Assignment 3
Logic and Recursion

CS 7400: Intensive Principles of Programming Languages
Chris Martens

Due Friday, November 8, 11:59pm
75 pts

This assignment is due on the above date and must be submitted electronically on Gradescope. Please use the LaTeX template on the website to typeset your assignment and make sure to include your full name and NU ID. For the written problems, you may (alternatively) submit handwritten answers that have been scanned and are **easily legible** (if you're not sure, check with a classmate).

You should submit *one* file, `hw03.pdf`, with your written solutions to the questions.

1 Constructive Logic

For these problems, use the proof rules in Appendix A.

Task 1 (10 points) Give a derivation of $\cdot \vdash ((\top \vee \bot) \supset A) \supset A$ true.

Task 2 (20 points) Recall that we define $\neg A$ to be $A \supset \bot$. You may recall from prior logic classes *DeMorgan's duals* as theorems in propositional logic of the following form, relating conjunction, disjunction, and negation:

1. $\mathsf{DM}_1 = \neg(A \vee B) \supset \neg A \wedge \neg B$
2. $\mathsf{DM}_2 = \neg(A \wedge B) \supset \neg A \vee \neg B$
3. $\mathsf{DM}_3 = (\neg A \wedge \neg B) \supset \neg(A \vee B)$
4. $\mathsf{DM}_4 = (\neg A \vee \neg B) \supset \neg(A \wedge B)$

For which of these formulas DM_i is it possible to construct a derivation of $\cdot \vdash \mathsf{DM}_i$ true using the rules of constructive propositional logic? Provide two things in your writeup:

1. A typeset derivation tree showing that one of the DeMorgan duals is true in the empty context. (There may be more than one, but you only need to write up one of them.)
2. An explanation of which DeMorgan duals *do not* hold in constructive propositional logic and why (intuitively; you do not need to give a formal proof).

2 Primitive Recursion

For these problems, see Appendix B for typing and evaluation rules.

Task 3 (10 points) The nth *triangular number* is the sum of the first $n+1$ natural numbers: $\triangle_0 = 0$, $\triangle_1 = 0+1 = 1$, $\triangle_2 = 0+1+2 = 3$, $\triangle_3 = 0+1+2+3 = 6$, and so on. Write a function $\mathsf{tri} : \mathsf{nat} \to \mathsf{nat}$ in System T such that $\mathsf{tri}(n)$ computes the nth triangle number.

Task 4 (10 points) Prove by natural number induction that your implementation is correct, i.e. that $\mathsf{tri}(\overline{n})$ computes $\sum_{i=0\ldots n} i$.

Task 5 (10 points) The Lucas function (https://en.wikipedia.org/wiki/Lucas_number) is defined mathematically by

$$\begin{array}{lcl} \mathsf{lucas}\ 0 & = & 2 \\ \mathsf{lucas}\ 1 & = & 1 \\ \mathsf{lucas}\ n+2 & = & \mathsf{lucas}\ (n+1) + \mathsf{lucas}\ n \end{array}$$

Define the lucas function in System T.

3 Fixed Point Recursion

For this problem, refer to Appendix C for typing rules and operational semantics.

Task 6 (15 points) Consider two functions, even and odd, defined by mutual recursion:

$$\begin{array}{lcl} \mathsf{even}\ 0 & = & \mathsf{true} \\ \mathsf{even}\ (n+1) & = & \mathsf{odd}\ n \\ & & \\ \mathsf{odd}\ 0 & = & \mathsf{false} \\ \mathsf{odd}\ (n+1) & = & \mathsf{even}\ n \end{array}$$

Define these functions using the fixed point operator in PCF, enriched with lazy pairs ($\tau_1 \ \& \ \tau_2$) and booleans. Your result should provide two definitions, $\mathsf{even} : \mathsf{nat} \rightharpoonup \mathsf{bool}$ and $\mathsf{odd} : \mathsf{nat} \rightharpoonup \mathsf{bool}$, potentially defined in terms of one or more auxiliary definitions.

Hint #1: PFPL Chapter 10 contains an implementation of these functions in System T.

Hint #2: Instead of thinking of even and odd as a *pair of mutually-recursive functions*, think of them as a *recursively-defined pair of functions*.

