Assignment 4

Dor Lebel 205872807 Nitzan Hochman 316264845

1:

((lambda (x1 y1) (if (> x1 y1) #t #f)) 8 3)

Rename bound variables:

((lambda (x y) (if (> x y) #t #f)) 8 3)

Assign type variables to all sub-exps:

Expression	Var
((lambda (x y) (if (> x y) #t #f)) 8 3)	T_0
(lambda (x y) (if (> x y) #t #f))	T_1
(if (> x y) #t #f)	T_{if}
(> x y)	T_2
>	$T_>$
x	T_x
y	T_y
#f	$T_{\#f}$
#t	$T_{\#t}$
8	T_{num8}
3	T_{num3}

Construct type equations:

Expression	Equation
((lambda (x y) (if (> x y) #t #f)) 8 3)	$T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$
(lambda (x y) (if (> x y) #t #f))	$T_1 = [T_x * T_y \rightarrow T_{if}]$
(if (> x y) #t #f)	$T_{if} = T_{\#t}, T_{\#t} = T_{\#f}$
(> x y)	$T_> = [T_x * T_y \rightarrow T_2]$
>	$T_> \rightarrow [N * N -> B]$
#f	$T_{\#f} = B$
#t	$T_{\#t} = B$
8	$T_{num8} = N$
3	$T_{num3} = N$

Solving the equations:

Expression	Substitution
$T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$	
$T_1 = [T_x * T_y \rightarrow T_{if}]$	
$T_{if} = T_{\#t}, T_{\#t} = T_{\#f}$	
$T_> = [T_x * T_y \rightarrow T_2]$	
$T_> \rightarrow [N * N -> B]$	
$T_{\#f} = B$	
$T_{\#t} = B$	
$T_{num8} = N$	
$T_{num3} = N$	

$T_1 = [T_{num8} * T_{num3} \rightarrow T_0] \circ Substitution = T_1 = [T_{num8} * T_{num3} \rightarrow T_0].$
$Substitution = Substitution \circ T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$

Expression	Substitution
~~$T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$~~	$\boldsymbol{T_1 = [T_{num8} * T_{num3} \rightarrow T_0]}$
$T_1 = [T_x * T_y \rightarrow T_{if}]$	
$T_{\#t} = T_{\#f}$	
$T_{if} = T_{\#t}$	
$T_> = [T_x * T_y \rightarrow T_2]$	
$T_> \rightarrow [N * N -> B]$	
$T_{\#f} = B$	
$T_{\#t} = B$	
$T_{num8} = N$	
$T_{num3} = N$	

$T_1 = [T_x * T_y \rightarrow T_{if}] \circ Substitution = (T_1 = [T_x * T_y \rightarrow T_{if}] = [T_{num8} * T_{num3} \rightarrow T_0]) \Rightarrow T_X = T_{num8}, T_Y = T_{num3}, T_0 = T_{if}$

Expression	Substitution
~~$T_1 = [T_x * T_y \rightarrow T_{if}]$~~	$T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$
$T_{\#t} = T_{\#f}$	
$T_{if} = T_{\#t}$	
$T_> = [T_x * T_y \rightarrow T_2]$	
$T_> \rightarrow [N * N -> B]$	
$T_{\#f} = B$	
$T_{\#t} = B$	
$T_{num8} = N$	
$T_{num3} = N$	
$\boldsymbol{T_x = T_{num8}}$	
$\boldsymbol{T_y = T_{num3},}$	
$\boldsymbol{T_0 = T_{if}}$	

$T_{if} = T_{\#t} \circ Substitution = (T_{if} = T_{\#t}). T_{\#t} = T_{\#f} \circ Substitution = (T_{\#t} = T_{\#f}).$
$Substitution = Substitution \circ (T_{if} = T_{\#t}) \circ (T_{\#t} = T_{\#f}).$

Expression	Substitution
~~$T_{if} = T_{\#t}, T_{\#t} = T_{\#f}$~~	$T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$
$\boldsymbol{T_> = [T_x * T_y \rightarrow T_2]}$	$\boldsymbol{T_{\#t} = T_{\#f}}$
$T_> \rightarrow [N * N -> B]$	$\boldsymbol{T_{if} = T_{\#t}}$
$T_{\#f} = B$	
$T_{\#t} = B$	
$T_{num8} = N$	
$T_{num3} = N$	
$T_x = T_{num8}$	
$T_y = T_{num3},$	
$T_0 = T_{if}$	

