Programming Languages [Fall 2018]
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 and essay questions include a guideline for how much to write. Respond in complete English sentences and avoid using bulleted and itemized lists.

4) For full credit on ML-response questions, implementations must avoid the @ operator and run in linear (O(n)) time.

5) This test is closed books, notes, laptops, phones, smartwatches, friends, neighbors, etc.

6) 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.

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

____

____

Signed: ____

1. [7 points]
What is a programming language? How is one defined? [2-3 sentences]

2. [18 points] [Essay]
$\forall P \in \{\lambda_{ST}, \lambda_{UT}$, diML, STERLING, L$\}$: explain (at the level discussed in class) why P is or is not Turing complete. L is the PL from the theory assignments.

3. [5 points]
State the standard substitution lemma and technique for proving it.

4. [20 points]
a) Encode a right-associative short-circuit NOR operator for Church booleans. Short-circuit means that the operator only evaluates operands when necessary (like && in C). *Don't* use abbreviations; e.g., don't write "true" as an abbreviation for a λ_{UT} expression.
e_1 NOR $e_2 \equiv$

b) Now trace evaluation of (false NOR false) NOR (false NOR false) using left-to-right, normal-order evaluation. Underline redexes and *do* use abbreviations when convenient.

5. [20 points]

For this problem, N always refers to a natural number, so you never need to write N nat.

a) Define inference rules for deriving sum and inequality relationships between natural numbers. The judgment forms are $N_1+N_2=N_3$ and $N_1 \neq N_2$.

b) Prove that different addends produce different sums.

Theorem. $\forall N_1, N_2, N_3, N_4, N_5$: $((N_1+N_2=N_3 \wedge N_1+N_4=N_5 \wedge N_2 \neq N_4) \Rightarrow N_3 \neq N_5)$

6. [20 points]
Define higher-order abstract syntax for λ_{ST} having base type nat.

7. [10 points]

Recall that porder L will “reorder L, placing it in ‘pong’ order, that is, bouncing between left and right elements. For example, porder [1,2,3,4,5] returns [1,5,2,4,3] and porder [1.1,2.2] returns [1.1,2.2].” Let’s generalize porder to porder’, which takes 3 (curried) arguments: a list L, a boolean flag b indicating whether to start on the left or right side, and an integer s indicating how many elements to skip at a time. For example,

- porder’ [1,2,3,4,5] true 1 returns [1,5,2,4,3]
- porder’ [1,2,3,4,5] true 2 returns [1,5,3]
- porder’ [1,2,3,4,5] false 1 returns [5,1,4,2,3]
- porder’ [1,2,3,4,5,6,7,8] false 2 returns [8,1,6,3]
- porder' L b s, where s<1, returns nil

Implement porder’ subject to the constraints of Assignment I (e.g., no library-fun. calls).

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

8. [12 points]
State, and prove any 3 cases of, the Weakening Lemma for λ_{ST} having base type nat.