

**CS 370**  
**Midterm Exam**  
**March 9, 2007**

The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish, but **no electronic aids** are allowed. Please note the following instructions. There will be a ten point deduction for failure to comply with them:

- start each problem on a new sheet of paper
- write your Banner ID number, but not your name, on each sheet of paper you turn in

Also, please note the following:

- show your work whenever appropriate. There can be no partial credit unless you show how you derived your answers
- be succinct. You may lose points for facts that, while true, are not relevant to the question at hand

You have until 2:20 to finish the exam.

1. (20 points) Draw a picture of a FSM that implements “I before E except after C and when said as A as in ‘neighbor’ and ‘weigh’”.

More precisely, the FSM should accept a token made up solely of the letters a-z, such that an e can only be followed by an i when it’s immediately preceded by a c or followed by gh, and the substrings cie and iegh cannot appear.

This is the only spelling rule that should be enforced; nonsensical “words” like *xyzzzy* are perfectly valid while totally reasonable words like *pfeiffer* are not. Here are some valid and some invalid strings:

<b>String</b>	<b>Valid</b>	<b>Note</b>
<i>xyzzzy</i>	Yes	<i>No e or i at all</i>
<i>aiex</i>	Yes	<i>i before e</i>
<i>aeix</i>	No	<i>e before i</i>
<i>ceix</i>	Yes	<i>e before i after c</i>
<i>ciex</i>	No	<i>i before e after c</i>
<i>eight</i>	Yes	<i>iegh combination</i>
<i>wiegh</i>	No	<i>iegh combination</i>

2. (20 points) An email address consists of a user name, followed by an @, followed by a domain name.

A user name consists of an arbitrary string of length at least one, taken from the characters a-z and 0-9.

A domain name consists of one or more domain name components. Each component except the last one is an arbitrary string of length at least one, taken from the characters a-z and 0-9, followed by a period. The last domain name component is a string of length two to four, taken from the characters a-z.

Write a regular expression for email addresses.

*Note: I realize this is a simplified description of email addresses. Trying to do a regular expression for real email addresses would be a bit ambitious for a midterm question!*

3. (40 points) BIND is the most common domain name server on the internet (and quite a few other programs have configuration files with a very similar file format). The BIND configuration file format can be described like this:

The configuration file has a list of zero or more entries.

An entry can be either a key-value pair or a nested complex type.

A key-value pair has a string for the key, an equal sign, a value (which can be either a string enclosed in quotes or an integer), and a semicolon.

A nested complex type has a string for a key, a left brace, a list of zero or more entries, and a right brace.

Spaces, newlines, etc. are not significant.

Here's an example:

```
options {
    foo = "bar";
    bar = -1;
    baz {
        what = "hum";
        hum = "foo" ;
    }
}
darnit = "bar";
```

- (a) (20 points) Write a context free grammar for BIND configuration files. Assume tokens have been defined for all the relevant single-character punctuation (`{`, `;`, `=`, etc), for strings, and for integers.
- (b) (20 points) Give an abstract syntax tree for the example configuration file in the question.
4. (20 points) Give a PDA that will recognize the set of all strings consisting of the letters `a` and `b`, such that the string starts with `a`, ends with `b`, and contains twice as many `b`'s as `a`'s. You can use any representation of the PDA that's clear and unambiguous.