1. **(20 points)** [Kozen HW2, Q2 $\equiv$ Sipser problem 1.24]
   Let $s \in \Sigma^*$ be a string. Define $\mathsf{rev}(s)$ to be the *reverse* of s:

$$\begin{array}{rcl} \mathsf{rev}(\epsilon) & = & \epsilon \\ \mathsf{rev}(s\mathsf{c}) & = & \mathsf{c} \cdot \mathsf{rev}(s), \quad \mathsf{c} \in \Sigma,\ s \in \Sigma^* \end{array}$$

   Let $s^{\mathcal{R}} = \mathsf{rev}(s)$.

   Let B be a language. Define $B^{\mathcal{R}} = \{s | s^{\mathcal{R}} \in B\}$. Prove that if B is regular, then $B^{\mathcal{R}}$ is regular as well.

2. **(20 points)** [Sipser problem 1.26]
   Let $\Sigma = \{0,1\} \times \{0,1\}$. Define $\mathsf{val} : \Sigma^* \rightarrow \mathbb{Z} \times \mathbb{Z}$ as in the September 12 notes:

$$\begin{array}{rcl} \mathsf{val}(\epsilon) & = & (0,0) \\ \mathsf{val}(s \cdot (\mathsf{a},\mathsf{b})) & = & (2 * \mathsf{first}(\mathsf{val}(s)) + \mathsf{a}, 2 * \mathsf{second}(\mathsf{val}(s)) + b) \end{array}$$

   Let

$$B \quad = \quad \{s \in \Sigma^* | \mathsf{first}(\mathsf{val}(s)) = 3 * \mathsf{second}(\mathsf{val}(s))\}$$

   Show that B is regular.

3. **(20 points)** Let Σ be an alphabet, and let

$$\begin{array}{rcll} \mathsf{half}(\epsilon) & = & \epsilon & \\ \mathsf{half}(\mathsf{c}) & = & \mathsf{c}, & \mathsf{c} \in \Sigma \\ \mathsf{half}(x\mathsf{c}_1\mathsf{c}_2) & = & \mathsf{half}(x)\mathsf{c}_1, & \mathsf{c}_1 \in \Sigma,\ x \in \Sigma^* \end{array}$$

   Let $B \subseteq \Sigma^*$ be a language. Define

$$B^{\frac{1}{2}} \quad = \quad \{x \in \Sigma^* |\ \exists y \in \Sigma^*.\ (y \in B) \wedge (x = \mathsf{half}(y))\}$$

   Prove that if B is regular, then $B^{\frac{1}{2}}$ is regular as well.

4. **(20 points)** The textbook for CpSc 121 (Rosen: *Discrete Mathematics and Its Applications,* 5^{th} *edition*, p. 12), suggests searching the web for universities in Mexico by looking for pages that contain the word "MEXICO" but not the word "NEW" (to exclude pages about universities in New Mexico). He writes his search as

   (MEXICO *AND* UNIVERSITY) *AND NOT* NEW

   Searches are assumed to be case-insensitive.

   (a) (5 points) What is wrong with Rosen's proposed search criterion? Give an example of text that could appear on a web page for which this search would do something different than its informal description.

   (b) (10 points) Now, write a regular expression that much better matches the informal specification. In particular, it shouldn't have the problem that you identified for Rosen's query in part (a).

   (c) (5 points) Presumably, you haven't solved the natural language understanding problem, so your expression will also fail to meet the informal specification for some web pages. Describe a web page for which your regular expression does the "wrong" thing.

   (d) (2 points, extra credit) Write a program in a language for which there is a compiler on the CS department undergraduate machines that solves the natural language understanding problem. This program should be able to take as input books, journal articles, and other English text and answer questions posed in natural English about their contents.

