ECE 468 — Midterm 1

September 30, 2010

Name: _____

Purdue email: _____

Please sign the following:
I affirm that the answers given on this test are mine and mine alone. I did not receive help from any person or material (other than those explicitly allowed).

X _____

Part	Points	Score
1	8	
2	20	
3	10	
4	22	
5	40	
Total	100	

Part 1: Short answers (8 points)

1) Explain (in *at most* 30 words) why it is useful to split the phases of a compiler into a *front end* and a *back end* (2 points)

-1 point for only defining what front and back end are, without explaining why we might want to split them.

2) Give an example of an English sentence that is *syntactically* correct but not *semantically* correct. (1 point)

-1 point for no/incorrect answer.

3) Briefly explain (it shouldn't take you more than two sentences) why the following grammar is not LL(k) for any k (3 points)

S → E $
E → int
E → (E + E)
E → (E – E)

Many people pointed out that rules 3 & 4 both started with '('. That merely means that it is not LL(1). To get full credit, your answer needed to include some discussion of the recursion: there could be an unlimited number of '('s.

-2 for only explaining why it is not LL(1)

4) Give a grammar that is equivalent to the grammar in problem 3, but *is* LL(1) (2 points)

-2 for grammar that is not equivalent to grammar in (3), or not LL(1)

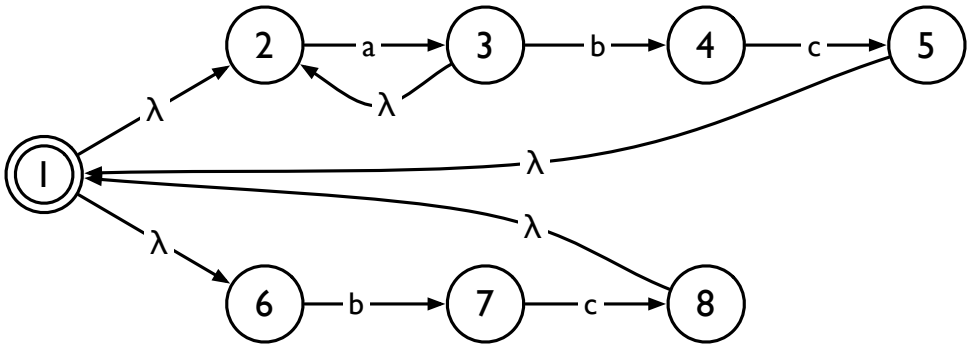
Part 2: Regular expressions, finite automata and scanners (20 points)

1) Describe, in one sentence, the strings captured by the following regular expression (2 points):

$$(ac|ab)^*$$

-2 for incorrect answer

2) Consider the following non-deterministic finite automaton. Fill in the transition table of its deterministic equivalent. (14 points)



I attempted to give as much partial credit as possible for this question. The rubric was as follows:

-2 points for incorrect answer

This left 12 points on the question. I divided those 12 points among the number of states you had in your answer, and then took off points for errors in individual states (at most two errors per state). If you made a mistake in one state, I did not take off additional points if that error carried over to subsequent states.

2) Draw the *reduced* version of the DFA you produced in the previous step (4 points)

This DFA had to be consistent with the DFA you produced in part (1).

-2 points for only partially reduced DFA

Part 3: Grammars (10 points)

Let G be the grammar:

$$\begin{aligned} S &\rightarrow ABC \\ A &\rightarrow xB \mid \lambda \\ B &\rightarrow yC \\ C &\rightarrow Az \end{aligned}$$

Using this grammar, answer the following questions.

1) What are the terminals and non-terminals of this grammar? (1 point)
-1 point for incorrect ansewr

2) Give 4 examples of strings in the language defined by this grammar. (2 points)

-0.5 points per wrong example

3) Draw the parse tree for the following partial derivation (*i.e.*, some of the leaves of your parse tree may be non-terminals) (4 points)

$$S \quad \Rightarrow^* \quad yAzC$$

-4 points for wrong parse tree

4) Did this partial derivation get produced by left-derivation or right-derivation? (1 points)

-1 point for wrong answer

5) Give an example of a partial derivation produced after 2 steps of *right*-derivation (2 points)

-2 points for wrong kind of derivation

I did not take off points for providing more than 2 steps of right-derivation, as long as the intermediate steps were shown.

Part 4: LL parsers (22 points)

Answer the questions in this part using the following grammar:

$$
\begin{array}{lcl}
S & \rightarrow & Ax\$ \\
A & \rightarrow & yz \\
A & \rightarrow & zAA \\
A & \rightarrow & \lambda
\end{array}
$$

1) Define the following sets: (8 points)

-2 points per wrong entry

A lot of people forgot that rule 3 had an ‘A’ followed by an ‘A’. This means that Follow(A) has to include First(A).

2) Give the predict sets for the productions: (8 points)

-2 points per wrong entry

I only took off points if your answers were inconsistent with the answers to question (1).

3) Fill in the LL(1) parse table based on your predict sets (4 points)

-0.5 points per wrong entry

I only took off points if your answers were inconsistent with the answers to question (2). Note that if you got the right answer (according to the key), but if your answers were inconsistent with each other, you lost points (partial credit goes both ways!)

4) Is this an LL(1) grammar? Why or why not? (2 points)

-1 point for answer without explanation

I only took off points if your answer was inconsistent with the parse table you came up with in question (3)

Part 5: LR(0) Parsers (40 points)

Use the following grammar for the next two questions:

$$
\begin{array}{lcl}
S & \rightarrow & Ax\$ \\
A & \rightarrow & yz \\
A & \rightarrow & zAA \\
A & \rightarrow & w
\end{array}
$$

1) Fill in the missing states for the for the following CFSM (22 points) and fill in the missing edge labels (10 points)

-2 points per missing configuration
-1 point per missing edge label

2) List the actions the parser will take when parsing the following string. For shift actions, indicate which state the parser will go to; for reduce actions, indicate which rule is being reduced and which state the parser will go to after reducing. You do not have to show the parse stack or the remaining input—though it may help. Assume the parser accepts when it gets to state 8. (8 points)

zwwx $

There were 8 steps (including 'Accept') in the correct solution, 1 point per step. I awarded points up until the first mistake. I also took off 0.5 points for listing the wrong rule to reduce using.