Lab 5: Lambda Calculus and Grammars

CSI 3120

Problem 1

The OCaml program

```
let f (x:int) = x+4 in
let g (y:int) = 3-y in
f(g(1))
```

can be written as the following lambda expression:

$$\left(\underbrace{(\lambda f.\lambda g.f(g\ 1))}_{\text{main}}\underbrace{(\lambda x.x+4)}_{\text{f}}\right)\underbrace{(\lambda y.3-y)}_{\text{g}}.$$

Reduce the expression to a normal form in two different ways, as described below.

(a) Reduce the expression by choosing, at each step, the reduction that eliminates a λ as far to the *left* as possible.

(b) Reduce the expression by choosing, at each step, the reduction that eliminates a λ as far to the *right* as possible.

Problem 2

Here is a "sugared" lambda expression that uses let declarations:

$$\begin{array}{l} \mathrm{let}\ compose = \lambda f.\lambda g.\lambda x.f(g\ x)\ \mathrm{in} \\ \quad \mathrm{let}\ h = \lambda x.x + x\ \mathrm{in} \\ \qquad compose\ h\ h\ 3 \end{array}$$

The "desugared" lambda expression, obtained when each $\mathrm{let}\ x = U\ \mathrm{in}\ V$ is replaced with $(\lambda z.V)U$ is

$$\begin{array}{l} (\lambda compose. \\ \quad (\lambda h.compose\ h\ h\ 3)\ \lambda x.x + x) \\ \quad \lambda f.\lambda g.\lambda x.f(g\ x). \end{array}$$

This is written with the same variable names as those of the let form to make it easier to read the expression.

Simplify the desugared lambda expression by using reduction. Make sure you understand why the simpified expression is the answer that is expected.