UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE THREE

LANGUAGE SEMANTICS AND IMPLEMENTATION

Tuesday 5 June 2001

14:00 to 15:30

Examiners:

K. Turner (External)

G. Plotkin (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. Axioms and rules for LC evaluation include the following:

$$\langle !l, s\rangle \Downarrow \langle n, s\rangle \text{ if } s(l) = n$$

$$\langle \mathbf{skip}, s\rangle \Downarrow \langle \mathbf{skip}, s\rangle$$

$$\frac{\langle E, s\rangle \Downarrow \langle n, s'\rangle}{\langle l := E, s\rangle \Downarrow \langle \mathbf{skip}, s'[l \mapsto n]\rangle}$$

(a) Give the LC evaluation rules for sequential composition $C_1; C_2$ and for while loops **while** B **do** C. *[5 marks]*

(b) Assuming the standard LC rules for integers and for booleans, give full derivations for the LC evaluation relation for the following configurations where state s is $\{l \mapsto 3, l' \mapsto 4\}$: *[11 marks]*

i. $\langle l := !l + !l', s\rangle$

ii. $\langle l := !l'; l' := !l, s\rangle$

iii. $\langle \mathbf{while}\ !l > 2\ \mathbf{do}\ l' := !l' + 1; l := !l - 1, s\rangle$

(c) Some languages have simultaneous assignment, $l_1, l_2 := E_1, E_2$, where at the same time l_1 is updated to the value of E_1 and l_2 to the value of E_2. *[9 marks]*

i. Give the evaluation rule for this new construct.

ii. Assuming s is the state $\{l \mapsto 3, l' \mapsto 4\}$, give derivations for the following configurations.

A. $\langle l, l' := !l', !l, s\rangle$

B. $\langle l, l := !l', !l, s\rangle$

2. (a) Define semantic equivalence for closed LPC terms. [*2 marks*]

(b) Prove the following call-by-name LPC equivalence:

$$\textbf{if } M_1 \textbf{ then } (\textbf{if } M_2 \textbf{ then } M_1' \textbf{ else } M_2') \textbf{ else } (\textbf{if } M_2 \textbf{ then } M_3' \textbf{ else } M_4')$$
$$\cong$$
$$\textbf{if } M_2 \textbf{ then } (\textbf{if } M_1 \textbf{ then } M_1' \textbf{ else } M_3') \textbf{ else } (\textbf{if } M_1 \textbf{ then } M_2' \textbf{ else } M_4')$$

You may assume that each M_i and M_i' has type bool. You will find that there are four subcases. Prove one of them in detail and state the others. [*6 marks*]

(c) Consider adding the following *iteration* construct to LPC:

$$\textbf{for } i = M_1, M_2 \textbf{ do } M_3$$

It is evaluated by first evaluating M_1 and then M_2; if the results are integers n_1 and n_2 such that $n_2 > n_1$ then M_2 is evaluated $(n_2 + 1) - n_1$ times.

(i) Give a formula for $fv(\textbf{for } i = M_1, M_2 \textbf{ do } M_3)$ in terms of $fv(M_1)$, $fv(M_2)$ and $fv(M_3)$. Use it to calculate $fv(\mathit{Example})$ where

$$\begin{array}{lll} \mathit{Example} \stackrel{\text{def}}{=} & \textbf{for} & i = 1, m \\ & \textbf{do} & \textbf{for } j = 1, n \\ & & \textbf{do } Aij := Bji \end{array}$$

(here i, j, m, n, A, B are variables). [*4 marks*]

(ii) Give a typing rule for the iteration construct and use it to show that $\Gamma \vdash \mathit{Example} : \mathit{com}$ for a certain choice of Γ. You can assume typings of LPC expressions which do not involve the iteration construct. [*5 marks*]

(iii) Give a structural operational semantics for transitions (not evaluation) for the iteration construct.
[**Hint:** There are four rules] [*8 marks*]

3. (a) Featherweight Java's expression syntax is:

$$e ::= x \mid e.f \mid e.m(\overline{e}) \mid \textbf{new}\ C(\overline{e}) \mid (C)e$$

Give the computation rules for attributes and methods. You may assume that the functions *fields*(C) and *mbody*(m, C) have already been defined. [*4 marks*]

(b) Suppose we extend Featherweight Java by adding a location class Loc such that **new** Loc() creates an object l where l is a new integer location. Loc has an attribute *val* which looks up the integer stored in l and a method *update*(m) which changes the integer stored in l to m. To accommodate these we add new expressions to the syntax:

$$e ::= l \mid n$$

Computation rules should now operate on configurations $\langle e, s\rangle$ where s is a state. Give rules for $\langle l.val, s\rangle$ and $\langle l.update(n), s\rangle$. Does this introduction of state create a problem with using computation rules? Would structural operational semantics like that given for LC and LPC help? Explain your reasoning. [*8 marks*]

(c) The following does not depend on previous parts of the question.

i. Explain rule induction. [*5 marks*]

ii. Assume the two axioms, 0 and 11, and the following two rules, where w is a non-empty word in $\{0, 1\}^*$:

$$\frac{w}{w00} \qquad \frac{w}{w11}$$

Prove by rule induction that any word generated by these axioms and rules contains an odd number of 0s and an even number of 1s. [*8 marks*]