Midterm Exam

Name: _____

Signature: _____

Instructions

Print your name on the exam. Sign your name below your printed name.

You have 75 minutes to complete the exam.

If you need more paper to write your answers, raise your hand and I will pass out more blank sheets.

You may only use this exam, the textbook, and one page of notes as reference.

You may not use electronic devices of any kind, or communicate with other students in any way.

If you finish early, you may hand in the exam and leave early.

There are two parts to the exam: A and B. There is a reference in the back of the exam for each part which you may find useful.

Parts A and B are each worth 50 points. Each part has two sub-problems worth 25 points each. There are 100 total possible points on this exam.

There is a dedicated answer box for each question. Please write your question in this answer box. If you need more space, you may continue your answer onto a blank sheet that is clearly labeled for that answer.

Part A (50 Points)

Consider the language of typed boolean expressions extended with variables, sums, and products. See Reference A for the syntax of terms e, the one-step relation $e \longrightarrow e$, the type system $\Gamma \vdash e : \tau$, and definitions for *stuck*, *unsafe*, and *safe* terms.

Consider the following terms:

$$e_1 \coloneqq \texttt{if F then } \langle \texttt{T}, \texttt{F} \rangle \texttt{ else T} \qquad e_2 \coloneqq \langle \texttt{inl T}, \texttt{F} \rangle \qquad e_3 \coloneqq \texttt{projl (if T then F else T)}$$

$$e_4 \coloneqq \texttt{case}(\texttt{inr } \langle \texttt{T}, \texttt{F} \rangle)\{x.\ x\}\{y.\ \texttt{projl } y\} \qquad e_5 \coloneqq \texttt{inl } \langle \texttt{projl T}, \texttt{projr F} \rangle$$

Question A.1 (25 Points)

For each of the above terms e_1–e_5, do *one* of the following:

- Identify another term e' and draw a derivation tree for $e_i \longrightarrow e'$.
- Identify that e_i is a *value*.
- Identify that e_i is *stuck* (i.e., it cannot take a step, and is not a value).

(Write your answer in the answer box on the next page.)

Question A.2 (25 Points)

For each of the above terms e_1–e_5, do *one* of the following:

- Identify a type τ and draw a derivation tree for $[] \vdash e_i : \tau$
- Identify that e_i is untypeable and *unsafe*
- Identify that e_i is untypeable and *safe*

(Write your answer in the answer box on the next page.)

Answer A.1

$e_1 \coloneqq \texttt{if F then } \langle \texttt{T}, \texttt{F} \rangle \texttt{ else T}$ $e_2 \coloneqq \langle \texttt{inl T}, \texttt{F} \rangle$ $e_3 \coloneqq \texttt{projl (if T then F else T)}$

$e_4 \coloneqq \texttt{case}(\texttt{inr } \langle \texttt{T}, \texttt{F} \rangle)\{x.\ x\}\{y.\ \texttt{projl } y\}$ $e_5 \coloneqq \texttt{inl } \langle \texttt{projl T}, \texttt{projr F} \rangle$

Solution

1. $$\frac{}{\texttt{if F then } \langle \texttt{T}, \texttt{F} \rangle \texttt{ else T} \longrightarrow \texttt{T}}\ \textsc{If-F}$$

2. Value

3. $$\frac{\dfrac{}{\texttt{if T then F else T} \longrightarrow \texttt{F}}\ \textsc{If-T}}{\texttt{projl (if T then F else T)} \longrightarrow \texttt{projl F}}\ \textsc{Projl-C}$$

4. $$\frac{}{\texttt{case}(\texttt{inr } \langle \texttt{T}, \texttt{F} \rangle)\{x.\ x\}\{y.\ \texttt{projl } y\} \longrightarrow \texttt{projl } \langle \texttt{T}, \texttt{F} \rangle}\ \textsc{Case-Inr}$$

