A	A	B
B	A	C
C	C	C

   (a) Draw the state diagram

   (b) Describe informally the language accepted and write the language accepted as a regular expression

   (c) Prove by induction on the length of the string that your informal description is correct. (Hint: in induction step, consider one at a time all states that the string $w = w_1 w_2 .. w_n$ could be in as the characters are processed one at a time)

6. (10 points) Define formally a DFA that accepts signed binary numbers made up of the following:

   - An optional leading “+” or “–”
   - A string of “0”s and “1”s of length at least 1 digit but at most one leading 0.
   - An optional “.” followed by an arbitrary length string of “0”s and “1”s. (only one “.” allowed in a number). A fraction of all 0s is allowed. A fraction with no digits is allowed.
   - An optional “E” followed a string of “1”s and “0”’s of length at least 1. Thus -1 1.0, 11.101E100, 101., and 0.1 are allowed, but not +-3, 00, 0.1.2, or E10.

   Make sure you explicitly define **all** of the components of a DFA (i.e. all 5 parts). For the transitions, you can show either a state diagram or a transition table, and you may assume that from any state if you do not show all possible characters, the ones not shown are on an arc to some trap state. If you draw a state diagram, you can assume that the list of states, starting state, and final states, are obvious, and you need not explicitly “name” all of them and need not write them out as a set.