Mid-term examination #2 — given Tuesday 12th November

General instructions

Closed-book, closed-notes, closed-computer, in-class exam.

Time allowed: 75 minutes.

Total points available: 150 pts.

Answer in the spaces provided.

Your name (print):

I pledge my honor that in the preparation of this assignment I have complied with the University of New Mexico Board of Regents' Policy Manual.

Please sign and date:

2.1 Reduction in the untyped lambda-calculus (30 pts)

1. (10 pts) Write down an untyped lambda-term whose reduction diverges, and demonstrate this by sketching **the first two steps in the reduction sequence** for that term using the evaluation rules for the call-by value lambda-calculus (see Appendix A).

   Full derivations of the reductions are *not* required.

2. (20 pts) Briefly explain the similarities and differences between:

   - full beta reduction,
   - call-by-name reduction, and
   - call-by-value reduction.

2.2 Programming in the untyped lambda-calculus (50 pts)

1. (20 pts) Define an untyped lambda-term nor that represents the corresponding two-input Boolean operation, such that the following all hold:

$$(\mathsf{nor}\ \mathsf{fls})\ \mathsf{fls} \longrightarrow_\beta^* \mathsf{tru} \qquad (\mathsf{nor}\ \mathsf{fls})\ \mathsf{tru} \longrightarrow_\beta^* \mathsf{fls}$$

$$(\mathsf{nor}\ \mathsf{tru})\ \mathsf{fls} \longrightarrow_\beta^* \mathsf{fls} \qquad (\mathsf{nor}\ \mathsf{tru})\ \mathsf{tru} \longrightarrow_\beta^* \mathsf{fls}.$$

where the lambda-terms tru and fls are as defined in the lectures.

You must fully define any helper functions that you use in your answer.

2. (10 pts) Write down definitions for the untyped lambda-terms corresponding to the Church numerals c_0, c_1, c_2, and c_3, which represent the natural numbers 0, 1, 2, and 3, respectively.

3. (20 pts) Define an untyped lambda-term pred that implements truncated subtraction of one from a Church numeral. That is, the following should both hold:

$$\mathsf{pred}\ \mathsf{c}_0 \longrightarrow_\beta^* \mathsf{c}_0 \qquad\qquad \mathsf{pred}\ \mathsf{c}_{i+1} \longrightarrow_\beta^* \mathsf{c}_i.$$

You may assume the existence of lambda-terms *pair*, *fst* and *snd*, such that

$$\mathit{fst}\ (\mathit{pair}\ f s) \longrightarrow_\beta^* f \qquad\qquad \mathit{snd}\ (\mathit{pair}\ f s) \longrightarrow_\beta^* s,$$

along with any arithmetic functions that you need in your answer.

2.3 Simply typed lambda-calculus—typing (30 pts)

1. (10 pts) *Briefly* explain why any attempt to derive a typing judgment of the form $\varnothing \vdash t : T$ will fail if t contains one or more free variables.

2. (20 pts) Using the typing rules for the simply-typed lambda calculus extended with numbers and booleans, as defined in Appendix B, identify the type T such that

$$\varnothing \vdash (\lambda f : \mathsf{Nat} \to \mathsf{Bool}.\, \mathsf{if}\ f\ 0\ \mathsf{then\ false\ else\ true})\ (\lambda x : \mathsf{Nat}.\, \mathsf{iszero}\ (\mathsf{succ}\ x)) : T$$

holds, if such a type exists, and **construct a full typing derivation for this judgment.**

2.4 Simply typed lambda-calculus—evaluation (40 pts)

Using the evaluation rules for the simply-typed lambda calculus extended with numbers and booleans, as defined in Appendix B, **construct a full derivation for *every* step of evaluation** of the term

$$(\lambda f : \mathsf{Nat} \to \mathsf{Bool}.\, \mathsf{if}\ f\ 0\ \mathsf{then}\ \mathsf{false}\ \mathsf{else}\ \mathsf{true})\ (\lambda x : \mathsf{Nat}.\, \mathsf{iszero}\ (\mathsf{succ}\ x))$$

and identify when you have reduced the term to a value.

