Tutorial-10

TA-Sinong

Quiz-9

QUIZ: GENERATING TYPES

Generate the polymorphic types for the following function:

```
fun fold (f, a, l) =
  case l of
        nil => a
       | h::t => fold (f, f (h, a), t)
```

Please show the intermedia steps and the equations that you are solving.

```
fun fold (f:A, a:B, l:C):D =
  case l of
     nil => a:B
   | h::t:E list => fold (f:A, f (h:E, a:B):((E*B)->B), t:E list)
```

D=B

C=E list

A=E*B->B

fold: ((E*B)->B)*B*(E list) -> B

Homework-9

Problem 1. Given the following variant of untyped lambda calculus:

```
e::=
x (variables)
| c (constants)
| \x.e
| e1 e2
| e1 bop e2 (binary op)
| uop e (unary op)
| let x = e1 in e2
| if e1 then e2 else e3
| letfun f(x) = e1 in e2  (defining a recursive function f(x) for use in e2)
| {e1, e2}
| e.1
| e.2
| inl e
| inr e
| case e1 of inl x => e2 | inr x => e3
| nil
| e1 :: e2
| case e1 of nil => e2 | x1 :: x2 => e3
| (e)
```

Problem-1

(a) Inductively define the constraint generation judgement:

```
G |- u ==> e:t, q
```

Solution.

$$\frac{G(x) = t}{G \vdash x \Rightarrow x : t, \{\}} \qquad (CT - Var)$$

$$\frac{}{G \vdash c \Rightarrow c : int, \{\}} (c\ is\ integer) \qquad (CT - Int)$$

$$\frac{}{G \vdash c \Rightarrow c : bool, \{\}} (c\ is\ true\ or\ false) \qquad (CT - Bool)$$

$$\frac{G, x : t_1 \vdash t \Rightarrow t : t_2, q}{G, \lambda x : t_1.t : t_1 \rightarrow a,\ \ q \cup \{a = t_2\}} \qquad (CT - Abs)$$

Problem-1

$$\frac{G \vdash u_1 \Rightarrow e_1 : t_1, q_1 \quad G \vdash u_2 \Rightarrow e_2 : t_2, q_2}{G \vdash u_1\ u_2 \Rightarrow e_1\ e_2 : a,\ q_1 \cup q_2 \cup \{t_1 = t_2 \rightarrow a\}} \qquad (CT - App)$$

$$\frac{G \vdash u_1 \Rightarrow e_1 : t_1, q_1 \quad G \vdash u_2 \Rightarrow e_2 : t_2, q_2}{G \vdash u_1\ bop\ u_2 \Rightarrow e_1\ bop\ e_2 : a,\ q_1 \cup q_2 \cup \{t_1 = t_2 = a\}} \qquad (CT - Bop)$$

$$\frac{G \vdash u \Rightarrow e : t, q}{G \vdash uop\ u \Rightarrow uop\ e : a,\ q \cup \{t = a\}} \qquad (CT - Uop)$$

$$\frac{G \vdash u_1 \Rightarrow e_1 : t_1, q_1 \quad G \vdash u_2 \Rightarrow e_2 : t_2, q_2}{G \vdash let\ x = u_1\ in\ u_2 \Rightarrow let\ x = e_1\ in\ e_2 : a,\ q_1 \cup q_2 \cup \{t_2 = a\}} \qquad (CT - Let)$$

$$\frac{G \vdash u_1 \Rightarrow e_1 : t_1, q_1 \quad G \vdash u_2 \Rightarrow e_2 : t_2, q_2, \quad G \vdash u_2 \Rightarrow e_2 : t_2, q_2}{\begin{array}{c} G \vdash if\ u_1\ then\ u_2\ else\ u_3 \Rightarrow if\ e_1\ then\ e_2\ else\ e_3 : a, \\ q_1 \cup q_2 \cup q_3 \cup \{t_1 = bool, t_2 = t_3 = a\} \end{array}} \qquad (CT - If)$$