5. Stuck

Answer A.2

$e_1 \coloneqq \texttt{if F then } \langle \texttt{T}, \texttt{F} \rangle \texttt{ else T}$ $e_2 \coloneqq \langle \texttt{inl T}, \texttt{F} \rangle$ $e_3 \coloneqq \texttt{projl (if T then F else T)}$

$e_4 \coloneqq \texttt{case}(\texttt{inr } \langle \texttt{T}, \texttt{F} \rangle)\{x.\ x\}\{y.\ \texttt{projl } y\}$ $e_5 \coloneqq \texttt{inl } \langle \texttt{projl T}, \texttt{projr F} \rangle$

Solution

1. Untypeable and Safe

2. $$\dfrac{\dfrac{\dfrac{}{[] \vdash \texttt{T} : \texttt{bool}}\textsc{Bool-I-T}}{[] \vdash \texttt{inl T} : \texttt{bool} + \tau}\textsc{Sum-I} \quad \dfrac{}{[] \vdash \texttt{F} : \texttt{bool}}\textsc{Bool-I-F}}{[] \vdash \langle \texttt{inl T}, \texttt{F} \rangle : (\texttt{bool} + \tau) \times \texttt{bool}}$$

3. Untypeable and Unsafe

4. $$\dfrac{\dfrac{\dfrac{}{[] \vdash \texttt{T} : \texttt{bool}} \quad \dfrac{}{[] \vdash \texttt{F} : \texttt{bool}}}{[] \vdash \langle \texttt{T}, \texttt{F} \rangle : \texttt{bool} \times \texttt{bool}}}{[] \vdash \texttt{inr } \langle \texttt{T}, \texttt{F} \rangle : \texttt{bool} + (\texttt{bool} \times \texttt{bool})} \quad \dfrac{x : \texttt{bool} \in [], x : \texttt{bool}}{[], x : \texttt{bool} \vdash x : \texttt{bool}} \quad \dfrac{\dfrac{y : (\texttt{bool} \times \texttt{bool}) \in [], y : (\texttt{bool} \times \texttt{bool})}{[], y : (\texttt{bool} \times \texttt{bool}) \vdash y : \texttt{bool} \times \texttt{bool}}}{[], y : (\texttt{bool} \times \texttt{bool}) \vdash \texttt{projl } y : \texttt{bool}}$$
$$[] \vdash \texttt{case}(\texttt{inr } \langle \texttt{T}, \texttt{F} \rangle)\{x.\ x\}\{y.\ \texttt{projl } y\} : \texttt{bool}$$

5. Untypeable and Unsafe

Part B (50 Points)

Consider the untyped lambda calculus, where β-reduction may occur in any context (i.e., full-β). See Reference B for the syntax of lambda terms e, the β rule, the free variables metafunction FV, the substitution metafunction $[x \mapsto e]$, and the standard encoding of boolean operations true, false and cond.

Question B.1 (25 Points)

Consider the following terms:

$$e_1 \coloneqq x\ x \qquad e_2 \coloneqq (\lambda x.\ x)\ (\lambda y.\ y)\ (\lambda z.\ z) \qquad e_3 \coloneqq (\lambda x.\ \lambda x.\ \lambda x.\ x\ x)\ y\ z$$

$$e_4 \coloneqq (\lambda x.\ x\ x)\ ((\lambda y.\ y\ y)\ z) \qquad e_5 \coloneqq y\ (\lambda y.\ y)\ z$$

For each of the above terms e_1–e_5, do *all* of the following:

- In the answer box on the next page: circle each free variable.
- In the answer box on the next page: for each bound variable, draw an arrow from the variable to the λ that binds it.
- Apply the β rule as many times as possible, and show a reduction sequence to a term which cannot be reduced any more. Underline each sub-expression that you apply the β rule to in each step. If the initial term e_i cannot be reduced, write "Already Reduced". If you use α-conversion, you must be explicit about the renaming. (None of the solutions require α-conversion, although you are free to use it if you want.)

(Write your answer in the answer box on the next page.)

