Student Number: _____

Family Name: _____

Given Names: _____

Signature: _____

THE UNIVERSITY OF NEW SOUTH WALES

Sample Exam

Session 2 2011

COMP3161/COMP9161

Concepts of Programming Languages

- Time allowed: **2 hours**
- Total number of questions: **4**
- Answer **all** questions
- The questions **are** of equal value
- Answer *each* question in a **separate** answer booklet
- This paper may **not** be retained by the candidate
- **Answers must be written in ink**, with the exception of graphs
- Drawing instruments or rules may be used
- There is a 3% penalty if you do not fill in your student number and name correctly

Question 1 [25 Marks]

Consider the following inductive definition of evaluation rules for a restricted form of boolean expressions.

Boolean expressions:

$$\frac{}{\texttt{true}\ \textbf{bool}} \qquad \frac{}{\texttt{false}\ \textbf{bool}}$$

$$\frac{b_1\ \textbf{bool} \quad b_2\ \textbf{bool}}{\texttt{and}(b_1, b_2)\ \textbf{bool}} \qquad \frac{b\ \textbf{bool}}{\texttt{not}(b)\ \textbf{bool}}$$

Evaluation rules:

$$\frac{}{\texttt{and}(\texttt{true}, b) \mapsto b} \qquad \frac{}{\texttt{and}(\texttt{false}, b) \mapsto \texttt{false}}$$

$$\frac{}{\texttt{not}(\texttt{false}) \mapsto \texttt{true}} \qquad \frac{}{\texttt{not}(\texttt{true}) \mapsto \texttt{false}}$$

$$\frac{b \mapsto b'}{\texttt{not}(b) \mapsto \texttt{not}(b')} \qquad \frac{b_1 \mapsto b_1'}{\texttt{and}(b_1, b_2) \mapsto \texttt{and}(b_1', b_2)}$$

A) **[7 marks]**

Give the derivation of the evaluation for the following expression:

- `and(not(false), and(true, not(true)))`

B) **[7 marks]**

Are the rules unambiguous? If so, briefly explain why. If not, give an example expression for which the set of rules allow more than a single derivation.

C) **[11 marks]**

The rules listed above give a small step semantics. List the inference rules which specify an equivalent big step semantics.

Question 2 [25 Marks]

A) **[10 marks]**

In the lecture, we discussed the E-machine as an example of an abstract machine which handles value bindings explicitly by maintaining a value environment. One of the possible return values of the E-machine are function closures.

i) What is a function closure?

ii) Give an example of an expression whose evaluation in the E-machine requires the creation of a closure.

B) **[15 marks]**

We discussed two distinct methods to handle exceptions: the first method required that, when an exception is thrown, the evaluation unrolls the stack until the matching catch-expression is found. The second method made it possible to directly jump to the matching catch-expression. Describe the second method:

i) What are the components of the state of the abstract machine?

ii) How does the state of the machine change when a `catch`-expression is evaluated?

iii) How does the state of the machine change when a `raise`-expression is evaluated?

For (ii) and (iii), you do not have to give the exact transition rule — it is sufficient to describe how the state is affected.

Question 3 [25 Marks]

A) **[6 marks]**

For each of the following three pairs of type expressions determine whether the pair has a most general unifier? If so, please provide it.

i) $(a, b) \rightarrow (b, a)$ and $(int, c) \rightarrow (c, c)$

ii) $a \rightarrow (a, a)$ and $(b, b) \rightarrow b$

iii) `int` $\rightarrow$ `int` and `float` $\rightarrow$ `int`

B) **[9 marks]**

Give the principal type of the following (polymorphic) MinHs expressions:

i)
```
Inr(Inl(True))
```

ii)
```
letfun f (x) is fst (snd (x));
```

iii)
```
letfun g (x) is
    case x of Inl(a) -> a
              Inr(b) -> b
    end
end
```

C) **[10 marks]**

Consider the following MinHs types:

- $\forall a.\forall b.(a * b \rightarrow c) \rightarrow (a \rightarrow b \rightarrow c)$
- $\forall a.\forall b.(a \rightarrow b) \rightarrow (b \rightarrow a)$
- $\forall a.\forall b.\forall c.(a \rightarrow b) \rightarrow (b \rightarrow c) \rightarrow (c \rightarrow a)$
- $\forall a.() \rightarrow a$
- $\forall a.a \rightarrow ()$

For which of these types exist terminating MinHs functions?

Question 4 [25 Marks]

A) **[10 marks]**

Progress and preservation are central concepts for strongly typed languages.

i) Give the definition of progress and of preservation in the context of a strongly typed language.

ii) The presence of partial functions can be problematic with respect to progress. Describe how they can be handled in a strongly typed language such that both progress and preservation still hold.

B) **[5 marks]**

Give an example each for a type constructor which is covariant and a type constructor which is contravariant in at least one of its argument positions.

C) **[10 marks]**

Java's array type is covariant. Why is this problematic?