A Rule Sheet: Constructive Propositional Logic

Truth:

$$\frac{}{\top\ \mathsf{true}}\ \top/I \qquad (\text{no } \top/E\ )$$

Conjunction:

$$\frac{A\ \mathsf{true} \quad B\ \mathsf{true}}{A \wedge B\ \mathsf{true}}\ \wedge/I \qquad \frac{A \wedge B\ \mathsf{true}}{A\ \mathsf{true}}\ \wedge/E_1 \qquad \frac{A \wedge B\ \mathsf{true}}{B\ \mathsf{true}}\ \wedge/E_2$$

Disjunction:

$$\frac{A\ \mathsf{true}}{A \vee B\ \mathsf{true}}\ \vee/I_1 \qquad \frac{B\ \mathsf{true}}{A \vee B\ \mathsf{true}}\ \vee/I_2$$

$$\frac{A \vee B\ \mathsf{true} \quad \begin{array}{c} A\ \mathsf{true} \\ \vdots \\ C\ \mathsf{true} \end{array} \quad \begin{array}{c} B\ \mathsf{true} \\ \vdots \\ C\ \mathsf{true} \end{array}}{C\ \mathsf{true}}\ \vee/E$$

Implication:

$$\frac{\begin{array}{c} A\ \mathsf{true} \\ \vdots \\ B\ \mathsf{true} \end{array}}{A \supset B\ \mathsf{true}}\ \supset/I \qquad \frac{A \supset B\ \mathsf{true} \quad A\ \mathsf{true}}{B\ \mathsf{true}}\ \supset/E$$

Falsehood:

$$(\text{no } \bot/I) \qquad \frac{\bot\ \mathsf{true}}{C\ \mathsf{true}}\ \bot/E$$

B Rule Sheet: System T

B.1 Type System

$$\frac{}{\mathsf{zero} : \mathsf{nat}}\ \mathsf{ty/zero} \qquad \frac{e : \mathsf{nat}}{\mathsf{succ}(e) : \mathsf{nat}}\ \mathsf{ty/succ}$$

$$\frac{e : \mathsf{nat} \quad e_0 : \tau \quad x{:}\mathsf{nat}, y{:}\tau \vdash e_s : \tau}{\mathsf{rec}(e, e_0, x.y.\ e_s) : \tau}\ \mathsf{ty/rec}$$

$$\frac{}{x{:}\tau \vdash x : \tau}\ \mathsf{ty}/x \qquad \frac{x : \tau_1 \vdash e : \tau_2}{\lambda x.\ e : \tau_1 \to \tau_2}\ \mathsf{ty/lam} \qquad \frac{f : \tau_1 \to \tau_2 \quad e : \tau_1}{f\ e : \tau_2}\ \mathsf{ty/app}$$

B.2 Operational Semantics

Values:

$$\frac{}{\mathsf{zero}\ \mathsf{value}}\ \mathsf{value/zero} \qquad \frac{e\ \mathsf{value}}{\mathsf{succ}(e)\ \mathsf{value}}\ \mathsf{value/succ}$$

$$\frac{}{\lambda x.\ e\ \mathsf{value}}\ \mathsf{val/lam}$$

Computation steps:

$$\frac{}{\mathsf{rec}(\mathsf{zero}, e_0, x.y.e_s) \mapsto e_0}\ \mathsf{step/rec/zero}$$

$$\frac{e\ \mathsf{value}}{\mathsf{rec}(\mathsf{succ}(e), e_0, x.y.e_s) \mapsto [e/x][\mathsf{rec}(e, e_0, x.y.e_s)/y]e_s}\ \mathsf{step/rec/succ}$$

$$\frac{e'\ \mathsf{value}}{(\lambda x.\ e)\ (e') \mapsto [e'/x']e}\ \mathsf{step/app/lam}$$

Congruence steps:

$$\frac{e \mapsto e'}{\mathsf{succ}(e) \mapsto \mathsf{succ}(e')}\ \mathsf{step/succ} \qquad \frac{e \mapsto e'}{\mathsf{rec}(e, e_0, x.y.e_s) \mapsto \mathsf{rec}(e', e_0, x.y.e_s)}\ \mathsf{step/rec/1}$$

$$\frac{f \mapsto f'}{f\ e \mapsto f'\ e}\ \mathsf{step/app/fn} \qquad \frac{f\ \mathsf{value} \quad e \mapsto e'}{f\ e \mapsto f\ e'}\ \mathsf{step/app/arg}$$

C Rule Sheet: PCF with Lazy Pairs and Booleans

C.1 Type System

Natural numbers:

$$\frac{}{\mathsf{zero} : \mathsf{nat}}\ \mathsf{ty/zero} \qquad \frac{e : \mathsf{nat}}{\mathsf{succ}(e) : \mathsf{nat}}\ \mathsf{ty/succ}$$

$$\frac{e : \mathsf{nat} \quad e_0 : \tau \quad x{:}\mathsf{nat} \vdash e_s : \tau}{\mathsf{ifz}(e, e_0, x.e_s) : \tau}\ \mathsf{ty/ifz}$$