A Reference: untyped call-by-value lambda-calculus

Syntax

$$
\begin{array}{rcll}
\text{Terms, } t & ::= & x & \textit{variable} \\
& | & \lambda x.\, t & \textit{abstraction} \\
& | & t\ t & \textit{application} \\
\\
\text{Values, } v & ::= & \lambda x.\, t & \textit{abstraction value}
\end{array}
$$

Evaluation Rules

$$\frac{t_1 \longrightarrow t_1'}{t_1\ t_2 \longrightarrow t_1'\ t_2}\ (\text{E-App1})$$

$$\frac{t_2 \longrightarrow t_2'}{v_1\ t_2 \longrightarrow v_1\ t_2'}\ (\text{E-App2})$$

$$\frac{}{(\lambda x.\, t_{12})\ v_2 \longrightarrow [x \mapsto v_2]t_{12}}\ (\text{E-AppAbs})$$

Capture-avoiding substitution

$$
\begin{array}{rcll}
[x \mapsto t]x & = & t & \\
[x \mapsto t]y & = & y & \text{if } x \neq y \\
[x \mapsto t](t_1\ t_2) & = & ([x \mapsto t]t_1)\ ([x \mapsto t]t_2) & \\
[x \mapsto t](\lambda x.\, t') & = & \lambda x.\, t' & \\
[x \mapsto t](\lambda y.\, t') & = & \lambda y.\, ([x \mapsto t]t') & \text{if } x \neq y
\end{array}
$$

NB: this definition is capture avoiding if we assume that t is a closed term.

B Reference: simply typed lambda calculus with booleans and numbers

Syntax

Terms, t	$::=$	x	*variable*
	$\mid$	$\lambda x : T.\, t$	*abstraction*
	$\mid$	$t\ t$	*application*
	$\mid$	true	*constant true*
	$\mid$	false	*constant false*
	$\mid$	$\mathsf{if}\ t\ \mathsf{then}\ t\ \mathsf{else}\ t$	*constant false*
	$\mid$	0	*constant zero*
	$\mid$	$\mathsf{succ}\ t$	*successor*
	$\mid$	$\mathsf{pred}\ t$	*predecessor*
	$\mid$	$\mathsf{iszero}\ t$	*zero test*
Values, v	$::=$	$\lambda x : T.\, t$	*abstraction value*
	$\mid$	true	*true value*
	$\mid$	false	*false value*
	$\mid$	nv	*numeric value*
Numeric values, nv	$::=$	0	*zero value*
	$\mid$	$\mathsf{succ}\ nv$	*successor value*
Types, T	$::=$	$T \to T$	*type of functions*
	$\mid$	Bool	*type of booleans*
	$\mid$	Nat	*type of natural numbers*
Typing contexts, Γ	$::=$	$\varnothing$	*empty context*
	$\mid$	$\Gamma, x : T$	*variable type assumption*

Capture-Avoiding Substitution

$$
\begin{array}{rcl}
[x \mapsto t]x & = & t \\
[x \mapsto t]y & = & y \qquad \text{if } x \neq y \\
[x \mapsto t](t_1\ t_2) & = & ([x \mapsto t]t_1)\ ([x \mapsto t]t_2) \\
[x \mapsto t](\lambda x\!:\!T.\, t') & = & \lambda x\!:\!T.\, t' \\
[x \mapsto t](\lambda y\!:\!T.\, t') & = & \lambda y\!:\!T.\, ([x \mapsto t]t') \qquad \text{if } x \neq y \\
[x \mapsto t]\mathsf{true} & = & \mathsf{true} \\
[x \mapsto t]\mathsf{false} & = & \mathsf{false} \\
[x \mapsto t]\mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 & = & \mathsf{if}\ ([x \mapsto t]t_1)\ \mathsf{then}\ ([x \mapsto t]t_2)\ \mathsf{else}\ ([x \mapsto t]t_3) \\
[x \mapsto t]\mathsf{0} & = & \mathsf{0} \\
[x \mapsto t](\mathsf{succ}\ t') & = & \mathsf{succ}\ ([x \mapsto t]t') \\
[x \mapsto t](\mathsf{pred}\ t') & = & \mathsf{pred}\ ([x \mapsto t]t') \\
[x \mapsto t](\mathsf{iszero}\ t') & = & \mathsf{iszero}\ ([x \mapsto t]t')
\end{array}
$$

NB: this definition is capture avoiding if we assume that t is a closed term.