$$\frac{G, f : a \rightarrow b, x : a \vdash u_1 \Rightarrow e_1 : t_1, q_1\ G, f(x) : b \vdash u_2 \Rightarrow e_2 : t_2, q_2}{\begin{array}{c} G \vdash letfun\ f(x) = u_1\ in\ u_2 \Rightarrow letfun\ f(x : a) : b = e_1\ in\ e_2 : c, \\ q_1 \cup q_2 \cup q_3 \cup \{t_1 = b, t_2 = c\} \end{array}} \qquad (CT - Letfun)$$

$$\frac{G \vdash u_1 \Rightarrow e_1 : t_1, q_1 \quad G \vdash u_2 \Rightarrow e_2 : t_2, q_2}{G \vdash \{u_1, u_2\} \Rightarrow \{e_1, e_2\} : a * b,\ q_1 \cup q_2 \cup \{t_1 = a, t_2 = b\}} \qquad (CT - Pair)$$

Problem-1

$$\frac{G \vdash u \Rightarrow \{e_1, e_2\} : t_1 * t_2, q}{G \vdash u.1 \Rightarrow e_1 : a,\ q \cup \{t_1 = a\}} \quad (CT - Proj1)$$

$$\frac{G \vdash u \Rightarrow \{e_1, e_2\} : t_1 * t_2, q}{G \vdash u.2 \Rightarrow e_2 : a,\ q \cup \{t_2 = a\}} \quad (CT - Proj2)$$

$$\frac{G \vdash u \Rightarrow e : t, q}{G \vdash inl[a+b]\ u \Rightarrow inl[a+b]\ e : a+b,\ q \cup \{t = a\}} \quad (CT - Inl)$$

$$\frac{G \vdash u \Rightarrow e : t, q}{G \vdash inr[a+b]\ u \Rightarrow inr[a+b]\ e : a+b,\ q \cup \{t = b\}} \quad (CT - Inr)$$

$$\frac{G \vdash u \Rightarrow e : t_1 + t_2, q_1\ G, x_1 : t_1 \vdash u_1 \Rightarrow e_1 : t, q_2\ G, x_2 : t_2 \vdash u_2 \Rightarrow e_2 : t, q_3}{\begin{array}{l} G \vdash (case\ u\ of\ inl\ x_1 \Rightarrow u_1 | inr\ x_2 \Rightarrow u_2) \Rightarrow (case\ e : a+b\ of\ inl\ x_1 \Rightarrow e_1 | \\ \qquad inr\ x_2 \Rightarrow e_2) : c, q_1 \cup q_2 \cup q_3 \cup \{t_1 = a, t_2 = b, t = c\} \end{array}} \quad (CT - Case)$$

$$\frac{}{G \vdash u \Rightarrow nil[t] : t\ list,} \quad (CT - Ni)$$

$$\frac{G \vdash u_1 \Rightarrow e_1 : t, q_1 \quad G \vdash u_2 \Rightarrow e_2 : t\ list, q_2}{G \vdash u_1 :: u_2 \Rightarrow e_1 :: e_2 : a,\ q_1 \cup q_2 \cup \{a = t\ list\}} \quad (CT - Cons)$$

$$\frac{G \vdash u \Rightarrow e : t_1\ list, q_1\ G \vdash u_1 \Rightarrow e_1 : t, q_2\ G, x_1 : t, x_2 : t_1\ list \vdash u_2 \Rightarrow e_2 : t, q_3}{\begin{array}{l} G \vdash (case\ u\ of\ nil[a] \Rightarrow u_1 | x_1 :: x_2 \Rightarrow u_2) \Rightarrow (case\ e : a\ list\ of\ nil[a] \Rightarrow e_1 | \\ \qquad x_1 :: x_2 \Rightarrow e_2) : b, q_1 \cup q_2 \cup q_3 \cup \{t_1 = a, t = b\} \end{array}} \quad (CT - Casel)$$

$$\frac{G \vdash u \Rightarrow e : t, q}{G \vdash u \Rightarrow e : unit, q \cup \{t = unit\}} \quad (CT - Unit)$$

