Solutions to Assignment 3.5

Problem 1. Extend Arithmetic Expressions with another case called an application. Namely, if e and f are ae's and x is a variable, and then the *application of* e*, regarded as a function of* x*, to* f, written

$$((\lambda(x)e)f),$$

is also an ae. The meaning of the application is given by the rule

$$[\![((\lambda(x)e)f)]\!]V ::= [\![e]\!](V[x \leftarrow [\![f]\!]V]) \tag{1}$$

(a) Extend the definition of the substitution operation, $[\sigma]$, so it applies to ae's with applications and so that the Substitution Lemma will continue to hold.

Hint: Substitution into an application will now generally have to include renaming lambda-bound identifiers to avoid "false capture." For example,

$$\begin{aligned}[\![((\lambda(y)\ ((\lambda(x)(x+y))\ 0)\ )\ x)]\!]V &::= [\![((\lambda(x)(x+y))\ 0)]\!]V[y \leftarrow V(x)]) \\ &= [\![x+y]\!]((V[y \leftarrow V(x)])[x \leftarrow 0]) \\ &= V(x),\end{aligned}$$

whereas if we try evaluating by naively substitute x for y inside the inner application, we get a different result:

$$\begin{aligned}[\![((\lambda(x)((x+y)[y := x]))\ 0)]\!]V &= [\![((\lambda(x)2x)\ 0)]\!]V \\ &= 0.\end{aligned}$$

The fix is to replace the x's in the inner application by a "fresh" variable, x' before substituting x for y:

$$\begin{aligned}[\![((\lambda(x')(x'+y)[y := x])\ 0)]\!]V &= [\![((\lambda(x')(x'+x))\ 0)]\!]V \\ &= [\![x'+x]\!](V[x' \leftarrow 0]) \\ &= V(x).\end{aligned}$$

Solution. Substitution into an application will be defined by the rule:

$$((\lambda(x)e)f)[\sigma] ::= ((\lambda(x')e[\sigma'])f[\sigma]) \tag{2}$$

where x' is a "fresh" variable — a variable not occurring in e, f, or σ — and

$$\sigma' ::= \sigma[x \leftarrow x'].$$

■

(b) Prove it.

Solution.

Lemma (General Substitution).

$$[\![M[\sigma]]\!]V = [\![M]\!]V_\sigma, \tag{3}$$

where

$$V_\sigma(y) ::= [\![\sigma(y)]\!]V.$$

The proof by structural induction on M of the Substitution Lemma for ordinary ae's carries over without change for ae's with applications, except that there is now one further induction case, namely when M is an application, $((\lambda(x)e)f)$. In this case, the lefthand side of (3) is

$$\begin{aligned}
[\![((\lambda(x)e)f)[\sigma]]\!]V &= [\![((\lambda(x')e[\sigma'])f[\sigma])]\!]V && \text{(by (2))}\\
&= [\![e[\sigma']]\!]U \quad \text{where } U ::= V[x' \leftarrow [\![f[\sigma]]\!]V] && \text{(by (1))}\\
&= [\![e]\!]U_{\sigma'} && \text{(ind. hypothesis for } e)
\end{aligned} \tag{4}$$

Similarly, the righthand side of (3) is

$$\begin{aligned}
[\![((\lambda(x)e)f)]\!]V_\sigma &= [\![e]\!](V_\sigma[x \leftarrow [\![f]\!]V_\sigma]) && \text{(by (1))}\\
&= [\![e]\!]W \qquad \text{where } W ::= V_\sigma[x \leftarrow [\![f]\!]V_\sigma].
\end{aligned} \tag{5}$$

So we need only prove that

$$U_{\sigma'} = W \tag{6}$$

to conclude from (4) and (5) that the General Substitution Lemma holds for an application.

To prove (6), we use a key property of "fresh" variables:

Lemma (Fresh Variable). *For any ae, M, valuation, V, and value, d, if x' does not occur in M, then*

$$[\![M]\!]V = [\![M]\!](V[x' \leftarrow d])$$

The proof of the Fresh Variable Lemma follows immediately by structural induction on M, as the reader can easily verify.

Now for a variable y distinct from x, we have

$$\begin{aligned}
U_{\sigma'}(y) &= [\![\sigma'(y)]\!]U && \text{(by def. of (valuation)}_{\sigma'}\text{)}\\
&= [\![\sigma(y)]\!]U && \text{(def. of } \sigma'\text{)}\\
&= [\![\sigma(y)]\!]V && \text{(by Fresh Var. Lemma since } x' \notin \sigma(y)\text{)}\\
&= V_\sigma(y) && \text{(def. of } V_\sigma\text{)}\\
&= W(y) && \text{(by def. of } W\text{, since } y \text{ is not } x\text{)}
\end{aligned}$$

Also,

$$\begin{aligned}
U_{\sigma'}(x) &= [\![\sigma'(x)]\!]U && \text{(by def. of (valuation)}_{\sigma'}\text{)}\\
&= [\![x']\!]U && \text{(def. of } \sigma'\text{)}\\
&= [\![f[\sigma]]\!]V && \text{(def. of } U\text{)}\\
&= [\![f]\!]V_\sigma && \text{(ind. hypothesis for } f\text{)}\\
&= W(x) && \text{(def. of } W\text{)}
\end{aligned}$$

So we have shown that $U_{\sigma'} = W$, completing the proof. ■