Throughout the semester, the teaching assistants and I will share responsibility for reviewing the homeworks you submit. The teaching assistants will comment on and assign provisional grades to most of the problems on your assignments based on sample solutions and grading criteria I provide. I will exclusively correct one of the problems, paying particular attention to your style and prose. In addition, I will review the solutions graded by the TAs.

We will employ anonymous grading this semester. To facilitate this, I will assign each of you an identification number for CS 361. You will include this number rather than your name on the work you submit. Only after both the TAs and I have reviewed your work and assigned grades will I translate your id number into your name so that your grades can be recorded and the papers returned. I will email your identification number to you before this assignment is due. You should use the same identification number for all of your assignments for this course, so please save this email.

To facilitate the distribution of your solutions to the TAs for grading, each problem's solution should be submitted on SEPARATE SHEETS OF PAPER. Make sure that your CS 361 id number and the number of the problem to which the solution corresponds is clearly visible on each page you submit. When I collect homeworks in class, there will be a separate pile for each problem on the assignment. You will be expected to separate the pages containing your work by problem and put them in the appropriate piles. This will be fun!

I ask that you format your homework solutions using LaTeX and/or other appropriate software. LaTeX is available both on the CS department's Linux workstations (you will probably want to use the pdflatex command) and on the Macs in TCL 217 and 216 (you will probably want to use TeXShop as an editor/previewer (http://pages.uoregon.edu/koch/texshop/index.html)). LaTeX is available for free to download to your personal machine as well. Information on how to do this can be found at https://www.latex-project.org/get/. Finally, there are websites where you can maintain and format documents using LaTeX in the cloud including Overleaf.[1]

A helpful introduction to LaTeX can be found at

http://tobi.oetiker.ch/lshort/lshort.pdf.

You may want to eventually invest in a copy of the reference guide for LaTex — LaTeX: A Document Preparation System (2nd Edition), by Leslie Lamport.

To make it easier for you to prepare your solutions using LaTeX, I will include on the course web site both a link to each assignment in PDF form and a link you can use to download the LaTeX source files used to create the assignment handout. This will enable you to see examples of how to format most of the symbols you are likely to need in your solutions. For each assignment this material will be available in both .tar and .zip archives of a directory containing one file holding the LaTeX source of each problem (with a name like `Prob2Hw1.tex`), and one file that combines all of the separate problems (with a name like `Hw1Template.tex`). There may also be a `figs` subdirectory holding

[1] Clicking on these names or the links in this PDF will take you to the associated web site.

any graphics used within the problem statements. Within the `Hw1Template.tex` file there will be a clearly labeled place where you can type your identification number to ensure that it appears on each page of the output produced.

For finite state machines drawings and other diagrams used as part of the answer to a homework problem, you have several options. You can draw them very neatly and just attach them to the assignment you submit. Better yet, you can create them with:

- a finite automaton simulator (like this one: http://jflap.org), or
- the "Made by Evan" Finite State Machine Designer (http://madebyevan.com/fsm/).

All questions on this assignment count equally and are graded on a 10 point scale as explained in the syllabus.

1. The power set $\mathcal{P}(X)$ of a set X is the set containing all subsets of X. For example, the power set of $\{a, b\}$ is $\{\emptyset, a, b, \{a, b\}\}$. It's well known that if the set X has n members then the power set of X has 2^n members. We'll prove this inductively. Let's start by defining a new binary function $\star$ called *set injection* as follows:

$$A \star c = \{Y \cup \{c\} \mid Y \in A\}$$

In other words, $\star$ injects c into the members of A. This assumes that A is a set of sets. For example

$$\{\emptyset, \{a\}, \{b\}, \{a, b\}\} \star c = \{\{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

(a) Assuming b is not already a member of any element of A, what is the cardinality of $A \star b$? You don't need to prove your answer, but you should at least explain why it is necessary to insist that b is not already a member of any element of A.

(b) Write an inductive definition for power set using the union and power set injection operations. I'll give you the base case:

$$\mathcal{P}(X) = \{\emptyset\} \text{ when } X = \emptyset$$

What's left is to define $\mathcal{P}(X)$ when X is non-empty. Here's a hint: Observe that when given a set X and some element $x \in X$, you can partition $\mathcal{P}(X)$ into two sets – one where all members contain x and one where all members do not contain x.

(c) Use induction on the size of X to prove $|\mathcal{P}(X)| = 2^{|X|}$ for all finite sets X. The key here is to use your definition from (b) to make $\mathcal{P}(X)$ more manageable so you can apply the inductive hypothesis. Part (a) will come in handy too.

2. Prove (by contradiction) that the set of prime numbers is infinite.

3. Show (by example) that the intersection of two uncountable sets can be empty, finite, countably infinite, or uncountably infinite.

4. We have studied several simple operations on languages including union, product, and closure. The study of languages recognized by various automata is full of less simple operations. For example, if L and M are languages over some alphabet Σ^*, we can define:

   - $L_{\frac{1}{3}-\frac{1}{3}} = \{uw \in \Sigma^* \,|\, \text{there exists } v \in \Sigma^* \text{ with } |u| = |v| = |w| \text{ and } uvw \in L\}$
   - $L/M = \{v \in \Sigma^* \,|\, \text{there exists } w \in \Sigma^* \text{ such that } w \in M \text{ and } vw \in L\}$

   Consider the languages

   - $L_{even-parity} = \{w \in \{0,1\}^* \,|\, \text{ the number of 1s in } w \text{ is even}\}$
   - $L_{odd-parity} = \{w \in \{0,1\}^* \,|\, \text{ the number of 1s in } w \text{ is odd}\}$
   - $L_{ba} = \{w\#v \,|\, w \in \{b\}^*, v \in \{a\}^*\}$

   Note that the alphabet for the language L_{ab} is $\{a, b, \#\}$. That is, # does not have any special meaning. It is simply one of three symbols in the language's alphabet (and it is used as a delimiter in the language).

   Provide an alternate description for each of the languages that would result from the following operations. Briefly justify your description.

   (a) $L_{even-parity}/L_{odd-parity}$

   (b) $L_{ba_{\frac{1}{3}-\frac{1}{3}}}$

   To give you a sense of the level of detail you should provide in your answer, suppose we had asked about $L_{even-parity_{\frac{1}{3}-\frac{1}{3}}}$. You could say it describes the set of all strings of binary digits of even length. To justify this, you would then need to show both that only strings of even length are in the set and than all strings of even length are in the set.

   To do this you could first observe that every string in the language must have even length since each such string is the concatenation of two strings u and w of equal length. To show that any string x of binary digits of even length belongs to $L_{even-parity_{\frac{1}{3}-\frac{1}{3}}}$ you could note that we can break any string x of even length up into a suffix and prefix of equal length. That is $x = uw$ with $|u| = |w|$. Now, if uw has even parity then the string $u0\ldots0w$ formed by inserting $|u|$ 0s between u and w must be in $L_{even-parity}$ and therefore $uw = x \in L_{even-parity_{\frac{1}{3}-\frac{1}{3}}}$. On the other hand, if uw has odd parity then the string $u10\ldots0w$ formed by inserting a 1 followed by $|u| - 1$ 0s between u and w must be in $L_{even-parity}$ and therefore $uw = x \in L_{even-parity_{\frac{1}{3}-\frac{1}{3}}}$.

5. Complete problems 1.6 f and i from Sipser. For the second question follow Sipser's convention of numbering input bits from left to right starting at 1.