Question B.2 (25 Points)

Consider an alternative encoding for booleans in the untyped lambda calculus:

$$\begin{aligned}\text{true} &\coloneqq \lambda f.\ f\ (\lambda a.\ \lambda b.\ a)\\ \text{false} &\coloneqq \lambda g.\ g\ (\lambda c.\ \lambda d.\ d)\\ \text{cond} &\coloneqq \lambda w.\ \lambda r.\ \lambda s.\ w\ (\lambda h.\ h\ r\ s)\end{aligned}$$

This is also a *valid* encoding of booleans. Partially prove this by showing:

$$\text{cond true}\ x\ y \longrightarrow^* x$$

You must show a full reduction sequence, where each step applies only one instance of the β rule.

(Write your answer in the answer box on the next page.)

Answer B.1

$e_1 \coloneqq x\ x$ $\quad$ $e_2 \coloneqq (\lambda x.\ x)\ (\lambda y.\ y)\ (\lambda z.\ z)$ $\quad$ $e_3 \coloneqq (\lambda x.\ \lambda x.\ \lambda x.\ x\ x)\ y\ z$

$e_4 \coloneqq (\lambda x.\ x\ x)\ ((\lambda y.\ y\ y)\ z)$ $\quad$ $e_5 \coloneqq y\ (\lambda y.\ y)\ z$

Solution

1. Already Reduced
2. $$\begin{aligned}&\underline{(\lambda x.\ x)\ (\lambda y.\ y)}\ (\lambda z.\ z)\\ &\longrightarrow \underline{(\lambda y.\ y)\ (\lambda z.\ z)}\\ &\longrightarrow \lambda z.\ z\end{aligned}$$
3. $$\begin{aligned}&\underline{(\lambda x.\ \lambda x.\ \lambda x.\ x\ x)\ y\ z}\\ &\longrightarrow \underline{(\lambda x.\ \lambda x.\ x\ x)\ z}\\ &\longrightarrow \lambda x.\ x\ x\end{aligned}$$
4. $$\begin{aligned}&(\lambda x.\ x\ x)\ \underline{((\lambda y.\ y\ y)\ z)}\\ &\longrightarrow (\lambda x.\ x\ x)\ (z\ z)\\ &\longrightarrow (z\ z)\ (z\ z)\end{aligned}$$
5. Already Reduced

Answer B.2

$$\begin{aligned}\text{true} &:= \lambda f.\ f\ (\lambda a.\ \lambda b.\ a)\\ \text{false} &:= \lambda g.\ g\ (\lambda c.\ \lambda d.\ d)\\ \text{cond} &:= \lambda w.\ \lambda r.\ \lambda s.\ w\ (\lambda h.\ h\ r\ s)\end{aligned}$$

Solution

$$\begin{aligned}&\text{cond true } x\ y\\ &= \underline{(\lambda w.\ \lambda r.\ \lambda s.\ w\ (\lambda h.\ h\ r\ s))\ (\lambda f.\ f\ (\lambda a.\ \lambda b.\ a))}\ x\ y\\ &\longrightarrow \underline{(\lambda r.\ \lambda s.\ (\lambda f.\ f\ (\lambda a.\ \lambda b.\ a))\ (\lambda h.\ h\ r\ s))\ x\ y}\\ &\longrightarrow \underline{(\lambda s.\ (\lambda f.\ f\ (\lambda a.\ \lambda b.\ a))\ (\lambda h.\ h\ x\ s))\ y}\\ &\longrightarrow \underline{(\lambda f.\ f\ (\lambda a.\ \lambda b.\ a))\ (\lambda h.\ h\ x\ y)}\\ &\longrightarrow \underline{(\lambda h.\ h\ x\ y)\ (\lambda a.\ \lambda b.\ a)}\\ &\longrightarrow \underline{(\lambda a.\ \lambda b.\ a)\ x\ y}\\ &\longrightarrow \underline{(\lambda b.\ x)\ y}\\ &\longrightarrow x\end{aligned}$$

