**Name:**

# Midterm 2

## CMSC 430
Introduction to Compilers
Spring 2015

April 23, 2015

## Instructions

**This exam contains 7 pages, including this one. Make sure you have all the pages. Write your name on the top of this page before starting the exam.**

Write your answers on the exam sheets. If you finish at least 15 minutes early, bring your exam to the front when you are finished; otherwise, wait until the end of the exam to turn it in. Please be as quiet as possible.

If you have a question, raise your hand. If you feel an exam question assumes something that is not written, write it down on your exam sheet. Barring some unforeseen error on the exam, however, you shouldn't need to do this at all, so be careful when making assumptions.

| Question | Score | Max |
|----------|-------|-----|
| 1        |       | 15  |
| 2        |       | 15  |
| 3        |       | 25  |
| Total    |       | 55  |

**Question 1. Short Answer (15 points).**

**a. (5 points)** Briefly explain the difference between type *inference* and type *checking.*

**b. (5 points)** In a few sentences, describe how a method invocation $o.m(a_1, \ldots, a_n)$ ("invoke method $m$ on object $o$ with arguments $a_1, \ldots, a_n$") could be carried out by a virtual machine.

**c. (5 points)** Briefly explain the purpose of a *closure*.

## Question 2. Program transformations (15 points).

**a. (15 points)** Apply defunctionalization to this program:

```
type bt =
 | Leaf
 | Node of int * bt * bt

let rec sumk b k =
  match b with
  | Leaf → k 0
  | Node (i, b1, b2) →
      sumk b1 (fun sb1 → sumk b2 (fun sb2 → k (i + sb1 + sb2)))
let sum b =
  sumk b (fun sb → sb)
```

**Question 3. Type Systems (25 points).**

**a. (8 points)** Assume that $int < float$. Write down every type $t$ such that $t \leq float \to int \to float$, following standard subtyping rules.

**b. (2 points)** Assume that $int < float$. Write down every type $t$ such that $t \leq int\ ref \to float\ ref$, following standard subtyping rules.

**c. (5 points)** Recall the simply typed lambda calculus:

$$e \quad ::= \quad n \mid x \mid \lambda x{:}t.e \mid e\ e$$
$$t \quad ::= \quad int \mid t \to t$$
$$A \quad ::= \quad \emptyset \mid x{:}t, A$$

$$\text{INT} \quad \frac{}{A \vdash n : int}$$

$$\text{VAR} \quad \frac{}{A \vdash x : A(x)}$$

$$\text{LAM} \quad \frac{x{:}t, A \vdash e : t'}{A \vdash (\lambda x{:}t.e) : t \to t'}$$

$$\text{APP} \quad \frac{A \vdash e_1 : t \to t' \qquad A \vdash e_2 : t}{A \vdash e_1\ e_2 : t'}$$

Draw a derivation that the following type judgment holds, where $A = +{:}\,int \to int \to int$. (You can draw the derivation upward from the judgment, and you can write $i$ instead of $int$ to save time):

$$A \vdash ((\lambda f{:}\,int \to int.f\ 1)\ (\lambda x{:}\,int. +\ 1\ x)) : int$$

**d. (10 points)** Perform type inference on the following program by listing the types that OCaml will infer for the blanks:

```
let rec bumble (f : ___) (xs : ___) : ___ =
  match xs with
  | [] → []
  | x :: xs →
      ((x ("fred" ^ (f 0))) + 1) :: (bumble f xs)
```