Booleans:

$$\frac{}{\mathsf{true} : \mathsf{bool}}\ \mathsf{ty/true} \qquad \frac{}{\mathsf{false} : \mathsf{bool}}\ \mathsf{ty/false}$$

$$\frac{e : \mathsf{bool} \quad e_t : \tau \quad e_f : \tau}{\mathsf{ite}(e, e_t, e_f) : \tau}\ \mathsf{ty/ite}$$

Fixed points:

$$\frac{x : \tau \vdash e : \tau}{\mathsf{fix}\{\tau\}(x.e) : \tau}\ \mathsf{ty/fix}$$

Partial functions:

$$\frac{x{:}\tau_1 \vdash e : \tau_2}{\lambda x.\ e : \tau_1 \rightharpoonup \tau_2}\ \mathsf{ty/parfun} \qquad \frac{f : \tau_1 \rightharpoonup \tau_2 \quad e : \tau_1}{f\ \ e : \tau_2}\ \mathsf{ty/parapp}$$

Lazy pairs (&):

$$\frac{e_1 : \tau_1 \quad e_2 : \tau_2}{\langle e_1, e_2 \rangle : \tau_1 \ \&\ \tau_2}\ \mathsf{ty/lpair} \qquad \frac{e : \tau_1 \ \&\ \tau_2}{e.1 : \tau_1}\ \mathsf{ty/proj_1} \qquad \frac{e : \tau_1 \ \&\ \tau_2}{e.2 : \tau_2}\ \mathsf{ty/proj_2}$$

C.2 Operational Semantics

Values:

$$\frac{}{\langle e_1, e_2\rangle \text{ value}}\ \textsf{val/lpair} \qquad \frac{}{\lambda x.\ e \text{ value}}\ \textsf{val/lam}$$

$$\frac{}{\textsf{zero value}}\ \textsf{value/zero} \qquad \frac{e \text{ value}}{\textsf{succ}(e) \text{ value}}\ \textsf{value/succ}$$

$$\frac{}{\textsf{true value}}\ \textsf{value/true} \qquad \frac{}{\textsf{false value}}\ \textsf{value/false}$$

Computation rules:

$$\frac{}{\textsf{fix}\{\tau\}(x.e) \mapsto [\textsf{fix}\{\tau\}(x.e)/x]e}\ \textsf{step/fix}$$

$$\frac{}{\textsf{ifz}(\textsf{zero}, e_0, x.e_s) \mapsto e_0}\ \textsf{step/ifz/z}$$

$$\frac{\textsf{succ}(e) \text{ value}}{\textsf{ifz}(\textsf{succ}(e), e_0, x.e_s) \mapsto [e/x]e_s}\ \textsf{step/ifz/s}$$

$$\frac{}{\textsf{ite}(\textsf{true}, e_t, e_f) \mapsto e_t}\ \textsf{step/if/true} \qquad \frac{}{\textsf{ite}(\textsf{false}, e_t, e_f) \mapsto e_f}\ \textsf{step/if/false}$$

$$\frac{}{\langle e_1, e_2\rangle.1 \mapsto e_1}\ \textsf{step/proj1} \qquad \frac{}{\langle e_1, e_2\rangle.2 \mapsto e_2}\ \textsf{step/proj2}$$

$$\frac{e' \text{ value}}{(\lambda x.\ e)\ (e') \mapsto [e'/x']e}\ \textsf{step/app/lam}$$

Congruence Rules:

$$\frac{e \mapsto e'}{e.1 \mapsto e'.1}\ \textsf{step/proj}_1\textsf{/1} \qquad \frac{e \mapsto e'}{e.2 \mapsto e'.2}\ \textsf{step/proj}_2\textsf{/1}$$

$$\frac{e \mapsto e'}{\textsf{succ}(e) \mapsto \textsf{succ}(e')}\ \textsf{step/succ}$$

$$\frac{f \mapsto f'}{f\ e \mapsto f'\ e}\ \textsf{step/app/fn} \qquad \frac{f \text{ value} \quad e \mapsto e'}{f\ e \mapsto f\ e'}\ \textsf{step/app/arg}$$

$$\frac{e \mapsto e'}{\textsf{ite}(e, e_t, e_f) \mapsto \textsf{ite}(e', e_t, e_f)}\ \textsf{step/ite/1}$$