Reference A

The language of boolean expressions extended with variables, and sum and product types:

$$\tau ::= \texttt{bool} \mid \tau + \tau \mid \tau \times \tau$$
$$\Gamma ::= [] \mid \Gamma, x : \tau$$

$$e ::= x \mid \texttt{T} \mid \texttt{F} \mid \texttt{if } e \texttt{ then } e \texttt{ else } e \mid \texttt{inl } e \mid \texttt{inr } e \mid \texttt{case}(e)\{x.\ e\}\{x.\ e\} \mid \langle e, e\rangle \mid \texttt{projl } e \mid \texttt{projr } e$$

$$v ::= \texttt{T} \mid \texttt{F} \mid \texttt{inl } v \mid \texttt{inr } v \mid \langle v, v\rangle$$

$\boxed{e \longrightarrow e}$

$$\frac{}{\texttt{if T then } e_2 \texttt{ else } e_3 \longrightarrow e_2}\ \textsc{If-T} \qquad \frac{}{\texttt{if F then } e_2 \texttt{ else } e_3 \longrightarrow e_3}\ \textsc{If-F}$$

$$\frac{e_1 \longrightarrow e_1'}{\texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 \longrightarrow \texttt{if } e_1' \texttt{ then } e_2 \texttt{ else } e_3}\ \textsc{If-C}$$

$$\frac{e \longrightarrow e'}{\texttt{inl } e \longrightarrow \texttt{inl } e'}\ \textsc{Inl-C} \qquad \frac{e \longrightarrow e'}{\texttt{inr } e \longrightarrow \texttt{inr } e'}\ \textsc{Inr-C}$$

$$\frac{}{\texttt{case}(\texttt{inl}(v))\{x_2.\ e_2\}\{x_3.\ e_3\} \longrightarrow [x_2 \mapsto v]e_2}\ \textsc{Case-Inl} \qquad \frac{}{\texttt{case}(\texttt{inr}(v))\{x_2.\ e_2\}\{x_3.\ e_3\} \longrightarrow [x_3 \mapsto v]e_3}\ \textsc{Case-Inr}$$

$$\frac{e_1 \longrightarrow e_1'}{\texttt{case}(e_1)\{x_2.\ e_2\}\{x_3.\ e_3\} \longrightarrow \texttt{case}(e_1')\{x_2.\ e_2\}\{x_3.\ e_3\}}\ \textsc{Case-C} \qquad \frac{e_1 \longrightarrow e_1'}{\langle e_1, e_2\rangle \longrightarrow \langle e_1', e_2\rangle}\ \textsc{Pair-C1}$$

$$\frac{e \longrightarrow e'}{\langle v, e\rangle \longrightarrow \langle v, e'\rangle}\ \textsc{Pair-C2} \qquad \frac{}{\texttt{projl } \langle v_1, v_2\rangle \longrightarrow v_1}\ \textsc{Projl} \qquad \frac{e \longrightarrow e'}{\texttt{projl } e \longrightarrow \texttt{projl } e'}\ \textsc{Projl-C}$$

$$\frac{}{\texttt{projr } \langle v_1, v_2\rangle \longrightarrow v_2}\ \textsc{Projr} \qquad \frac{e \longrightarrow e'}{\texttt{projr } e \longrightarrow \texttt{projr } e'}\ \textsc{Projr-C}$$

$\boxed{\Gamma \vdash e : \tau}$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\ \textsc{Var} \qquad \frac{}{\Gamma \vdash \texttt{T} : \texttt{bool}}\ \textsc{Bool-I-T} \qquad \frac{}{\Gamma \vdash \texttt{F} : \texttt{bool}}\ \textsc{Bool-I-F} \qquad \frac{\Gamma \vdash e_1 : \texttt{bool} \quad \Gamma \vdash e_2 : \tau \quad \Gamma \vdash e_3 : \tau}{\Gamma \vdash \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 : \tau}\ \textsc{If-E}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \langle e_1, e_2\rangle : \tau_1 \times \tau_2}\ \textsc{Prod-I} \qquad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \texttt{projl } e : \tau_1}\ \textsc{Prod-E1} \qquad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \texttt{projr } e : \tau_2}\ \textsc{Prod-E2} \qquad \frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \texttt{inl } e : \tau_1 + \tau_2}\ \textsc{Sum-I1}$$

