《程序设计语言的形式语义》期中考试试卷

考试时间：2026年10月28日10:10-12:00；考试方式：闭卷

姓名：_____ 学号：_____

题号	1	2	3	4	总分
满分	20	25	30	25	100
得分					

In this exam, we consider a programming language which extends simply-typed lambda calculus (STLC) with primitive Booleans, nondeterministic choice, and recursion.

Figure 1 shows the syntax of the programming language. Figure 2 gives all the evaluation rules. Figure 3 gives the typing rules. Contexts map variables to types; bound variables may be renamed to fresh variables before extending a context.

$$
\begin{array}{rrcl}
\text{(Types)} & \tau & ::= & \mathsf{Bool} \mid \tau \to \tau \\
\text{(Terms)} & M, N & ::= & x \mid \lambda x : \tau.\, M \mid M\, N \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2 \mid M \oplus N \mid \mathsf{fix}\, M \\
\text{(Values)} & v & ::= & \mathsf{true} \mid \mathsf{false} \mid \lambda x : \tau.\, M \\
\text{(Contexts)} & \Gamma & ::= & \bullet \mid \Gamma, x : \tau
\end{array}
$$

Figure 1: Syntax of the programming language

$$\frac{M \longrightarrow M'}{M\, N \longrightarrow M'\, N}\ (\textsc{app1}) \qquad \frac{N \longrightarrow N'}{v\, N \longrightarrow v\, N'}\ (\textsc{app2})$$

$$\frac{}{(\lambda x : \tau.M)\, v \longrightarrow M[v/x]}\ (\textsc{beta}) \qquad \frac{M \longrightarrow M'}{\mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2 \longrightarrow \mathsf{if}\ M'\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2}\ (\textsc{if-step})$$

$$\frac{}{\mathsf{if\ true\ then}\ M\ \mathsf{else}\ N \longrightarrow M}\ (\textsc{if-true}) \qquad \frac{}{\mathsf{if\ false\ then}\ M\ \mathsf{else}\ N \longrightarrow N}\ (\textsc{if-false})$$

$$\frac{}{M \oplus N \longrightarrow M}\ (\textsc{choose-l}) \qquad \frac{}{M \oplus N \longrightarrow N}\ (\textsc{choose-r})$$

$$\frac{M \longrightarrow M'}{\mathsf{fix}\, M \longrightarrow \mathsf{fix}\, M'}\ (\textsc{fix-step}) \qquad \frac{}{\mathsf{fix}\, (\lambda x : \tau.M) \longrightarrow M[\mathsf{fix}\, (\lambda x : \tau.M)/x]}\ (\textsc{fix})$$

Figure 2: Evaluation rules

Safety. The following hold for this language.

Theorem (Preservation)**.** *For any M, N and τ, if $\bullet \vdash M : \tau$ and $M \longrightarrow N$, then $\bullet \vdash N : \tau$.*

Theorem (Progress)**.** *For any M and τ, if $\bullet \vdash M : \tau$, then $M \in \mathit{Values}$ or $\exists M'.M \longrightarrow M'$.*

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\ (\textsc{t-var}) \qquad \frac{}{\Gamma \vdash \mathsf{true} : \mathsf{Bool}}\ (\textsc{t-true}) \qquad \frac{}{\Gamma \vdash \mathsf{false} : \mathsf{Bool}}\ (\textsc{t-false})$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma . M : \sigma \to \tau}\ (\textsc{t-abs}) \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M\, N : \tau}\ (\textsc{t-app})$$

$$\frac{\Gamma \vdash M : \mathsf{Bool} \quad \Gamma \vdash N_1 : \tau \quad \Gamma \vdash N_2 : \tau}{\Gamma \vdash \mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2 : \tau}\ (\textsc{t-if})$$

$$\frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash M \oplus N : \tau}\ (\textsc{t-choice}) \qquad \frac{\Gamma \vdash M : \tau \to \tau}{\Gamma \vdash \mathsf{fix}\, M : \tau}\ (\textsc{t-fix})$$

Figure 3: Typing rules

1. (本题共20分；你的得分：_____)

   Free variables and substitution.

   For arbitrary terms M, N and a variable x, define the set of free variables $\mathsf{fv}(M)$ and the capture-avoiding substitution $M[N/x]$.

2. (本题共25分；你的得分：_____)

Evaluation and typing.

Consider the term

$$M \stackrel{\text{def}}{=} (\lambda f : \mathsf{Bool} \to \mathsf{Bool}.\mathsf{true})\big(((\lambda g : \mathsf{Bool} \to \mathsf{Bool}.g)(\lambda x : \mathsf{Bool}.x)) \oplus ((\lambda y : \mathsf{Bool}.y)\mathsf{false})\big).$$

(a) Give *all* the complete evaluation sequences of M. **[15 points]**

(b) Determine whether there exists a type τ such that $\bullet \vdash M : \tau$. If so, give a complete typing derivation; otherwise, explain why no such derivation exists. **[10 points]**

3. (本题共30分；你的得分：_____)

Preservation and progress after changing rules.

Each variant below changes the original language *independently*. Except for the explicitly stated changes, all syntax, typing rules, and evaluation rules remain unchanged.

Variant 1. Remove the CHOOSE-R rule.

Variant 2. Remove both the CHOOSE-L and CHOOSE-R rules.

