Compilers [Fall 2015]
Test III

NAME: _____

Instructions:

1) This test is 9 pages in length.

2) You have 2 hours to complete and turn in this test.

3) Prose-response questions include a guideline for how much write. Respond in complete English sentences. Essays should be well organized and readable.

4) This test is closed books, notes, papers, friends, neighbors, etc.

5) Use the backs of pages in this test packet for scratch work. If you write more than a final answer in the area next to a question, circle your final answer.

6) Write and sign the following: "I pledge my Honor that I have not cheated, and will not cheat, on this test."

_____

_____

Signed: _____

1. [4 points]
Briefly explain Rice's Theorem, as discussed in class. [1-2 sentences]

2. [7 points]
What is copying garbage collection; how does it work? [1 paragraph]

3. [5 points]
What are IRs, what are other names for IRs, and what kinds of IRs do compilers use in practice? [2-3 sentences]

4. [10 points]
Sketch a proof that the halting problem is undecidable, at the level of detail discussed in class.

5. [14 points]
Show pseudocode to implement an LL(1) parser for the following grammar or a grammar equivalent to the following, or briefly explain why such an implementation is impossible.
S::= E$ E::=E(E) E::=F F::=

6. [15 points]
Optimize the following program for a DISM having only enough code memory to hold 12 instructions. Respond in a sequence of steps, with each step performing exactly one of the code optimizations we discussed in class. Indicate the optimization for each step.

```
mov 1 1
mov 2 1
mov 3 1
add 1 2 1
add 1 1 2
add 3 2 2
add 4 3 3
#L: mov 4 12
ptn 1
add 1 1 2
bgt 4 1 #L
hlt 2
```

7. [20 points]
The source program below is input to dj2dism, which manages memory as discussed in class. Draw a picture to illustrate the contents of DISM data memory as this program executes, *at the point where the most memory cells are in active use* (i.e., the space between the stack and heap is smallest). Be sure to indicate the values of FP, SP, and HP.

```
class C1 extends C2 {C2 x; C1 f(C1 c){nat m; null;}}
class C2 extends Object {nat n; C1 f(C1 c){c.f(c);}}
main { C1 c; C2 cc;
       cc = c = new C1();
       c.f(null);
       new C2().f(c);
       cc.f(c); }
```

8. [25 points] [Essay] [You have two pages to respond.]
Assuming dj2dism is set up as discussed in class, describe how each phase of the compiler needs to change (if at all) to accommodate methods having any natural number of arguments. Provide pseudocode when helpful.

(This page provides extra space for Problem 8.)

Undergraduates stop here. The remaining problem is for graduate students.

9. [12 points] [Essay]

(a) Describe how to convert any RE into an equivalent NFA (as we did in class).

(b) Suppose that, in your algorithm from Part (a), an RE of the form R* was converted into an NFA by recursively building an NFA N for R, and then adding (two) epsilon edges to go between N's initial and accepting states. Show that this alternate conversion would be problematic.