$$\frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \texttt{inr } e : \tau_1 + \tau_2}\ \textsc{Sum-I2} \qquad \frac{\Gamma \vdash e_1 : \tau_1 + \tau_2 \quad \Gamma, x_2 : \tau_1 \vdash e_2 : \tau \quad \Gamma, x_3 : \tau_2 \vdash e_3 : \tau}{\Gamma \vdash \texttt{case}(e_1)\{x_2.\ e_2\}\{x_3.\ e_3\} : \tau}\ \textsc{Sum-Elim}$$

Definitions of *stuck*, *unsafe*, and *safe* terms:

A term e is *stuck iff* e is not a value *and* there is no e' such that $e \longrightarrow e'$.
A term e is *unsafe iff* there exists some e' such that $e \longrightarrow^* e'$ and e' is stuck.
A term e is *safe iff* for every e' such that $e \longrightarrow^* e'$, e' is not stuck.
If a term is not safe, then it is unsafe. If a term is not unsafe, then it is safe.

Reference B

The untyped lambda calculus:

$$e ::= x \mid \lambda x.\ e \mid e\ e$$

When we write multiple terms in sequence:

$$e_1\ e_2\ e_3\ e_4$$

the implicit parenthesis placement is left-associative:

$$((e_1\ e_2)\ e_3)\ e_4$$

and when we write multiple lambda terms in sequence:

$$\lambda x.\ \lambda y.\ \lambda z.\ e$$

the implicit parenthesis placement is right-associative:

$$\lambda x.\ (\lambda y.\ (\lambda z.\ e))$$

The β rule:

$$(\lambda x.\ e_1)\ e_2 \longrightarrow [x \mapsto e_2]e_1$$

which we allow to apply inside any expression context, e.g., either of the following step sequences are valid, where the underlined term is the term to which the β rule is applied:

- $\underline{(\lambda a.\ (\lambda b.\ b)\ a\ a)\ d} \longrightarrow \underline{(\lambda b.\ b)\ d}\ d \longrightarrow d\ d$
- $(\lambda a.\ \underline{(\lambda b.\ b)\ a}\ a)\ d \longrightarrow \underline{(\lambda a.\ a\ a)\ d} \longrightarrow d\ d$

The free-variables meta-function for lambda terms:

$$\begin{aligned} \mathrm{FV} &\in \mathrm{exp} \to \wp(\mathrm{var}) \\ \mathrm{FV}(x) &:= \{x\} \\ \mathrm{FV}(\lambda x.\ e) &:= \mathrm{FV}(e) \setminus \{x\} \\ \mathrm{FV}(e_1\ e_2) &:= \mathrm{FV}(e_1) \cup \mathrm{FV}(e_2) \end{aligned}$$

The substitution meta-function for lambda terms:

$$\begin{aligned} [x \mapsto e_2] &\in \mathrm{exp} \rightharpoonup \mathrm{exp} \\ [x \mapsto e_2]y &:= e_2 && \text{if } x = y \\ [x \mapsto e_2]y &:= y && \text{if } x \neq y \\ [x \mapsto e_2](\lambda y.\ e_1) &:= \lambda y.\ [x \mapsto e_2](e_1) && \text{if } x \neq y \text{ and } y \notin FV(e_2) \\ [x \mapsto e_2](e_{11}\ e_{12}) &:= ([x \mapsto e_2]e_{11})\ ([x \mapsto e_2]e_{12}) \end{aligned}$$

The standard encoding for booleans in the untyped lambda calculus:

$$\begin{aligned} \mathrm{true} &:= \lambda x.\ \lambda y.\ x \\ \mathrm{false} &:= \lambda x.\ \lambda y.\ y \\ \mathrm{cond} &:= \lambda b.\ \lambda x.\ \lambda y.\ b\ x\ y \end{aligned}$$