Programming Languages [Spring 2019]
Test I

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 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 be simplified and efficient.

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. [4 points]
Defining a programming language requires defining two high-level things. What are they? [1-2 sentences]

2. [4 points]
In class, how did we distinguish the ML keywords `fun` and `fn`? [1-2 sentences]

3. [4 points]
What is the ML value restriction? [1-2 sentences]

4. [8 points]
Function c takes a list of booleans and computes their conjunction.
a) Implement c according to the constraints of Assignment 1 (i.e., with recursion).

b) Implement c according to the constraints of Assignment 2 (i.e., with map/fold).

5. [8 points]
Consider the following let-expression E.

```
let
  val c = 4
  val c = c+2
  fun a(b) = c + 3 + b
  fun q(c) = a(c)
  val c = 7
in
  q(c+1)
end
```

a) To what value does E evaluate, assuming lexical scoping? Show enough work to convince me that you are not guessing.

b) To what value does E evaluate, assuming dynamic scoping? Show enough work to convince me that you are not guessing.

6. [22 points]
All parts of this problem use the representation of polynomials as int list lists, as on Assignments 1-2.

a) Implement constplus according to the constraints of Assignment 2 such that the function always adds a new int list to the given int list list. (Hint: the type of constplus here is `int -> int list list -> int list list`.)

b) Re-implement constplus, this time according to the constraints of Assignment 2 such that the function only adds a new int list to the given int list list when necessary.

c) Complete the following to implement constmult according to the constraints of Assignment 2.

fun constmult c p = map

d) Complete the following to implement constmult according to the constraints of Assignment 2.

fun constmult c p = fold

e) Complete the following to implement constmult according to the constraints of Assignment 2.

fun constmult c = map

7. [35 points]

a) What were the inference rules we defined in class for the N nat and N_1-N_2=N_3 judgment forms?

b) A second, equally valid, set of inference rules for the N_1-N_2=N_3 judgment form is:

$\boxed{N_1\text{-}N_2\text{=}N_3}$

$$\frac{}{N\text{-}N\text{=}Z}\ 2Z \qquad\qquad \frac{N_1\text{'-}N_2\text{=}N_3\text{'}}{S(N_1\text{'})\text{-}N_2\text{=}S(N_3\text{'})}\ 2S$$

Define a third, equally valid, set of inference rules for the N_1-N_2=N_3 judgment form.

c) Using Rules 2Z and 2S given above, and the standard rules for deriving N Nat, prove:

Theorem. $\forall\ N_1, N_2, N_3$: (N_1-N_2=N_3 $\Rightarrow$ N_1-N_3=N_2)

The following page is blank, in case you need additional space. As on Assignment 2, assume that N always refers to a valid natural number. Hint: You may want to prove the following 2 lemmas. $\forall$ N: (N-Z=N) and $\forall\ N_1, N_2, N_3$: (N_1-N_2=N_3 $\Rightarrow$ $S(N_1)$-$S(N_2)$=N_3)

$\boxed{N_1\text{-}N_2\text{=}N_3}$

$$\frac{}{N\text{-}N\text{=}Z}\ 2Z \qquad \frac{N_1\text{'-}N_2\text{=}N_3\text{'}}{S(N_1\text{'})\text{-}N_2\text{=}S(N_3\text{'})}\ 2S$$

8. [15 points]
For each of the following ML expressions, write the expression's type, or, if the expression is ill typed, write "no type".

a) `fun i(j,k)=k::i`

b) `fun i(j,k)=k::j`

c) `fun i j k = j = k`

d) `fun i j k = j(k,i)`

e) `fun i j k = i j k`

f) `fun i j k = k j`

[Undergraduates stop here. The remaining problems are for graduate students.]

[6 points]

g) `fun i j k = k k j j`

h) `fun i j k = k j j j j`