Indian Institute of Technology, Delhi
Department of Computer Science and Engineering

CS 232 N Programming Languages Minor I
February 1, 2000 16:00-17:00 Maximum Marks: 20

Answer the questions on this question paper itself, in the spaces provided, using a black or blue PEN (no pencils or red pens, please). Write your NAME, ENTRY NUMBER and GROUP in the spaces provided as the top of the sheet, otherwise your paper will not be corrected. Please write neatly, striking off any rough work.
The objective of this test is to *specify* a <u>boolean</u> <u>calculator</u>, which you will later implement in SML as part of a programming assignment.
The boolean expressions are built up from *propositional variables* drawn from some set $\mathcal{X}$, the constants **true** and **false**, the unary negation connective $\neg$, and the binary connectives $\wedge$ ("and") and $\vee$ ("or").

Q1 [3 marks] **Syntax.** Define inductively the set BExp consisting of the *completely parenthesized* syntactic boolean expressions.

BExp is defined as the smallest set such that:

- $\underline{\mathbf{true}} \in \mathsf{BExp}$;
- $\underline{\mathbf{false}} \in \mathsf{BExp}$;
- For each $x \in \mathcal{X}$: $x \in \mathsf{BExp}$;
- If $be \in \mathsf{BExp}$, then $\underline{(}\underline{\neg} be\underline{)} \in \mathsf{BExp}$;
- If $be_1 \in \mathsf{BExp}$ and $be_2 \in \mathsf{BExp}$, then $\underline{(}be_1\underline{\wedge}be_2\underline{)} \in \mathsf{BExp}$;
- If $be_1 \in \mathsf{BExp}$ and $be_2 \in \mathsf{BExp}$, then $\underline{(}be_1\underline{\vee}be_2\underline{)} \in \mathsf{BExp}$.

Q2 [5 marks] **Denotational Semantics.** Specify a definitional interpreter for the language BExp with respect to the domain $Bool = \{\mathbf{tt}, \mathbf{ff}\}$ with the usual functions $\sim$ (negation), $\sqcap$ (conjunction) and $\sqcup$ (disjunction).

Syntactic Domain(s):

$$x \in \mathcal{X} \qquad be \in \mathsf{BExp}$$

Semantic Domain(s):

$$b \in Bool \qquad \rho \in SBEnv \overset{\triangle}{=} \mathcal{X} \to Bool$$

Semantic Function(s):

$$Truth : \mathsf{BExp} \to SBEnv \to Bool$$

where

$$\begin{array}{lcl}
Truth[\![\underline{\mathbf{true}}]\!]\rho & = & \mathbf{tt} \\
Truth[\![\underline{\mathbf{false}}]\!]\rho & = & \mathbf{ff} \\
Truth[\![x]\!]\rho & = & \rho(x) \\
Truth[\![\underline{(}\underline{\neg}be\underline{)}]\!]\rho & = & \sim Truth[\![be]\!]\rho \\
Truth[\![\underline{(}be_1\underline{\wedge}be_2\underline{)}]\!]\rho & = & Truth[\![be_1]\!]\rho \sqcap Truth[\![be_2]\!]\rho \\
Truth[\![\underline{(}be_1\underline{\vee}be_2\underline{)}]\!]\rho & = & Truth[\![be_1]\!]\rho \sqcup Truth[\![be_2]\!]\rho
\end{array}$$

Q3 [9 marks] **Big-step SOS.** Define the set of syntactic environments for boolean expressions as

$$\gamma \in BEnv \overset{\triangle}{=} \mathcal{X} \rightarrow \{\underline{\mathbf{true}}, \underline{\mathbf{false}}\}$$

Now specify the elementary syntactic boolean operations using *tables* and then specify a *Big-step* Structural Operational Semantics for BExp by inductively defining a relation

$$\hookrightarrow \quad \subset \quad BEnv \times \mathsf{BExp} \times \{\underline{\mathbf{true}}, \underline{\mathbf{false}}\}$$

NOT

a	$NOT(a)$
$\underline{\mathbf{false}}$	$\underline{\mathbf{true}}$
$\underline{\mathbf{true}}$	$\underline{\mathbf{false}}$

OR

a	b	$OR(a,b)$
$\underline{\mathbf{false}}$	$\underline{\mathbf{false}}$	$\underline{\mathbf{false}}$
$\underline{\mathbf{false}}$	$\underline{\mathbf{true}}$	$\underline{\mathbf{true}}$
$\underline{\mathbf{true}}$	$\underline{\mathbf{false}}$	$\underline{\mathbf{true}}$
$\underline{\mathbf{true}}$	$\underline{\mathbf{true}}$	$\underline{\mathbf{true}}$

AND

a	b	$AND(a,b)$
$\underline{\mathbf{false}}$	$\underline{\mathbf{false}}$	$\underline{\mathbf{false}}$
$\underline{\mathbf{false}}$	$\underline{\mathbf{true}}$	$\underline{\mathbf{false}}$
$\underline{\mathbf{true}}$	$\underline{\mathbf{false}}$	$\underline{\mathbf{false}}$
$\underline{\mathbf{true}}$	$\underline{\mathbf{true}}$	$\underline{\mathbf{true}}$

$$\gamma \vdash \underline{\mathbf{true}} \hookrightarrow \underline{\mathbf{true}}$$

$$\gamma \vdash \underline{\mathbf{false}} \hookrightarrow \underline{\mathbf{false}}$$

$$\gamma \vdash x \hookrightarrow \gamma(x)$$

$$\frac{\gamma \vdash be \hookrightarrow tv}{\gamma \vdash \underline{(}\underline{\neg}be\underline{)} \hookrightarrow NOT(tv)}$$

$$\frac{\gamma \vdash be_1 \hookrightarrow tv_1 \qquad \gamma \vdash be_2 \hookrightarrow tv_2}{\gamma \vdash \underline{(}be_1\underline{\wedge}be_2\underline{)} \hookrightarrow AND(tv_1, tv_2)}$$

$$\frac{\gamma \vdash be_1 \hookrightarrow tv_1 \qquad \gamma \vdash be_2 \hookrightarrow tv_2}{\gamma \vdash \underline{(}be_1\underline{\vee}be_2\underline{)} \hookrightarrow OR(tv_1, tv_2)}$$

Q4 [3 Marks] **Facts.**

State (without proof) the proposition that your operational semantics is *correct* with respect to your denotational semantics.

For all $be \in \mathsf{BExp}$, $\gamma \in BEnv$ and $tv \in \{\underline{\mathbf{true}}, \underline{\mathbf{false}}\}$:

$$\text{If} \quad \gamma \vdash be \hookrightarrow tv \quad \text{then}$$

for $\rho \in SBEnv$ such that for all $x \in \mathcal{X} : \rho(x) = Truth[\![\gamma(x)]\!]\rho'$:

$$Truth[\![be]\!]\rho = Truth[\![tv]\!]\rho''$$

(where ρ', ρ'' are arbitrary semantic boolean environments).