Kyle Burke
CS520

Assignment 3 Answer Key

Problem 1. eqfn = λg.λx.λy. if iszero(x)
then
if iszero(y) true else false
else
if iszero(y) then false else g (pred x) (pred y)

eq = Y eqfn

eq 2 2

$\Rightarrow$ (λx.eqfn(x x))(λx.eqfn(x x)) 2 2

$\Rightarrow^*$ eq 1 1

$\Rightarrow^*$ eq 0 0

$\Rightarrow^*$ true

Problem 2. We will conduct this proof by contradiction:

X Assume that $T$ is a tautology in $IL$.

Now, by the *formulas-as-types cooresponddence*, there is some term $t_0$ for which $\vdash t_0 : T$ is derivable.

Using the Normalization Theorem, there is a simply typable term $t_1$, which is in normal form under full beta reduction such that $t_0 \Rightarrow^* t_1$.

The Preservation Theorem states that $t_1$ has the same type, $(T)$ as $t_0$.

We see that $t_1$ has the form $\lambda x : (a \to b) \to a.A$

We examine the possibilities for the types for $A$:

• $A$ is a variable. Since it is in closed form, it cannot be a free variable, and would have to be $x$. Then, however, the type of $t_1$ would be $((a \to b) \to a) \to ((a \to b) \to a)$. Thus, $A$ cannot be a variable.

• $A$ is an abstraction, because then its type would not be $a$, but would have the form of a function. Thus, $A$ cannot be an abstraction.

• It must be then that $A$ is an application. Thus, $A$ has the form $t_2\ t_3$, where $t_2$ and $t_3$ are terms. Again, we investigate possible cases:

•• $t_2$ is an abstraction. This cannot be the case, though, because then we could have evaluated further on $t_3$, and $t_2\ t_3$ would not be in normal form.

$\bullet\bullet$ $t_2$ is an application. Now, instead of running through three more cases, and continuing endlessly, at some point, our chain of applications on the left must end in a variable (none can be an abstraction for the same reason as above). Since we haven't encountered any abstractions, the only bound variable choice we have is $x$. However, $x$ is of the type $(a \to b) \to a$, and only has one type as input. If we have descended through $n$ levels of applications (including the current one), i.e., $t_2$ is of the form $t_{2,1}\, t_{2,2} \cdots t_{2,n-1}$, then $x$ must have the type $T_1 \to T_2 \to \cdots \to T_n \to a$ (with $t_{2,1} : T_1, \ldots, t_{2,n_1} : T_{n-1}$ and $t_3 : T_n$). The only way this fits is if $n = 1$, forcing $T_1 = (a \to b)$ and $t_2$ be a variable.

$\bullet\bullet$ $t_2$ is a variable. Since $x$ is the only bound choice we have, $t_2 = x$. $x$ is of type $(a \to b) \to a$, requiring that $t_3$ be of type $a \to b$. We use the same reasoning as above to show that $t_3$ must be an application: in fact, a sequence of applications on the left. This sequence must end with the variable $x$ on the left at some point ($t_3$ itself cannot be $x$, because it is not of the correct type). Thus, after descending through $n$ levels of applications, $x$ will be of type $T_1 \to \cdots \to T_n \to (a \to b)$. However, since $a \to b$ cannot equal $a$, $x$ cannot have this type. $\rightarrow\!\leftarrow$ . Thus, $t_3$ cannot be an application, and $t_2$ cannot be a variable.

Thus, since $A$ cannot take any viable form, $T$ is not a tautology.

Problem 3. The solution is in the book.

Problem 4. The solution is in the book.

Problem 5. The solution is in the book.