Assignment 4

Part 1:

1)

(a) **False**. The type of function's g image is T2 and the type of function's f domain is T1, so the composition of f with g may cause a wrong parameter type passed to f. It will be true only if T2 is compatible with T1

(b) **True.** Under the assumption of left hands side we can see that both y and f's domain has type T2. So y can be passed as a parameter to f. The type of the expression (f y) is the same type of f's image. Thus by left hand side assumptions we can derive (f y): T1.

(c) **True.** Since the expression is a unitary procedure we know its type must be [T'->T'']. Because the only thing the procedure does is to pass the parameter to f, we can deduce that T'=T1. Since the returned value of the procedure is exactly the return value of f, we deduce T''=T2, and thus conclude [T'->T'']=[T1->T2] which is what we wanted to prove.

(d) **False.** T2 could be bound to a type not compatible with Number. Thus, the expression (f x 100) in this case will be a type error. If in fact T2 is indeed compatible with Number so the statement will be true.

(a) ((lambda (x1) (+ x1 1)) 4)

Stage I: Rename bound Variables:

((lambda (x1) (+ x1 1)) 4) turn to ((lambda (x) (+ x 1)) 4)

Stage II: Assign type Variables for every sub-expression:

Expression	Variable
((lambda (x) (+ x 1)) 4)	T0
(lambda (x) (+ x 1))	T1
(+ x 1)	T2
+	T+
x	Tx
1	Tnum1
4	Tnum4

Stage III: Construct type equations:

The equations for sub-expressions are:

Expression	Equation
((lambda(x)(+ x 1)) 4)	$T_1 = [T_{num4} \to T_0]$
(lambda(x)(+ x 1))	$T_1 = [T_x \to T_2]$
(+ x 1)	$T_+ = [T_x * T_{num1} \to T_2]$

The equations for the primitives are:

Expression	Equation
+	$T_+ = [Number * Number \to Number]$
1	$T_{num1} = Number$
4	$T_{num4} = Number$

Stage IV: Solve the equations:

Equasion	substitution
$T_1 = [T_{num4} \to T_0]$	{ }
$T_1 = [T_x \to T_2]$	
$T_+ = [T_x * T_{num1} \to T_2]$	
$T_+ = [Number * Number \to N]$	
$T_{num1} = Number$	
$T_{num4} = Number$	

Step 1: $(T_1 = [T_{num4} \to T_0]) \circ Sub = (T_1 = [T_{num4} \to T_0])$

$Sub = Sub \circ (T_1 = [T_{num4} \to T_0])$

Equation	Substitution
$T_1 = [T_x \to T_2]$	$\{T_1 := [T_{num4} \to T_0])\}$
$T_+ = [T_x * T_{num1} \to T_2]$	
$T_+ = [Number * Number \to Number]$	
$T_{num1} = Number$	
$T_{num4} = Number$	

Step 2: $(T_1 = [T_x \to T_2]) \circ Sub = ([T_{num4} \to T_0] = [T_x \to T_2])$

Both sides are composite so we split into two equations:

Equation	Substitution
$T_+ = [T_x * T_{num1} \to T_2]$	$\{T_1 := [T_{num4} \to T_0]\}$
$T_+ = [Number * Number \to N]$	
$T_{num1} = Number$	
$T_{um4} = Number$	
$T_x = T_{num4}$	
$T_2 = T_0$	

Step 3: $(T_+ = [T_x * T_{num1} \to T_2]) \circ Sub = (T_+ = [T_x * T_{num1} \to T_2])$

Equation	Substitution
$T_+ = [Number * Number \to Number]$	$\{T_1 := [T_{num4} \to T_0], T_+ := [T_x * T_{num1} \to T_2]\}$
$T_{num1} = Number$	
$T_{num4} = Number$	
$T_x = T_{num4}$	
$T_2 = T_0$	

Step 4: $(T_+ = [Number * Number \to Number]) \circ sub =$

$= ([T_x * T_{num1} \to T_2] = [Number * Number \to Number])$

We split the equation into three:

Equation	Substitution
$T_{num1} = Number$	$\{T_1 := [T_{num4} \to T_0], T_+ := [T_x * T_{num1} \to T_2]\}$
$T_{num4} = Number$	
$T_x = T_{num4}$	
$T_2 = T_0$	
$T_x = Number$	
$T_{num1} = Number$	
$T_2 = Number$	

