The University of New South Wales

COMP3161/9161
Concepts of Programming Languages

SAMPLE

Midsession Exam

Semester 2, 2018

Time Allowed: 50 Minutes

Total Number of Questions: 6

Answer **all** questions.

The questions are **not** of equal value.

You are permitted **one** hand-written, single-sided A4 sheet of notes.

Only write your answers on the provided booklets.

Answers must be written in ink, with the exception of diagrams.

Drawing instruments or rules may be used.

Excessively verbose answers may lose marks.

There is a 3% penalty if your name and student number are not filled in correctly.

Question 1 (4 marks)

Given the following expression e, written in the arithmetic expression language we defined in the lectures:

$$\begin{array}{l}\texttt{let } x = 5 \texttt{ in}\\ \quad \texttt{let } x = 10 + x \texttt{ in}\\ \quad\quad \texttt{let } y = 5 + x \texttt{ in}\\ \quad\quad\quad x + \texttt{let } x = 15 \texttt{ in } x \times y \texttt{ end}\\ \quad\quad \texttt{end}\\ \quad \texttt{end}\\ \texttt{end}\end{array}$$

Find a term that is α equivalent where each variable name is used only in one unique binding position (i.e. each variable name only occurs once on the left hand side of a `let`-binding).

Question 2 (6 marks)

In the lecture we discussed the role of the lexer, the parser, and the (static) semantic analyser. For each of these components, give an example of a program error that can be detected by that component. (The error may be in terms of C, Haskell or the language of arithmetic expressions with `let`-bindings discussed in the lecture).

Question 3 (6 marks)

What is the difference between concrete and abstract syntax of a programming language? (keep your answer brief — it may be easiest to describe the difference if you use the possible concrete and abstract syntax of an actual language construct as an example)

Question 4 (4 marks)

What is the β-normal form of this λ-term? What about the $\beta\eta$-normal form? Write down the chain of reductions.

$$(\lambda n.\ \lambda f.\ \lambda x.\ f\ (n\ x\ x))\ (\lambda f.\ \lambda x.\ x)$$

Question 5 (7 marks)

Given the following big-step semantics for arithmetic expressions:

$$\frac{}{(\texttt{Num}\ n) \Downarrow n}(1) \quad \frac{e_1 \Downarrow v_1 \qquad e_2 \Downarrow v_2}{(\texttt{Plus}\ e_1\ e_2) \Downarrow (v_1 + v_2)}(2) \quad \frac{e_1 \Downarrow v_1 \qquad e_2 \Downarrow v_2}{(\texttt{Times}\ e_1\ e_2) \Downarrow (v_1 \times v_2)}(3)$$

$$\frac{e_1 \Downarrow v_1 \qquad e_2[x := (\texttt{Num}\ v_1)] \Downarrow v_2}{(\texttt{Let}\ e_1\ (x.\ e_2)) \Downarrow v_2}(4)$$

Draw a derivation tree to describe the evaluation of the term:

$$(\texttt{Plus}\ (\texttt{Num}\ 5)\ (\texttt{Let}\ (\texttt{Num}\ 7)\ (x.\ (\texttt{Plus}\ x\ (\texttt{Num}\ 1)))))$$

Question 6 (13 marks)

Here are some rules to inductively define the judgements **Nat** and **Unnat**:

$$\frac{}{0\ \mathbf{Nat}}N_1 \quad \frac{n\ \mathbf{Nat}}{(\mathsf{S}\ n)\ \mathbf{Nat}}N_2$$

$$\frac{}{0\ \mathbf{Unnat}}U_1 \quad \frac{}{(\mathsf{S}\ 0)\ \mathbf{Unnat}}U_2 \quad \frac{n\ \mathbf{Unnat}}{(\mathsf{S}\ (\mathsf{S}\ n)\ \mathbf{Unnat}}U_3$$

Using rule induction, show that both definitions are equivalent, that is, for all x, x **Nat** is derivable if and only if x **Unnat** is derivable. Clearly state which cases you have to consider, and what the induction hypothesis is for each case. Annotate derivations with the name of the rule you used. One direction should be comparatively straightforward. For the other direction, you may find it helpful to prove a lemma.