Assignment 1

CSI 4336

Due September 4, 2018

Submitting your assignment

All written portions of the assignment should be prepared in LaTeX.

Submit this assignment by the due date in two ways: by email (before class) and printed (at the beginning of class). Don't put any code in the printed copy. Proofread your document for style before submitting it.

Send the email to `hamerly@cs.baylor.edu` with the subject "CSI 4336 assignment X" (where X is the assignment number). The email should have one attachment (plain text, .zip, or .tar.gz format) containing:

- the .tex document you wrote named "lastname.tex" (where 'lastname' is your last name),
- a compiled .pdf from the .tex document named "lastname.pdf" (where 'lastname' is your last name),
- any additional files used in your LaTeX document, named "lastname_fig1.pdf" (or similar), and
- all source code used for any programs.

1 Sipser exercises (10 points each, 20 points total)

Do the following exercises from your textbook:

- Exercise 1.6 (parts a, h)
- Exercise 1.17

Use the 'dot' program from GraphViz (`http://www.graphviz.org/`) to render your finite automata. You can make a text file containing a description of a graph, and dot will produce a graphical rendering of that description in PostScript or PDF. You can include the image in your assignment with the `\includegraphics` command (use `\usepackage{graphicx}` in the header of your document).

2 Proofs (10 points each, 20 points total)

Do the following proofs from your textbook:

- Problem 0.13 (note: this is 0.12 in the second edition)
- Problem 1.31

For each proof, include the theorem statement and a proof sketch (high-level outline) before the actual proof. The proof itself should be clear, complete, and concise.

3 Enumerating pairs (10 points)

This problem is to turn in electronically via email, rather than on Kattis. It seems simple, and it is, but take some time to think about it.

Write a C++ or Java program that enumerates the set of all pairs of natural numbers $(1, 2, 3, \ldots)$. For this program you should *pretend* the `int` type supports unbounded magnitude (it can represent 'arbitrarily large' numbers). Thus, there's no point at which an integer overflows. This makes it easier to think about our program enumerating an infinite set. Of course a consequence of pretending that an integer is unbounded is that we'll also pretend that a loop like the following will never terminate (hint: this is important):

```
int i = 1;
while (i > 0)
    i++;
```

To enumerate pairs, just print out lines like the following (just the pairs, nothing else):

```
1 1
1 2
...
```

I will run your program for a while and check that its output is correct.

Put your program in one file and name it 'lastname_pairs.cpp' or 'lastname_pairs.java' (where 'lastname' is your last name).

4 Simulating an NFA (20 points)

Do the problem 'NFA Simulator' on Kattis.