Problem-1

(b) Give the detailed derivation of the following expressions and obtain the set of equations, then solve these equations to get the principle solution and give the universal polymorphic types:

(1)
```
letfun sum(l) = case l of nil => 0 | x1 :: x2 => x1 + sum(x2)
in sum(12::10::0::nil)
```

Problem-1

Solution.
derivation:

$(letfun\ sum(l : a) = case\ l\ of\ nil : b\ list => 0|x1 : c :: x2 : d => x1 + sum(x2) : d \to e$
$in\ sum(12 :: 10 :: 0 :: nil), \{\})$
(by CT-Bop)
$\to(letfun\ sum(l : a) = case\ l\ of\ nil : b\ list => 0|x1 : c :: x2 : d => (x1 + sum(x2)) : f$
$in\ sum(12 :: 10 :: 0 :: nil), \{d = a, c = e, f = c\}$
(by CT-Casel)
$\to(letfun\ sum(l : a) = (case\ l\ of\ nil\ list => 0|x1 :: x2 => x1 + sum(x2)) : g$
$in\ sum(12 :: 10 :: 0 :: nil) : e,$
$\{d = a, c = e, f = c, a = b\ list, c = int, f = int, int = g\}$
(by CT-Letfun)
$\to((letfun\ sum(l : a) = (case\ l\ of\ nil\ list => 0|x1 :: x2 => x1 + sum(x2)) : g$
$in\ sum(12 :: 10 :: 0 :: nil) : e)h,$
$\{d = a, c = e, f = c, a = b\ list, c = int, f = int, int = g, g = e, h = e\}$
(by CT-Cons)
$\to(letfun\ sum(l : a) = (case\ l\ of\ nil\ list => 0|x1 :: x2 => x1 + sum(x2)) : g$
$in\ sum(12 :: 10 :: 0 :: nil) : e,$
$\{d = a, c = e, f = c, a = b\ list, c = int, f = int, int = g, g = e, h = e, d = int\ list\}$

Problem-1

solve constraint set:

$$(I, \{d = a, c = e, f = c, a = b\ list, c = int, f = int, int = g, g = e, h = e, d = int\ list\}$$
$$\rightarrow([d = a] \circ I, \{c = e, f = c, a = b\ list, c = int, f = int, int = g, g = e, h = e, a = int\ list\})$$
$$\rightarrow([c = e] \circ [d = a] \circ I, \{f = e, a = b\ list, e = int, f = int, int = g, g = e, h = e, a = int\ list\})$$
$$\rightarrow([f = e] \circ [c = e] \circ [d = a] \circ I, \{a = b\ list, e = int, int = g, g = e, h = e, a = int\ list\})$$
$$\rightarrow([a = b\ list] \circ [f = e] \circ [c = e] \circ [d = a] \circ I, \{e = int, int = g, g = e, h = e, b\ list = int\ list\})$$
$$\rightarrow([e = int] \circ [a = b\ list] \circ [f = e] \circ [c = e] \circ [d = a] \circ I, \{int = g, h = int, b\ list = int\ list\})$$
$$\rightarrow([g = int] \circ [e = int] \circ [a = b\ list] \circ [f = e] \circ [c = e] \circ [d = a] \circ I, \{h = int, b\ list = int\ list\})$$
$$\rightarrow([h = int] \circ [g = int] \circ [e = int] \circ [a = b\ list] \circ [f = e] \circ [c = e] \circ [d = a] \circ I, \{b\ list = int\ list\})$$
$$\rightarrow([b = int] \circ [h = int] \circ [g = int] \circ [e = int] \circ [a = b\ list] \circ [f = e] \circ [c = e] \circ [d = a] \circ I, \{\})$$

pincipal solution: S(b)=S(c)=S(e)=S(f)=S(g)=S(h)int, S(d)=S(a)int list

universal polymorphic types:

$$letfun\ sum(l : int\ list) : int = case\ l\ of\ nil : int\ list => 0 | x1 : int :: x2 : int => x1 + sum(x2)$$
$$in\ sum(12 :: 10 :: 0 :: nil)$$
$$: int$$