$T_> = [T_x * T_y \rightarrow T_2] \circ Substitution = (T_> = [T_x * T_y \rightarrow T_2]).$
$Substitution = Substitution \circ (T_> = [T_x * T_y \rightarrow T_2]).$

Expression	Substitution
~~$T_> = [T_x * T_y \rightarrow T_2]$~~	$T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$
$T_> \rightarrow [N * N -> B]$	$T_{\#t} = T_{\#f}$
$T_{\#f} = B$	$T_{if} = T_{\#t}$
$T_{\#t} = B$	$\boldsymbol{T_> = [T_x * T_y \rightarrow T_2]}$
$T_{num8} = N$	
$T_{num3} = N$	
$T_x = T_{num8}$	
$T_y = T_{num3}$,	
$T_0 = T_{if}$	

$T_> = [T_x * T_y \rightarrow T_2] \circ Substitution = (T_> = [T_x * T_y \rightarrow T_2] = T_> \rightarrow [N * N -> B]) => T_x = N, T_y = N, T_2 = B.$

Expression	Substitution
~~$T_> \rightarrow [N * N -> B]$~~	$T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$
$T_{\#f} = B$	$T_{\#t} = T_{\#f}$
$T_{\#t} = B$	$T_{if} = T_{\#t}$
$T_{num8} = N$	$T_> = [T_x * T_y \rightarrow T_2]$
$T_{num3} = N$	
$T_x = T_{num8}$	
$T_y = T_{num3}$,	
$T_0 = T_{if}$	
$\boldsymbol{T_x = N}$	
$\boldsymbol{T_y = N}$	
$\boldsymbol{T_2 = B}$	

$T_{\#f} = B \circ Substitution = (T_{\#f} = B).\ T_{\#t} = B \circ Substitution = (T_{\#t} = B).$
$Substitution = Substitution \circ (T_{\#t} = B) \circ (T_{\#f} = B).$

Expression	Substitution
~~$T_> \rightarrow [N * N -> B]$~~	$T_1 = [T_{num8} * T_{num3} \rightarrow T_0]$
~~$T_{\#f} = B$~~	$T_{\#t} = \boldsymbol{B}$
~~$T_{\#t} = B$~~	$T_{if} = \boldsymbol{B}$
$T_{num8} = N$	$T_> = [T_x * T_y \rightarrow T_2]$
$T_{num3} = N$	$\boldsymbol{T_{\#f} = B}$
$T_x = T_{num8}$	$\boldsymbol{T_{\#t} = B}$
$T_y = T_{num3}$,	
$T_0 = T_{if}$	
$T_x = N$	
$T_y = N$	
$T_2 = B$	

$T_{num8} = N \circ Substitution = (T_{num8} = N).\ T_{num3} = N \circ Substitution = (T_{num3} = N).$
$Substitution = Substitution \circ (T_{num3} = N) \circ (T_{num8} = N).$

Expression	Substitution

~~$T_{num8} = N$~~	$T_1 = [\boldsymbol{N} * \boldsymbol{N} \rightarrow T_0]$
~~$T_{num3} = N$~~	$T_{\#t} = B$
$T_x = T_{num8}$	$T_{if} = B$
$T_y = T_{num3}$	$T_> = [T_x * T_y \rightarrow T_2]$
$T_0 = T_{if}$	$T_{\#f} = B$
$T_x = N$	$T_{\#t} = B$
$T_y = N$	$\boldsymbol{T_{num8} = N}$
$T_2 = B$	$\boldsymbol{T_{num3} = N}$

$T_x = T_{num} \circ Substitution = (T_x = N)$. $T_y = T_{num3} \circ Substitution = (T_y = N)$.
$Substitution = Substitution \circ (T_x = N) \circ (T_y = N)$.

