Homework 2 (Solutions)

Due: Monday, Feb 12, 11:59pm

Preface

Discussing high-level approaches to homework problems with your peers is encouraged. You must include at the top of all of your assignment documents a Collaboration Statement which declares any other people with whom you discussed homework problems. For example:

> *Collaboration Statement: I discussed problems 1 and 3 with Jamie Smith. I discussed problem 2 with one of the TAs. I discussed problem 4 with a personal tutor.*

If you did not discuss the assignment with anyone, you still must declare:

> *Collaboration Statement: I did not discuss homework problems with anyone.*

Copying answers or doing the work for another student is not allowed. If there are multiple documents to submit (e.g., a written portion and programming portion), you must write a collaboration statement at the top of each portion. Please do not write your Collaboration Statement in the text of an email; you must include it directly it in your assignment.

For the written portion of the assignment, you must write your name at the top of your assignment. For the programming portion, you must write your name at the top *and the bottom* of your submitted *.ml file in an OCaml comment. E.g.:

```
<beginning of file>
(* Name: Jamie Smith *)
... your solutions to the assignment ...
(* Name: Jamie Smith *)
<end of the file>
```

Submitting

Written Portion

Prepare the written portion of your assignment as either handwritten or using LaTeX. I will not accept submissions written in Word, Google Docs, or using any other text processing software. Handwritten assignments must be written neatly or they will receive a 0 grade. Scanned pdfs of handwritten work must be created using a scanner and not a camera (e.g., from your phone).

Submit the written portion of your assignment either (1) via scanned pdf email to me: David.Darais@uvm.edu with "CS 225 HW2" in the subject line, or (2) placed under my office door (Votey 319) at any hour before the deadline.

Programming Portion

Prepare the programming portion of your assignment in a text editor of your choice. You will be working from a provided file named `hw2.ml`. Do not change the file name when you submit your assignment.

Submit the programming portion of your assignment by emailing your `hw2.ml` file to me: David.Darais@uvm.edu with "CS 225 HW2" in the subject line. I will not accept a printed or handwritten version of your `hw2.ml`.

HW2: Written Portion

Recall the definition for the syntax of the lambda calculus:

$$\begin{array}{lll} t ::= x & & \text{variable} \\ \quad \mid \lambda x.\, t & & \text{abstraction} \\ \quad \mid t\ t & & \text{application} \end{array}$$

And the definitions for free variables and closed terms.

Definition. *A variable x is free if it is not bound by an enclosing abstraction.*

Definition. *A term t is closed if it contains no free variables.*

Also consider the following terms:

$$t_1 := x \qquad t_2 := \lambda y.\, y \qquad t_3 := \lambda x.\, x\ x \qquad t_4 := (\lambda x.\, x)\ x$$

$$t_5 := \lambda x.\, \lambda y.\, x\ y \qquad t_6 := \lambda x.\, x\ y \qquad t_7 := (\lambda y.\, x)\ y \qquad t_8 := ((\lambda x.\, x)\ z\ x)((\lambda y.\, z\ y)\ y)$$

Problem 1 (5 Points)

For each of the eight terms t_i, identify the free variables in each term.

Solution

1. x
2. none
3. none
4. the second x (left-to-right)
5. none
6. y
7. x and y
8. the second x, the second y, and both zs (left-to-right)

Problem 2 (5 Points)

For each of the eight terms t_i, identify which terms are closed.

Solution

1. open
2. closed
3. closed

4. open
5. closed
6. open
7. open
8. open

Problem 3 (10 Points)

Write the term that is equal to each of the following substitutions:

a. $[x \mapsto t_2]t_1$

b. $[y \mapsto t_3]t_2$

c. $[x \mapsto t_3]t_4$

d. $[y \mapsto t_5]t_6$

e. $[z \mapsto t_2]t_8$

Solution

a.

$$\begin{aligned}&[x \mapsto t_2]t_1\\ &= [x \mapsto (\lambda y.\ y)]x\\ &= \lambda y.\ y\end{aligned}$$

b.

$$\begin{aligned}&[y \mapsto t_3]t_2\\ &= [y \mapsto (\lambda x.\ x\ x)](\lambda y.\ y)\\ &= \lambda y.\ y\end{aligned}$$

c.

$$\begin{aligned}&[x \mapsto t_3]t_4\\ &= [x \mapsto (\lambda x.\ x\ x)]((\lambda x.\ x)\ x)\\ &= (\lambda x.\ x)(\lambda x.\ x\ x)\end{aligned}$$

d.

$$\begin{aligned}&[y \mapsto t_5]t_6\\ &= [y \mapsto (\lambda x.\ \lambda y.\ x\ y)](\lambda x.\ x\ y)\\ &= \lambda x.\ x\ (\lambda x.\ \lambda y.\ x\ y)\end{aligned}$$

e.

$$\begin{aligned}&[z \mapsto t_2]t_8\\ &= [z \mapsto (\lambda y.\ y)](((\lambda x.\ x)\ z\ x)((\lambda y.\ z\ y)\ y))\\ &= ((\lambda x.\ x)\ (\lambda y.\ y)\ x)((\lambda y.\ (\lambda y.\ y)\ y)\ y)\end{aligned}$$

Problem 4 (15 Points)

Show a reduction sequence $[z \mapsto t_3]t_8 \longrightarrow^* t'$ for some t' using the full-β reduction strategy. The final term t' must be a stuck term.