Problem-1

(2)
```
let x = inr (5::4::3) in
case x of inl y => y.1 + y.2 |
inr y => (case y of nil => 0 | h::l => h)
```

Problem-1

$$(let\ x : b = inr(5 :: 4 :: 3)\ in$$
$$case\ x : b\ of\ inl\ y : c \Rightarrow y.1 + y.2|$$
$$inr\ y' : d \Rightarrow (case\ y'\ of\ nil : e\ list \Rightarrow 0|h : f :: l : g \Rightarrow h), \{\})$$

(by CT-Proj1 CT-Proj2 and CT-Bop)

$$\rightarrow (let\ x : b = inr(5 :: 4 :: 3)\ in$$
$$case\ x : b\ of\ inl\ y : c \Rightarrow (y.1 + y.2) : j|$$
$$inr\ y' : d \Rightarrow (case\ y'\ of\ nil : e\ list \Rightarrow 0|h : f :: l : g \Rightarrow h), \{c = c_1 * c_2, c_1 = c_2, j = c_1\})$$

(by CT-Casel)

$$\rightarrow (let\ x : b = inr(5 :: 4 :: 3)\ in$$
$$case\ x : b\ of\ inl\ y : c \Rightarrow (y.1 + y.2) : j|$$
$$inr\ y' : d \Rightarrow (case\ y'\ of\ nil : e\ list \Rightarrow 0|h : f :: l : g \Rightarrow h) : k,$$
$$\{c = c_1 * c_2, c_1 = c_2, j = c_1, d = e\ list, g = e\ list, f = int, k = int\})$$

Problem-1

CT-Case

$$\rightarrow(let\ x : b = inr(5 :: 4 :: 3)\ in$$
$$case\ x : b\ of\ inl\ y : c \Rightarrow (y.1 + y.2) : j|$$
$$inr\ y' : d \Rightarrow (case\ y'\ of\ nil : e\ list \Rightarrow 0|h : f :: l : g \Rightarrow h) : k,$$
$$\{c = c_1 * c_2, c_1 = c_2, j = c_1, d = e\ list, g = e\ list, f = int, k = int, b = n + m,$$
$$c = n, d = m, j = k\})$$

CT-inr

$$\rightarrow(let\ x : b = inr(5 :: 4 :: 3)\ in$$
$$case\ x : b\ of\ inl\ y : c \Rightarrow (y.1 + y.2) : j|$$
$$inr\ y' : d \Rightarrow (case\ y'\ of\ nil : e\ list \Rightarrow 0|h : f :: l : g \Rightarrow h) : k,$$
$$\{c = c_1 * c_2, c_1 = c_2, j = c_1, d = e\ list, g = e\ list, f = int, k = int, b = n + m,$$
$$c = n, d = m, j = k, m = int\ list\})$$

CT-Let

$$\rightarrow((let\ x : b = inr(5 :: 4 :: 3)\ in$$
$$case\ x : b\ of\ inl\ y : c \Rightarrow (y.1 + y.2) : j|$$
$$inr\ y' : d \Rightarrow (case\ y'\ of\ nil : e\ list \Rightarrow 0|h : f :: l : g \Rightarrow h) : k) : o,$$
$$\{c = c_1 * c_2, c_1 = c_2, j = c_1, d = e\ list, g = e\ list, f = int, k = int, b = n + m,$$
$$c = n, d = m, j = k, m = int\ list, o = k\})$$

