CSE 34151 Theory of Computing: Homework 2
DFAs and NFAs

Version 1: Jan. 31, 2018

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 is different, it will be listed as V3:x.yy/V2:x.zz. If either IE or V2 do 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 **legible PDF**. You can scan written solutions on the printer in the back of the classroom, or using a smartphone (with a scanner app like CamScanner). It is up to you to ensure that submissions are legible. REMEMBER THAT IF WE CAN"T READ IT OR SCAN IS CUT OFF, YOU DON"T GET A GRADE FOR IT.
- The problems marked as "TEAM" may be solved in a collaborative fashion with up to 2 other students. In such cases, your submission should have the word "TEAM" at the start of the problem, followed by the names of your collaborators (must be other students in this class). When such problems are graded, the first submission encountered by the grader for the team will be used for a common grade for all identified team members.
- Please give every PDF file a unique filename.
  - If you're making a complete submission (all problems), name your PDF file `netid-hw#.pdf`, where `netid` is replaced with your `NetID` and # is the homework number.
  - If you're submitting some problems now and other problems later, name your file `netid-hw-1-2-3.pdf`, where `1-2-3` is replaced with just the problems you are submitting now. This may be useful for team submissions.
  - 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".
- You may also find the website http://madebyevan.com/fsm/ a useful tool for drawing state diagrams via drop and drag. It will output both .png image files and latex in the tikz format.
- Submit your PDF file in Sakai to the appropriate directory. 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

5. 1.7a,f NFA state diagrams with fixed number of states
6. 1.11 NFAs need only 1 accept state
7. 1.16a NFA to DFA

Book Exercises

These problems are found in the text book and are to be answered and submitted by each student. **If they are not marked as "TEAM," you are to solve them individually.** If they are marked as "TEAM" you may submit the same answer as the others in your team. In any case, 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 not part of your TEAM.

1. (5pt) 1.4g: construction of a DFA from intersection of smaller ones. Note you should define both smaller languages and build the DFAs for them before constructing the intersection.
   Solution:
   Two languages: $L_1 = \{w||w| \text{ is even}\}$ and $L_2 = \{w|w \text{ has an odd number of a's }\}$
   Most direct solution to DFA: first is to build a 2D table with the states from M_1 as row labels and M_2 as column labels. δ for combined machine constructed as discussed in class, with final states all states where the row label was in F_1 and the column label is in F_2.
2. (10pt) Book 1.14a,b Hint for a): assume you have any DFA M, and when you swap the final states you get M'. Assume L(M) is the language accepted by M. Discuss what L(M') is.
   Solution:
   a. Assume L is language accepted by M, M' is M with final states reversed, and L' is language is accepted by M'. Assume w is in L. Thus when given to M, M ends up in an accept state. Consequently, if given to M', it must end up in a non-final state. The reverse is true if w is not in L (i.e. it must be in L'). Then w not accepted by M, and thus accepted by M'.
   b.Consider NFA from Exercise 1.16a. The string a is accepted because of the self loop. If we switch the final state to state 2, a is still accepted. Thus the simple approach to switching final states doesn't prove regexes is closed under complementation.
   The class of languages accepted by NFAs is closed under complementation because NFAs are equivalent to DFAs and DFA are closed.
3. (TEAM - 10pt) (5 points) 1.17 Build an NFA and convert to DFA
   Solution: The formal definition of a NFA is given by:

$$(Q,\ \Sigma,\ \delta,\ q_0,\ F)$$

For 1.17a we have the language:

$(01 \cup 001 \cup 010)^*$ For this NFA, we have:
$Q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$
$\Sigma = \{0, 1\}$
$q_0 = q_1$
$F = \{q_1\}$
$\delta =$

state	0	1	ε
q_1	$\{q_2\}$	$\emptyset$	$\emptyset$
q_2	$\{q_5\}$	$\{q_1, q_3\}$	$\emptyset$
q_3	$\{q_1, q_4\}$	$\emptyset$	$\{q_1\}$
q_4	$\emptyset$	$\emptyset$	$\{q_1\}$
q_5	$\emptyset$	$\{q_1, q_6\}$	$\emptyset$
q_6	$\emptyset$	$\emptyset$	$\{q_1\}$

For 1.17b we are converting the NFA we just designed into a DFA. The formal definition of a DFA is as follows:

$$(Q,\ \Sigma,\ \delta,\ q_0,\ F)$$

For this NFA converted to a DFA, we will have the following:
$Q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$
$\Sigma = \{0, 1\}$
$q_0 = q_1$
$F = \{q_1, q_4\}$
$\delta =$

state	0	1
q_1	q_2	q_6
q_2	q_4	q_3
q_3	q_6	q_1
q_4	q_5	q_6
q_5	q_4	q_2
q_6	q_6	q_6

Non-book Problems

The following problems are not found in the text book. **If they are not marked as "TEAM," you are to solve them individually.** If they are marked as "TEAM" you may submit the same answer as the others in your team. 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.

4. (TEAM - 10pt) Define an NFA 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.

   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.

