SI 413 Fall 2012: Homework 5

Your name:

Due: Friday, 28 September, before class

Instructions: Review the course honor policy for written homeworks.

This cover sheet must be the front page of what you hand in. Fill out the left column in the table to the right after we go over each problem in class, according to the rubric below.

This rubric is also on the website, in more detail, under "Other Stuff"→"Grading Rubrics".

Make sure all problems are submitted IN ORDER.

- **5**: Solution is completely correct, concisely presented, and neatly written.
- **4**: The solution is mostly correct, but one or two minor details were missed, or the presentation could be more concise.
- **3**: The main idea is correct, but there are some significant mistakes. The presentation is somewhat sloppy or confused.
- **2**: A complete effort was made, but the result is mostly incorrect. There may be some basic misunderstandings of the topic or the problem.
- **1**: The beginning of an attempt was made, but the work is clearly incomplete.
- **0**: Not submitted.

Problem	Self-assessment	Final assessment
1		
2		
3		
4		
5		
6		

Comments or suggestions about this homework:

Comments or suggestions about the course so far:

Citations (other students, websites, . . .):

Use a separate sheet of paper for your answers! Everything should be submitted in one packet, all printed out for me to see.

1 Homographs and Synonyms

Pick a pair of two programming languages that you know, and come up with an example of each of the following:

a) A **homograph** is a code fragment that is the same *syntactically* between the two languages, but has different *semantics* in each..

b) A **synonym** is a code fragment that is the same *semantically* between the two languages, but has different *syntax*.

2 Scanner DFA

Draw the DFA for the scanner that accepts the following tokens (as specified by regular expressions). Label accepting states with the name of the token.

```
FLOAT: [0-9]*"."[0-9]+
  INT: [0-9]+
RANGE: [0-9]+":"[0-9]+
  HEX: 0x[0-9a-f]+
```

(Remember, a `+` means "one or more of the previous thing", and a `*` means "zero or more of the previous thing". Square brackets `[]` indicate a range of characters (or multiple ranges), and anything in quotes `""` means that literal character.)

3 Modified Scanner

Now suppose we want to add a new token `OCT` to the language above, matching the regular expression `0[0-7]+` (that is, a zero followed by one or more digits between 0 and 7).

a) Draw the modified DFA that includes this new token. (Don't just modify your picture from the previous problem, draw a new one.)

b) Compare the work you had to do for (a) to the changes you would have to make to the specification for an automatic scanner generator like `flex`.

4 Modifying a Grammar

The following grammar is ambiguous (in terms of parsing), but it still defines a language:

$S \to exp$
$exp \to exp$ `OPA` exp
$exp \to$ `NUM`

Write a new grammar for *the same language* which is unambiguous and would work well for top-down parsing. (In particular, your new grammar should be LL(1).)

5 Top-Down Parsing

Using your modified grammar from the previous problem, draw the parse tree that would be generated by a top-down parser for the following string:

```
53 + 89 - 103
```

6 PREDICT and FOLLOW

a) Give the PREDICT and FOLLOW sets for your modified grammar from two problems ago.

b) State how these would be used to create a recursive-descent parser.

c) State how these would be used to create a table-driven top-down parser.