Problem-1

solve constraint set:

$$
\begin{aligned}
&(I, \{c = c_1 * c_2, c_1 = c_2, j = c_1, d = e\ list, g = e\ list, f = int, k = int, b = n + m,\\
&c = n, d = m, j = k, m = int\ list, o = k\})\\
\rightarrow&([c = c_1 * c_2] \circ I, \{c_1 = c_2, j = c_1, d = e\ list, g = e\ list, f = int, k = int, b = n + m,\\
&c_1 * c_2 = n, d = m, j = k, m = int\ list, o = k\})\\
\rightarrow&([c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{j = c_2, d = e\ list, g = e\ list, f = int, k = int, b = n + m,\\
&c_2 * c_2 = n, d = m, j = k, m = int\ list, o = k\})\\
\rightarrow&([j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{d = e\ list, g = e\ list, f = int, k = int, b = n + m,\\
&c_2 * c_2 = n, d = m, c_2 = k, m = int\ list, o = k\})\\
\rightarrow&([d = e\ list] \circ [j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{g = e\ list, f = int, k = int, b = n + m,\\
&c_2 * c_2 = n, e\ list = m, c_2 = k, m = int\ list, o = k\})\\
\rightarrow&([g = e\ list] \circ [d = e\ list] \circ [j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{f = int, k = int, b = n + m,\\
&c_2 * c_2 = n, e\ list = m, c_2 = k, m = int\ list, o = k\})\\
\rightarrow&([f = int] \circ [g = e\ list] \circ [d = e\ list] \circ [j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{k = int,\\
&b = n + m, c_2 * c_2 = n, e\ list = m, c_2 = k, m = int\ list, o = k\})\\
\rightarrow&([k = int] \circ [f = int] \circ [g = e\ list] \circ [d = e\ list] \circ [j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I,\\
&\{b = n + m, c_2 * c_2 = n, e\ list = m, c_2 = int, m = int\ list, o = int\})\\
\rightarrow&([b = n + m] \circ [k = int] \circ [f = int] \circ [g = e\ list] \circ [d = e\ list] \circ [j = c_2] \circ [c_1 = c_2] \circ\\
&[c = c_1 * c_2] \circ I, \{c_2 * c_2 = n, e\ list = m, c_2 = int, m = int\ list, o = int\})
\end{aligned}
$$

Problem-1

$$\rightarrow([n = c_2 * c_2] \circ [b = n + m] \circ [k = int] \circ [f = int] \circ [g = e\ list] \circ [d = e\ list] \circ [j = c_2] \circ$$
$$[c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{e\ list = m, c_2 = int, m = int\ list, o = int\})$$
$$\rightarrow([m = e\ list][n = c_2 * c_2] \circ [b = n + m] \circ [k = int] \circ [f = int] \circ [g = e\ list] \circ [d = e\ list] \circ$$
$$[j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{c_2 = int, e\ list = int\ list, o = int\})$$
$$\rightarrow([c_2 = int] \circ [m = e\ list][n = c_2 * c_2] \circ [b = n + m] \circ [k = int] \circ [f = int] \circ [g = e\ list] \circ$$
$$[d = e\ list] \circ [j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{e\ list = int\ list, o = int\})$$
$$\rightarrow([e = int] \circ [c_2 = int] \circ [m = e\ list][n = c_2 * c_2] \circ [b = n + m] \circ [k = int] \circ [f = int] \circ$$
$$[g = e\ list] \circ [d = e\ list] \circ [j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{o = int\})$$
$$\rightarrow([o = int] \circ [e = int] \circ [c_2 = int] \circ [m = e\ list][n = c_2 * c_2] \circ [b = n + m] \circ [k = int] \circ$$
$$[f = int] \circ [g = e\ list] \circ [d = e\ list] \circ [j = c_2] \circ [c_1 = c_2] \circ [c = c_1 * c_2] \circ I, \{\})$$

pincipal solution: S(o)=S(e)=S(c_2)=S(k)=S(f)=S(j)=S(c_1)=int, S(m)=S(g)=S(d)=int list, S(n)=S(c)=int*int, S(b)=int*int+int list

universal polymorphic types:

$$let\ x : int * int + int\ list = inr(5 :: 4 :: 3)\ in$$
$$case\ x\ of\ inl\ y : int * int \Rightarrow y.1 + y.2|$$
$$inr\ y' : int\ list \Rightarrow (case\ y'\ of\ nil : int\ list \Rightarrow 0|h : int :: l : int\ list \Rightarrow h)$$
$$: int$$