5. **(20 points)** Let M be a two-input-tape finite automaton as described in the September 14 lecture notes. Define

$$L_\forall(M) \quad = \quad \left\{ s_1 \in \Sigma_1^* \;\middle|\; \forall s_2 \in \Sigma_2^{|s_1|}.\ \mathsf{weave}(s_1, s_2) \in L(M) \right\}$$

We call a machine with this acceptance condition a universally quantified, two-input-tape, finite automaton.

(a) (10 points) Prove that $L_\forall(M)$ is regular.

(b) (10 points) In the September 14 notes, we showed that we could simplify the presentation of a state transition diagram for an existentially quantified, two-input-tape finite automaton by dropping the Σ_2 component of each label and omitting arcs to terminally non-accepting states. The equivalent simplifications for universally quantified, two-input-tape, finite automaton are dropping the Σ_2 component of each label and omitting arcs to terminally accepting states. If a state has no outgoing arc for some input symbol, it is assumed that the machine transitions to a terminally accepting state.

The automaton corresponding to the simplified state-transition diagram is called a $\forall$-automaton (pronounced "for all automaton"). This is a generally accepted technical term (whereas I made up the names "existentially-" and "universally-quantified, two-input-tape, finite automata" just so we could talk about them).

Draw the state transition diagram for a $\forall$-automaton that accepts a string $s \in \{\mathtt{a}, \mathtt{b}\}^*$ iff:

every '`a`' is followed immediately by a '`b`';
and the number of '`a`' symbols in the input is even;
and the number of '`b`' symbols in the input is a multiple of three.

6. **[10 points, Extra Credit]** Here's your chance to provide some feedback on this course. To allow you to do so anonymously, I'm printing this on a separate page. You can write your answers on a separate page as well and staple it to the end of your regular solutions. Don't put your name on the survey page (do put it on the first page of the rest of your solution). The TAs will note if you filled out the survey, and if so give you credit. They won't examine the content of your answers. Instead, they will remove the survey pages and give them to me. I'll look over the results. Thanks.

   (a) **Lectures**

   The pace of lectures is:

   ☐ Too slow

   ☐ Too fast

   ☐ Just about right

   I would like lectures to (check all that apply):

   ☐ Stick closer to the book's presentation of the material.

   ☐ Bring in more connections to related topics – I already understand the material when I've read the book and would like something new.

   ☐ Other (please describe)

   There should be some extra time left over once we've covered finite automata, context free languages, and Turing machines. I'm hoping to have about two weeks (6 lectures) after we've covered the basics. Indicate how you would like this time to be used:

   ☐ Give some applications of each at the end of the corresponding section. For example, show how regular and context-free languages are used to describe the syntax of programming languages and automate the generation of compiler front ends.

   ☐ Cover more standard topics in computability theory: NP-completeness, time and space complexity hierarchies, computing with oracles.

   ☐ Introduce some advanced topics of computability theory: e.g. interactive proofs, zero-knowledge proofs, cryptography, quantum computing.

   ☐ Describe some of my research that's related to this topic: hybrid automata (automata with continuous state), hardware verification, computer aided theorem proving, dynamical systems and computation.

   (b) Homework

   How much time did you spend on this assignment? ____________

   Were the questions (check all that apply):

   ☐ Life changing

   ☐ Interesting, I definitely learned new things.

   ☐ A good check to make sure that I understood the material.

   ☐ The questions were too hard given what we had covered in the readings and lectures.

   ☐ Too tedious.

(c) Daily Questions

How much time do you spend on average to do the Daily Question? ____________

Are the questions (check all that apply):

- ☐ Helpful to understanding the reading.
- ☐ Helpful to understanding the lectures.
- ☐ Helpful warm-ups for the homework.
- ☐ I'd prefer to have one homework assignment a week and no daily questions.
- ☐ Not particularly educational, but they don't take much time.
- ☐ A waste of time.
- ☐ Way too easy.
- ☐ Way too hard.

(d) Textbook

Which text are you reading (check all that apply):

- ☐ Kozen
- ☐ Sipser
- ☐ Other: ____
- ☐ none

How much time do you spend on average to do the reading for a lecture? ____________

(e) Any other comments?