עבודת בית 4

מגישים: עומרי גולדברג 208938985, נעה חרדון 312164668

Part 1:

1.

a. $\{f : [T1 \to T2], g : [T1 \to T2], a : T1\} \vdash (f\ (g\ a)) : T2$ – false, the type of a is T1, g receives T1 and returns T2, f takes T1 as input and not T2.

b. $\{x : T1,\ y : T2,\ f : [T2 \to T1]\} \vdash (f\ y) : T1$ – true, f receives as input the type T2 like the parameter y type, and returns T1 as the final return.

c. $\{f : [T1 \to T2]\} \vdash (lambda\ (x)\ (f\ x)) : [T1 \to T2]$ – false, we don't have any information about the type of the parameter x therefore it can be of a type that doesn't match the input type of f, also the lambda returns T2 and not $[T1 \to T2]$.

d. $\{f : [T1 * T2 \to T3]\} \vdash (lambda\ (x)\ (f\ x\ 100)) : [T1 \to T3]$ – false, we don't have any information about the type of the parameter x therefore it can be of a type that doesn't match the input type of f, also the lambda returns T3 and not $[T1 \to T3]$.

2.

a. $((lambda\ (x1)\ (+\ x1\ 1))\ 4)$

Stage 1: rename bound variables –

$((lambda\ (x1)\ (+\ x1\ 1))\ 4) \to ((lambda\ (x)\ (+\ x\ 1))\ 4)$

Stage 2: 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 3: construct type equations

Expression	Equation
$((lambda\ (x)\ (+\ x\ 1))\ 4)$	$[Tnum4 \to T0]$
$(lambda\ (x)\ (+\ x\ 1))$	T1=[Tx →T2]
$(+\ x\ 1)$	T+ = $[Tx * Tnum1 \to T2]$
$+$	T+= $[Number * Number \to Number]$
1	Tnum1 = Number
4	Tnum4 = Number

Stage 4: solve the equations

Equation	Substitution
T1 = $[Tnum4 \to T0]$	{ }
T1 = $[Tx \to T2]$	
T+ = $[Tx * Tnum1 \to T2]$	
T+= $[Number * Number \to Number]$	
Tnum1 = Number	
Tnum4 = Number	

Step 1: T1 = [Tnum4 → T0] ∘ substitution = (T1 = [Tnum4 → T0]) →
substitution = substitution ∘ (T1 = [Tnum4 → T0])

Equation	Substitution
T1 = $[Tx \rightarrow T2]$	{T1 = $[Tnum4 \rightarrow T0]$}
T+ = $[Tx * Tnum1 \rightarrow T2]$	
T+= $[Number * Number \rightarrow Number]$	
Tnum1 = Number	
Tnum4 = Number	

Step 2: T1 = [Tx → T2] = [Tnum4 → T0], *both sides should be splitted*.

Equation	Substitution
T+ = $[Tx * Tnum1 \rightarrow T2]$	{T1 = $[Tnum4 \rightarrow T0]$}
T+= $[Number * Number \rightarrow Number]$	
Tnum1 = Number	
Tnum4 = Number	
Tx = Tnum4	
T2 = T0	

Step 3: T+= [Tx ∗ $Tnum1$ → T2] ∘ $substitution$ = (T+= [$Tx * Tnum1 \rightarrow T2$]) →
$substitution = substitution \circ (T+= [Tx \cdot Tnum1 \rightarrow T2])$.

Equation	Substitution
T+= $[Number * Number \rightarrow Number]$	$\left\{\begin{matrix} T1 = [Tnum4 \rightarrow T0] \\ T+= [Tx * Tnum1 \rightarrow T2] \end{matrix}\right\}$
Tnum1 = Number	
Tnum4 = Number	
Tx = Tnum4	
T2 = T0	

Step 4: T+= [Number ∗ $Number$ → Number] = $[Tx * Tnum1 \rightarrow T2]$), *split it now*

Equation	Substitution
Tnum1 = Number	$\left\{\begin{matrix} T1 = [Tnum4 \rightarrow T0] \\ T+= [Tx * Tnum1 \rightarrow T2] \end{matrix}\right\}$
Tnum4 = Number	
Tx = Tnum4	
T2 = T0	
Tx = Number	
$T2$ = Number	

