CSCE 355, Spring 2024, Assignment 3
Due February 12, 2024 at 11:30pm

1. For the ϵ-NFA of textbook Exercise 2.5.2,

	ϵ	a	b	c
$\rightarrow p$	$\{q, r\}$	$\emptyset$	$\{q\}$	$\{r\}$
q	$\emptyset$	$\{p\}$	$\{r\}$	$\{p, q\}$
$*r$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

   find an equivalent NFA (without ϵ-moves) using the method explained in class. This is also Method 2 described in the COURSE NOTES (link from the course homepage) in Section 10.4.

2. Do Exercise 2.5.3(a): Design an ϵ-NFA for the following language: the set of all strings consisting of zero or more a's followed by zero or more b's, followed by zero or more c's. Try to use ϵ-transitions to simplify your design.

3. Do Problem 2.3 (pp. 81–82). This illustrates a proof by string induction.

4. (a) Show that every regular language is recognized by an ϵ-NFA where out of each state there is *no more than one* ϵ-transition and *no more than one* non-ϵ-transition (i.e., a transition on a symbol from the alphabet).

   (b) Show that every regular language is recognized by an ϵ-NFA where out of each state there is *exactly one* ϵ-transition and *exactly one* non-ϵ-transition (i.e., a transition on a symbol from the alphabet). (A solution to this part is obviously also a solution to the previous part.)

5. Do Exercise 3.1.1(b,c): Write regexes for the following languages:

   b) The set of strings of 0's and 1's whose tenth symbol from the right end is 1.

   c) The set of strings of 0's and 1's with at most one pair of consecutive 1's.

6. (Optional) Do Exercises 3.1.2(b,c) and 3.1.3(a,b,c)

7. Write a regular expression for the language of strings over $\{a, b, c\}$ where no a appears after any b or c.

8. Do Exercise 3.2.3: Convert the following DFA to a regular expression, using the state-elimination technique of Section 3.2.2.

	0	1
$\rightarrow *p$	s	p
q	p	s
r	r	q
s	q	r

9. Do Exercise 3.2.4(c): Convert the following regex to an ϵ-NFA: $\mathbf{00}(\mathbf{0}+\mathbf{1})^*$.

10. Recall the DFA D we constructed that accepts a binary string iff it has an odd number of 1's:

	0	1
$\rightarrow A$	A	B
$*B$	B	A

    (a) Convert D into an equivalent clean ϵ-NFA using the clean-up procedure in class (add a new start state, a new final state, and some ϵ-transitions).

    (b) Use the state elimination method to convert D to a regular expression. Eliminate state A first, then B.

11. Same exercise as before, except make A the final state (so that D accepts a string iff it has an *even* number of 1's).

12. (Optional) Recall the product DFA P that counts an even number of zeros and an odd number of ones:

	0	1
$\rightarrow EE$	OE	EO
OE	EE	OO
$*EO$	OO	EE
OO	EO	OE

    Use the state elimination method to convert P to a regular expression. (To control the complexity, you may wish to define names for intermediate regexes.)

13. Draw the transition diagram of an ϵ-NFA equivalent to the regex $(a+bc)^*aa$. You may (but are not required to) contract ϵ-transitions provided it is safe to do so.

14. Write a regular expression for the language of strings over $\{a, b, c\}$ where no a appears after any b or c.