   Solution:
   $Q = \{s_s, s_f, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$
   $\Sigma = \{0, 1, +, -, ., E\}$
   $q_0 = s_s \in Q$
   $F = \{s_2, s_4, s_5, s_7\}$
   $\delta : Q \times \Sigma \to Q$ given by table

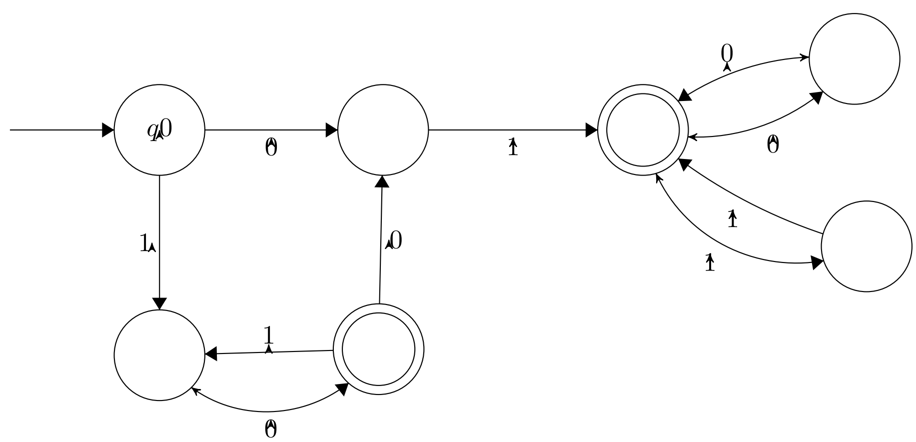
Current State	0	1	+	-	.	E
s_s	s_2	s_2	s_1	s_1	s_f	s_f
s_f	s_f	s_f	s_f	s_f	s_f	s_f
s_1	s_3	s_2	s_f	s_f	s_f	s_f
s_2*	s_2	s_2	s_f	s_f	s_4	s_6
s_3	s_f	s_2	s_f	s_f	s_4	s_6
s_4*	s_5	s_5	s_f	s_f	s_f	s_f
s_5*	s_5	s_5	s_f	s_f	s_f	s_6
s_6	s_7	s_7	s_f	s_f	s_f	s_f
s_7*	s_7	s_7	s_f	s_f	s_f	s_f

   Common Grading comments:

   - States defined: 1
   - Accept states defined: 1‘
   - Alphabet defined:1
   - Start state defined: 1

- Transitions defined: 6
  - +/- 1
  - integer 1
  - decimal 1
  - exponent 1
  - combination 2

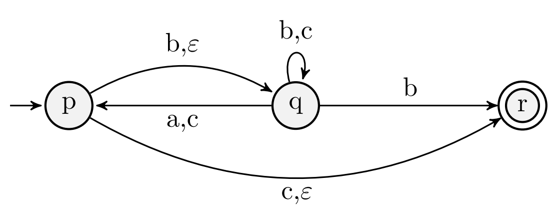
5. (10pt) Define $L_1 = \{w | w = a_0b_0...a_ib_i...a_{n-1}b_{n-1},\ n \geq 1,\ \Sigma = \{0,1\}$ where both bit strings $a_0a_1a_2..a_{n-1}$ and $b_0b_1b_2..b_{n-1}$ are binary numbers with the least significant digit to the *left* rather than right, and where the value represented by $b_0b_1b_2..b_{n-1}$ is one more than $a_0a_1a_2..a_{n-1}\}$. This is essentially a representation of binary incrementation done serially a bit at a time. As an example, the normal binary equivalent of 13_{10} is 1101 so this reversed representation is $a_0a_1a_2a_3 = 1011$. Likewise the normal representation of 14_{10} is 1110 so this reversed representation is $b_0b_1b_2b_3 = 0111$. Since 14 is one more than 13, the string from L_1 that interleaves these two strings is $a_0b_0a_1b_1a_2b_2a_3b_3 = 10011111$. Note also that only the lowest n digits of the result are represented in b. Develop a DFA that accepts L_1. Either a state diagram or transition table is ok. Hint: there is a simple 7 state DFA solution.
   Solution:



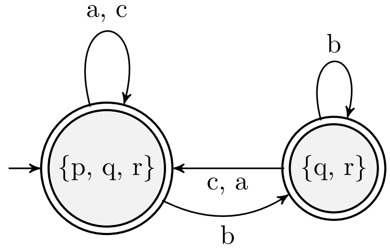
   Scoring notes:

   - +1 answer is an FA (state table or diagram) that can be read
   - +2 for start state and a complete diagram (diagram represents a complete thought)
   - +2 for correct accept states, +1 for any right accept states that did not also accept wrong strings along the minimal path to get to the accept state
   - +5 all correct transitions, +4 near perfect (1-2 mistakes), +3 some missing/incorrect transitions, +2 many mistakes, +1 a couple okay transitions with an understanding that input should be handled in pairs

6. (5pt) Convert the following NFA to a DFA. Note that there are εs. Use the approach of Theorem 1.39 and explicitly show the corresponding E(q) for each q.



   Solution: Note first that E(q) = {q}. Let M' be the DFA corresponding to M. $M' = (Q', \Sigma', \delta', q_0', F'$. Where $Q' = p, q, r, q, r$, $\Sigma' = \Sigma$, δ' is given by the below transition table, $q_0 = p, q, r$, and $F' = Q'$.



Remark	Points
No formal definition	-1
Wrong states	-1
Wrong transitions	-1
Wrong accept states	-1
Wrong epsilon closures	-1