Step 5: $(T_{num1} = Number) \circ sub = (T_{num1} = Number)$

$sub = sub \circ (T_{num1} = Number)$

Equation	Substitution
$T_{num4} = Number$	$\{T_1 := [T_{num4} \to T_0], T_+ := [T_x * Number \to T_2], T_{num1} := Number\}$
$T_x = T_{num4}$	
$T_2 = T_0$	
$T_x = Number$	
$T_{num1} = Number$	
$T_2 = Number$	

Step 6: $(T_{num4} = Number) \circ sub = (T_{num4} = Number)$, $sub = sub \circ (T_{num4} = Number)$

Equation	Substitution
$T_x = T_{num4}$	$\{T_1 := [Number \to T_0], T_+ := [T_x * Number \to T_2], T_{num1} := Number, T_{num4} := Number\}$
$T_2 = T_0$	
$T_x = Number$	
$T_{num1} = Number$	
$T_2 = Number$	

Step 7: $(T_x = T_{num4}) \circ Sub = (T_x = Number)$, $Sub = Sub \circ (T_x = Number)$

Equation	Substitution
$T_2 = T_0$	$\{T_1 := [Number \to T_0],\ T_+ := [Number \to T_2],$ $T_{num1} := Number,\ T_{num4} := Number,\ T_x := Number\}$
$T_x = Number$	
$T_{num1} = Number$	
$T_2 = Number$	

Step 8: $(T_2 = T_0) \circ Sub = (T_2 = T_0)$, $Sub = Sub \circ (T_2 = T_0)$

Equation	Substitution
$T_x = Number$	$\{T_1 : [Number \to T_0],\ T_+ := [Number \to T_0],$ $T_{num1} = Number,\ T_{num4} := Number,\ T_x := Number,$ $T_2 = T_0\}$
$T_{num1} = Number$	
$T_2 = Number$	

Step 9:

~~$(T_x = Number) \circ Sub = (T_x = Number)$~~

$(T_x = Number) \circ Sub = (Number = Number)$

Equation	Substitution
$T_{num1} = Number$	$\{T_1 : [Number \to T_0],\ T_+ := [Number \to T_0],$ $T_{num1} = Number,\ T_{num4} := Number,\ T_x := Number,$ $T_2 = T_0\}$
$T_2 = Number$	

Step 10: $(T_{num1} = Number) \circ Sub = (Number = Number)$

Equation	Substitution
$T_2 = Number$	$\{T_1 := [Number \to T_0],\ T_+ := [Number \to T_0],\ T_{num1} := Number,$ $T_{num4} := Number,\ T_x = Number,\ T_2 := T_0\}$

Step 11: $(T_2 = Number) \circ Sub = (T_0 = Number)$, $Sub = Sub \circ (T_0 = Number)$

Equation	Substitution
	$\{T_1 := [Number \to Number],\ T_+ := [Number \to Number],$ $T_{num1} := Number,\ T_{num4} := Number,\ T_x = Number,$ $T_2 = Number,\ T_0 = Number\}$

(b) ((lambda (f1 x1) (f1 x1 1)) 4 +)

Stage I: Rename bound variables:

((lambda(f1 x1)(f1 x1 1)) 4 +) turn to:

((lambda(f x)(f x 1)) 4 +)

Stage II: Assign type variable for every sub-expression:

Expression	Variable
((lambda(f x)(f x 1)) 4 +)	T0
(lambda(f x)(f x 1))	T1
(f x 1)	T2
f	Tf
x	Tx
1	Tnum1
4	Tnum4
+	T+

Stage III: Construct type equations:

The equations for sub-expressions are:

Expression	Equation
((lambda (f x) (f x 1)) 4 +)	$T1 = [T_{num4} * T_{+} \rightarrow T0]$
(lambda (f x) (f x 1))	$T1 = [T_f * T_x \rightarrow T_2]$
(f x 1)	$T_f = [T_x * T_{num1} \rightarrow T_2]$

The equations for the primitives are:

Expression	Equations
1	$T_{num1} = Number$
4	$T_{num4} = Number$
+	$T_{+} = [Number * Number \rightarrow Number]$

Stage IV: Solve the equations:

Equation	Substitution
$T_1 = [T_{num4} * T_{+} \rightarrow T0]$	{ }
$T_1 = [T_f * T_x \rightarrow T_2]$	
$T_f = [T_x * T_{num1} \rightarrow T_2]$	
$T_{num1} = Number$	
$T_{num4} = Number$	
$T_{+} = [Number * Number \rightarrow Number]$	

Step 1: $(T_1 = [T_{num4} * T_+ \to T_0]) \circ Sub = (T_1 = [T_{num4} * T_+ \to T_0])$

$$Sub = Sub \circ (T_1 = [T_{num4} * T_+ \to T_0])$$

Equation	Substitution
$T_1 = [T_f * T_x \to T_2]$	$\{T_1 := [T_{num4} * T_+ \to T_0]\}$
$T_f = [T_x * T_{num1} \to T_2]$	
$T_{num1} = Number$	
$T_{num4} = Number$	
$T_+ = [Number * Number \to Number]$	

Step 2: $(T_1 = [T_f * T_x \to T_2]) \circ Sub = ([T_{num4} * T_+ \to T_0] = [T_f * T_x \to T_2])$

Both sides are composite so we split into three equations:

Equation	Substitution
$T_f = [T_x * T_{num1} \to T_2]$	$\{T_1 := [T_{num4} * T_+ \to T_0]\}$
$T_{num1} = Number$	
$T_{num4} = Number$	
$T_+ = [Number * Number \to Number]$	
$T_{num4} = T_f$	
$T_+ = T_x$	
$T_2 = T_0$	

Step 3: $(T_F = [T_x * T_{num1} \to T_2]) \circ sub = (T_F = [T_x * T_{num1} \to T_2])$

$\cancel{sub = sub \circ (T_1 = [T_{num4} * T_+ \to T_0}$

$sub = sub \circ (T_F = [T_x * T_{num1} \to T_2])$

Equation	Substitution
$T_{num1} = Number$	$\{T_1 := [T_{num4} * T_+ \to T_0],\ T_F := [T_x * T_{num1} \to T_2]\}$
$T_{num4} = Number$	
$T_+ = [Number * Number \to Number]$	
$T_{num4} = T_F$	
$T_+ = T_x$	
$T_2 = T_0$	

Step 4: $(T_{num1} = Number) \circ sub = (T_{num1} = Number)$

$sub = sub \circ (T_{num1} = Number)$

Equation	Substitution
$T_{num4} = Number$	$\{T_1 := [T_{num4} * T_+ \to T_0],\ T_F := [T_x * Number \to T_2],$ $T_{num1} := Number\}$
$T_+ = [Number * Number \to Number]$	
$T_F = T_{num4}$	
$T_+ = T_x$	
$T_2 = T_0$	

Step 5: $(T_{num4} = Number) \circ sub = (T_{num4} = Number)$, $sub = sub \circ (T_{num4} = Number)$

Equation	Substitution
$T_+ = [Number * Number \to Number]$	$\{T_1 := [Number * T_+ \to T_0],\ T_F := [T_x * Number \to T_2],$ $T_{num1} := Number,\ T_{num4} := Number\}$
$T_F = T_{num4}$	
$T_+ = T_x$	
$T_2 = T_0$	

Step 6: $(T_+ = [Number * Number \to Number]) \circ sub = (T_+ = [Number * Number \to Number])$

Equation	substitution
$T_f = T_{num4}$	$\{T_1 := [Number * [Number * Number \to Number] \to Number],\ T_f := [T_x * Number \to T_2],\ T_{num1} := Number,\ T_{num4} := Number,\ T_+ := [Number * Number \to Number]\}$
$T_+ = T_x$	
$T_2 = T_0$	

Step 7: $(T_f = T_{num4}) \circ sub = ([T_x * Number \to T_2] = Number)$

At this point of the algorithm, the Procedure "checkEqualType" will fail and return an error.

In the abstract algorithm we saw in class, we are at the "Solving the equations" part in the case where:

if [te1 ∘ sub] is an atomic and [te2 ∘ sub] is a composite type:

$$\boxed{substitution := FAIL}$$

Part 2:

2.2)

(b) The reason that 'asycMemo' returns a promise is because we return the value of f(param) set in the 'PromisedStore', where the value of f(param) is saved there as a promise. So the PromiseStore returns us a promise of a value and thus the return type of 'asycMemo' must return a promise.