Solution

$$
\begin{aligned}
&[z \mapsto t_3]t_8 \\
&= [z \mapsto (\lambda x.\ x\ x)](((\lambda x.\ x)\ z\ x)((\lambda y.\ z\ y)\ y)) \\
&= (\underline{(\lambda x.\ x)\ (\lambda x.\ x\ x)}\ x)((\lambda y.\ (\lambda x.\ x\ x)\ y)\ y) \\
&\longrightarrow (\underline{(\lambda x.\ x\ x)\ x})((\lambda y.\ (\lambda x.\ x\ x)\ y)\ y) \\
&\longrightarrow (x\ x)(\underline{(\lambda y.\ (\lambda x.\ x\ x)\ y)\ y}) \\
&\longrightarrow (x\ x)(\underline{(\lambda y.\ y\ y)\ y}) \\
&\longrightarrow (x\ x)(y\ y)
\end{aligned}
$$

Problem 5 (10 Points)

Recall the definition of the Y-combinator:

$$Y \coloneqq \lambda f.\ (\lambda x.\ f\ (x\ x))(\lambda x.\ f\ (x\ x))$$

Define a term t such that $Y\ t$ is a looping term using full-β evaluation. Justify the fact that $Y\ t$ is looping by showing a reduction sequence $Y\ t \longrightarrow^* Y\ t$.

Solution

$$t \coloneqq \lambda x.\ x$$

$$
\begin{aligned}
&Y\ t \\
&\longrightarrow^* t\ (Y\ t) \\
&= (\lambda x.\ x)\ (Y\ t) \\
&\to Y\ t
\end{aligned}
$$

Problem 6 (10 Points)

1. Recall the encoding of booleans in pure lambda calculus:

$$
\begin{aligned}
\text{true} &\coloneqq \lambda x.\ \lambda y.\ x \\
\text{false} &\coloneqq \lambda x.\ \lambda y.\ y \\
\text{cond} &\coloneqq \lambda b.\ \lambda x.\ \lambda y.\ b\ x\ y
\end{aligned}
$$

   Imagine instead cond was defined as $\text{cond} \coloneqq \lambda x.\ x$. Assuming true and false are defined as before, does this result in a valid encoding of booleans? Explain your answer.

2. Define a pure lambda calculus encoding for the following boolean connectives: negation ($\neg$), logical or ($\vee$), and implication ($\Rightarrow$). (Recall that in boolean logic, $x \Rightarrow y$ *iff* $\neg x \vee y$.) You may use true, false and cond by name in your definitions if you want.

Solution

1. Yes.

$$
\begin{aligned}
&\text{cond true } t_1\ t_2 \\
&= \underline{(\lambda x.\ x)\ (\lambda x.\ \lambda y.\ x)}\ t_1\ t_2 \\
&\longrightarrow \underline{(\lambda x.\ \lambda y.\ x)\ t_1}\ t_2 \\
&\longrightarrow \underline{(\lambda y.\ t_1)\ t_2} \\
&\longrightarrow t_1
\end{aligned}
$$

$$
\begin{aligned}
&\text{cond false } t_1\ t_2 \\
&= \underline{(\lambda x.\ x)\ (\lambda x.\ \lambda y.\ y)}\ t_1\ t_2 \\
&\longrightarrow \underline{(\lambda x.\ \lambda y.\ y)\ t_1}\ t_2 \\
&\longrightarrow \underline{(\lambda y.\ y)\ t_2} \\
&\longrightarrow t_2
\end{aligned}
$$

2. *(Many answers possible...)*

$$
\begin{aligned}
\neg &:= \lambda b.\ b\ \text{false}\ \text{true} \\
\vee &:= \lambda b_1.\ \lambda b_2.\ b_1\ \text{true}\ b_2 \\
\Rightarrow &:= \lambda b_1.\ \lambda b_2.\ b_1\ b_2\ \text{true}
\end{aligned}
$$

(Another solution...)

$$
\begin{aligned}
\neg &:= \lambda b.\ \text{cond}\ b\ \text{false}\ \text{true} \\
\vee &:= \lambda b_1.\ \lambda b_2.\ \text{cond}\ b_1\ \text{true}\ b_2 \\
\Rightarrow &:= \lambda b_1.\ \lambda b_2.\ \vee\ (\neg\ b_1)\ b_2
\end{aligned}
$$

Problem 7 (5 Points)

Approximately how many hours did you spend working on this assignment?

HW2: Programming Portion (40 Points)

First, make sure you have installed `ocaml` on your personal machine. See the Piazza post for instructions, and ask a Piazza question if you are stuck. I recommend getting `merlin` working (again, see instructions in Piazza) for useful interactive feedback while writing OCaml code.

Next, download hw2.ml from the course webpage and follow all of the directions in the file. See submission instructions for this portion of the assignment at the beginning of this writeup.

There are two problems in the programming portion. Each problem is wrapped around header indicators: e.g.,

```
(* ########## Problem 1 (15 Points) ########## *)
... your solution here ...
(* ########## End Problem 1 (15 Points) ########## *)
```

You must put your entire solution within these headers. Do not put code or modify code outside of these headers.