Name: _____

USC ID:_____

Midterm Exam
CS 410, Fall 2000 [Bono]

October 19, 2000

There are 6 problems on the exam, with 80 points total available. There are 8 pages to the exam, including this one; make sure you have all of them. If you need additional space to write any answers, you may use the backs of pages (just direct us where to look).

Put your name and USC ID number at the top of the exam. Please read over the whole test before beginning. Good luck!

	value	score
Problem 1	10 pts.	
Problem 2	12 pts.	
Problem 3	12 pts.	
Problem 4	15 pts.	
Problem 5	12 pts.	
Problem 6AB	14 pts.	
Problem 6C	5 pts.	
TOTAL	80 pts.	

Problem 1 [10 pts.]

Part A [2]. When using subset construction there can be as many as 2^n states in a

_____ that's equivalent to a _____

with n states.

Part B [2]. `flex` is a tool for automatically building a

_____ from _____.

Part C [2]. Thompson's construction is a method for automatically building a

_____ from a _____.

Part D [2]. When running a SLR(1) parser, when we have a handle for a production whose left-hand-side is A, for what inputs can we reduce by this production?

Part E [2]. The actions of a LL(1) parser correspond to what kind of derivation?

Problem 2 [12 pts.]

Part A [10]. Use subset construction to build a DFA equivalent to the NFA below. Show your work.

	a	**b**	ε
1	{ }	{ }	{ 2, 8 }
2	{ }	{ }	{ 3, 6 }
3	{ 4 }	{ }	{ }
4	{ }	{ }	{ 5 }
5	{ }	{ }	{ 2, 8 }
6	{ }	{ 7 }	{ }
7	{ }	{ }	{5 }
8	{ }	{ }	{ }

Part B [2]. Give a regular expression for the language recognized by the DFA (or NFA) from part A. Note: It may help to draw the state diagram first.

Problem 3 [12 pts.]

Find the FIRST and FOLLOW sets for each of the nonterminals in the grammar below. Show your work. Note: ε denotes the empty string.

```
S → n B v
  | C B v
  | S z

B → C y
  | ε

C → w B B y
  | w S
  | ε
```

Problem 4 [15 pts.]

If we attempt to build an SLR parser with the grammar below for sets of matched parentheses, we'll get a shift-reduce error. Show exactly where and how this error occurs. I.e., show the relevant sets of items in the DFA, and the details and rationale for the shift and for the reduce.

1) S → S S

2) S → (S)

3) S → ( )

Problem 5 [12 pts.]

Consider the following grammar:

```
S →  m S m
   |  B
B →  x B
   |  x
```

Part A. Eliminate left recursion from the grammar, if necessary, or write "Same", if not necessary.

Part B. Left-factor the grammar you gave for part A, or write "Same", if not necessary.

Part C. Show the first and follow sets for all the non-terminals in the grammar you gave in part B.

Part D. Show the LL(1) parse table for the grammar you gave in part B.

Circle and label (A,B,C,D) your answers to each of the parts.

Problem 6 [19 pts.]

Part A [10]. The following grammar is for Boolean expressions with logical operators, **and**, **or**, and **not**. Write semantic rules to evaluate the expression being parsed (put them alongside the productions). You will use a semantic attribute `E.val`, which is the (truth) value of the expression. This value will either be `1` (for true) or `0` (for false). Note: the subscripts are only to distinguish multiple instances of the same nonterminal.

$E \rightarrow E_1$ **or** E_2

| E_1 **and** E_2

| **not** E_1

| **(** E_1 **)**

| **true**

| **false**

Part B [4]. Using the above grammar and semantic rules show the parse tree and the value of `E.val` at each node for the following sentence:

not (true and false)

Problem 6 (cont.)

Part C [5]. Show that the grammar (reprinted here) is ambiguous. Note: this part does not concern semantic rules.

```
E →  E or E
   |  E and E
   |  not E
   |  ( E )
   |  true
   |  false
```