Problem-1

(3)
```
let x = inl {3::2::1, nil} in
case x of inl y => (if y.2 == nil
then case y.1 of nil => 0 | h::l => h
else 0)
inr y => (case y of nil => 0 | h::l => h)
```

Problem-1

Solution.
derivation:

$$(let\ x : a = inl\ \{3 :: 2 :: 1, nil : b\ list\}\ in$$
$$case\ x\ of\ inl\ y : c => (if\ y.2 == nil$$
$$then\ case\ y.1\ of\ nil => 0 | h : d :: l : e => h$$
$$else\ 0)$$
$$inr\ y' : f => (case\ y\ of\ nil => 0 | h' : d' :: l' : e' => h), \{\})$$

CT-Case and CT-Proj1 and CT-cons

$$\rightarrow (let\ x : a = inl\ \{3 :: 2 :: 1, nil : b\ list\}\ in$$
$$case\ x\ of\ inl\ y : c => (if\ y.2 == nil$$
$$then\ (case\ y.1\ of\ nil => 0 | h : d :: l : e => h) : t_1$$
$$else\ 0)$$
$$inr\ y' : f => (case\ y\ of\ nil => 0 | h' : d' :: l' : e' => h), \{c = c_1 * c_2, c_1 = b\ list$$
$$, e = d\ list, c_1 = d\ list, d = int, t_1 = int\})$$

CT-If and CT-Bop

$$\rightarrow (let\ x : a = inl\ \{3 :: 2 :: 1, nil : b\ list\}\ in$$
$$case\ x\ of\ inl\ y : c => (if\ y.2 == nil$$
$$then\ (case\ y.1\ of\ nil => 0 | h : d :: l : e => h) : t_1$$
$$else\ 0) : t_2$$
$$inr\ y' : f => (case\ y\ of\ nil => 0 | h' : d' :: l' : e' => h), \{c = c_1 * c_2, c_1 = b\ list$$
$$, e = d\ list, c_1 = d\ list, d = int, t_1 = int, c_2 = b\ list, t_2 = int\})$$

Problem-1

CT-Case and CT-cons

$\rightarrow (let\ x : a = inl\ \{3 :: 2 :: 1, nil : b\ list\}\ in$

$case\ x\ of\ inl\ y : c => (if\ y.2 == nil$

$then\ (case\ y.1\ of\ nil => 0 | h : d :: l : e => h) : t_1$

$else\ 0) : t_2$

$inr\ y' : f => (case\ y\ of\ nil => 0 | h' :: l' => h) : t_3, \{c = c_1 * c_2, c_1 = b\ list, e = d\ list$

$, c_1 = d\ list, d = int, t_1 = int, c_2 = b\ list, t_2 = int, f = b\ list, e' = d'\ list, d' = int,$

$d'\ list = b\ list, t_3 = int\})$

CT-Case

$\rightarrow (let\ x : a = inl\ \{3 :: 2 :: 1, nil : b\ list\}\ in$

$(case\ x\ of\ inl\ y : c => (if\ y.2 == nil$

$then\ (case\ y.1\ of\ nil => 0 | h : d :: l : e => h) : t_1$

$else\ 0) : t_2$

$inr\ y' : f => (case\ y\ of\ nil => 0 | h' :: l' => h) : t_3) : t_4, \{c = c_1 * c_2, c_1 = b\ list, e = d\ list$

