CIS (4|5)61
Spring 2015 Midterm Exam

Your name: _____

Total: _____ of 35 (undergrads) or 45 (grads)

1. [10 points]

Consider the regular expression `a*b?a*b?a*`, where `x?` is a shorthand for `x` | ϵ, as in lex or JFlex.

(a) [3 points] Which of these strings belong to the language denoted by the regular expression? (Circle them.)

- "aaa"
- "abba"
- "" (the empty string)
- "bb"
- "bbaabb"
- "b"

(b) [7 points] Draw an NFA and DFA that accepts strings in the language of `a*b?a*b?a*`.

Graduate students answer both questions 2 and 3. Undergraduates may answer either one.

2. [10 points] Flex says I probably made a mistake in this scanner definition file. (JFlex would also complain about a JFlex version.) What's wrong with it?

```
%%
[a-zA-Z]+  { return IDENT; }
"if"    { return IF;  }
%%
```

3. [10 points] I like to write file processing scripts in flex or JFlex, because they are really fast. I wrote the following flex scanner.

```
%%
a { return 1; }
a*x { return 2; }
%%
```

It's fast enough when I give it a single string containing only 100,000 repetitions of the character 'a' followed by an 'x', ('aaaaaa... aaax'), but super slow when I omit the 'x' ('aaaaaa... aaa'). Why?

(score)

4. [15 points]

Here is a fragment of a grammar for expressions that may include references to variables or calls to functions. The terminal symbols are i (identifier), + (addition), and parentheses (and).

E → **E** + **T**

E → **T**

T → **Call**

T → **Var**

Var → i

Call → i ( )

Show that the grammar is not LR(0), but it is LALR(1). (You can omit boring parts of the CFSM; build just enough to show that LALR(1) lookahead resolves the LR(0) conflict.)

(score)