Step 5: substitution of Tnum1

Equation	Substitution
Tnum4 = Number	$\left\{\begin{array}{c} \mathrm{T1} = [Tnum4 \rightarrow T0] \\ T\mathrm{+}= [Tx * Number \rightarrow T2] \\ Tnum1 = Number \end{array}\right\}$
Tx = Tnum4	
T2 = T0	
Tx = Number	
$T2$ = Number	

Step 6: substitution of Tnum4

Equation	Substitution
Tx = Tnum4	$\left\{\begin{array}{c} \mathrm{T1} = [Number \rightarrow T0] \\ T\mathrm{+}= [Tx * Number \rightarrow T2] \\ Tnum1 = Number \\ Tnum4 = Number \end{array}\right\}$
T2 = T0	
Tx = Number	
$T2$ = Number	

Step 7: substitution of Tx, *then substitution of each equation that includes* Tx

Equation	Substitution
T2 = T0	$\left\{\begin{array}{c} \mathrm{T1} = [Number \rightarrow T0] \\ T\mathrm{+}= [Tx * Number \rightarrow T2] \\ Tnum1 = Number \\ Tnum4 = Number \\ Tx = Number \end{array}\right\}$
Tx = Number	
$T2$ = Number	

Step 8: substitution of T2

Equation	Substitution
Tx = Number	$\left\{\begin{array}{c} \mathrm{T1} = [Number \rightarrow T0] \\ T\mathrm{+}= [Tx * Number \rightarrow T2] \\ Tnum1 = Number \\ Tnum4 = Number \\ Tx = Number \\ T2 = T0 \end{array}\right\}$
$T2$ = Number	

Step 9: substitution of Tx *with* $Number$

Equation	Substitution
$T2$ = Number	$\left\{\begin{array}{c} \mathrm{T1} = [Number \rightarrow T0] \\ T\mathrm{+}= [Tx * Number \rightarrow T2] \\ Tnum1 = Number \\ Tnum4 = Number \\ Tx = Number \\ T2 = T0 \end{array}\right\}$

Step 10: $substitution\ of\ T2\ with\ T0,\ T0 = Number, substitution\ the\ equations\ with\ T0\ with\ Number$

Equation	Substitution
	$\left\{\begin{array}{c} T1 = [Number \to Number] \\ T+= [Tx * Number \to Number] \\ Tnum1 = Number \\ Tnum4 = Number \\ Tx = Number \\ T2 = T0 \\ T0 = Number \end{array}\right\}$

In conclusion the type inference succeeds, the inferred type of T0 is: Number

b. $((lambda\ (f1\ x1)\ (f1\ x1\ 1))\ 4\ +)$

Stage 1: rename bound variables –

$((lambda\ (f1\ x1)\ (f1\ x1\ 1))\ 4\ +) \to ((lambda\ (f\ x)\ (f\ x\ 1))\ 4\ +)$

Stage 2: assign type variables 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 3: construct type equations

Expression	Equation
$((lambda\ (f\ x)\ (f\ x\ 1))\ 4\ +)$	T1=[Tnum4 $*$ T $+* Tnum1 \to$T0]
$(lambda\ (f\ x)\ (f\ x\ 1))$	T1=[$Tf *$ T$x * Tnum1 \to$T2]
$(f\ x\ 1)$	$Tf = [Tx * Tnum1 \to T2]$
1	Tnum1 = Number
4	Tnum4 = Number
+	T+= $[Number * Number \to Number]$

Stage 4: solve the equations

Equation	Substitution
T1=[Tnum4 $*$ T $+* Tnum1 \to$T0]	{ }
T1=[$Tf *$ T$x * Tnum1 \to$T2]	
$Tf = [Tx * Tnum1 \to T2]$	
Tnum1 = Number	
Tnum4 = Number	
T+= $[Number * Number \to Number]$	

Step 1: T1 = [Tnum4 ∗ T +∗ $Tnum1$ → T0] ∘ substitution =
(T1 = [Tnum4 ∗ T +∗ $Tnum1$ → T0]) →
substitution = substitution ∘ (T1 = [Tnum4 ∗ T +∗ $Tnum1$ → T0])

Equation	Substitution
T1=[Tf ∗ Tx ∗ $Tnum1$ →T2]	{T1 = [Tnum4 ∗ T +∗ $Tnum1$ → T0]}
Tf = [Tx ∗ $Tnum1$ → $T2$]	
Tnum1 = Number	
Tnum4 = Number	
T+= [$Number$ ∗ $Number$ → $Number$]	

Step 2: T1 = [Tnum4 ∗ T +∗ $Tnum1$ → T0] = [Tf ∗ Tx → T2], *both sides should be splitted*.