Variant 3. Remove both the CHOOSE-L and CHOOSE-R rules, and extend the value syntax by

$$(\mathsf{Values})\ v \ ::= \ \cdots \mid M \oplus N$$

This extended syntax is used in BETA and APP2.

Variant 4. Replace T-CHOICE by

$$\frac{\Gamma \vdash M : \tau}{\Gamma \vdash M \oplus N : \tau}\ (\text{T-CHOICE-LEFT})$$

Variant 5. Replace T-FIX by

$$\frac{\Gamma \vdash M : \sigma \to \tau}{\Gamma \vdash \mathsf{fix}\, M : \tau}\ (\text{T-FIX-GENERAL})$$

Fill the table with *holds* or *fails*. **[10 points]**

For each failed property, give a counterexample and explain why it violates that property. No explanation is required for a *holds* entry. The counterexamples and explanations carry **20 points in total**.

Variant	Preservation	Progress
1		
2		
3		
4		
5		

(This is a blank page for you to write down your answers.)

4. (本题共25分；你的得分：_____)

Nondeterministic choice and termination.

Multistep evaluation ($k \geq 0$) is defined by

$$\begin{aligned} M \longrightarrow^0 M' \quad &\text{iff} \quad M = M', \\ M \longrightarrow^{k+1} M' \quad &\text{iff} \quad \exists M''.\ M \longrightarrow M'' \wedge M'' \longrightarrow^k M', \\ M \longrightarrow^* M' \quad &\text{iff} \quad \exists k \geq 0.\ M \longrightarrow^k M'. \end{aligned}$$

May termination is defined by

$$M \Downarrow_{\text{may}} \quad \text{iff} \quad \exists v.\ M \longrightarrow^* v.$$

Must termination is defined inductively by the following rules:

$$\frac{M \in \mathsf{Values}}{M \Downarrow_{\text{must}}} \ (\textsc{must-value})$$

$$\frac{M \longrightarrow M' \qquad \forall M''.\ (M \longrightarrow M'') \Rightarrow (M'' \Downarrow_{\text{must}})}{M \Downarrow_{\text{must}}} \ (\textsc{must-step})$$

For parts (a) and (b), let R, S satisfy

$$x : \mathsf{Bool} \to \mathsf{Bool}, b : \mathsf{Bool} \vdash R : \mathsf{Bool}, \qquad x : \mathsf{Bool} \to \mathsf{Bool}, b : \mathsf{Bool} \vdash S : \mathsf{Bool}.$$

Define

$$\begin{aligned} P_{R,S} &\stackrel{\text{def}}{=} \mathsf{fix}\,(\lambda x : \mathsf{Bool} \to \mathsf{Bool}.\lambda b : \mathsf{Bool}.(R \oplus S)), \\ Q_{R,S} &\stackrel{\text{def}}{=} \mathsf{fix}\,\big((\lambda x : \mathsf{Bool} \to \mathsf{Bool}.\lambda b : \mathsf{Bool}.R) \oplus (\lambda x : \mathsf{Bool} \to \mathsf{Bool}.\lambda b : \mathsf{Bool}.S)\big). \end{aligned}$$

(a) For Boolean constants r, s, define **[10 points]**

$$\begin{aligned} R_r &\stackrel{\text{def}}{=} \mathsf{if}\ b\ \mathsf{then}\ (x\,r)\ \mathsf{else}\ \mathsf{true}, \\ S_s &\stackrel{\text{def}}{=} x\,s. \end{aligned}$$

Find one pair of Boolean constants r and s such that

$$P_{R_r,S_s}\ \mathsf{true} \Downarrow_{\text{may}} \quad \wedge \quad \neg\big(Q_{R_r,S_s}\ \mathsf{true} \Downarrow_{\text{may}}\big).$$

Using the evaluation rules and the definition of may termination given above, formally prove that $P_{R_r,S_s}\ \mathsf{true} \Downarrow_{\text{may}}$ and $\neg\big(Q_{R_r,S_s}\ \mathsf{true} \Downarrow_{\text{may}}\big)$ hold for your chosen r, s.

(This is a blank page for you to write down your answers.)

(b) For Boolean constants r, s, define **[10 points]**

$$R_r \stackrel{\text{def}}{=} \mathsf{if}\ b\ \mathsf{then}\ (x\, r)\ \mathsf{else}\ \mathsf{true},$$
$$S_s \stackrel{\text{def}}{=} \mathsf{if}\ b\ \mathsf{then}\ \mathsf{true}\ \mathsf{else}\ (x\, s).$$

Find one pair of Boolean constants r and s such that

$$Q_{R_r,S_s}\ \mathsf{true} \Downarrow_{\text{must}} \quad \wedge \quad \neg\big(P_{R_r,S_s}\ \mathsf{true} \Downarrow_{\text{must}}\big).$$

Using the evaluation rules and the definition of must termination given above, formally prove that $Q_{R_r,S_s}\ \mathsf{true} \Downarrow_{\text{must}}$ and $\neg\big(P_{R_r,S_s}\ \mathsf{true} \Downarrow_{\text{must}}\big)$ hold for your chosen r, s.

(c) Determine whether the following claim holds. **[5 points]**

> For every closed term M (i.e., M has no free variables), if $M \Downarrow_{\text{may}}$ and there is no infinite evaluation sequence starting from M, then $M \Downarrow_{\text{must}}$.

If it holds, just answer yes. Otherwise, give a counterexample and formally justify why it satisfies both premises but not the conclusion, using the evaluation rules and the definitions of may and must termination.