Programming Languages [Fall 2016]
Test II

NAME: ____

Instructions:

1) This test is 7 pages in length.

2) You have 75 minutes to complete and turn in this test.

3) Short-answer questions include a guideline for how many sentences to write. Respond in complete English sentences.

4) This test is closed books, notes, papers, friends, laptops, phones, smartwatches, 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. [10 points]
How did we define "programming language" in class? Based on that definition, how would one go about defining a particular programming language—what exactly needs to be defined? [2-4 sentences]

2. [5 points]
Show how to encode the ML-style expression (e_1; e_2; e_3; e_4) in λ_{UT}.

3. [9 points]
For the remainder of this test, you can assume that all uses of N refer to natural numbers (either zero or the successor of a natural number), so none of your solutions need to contain explicit judgments of the form N nat.

Define inference rules for subtraction. The judgment form is N_1-N_2=N_3. Subtracting a larger number from a smaller number should produce 0; e.g., 0-1=0.

4. [16 points]

For each of the following ML functions, write the function's type; if the function is ill typed, write "no type".

a) fun f x y z = f x y z x y z

b) fun f x y z = f (x (f x y z)) (f x y x)

c) fun f x y z = y (x (f x y x)) (f x y z)

d) fun f x y z = z (x z) y

5. [10 points]

Define a call-by-name operational semantics for λ_{UT}.

6. [20 points]

We want to prove this theorem: If ($N1+N2 \geq N1+N3$) then ($N2 \geq N3$), for all N1, N2, and N3. Define deductive systems for concluding $N1+N2=N3$ and $N1 \geq N2$; then prove the theorem or provide a counterexample.

7. [22 points]

a) Define a function FV that returns the set of free (i.e., used but undeclared) variables in a given λ_{UT} expression.

b) Define a function BV that returns the set of bound (i.e., declared) variables in a given λ_{UT} expression.

c) Define a function V that returns the set of (used or declared) variables in a given λ_{UT} expression.

d) Using your definitions of FV, BV, and V, prove the following theorem or provide a counterexample.

Theorem. For all λ_{UT} expressions e: $|FV(e)| + |BV(e)| = |V(e)|$

For this problem you don't need to define deductive systems for addition or set-size operators; please just use our normal rules and understanding of these judgments.

The next page has additional space for your proof/counterexample.

8. [8 points]
Assuming Church Booleans as discussed in class, define an xor operator in λ_{UT}.

[Undergraduates stop here. The following problem is for graduate students.]

9. [12 points]

Recall the list-splitting function `f` from Test I, which evaluates to ([1,2,3],[4,5,6]) on input [1,2,3,4,5,6] and evaluates to ([1],[2,3]) on input [1,2,3]. Implement this function in ML such that its running time is linear in the size of the input (i.e., O(n)). Avoid library functions, e.g., folds.