Equation	Substitution
Tf = [Tx ∗ $Tnum1$ → $T2$]	{T1 = [Tnum4 ∗ T +∗ $Tnum1$ → T0]}
Tnum1 = Number	
Tnum4 = Number	
T+= [$Number$ ∗ $Number$ → $Number$]	
Tf = $Tnum4$	
Tx = T +	
$Tnum1$ = $Tnum1$	

Step 3: Tf = [Tx ∗ $Tnum1$ → T2] ∘ substitution = (Tf = [Tx ∗ $Tnum1$ → T2]) → substitution = substitution ∘ (Tf = [Tx ∗ $Tnum1$ → T2]).

Equation	Substitution
Tnum1 = Number	$\begin{Bmatrix} \text{T1 = [Tnum4 ∗ T +∗ } Tnum1 \text{ → T0]} \\ \text{T}f \text{ = [T}x \text{ ∗ } Tnum1 \text{ → T2]} \end{Bmatrix}$
Tnum4 = Number	
T+= [$Number$ ∗ $Number$ → $Number$]	
Tf = $Tnum4$	
Tx = T +	
$Tnum1$ = $Tnum1$	

Step 4: substitution of Tnum1

Equation	Substitution
Tnum4 = Number	$\begin{Bmatrix} T1 = [Tnum4 * T +* Number \rightarrow T0] \\ Tf = [Tx * Number \rightarrow T2] \\ Tnum1 = Number \end{Bmatrix}$
T+= [$Number$ ∗ $Number$ → $Number$]	
Tf = $Tnum4$	
Tx = T +	
$Tnum1$ = $Tnum1$	

Step 5: substitution of Tnum4

Equation	Substitution
T+= [$Number$ ∗ $Number$ → $Number$]	$\begin{Bmatrix} T1 = [Number * T +* Number \rightarrow T0] \\ Tf = [Tx * Number \rightarrow T2] \\ Tnum1 = Number \\ Tnum4 = Number \end{Bmatrix}$
Tf = $Tnum4$	
Tx = T +	
$Tnum1$ = $Tnum1$	

Step 6: substitution of T +

Equation	Substitution
$Tf = Tnum4$	$\left\{\begin{array}{c} T1 = [Number * [Number * Number \rightarrow Number] * Number \rightarrow T0] \\ Tf = [Tx * Number \rightarrow T2] \\ Tnum1 = Number \\ Tnum4 = Number \\ T+= [Number * Number \rightarrow Number] \end{array}\right\}$
$Tx = T +$	
$Tnum1 = Tnum1$	

Step 7: $Tf = Tnum4 = [Tx * Number \rightarrow T2]$, *number cannot be a composite expression.*

In conclusion the type inference failed, $(f\ x\ 1)\ =\ (4 + 1)$, isn't a legal procedure.

Question 2.2

b. the wrapped function returns Promise<R> because a promise represents the result of an asynchronous operation and in this function we want to know if we succeed to know the type or not.

Part 3:

Question 3.1

Typing rule define:
For every: type environment _Tenv,
variable _x1
expressions _e1 and
type expressions _S1, _U1:
If _Tenv |- _x1 : _S1
  _Tenv |- _e1 : _U1
  _Tenv |- _S1 = _U1
Then _Tenv |- (define _x1 _e1) : void

Typing rule set:
For every: type environment _Tenv,
variable _x1
expressions _e1 and
type expressions _S1:
If _Tenv |- _x1 : _S1
  _Tenv |- _e1 : _S1
Then _Tenv |- (set! _x1 _e1) : void