CS Theory (Fall '25) Assigned: Sept 23, 2025

Homework 2

Instructors: *William Pires and Toniann Pitassi* Due: Oct 2, 2025 at 11am

Collaboration: Collaboration with other students in the class homework is allowed. However, you must write up solutions by yourself and understand everything that you hand in. You may also consult other reference materials, but you may not seek out answers from other sources or use AI tools. **You are required to list who you collaborated with on each problem, including any TAs or students you discussed the problems with in office hours. Also list any reference materials consulted other than the lectures and textbook for our class.** See the course webpage for more details.

Formatting: Write the solution to each part of each problem on a separate page. Be sure to correctly indicate which page each problem appears on in GradeScope. Please do not write your name on any page of the submission; we are using anonymous grading in GradeScope.

Grading: There are 50 total possible points, not including the Bonus problem which is optional and for extra credit. For all problems except for the Bonus problem, you have the option of answering "Don't know" on any parts of the question, and you will receive 20 percent of the total marks for those parts.

0 Exercises

Here are some recommended exercises. You should not turn in your solutions for these, and we will not grade them.

(a) Exercises 1.11,1.14b,1.17

For Problem 1.11, the solution is in the book.

(b) Prove that a regular language L is finite if and only if it can be written as a regular expression that does not involve the '*' operation.

1 Problems

1.1 NFA to DFA (10 points)

Consider the following NFA over the alphabet $\{a, b, c\}$.

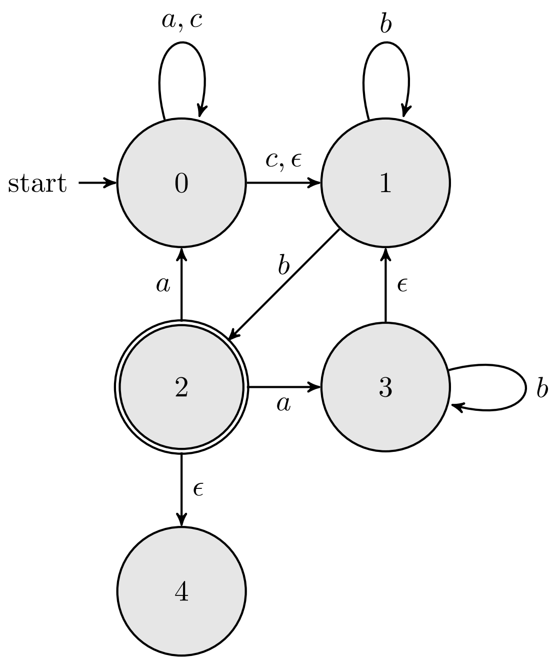

Figure 1: NFA for problem (1).

Convert this NFA into a DFA. Draw the final DFA after removing states that can't be reached from the start state. In the DFA, your states should be named as subsets of the states of the NFA. (I.e. your states should have names like $\varnothing, \{1\}, \{2, 3, 4\}$ etc...).

1.2 Strange Regular Language (8 points)

Let L_1, L_2 be regular languages over alphabet Σ. Define a new language

$$L = \{w \mid w = rst \text{ where } r \in L_1, s \in L_2 \text{ and } t \in L_1 \setminus L_2\}.$$

That is, L consists of all strings $w \in \Sigma^*$ such that w can be written as the concatenation of three strings, r, s, t such that r is in L_1, s is in L_2 and t is in L_1 but not in L_2. Prove that L is regular. (Hint: you may use without proof the four closure properties that we proved in class, but any other property or regular languages that you use, you need to prove.).

1.3 Some weird properties of regular language (20 points)

The following two problems ask you to explore new ways to construct a new regular language L' from an existing regular language, L. For both questions, your proof should follow the same format as the proofs we did in class/book for showing that regular languages are closed under complement, union, concatenation, and star. First give a construction of an NFA for L', using a DFA for L, and then explain/prove correctness of your construction. Be sure to argue both directions of correctness: (i) first argue that every string in L' will be accepted by your NFA, and (ii) secondly argue that every string not in L' will be rejected by your NFA.

(a) Let L be a language over $\Sigma = \{a, b\}$. Define $\text{Swap}(L)$ to be the language consisting of all strings over $w \in \{a, b, c\}$ such that by replacing each c of w by a or b, we can obtain a string in L. Prove that if L is regular, then $\text{Swap}(L)$ is regular.

For example if $L = \{aab, bb\}$. Then $w = cc \in \text{Swap}(L)$ since we can swap the c's to get $bb \in L$. The string $w = cac$ is also in $\text{Swap}(L)$ since we can replace the first c by a, the second c by b and get $aab \in L$. However, the strings $cccc$, bcc, ac aren't in $\text{Swap}(L)$.

(b) Define $L' = \text{Leave-1-Out } (L)$ to be the language consisting of all strings $w \in \{0,1\}^*$ such that by inserting *exactly* one symbol somewhere in w, we can obtain a string in L. Prove that if L is a regular language, then so is $\text{Leave-1-Out}(L)$.

For example, if L consists of all strings that start with 110, then $w = 1111$ is in $\text{Leave-1-Out}(L)$ since we can insert a 0 after the 2rd symbol of w to obtain 11011 which is in L. But $w = 001$ isn't in $\text{Leave-1-Out}(L)$ since we can't insert a symbol in w to make it start with 110.

If $L = \{1\}$, then $w = 1$ is *not* in $\text{Leave-1-Out}(L)$ since we need to insert *exactly* one symbol.

1.4 Regular Expressions (12 points)

Prove of disprove the following for *regular expressions* r, s and t.

(a) $s(rs \cup s)^* = s(rr^*s)^*$

(b) $(s^*r^* \cup rs) = (r \cup s^*)(s \cup r^*)$

2 Bonus Problem

A co-NFA is like an NFA, except it accepts an input $w \in \Sigma^*$ if and only if every possible state it could end up in when reading w (if the NFA does not abort) is an accept state. If a language L is recognized by a co-NFA, is L always regular?