Expression	Substitution
~~$T_x = T_{num8} \rightarrow T_x = N$~~	$T_1 = [N * N \rightarrow T_0]$
~~$T_y = T_{num3}, \rightarrow T_y = N$~~	$T_{\#t} = B$
$T_0 = T_{if}$	$T_{if} = B$
~~$T_x = N$~~	$T_> = [\boldsymbol{N} * \boldsymbol{N} \rightarrow T_2]$
~~$T_y = N$~~	$T_{\#f} = B$
$T_2 = B$	$T_{\#t} = B$
	$T_{num8} = N$
	$T_{num3} = N$
	$\boldsymbol{T_x = N}$
	$\boldsymbol{T_y = N}$

$T_0 = T_{if} \circ Substitution = \left(T_0 = T_{if} = B\right)$.
$Substitution = Substitution \circ (T_0 = B)$.

Expression	Substitution
~~$T_0 = T_{if}$~~	$T_1 = [N * N \rightarrow \boldsymbol{B}]$
$T_2 = B$	$T_{\#t} = B$
	$T_{if} = B$
	$T_> = [N * N \rightarrow T_2]$
	$T_{\#f} = B$
	$T_{\#t} = B$
	$T_{num8} = N$
	$T_{num3} = N$
	$T_x = N$
	$T_y = N$
	$\boldsymbol{T_0 = B}$

$T_2 = B \circ Substitution = (T_2 = B)$.
$Substitution = Substitution \circ (T_2 = B)$.

Expression	Substitution

~~$T_2 = B$~~	$T_1 = [N * N \rightarrow T_0]$
	$T_{\#t} = B$
	$T_{if} = B$
	$T_{>} = [N * N \rightarrow \boldsymbol{B}]$
	$T_{\#f} = B$
	$T_{\#t} = B$
	$T_{num8} = N$
	$T_{num3} = N$
	$T_x = N$
	$T_y = N$
	$T_2 = B$

Conclusion: ((lambda (x y) (if (> x y) #t #f)) 8 3) = T0 = Boolean

2:

a. **{f:[T1->T2], x: T1} |- (f x): T2 – True.**
The function f gets a parameter of type T1, and its return value is of type T2. x type is T1, therefore x is a valid parameter for f, f returns T2 so (f x): T2 is a true statement.

b. **{f:[T1->T2],g: [T2->T3]}, x: T2 |- (f g x): T3 – False.**
The function f gets only one parameter of type T1, but in the statement (f g x): T3 f receives two parameters.

c. **{f:[T2->T1],g: [T1->T2], x: T1} |- (f (g x)): T1 – True**
The function g gets a parameter of type T1, and its return value is of type T2. x type is T1, therefore x is a valid parameter for g, so (g x) is a true statement which returns a value of type T2. This value is received as a parameter to the function f, which gets a parameter of type T2 (so the returns value from g function is a valid parameter for f) and returns a value of type T1. So (f(g x)): T1 is a true statement.

d. **{f:[T2->Number], x: Number} |- (f x x): Number - False**
The function f gets only one parameter of type T2, but in the statement (f x x): Number f receives two parameters.

3:

a. cons – type: [T1*T2 -> pair(T1, T2)]
b. car – type: [pair(T1,T2) -> T1]
c. cdr – type: [pair(T1,T2) -> T2]

4:

(Define f (lambda (x) (values x x x))) – type: [T1->(T1*T1*T1)]

5:

a. {**T1, T2**} **- MGU**:
{T1=T2}

b. {**Number, Number**} **- MGU**:
{ } (no change is needed)

c. {**[T1*[T1->T2]->Number], [[T3->Number]*[T4->Number]->N]**} **- MGU**:
{T1 = [T3 ->Number], T4 = [T3 -> Number], T2 = Number}

d. {**[T1->T1], [T1->[Number->Number]**} **– MGU:**
{T1 = [Number -> Number]}

Part 2 question 2.3:

a. (define (f : [number -> number * number]) (lambda (x : number) : (number * number) (values x (+ x 1))))

b. (define (g : [T1 -> String * T1]) (lambda (x: T1) : (String * T1) (values "x" x)))

Part 4 question 4.b:

Using promises, we can achieve 3 main benefits:

• The type of functions returning Promises is more informative and similar to the simple types of synchronous versions.

• We can chain sequences of asynchronous calls in a chain of .then() calls.

• We can aggregate error handling in a single handler for a chain of calls, in a way similar to exception handling.