Evaluation Rules

$$\frac{t_1 \longrightarrow t_1'}{t_1\ t_2 \longrightarrow t_1'\ t_2}\ (\text{E-App1})$$

$$\frac{t_2 \longrightarrow t_2'}{v_1\ t_2 \longrightarrow v_1\ t_2'}\ (\text{E-App2})$$

$$\frac{}{(\lambda x\!:\!T_{11}.\, t_{12})\ v_2 \longrightarrow [x \mapsto v_2]t_{12}}\ (\text{E-AppAbs})$$

$$\frac{}{\mathsf{if\ true\ then}\ t_2\ \mathsf{else}\ t_3 \longrightarrow t_2}\ (\text{E-IfTrue})$$

$$\frac{}{\textsf{if false then } t_2 \textsf{ else } t_3 \longrightarrow t_3} \text{ (E-IfFalse)}$$

$$\frac{t_1 \longrightarrow t_1'}{\textsf{if } t_1 \textsf{ then } t_2 \textsf{ else } t_3 \longrightarrow \textsf{if } t_1' \textsf{ then } t_2 \textsf{ else } t_3} \text{ (E-If)}$$

$$\frac{t_1 \longrightarrow t_1'}{\textsf{succ } t_1 \longrightarrow \textsf{succ } t_1'} \text{ (E-Succ)}$$

$$\frac{}{\textsf{pred } 0 \longrightarrow 0} \text{ (E-PredZero)}$$

$$\frac{}{\textsf{pred } (\textsf{succ } nv_1) \longrightarrow nv_1} \text{ (E-PredSucc)}$$

$$\frac{t_1 \longrightarrow t_1'}{\textsf{pred } t_1 \longrightarrow \textsf{pred } t_1'} \text{ (E-Pred)}$$

$$\frac{}{\textsf{iszero } 0 \longrightarrow \textsf{true}} \text{ (E-IszeroZero)}$$

$$\frac{}{\textsf{iszero } (\textsf{succ } nv_1) \longrightarrow \textsf{false}} \text{ (E-IszeroSucc)}$$

$$\frac{t_1 \longrightarrow t_1'}{\textsf{iszero } t_1 \longrightarrow \textsf{iszero } t_1'} \text{ (E-Iszero)}$$

Typing Rules

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T} \text{ (T-VAR)}$$

$$\frac{\Gamma, x : T_1 \vdash t_2 : T_2 \qquad x \notin \mathrm{dom}(\Gamma)}{\Gamma \vdash \lambda x{:}T_1.\, t_2 : T_1 \to T_2} \text{ (T-ABS)}$$

$$\frac{\Gamma \vdash t_1 : T_{11} \to T_{12} \qquad \Gamma \vdash t_2 : T_{11}}{\Gamma \vdash t_1\ t_2 : T_{12}} \text{ (T-APP)}$$

$$\frac{}{\Gamma \vdash \mathsf{true} : \mathsf{Bool}} \text{ (T-TRUE)}$$

$$\frac{}{\Gamma \vdash \mathsf{false} : \mathsf{Bool}} \text{ (T-FALSE)}$$

$$\frac{\Gamma \vdash t_1 : \mathsf{Bool} \qquad \Gamma \vdash t_2 : T \qquad \Gamma \vdash t_3 : T}{\Gamma \vdash \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 : T} \text{ (T-IF)}$$

$$\frac{}{\Gamma \vdash 0 : \mathsf{Nat}} \text{ (T-ZERO)}$$

$$\frac{\Gamma \vdash t : \mathsf{Nat}}{\Gamma \vdash \mathsf{succ}\ t : \mathsf{Nat}} \text{ (T-SUCC)}$$

$$\frac{\Gamma \vdash t : \mathsf{Nat}}{\Gamma \vdash \mathsf{pred}\ t : \mathsf{Nat}} \text{ (T-PRED)}$$

$$\frac{\Gamma \vdash t : \mathsf{Nat}}{\Gamma \vdash \mathsf{iszero}\ t : \mathsf{Bool}} \text{ (T-ISZERO)}$$