$, c_1 = d\ list, d = int, t_1 = int, c_2 = b\ list, t_2 = int, f = b\ list, e' = d'\ list, d' = int,$

$d'\ list = b\ list, t_3 = int, a = a_1 + a_2, a_1 = c, a_2 = f, t_2 = t_3, t_4 = t_2\})$

Problem-1

CT-inl and CT-Cons

$\rightarrow$$(let\ x : a = inl\ \{3 :: 2 :: 1, nil : b\ list\}\ in$
$(case\ x\ of\ inl\ y : c => (if\ y.2 == nil$
$then\ (case\ y.1\ of\ nil => 0 | h : d :: l : e => h) : t_1$
$else\ 0) : t_2$
$inr\ y' : f => (case\ y\ of\ nil => 0 | h' :: l' => h) : t_3) : t_4, \{c = c_1 * c_2, c_1 = b\ list, e = d\ list$
$, c_1 = d\ list, d = int, t_1 = int, c_2 = b\ list, t_2 = int, f = b\ list, e' = d'\ list, d' = int,$
$d'\ list = b\ list, t_3 = int, a = a_1 + a_2, a_1 = c, a_2 = f, t_2 = t_3, t_4 = t_2, a_1 = int\ list * b\ list\})$

CT-Let

$\rightarrow$$((let\ x : a = inl\ \{3 :: 2 :: 1, nil : b\ list\}\ in$
$(case\ x\ of\ inl\ y : c => (if\ y.2 == nil$
$then\ (case\ y.1\ of\ nil => 0 | h : d :: l : e => h) : t_1$
$else\ 0) : t_2$
$inr\ y' : f => (case\ y\ of\ nil => 0 | h' :: l' => h) : t_3) : t_4) : t_5, \{c = c_1 * c_2, c_1 = b\ list,$
$e = d\ list, c_1 = d\ list, d = int, t_1 = int, c_2 = b\ list, t_2 = int, f = b\ list, e' = d'\ list, d' = int,$
$d'\ list = b\ list, t_3 = int, a = a_1 + a_2, a_1 = c, a_2 = f, t_2 = t_3, t_4 = t_2, a_1 = int\ list * b\ list,$
$t_5 = t_4\})$

Problem-1

solve constraint set:

$$(I, \{c = c_1 * c_2, c_1 = b\ list, e = d\ list, c_1 = d\ list, d = int, t_1 = int, c_2 = b\ list, t_2 = int,$$
$$f = b\ list, e' = d'\ list, d' = int, d'\ list = b\ list, t_3 = int, a = a_1 + a_2, a_1 = c_1 * c_2, a_2 = f,$$
$$t_2 = t_3, t_4 = t_2, a_1 = int\ list * b\ list, t_5 = t_4\})$$

$$\rightarrow([c = c_1 * c_2] \circ I, \{c_1 = b\ list, e = d\ list, c_1 = d\ list, d = int, t_1 = int, c_2 = b\ list, t_2 = int,$$
$$f = b\ list, e' = d'\ list, d' = int, d'\ list = b\ list, t_3 = int, a = a_1 + a_2, a_1 = c_1 * c_2, a_2 = f,$$
$$t_2 = t_3, t_4 = t_2, a_1 = int\ list * b\ list, t_5 = t_4\})$$

$$\rightarrow([c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{e = d\ list, b\ list = d\ list, d = int, t_1 = int, c_2 = b\ list,$$
$$t_2 = int, f = b\ list, e' = d'\ list, d' = int, d'\ list = b\text{''}\ list, t_3 = int, a = a_1 + a_2, a_1 = b\ list * c_2$$
$$, a_2 = f, t_2 = t_3, t_4 = t_2, a_1 = int\ list * b\ list, t_5 = t_4\})$$

$$\rightarrow([e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{b\ list = d\ list, d = int, t_1 = int, c_2 = b\ list,$$
$$t_2 = int, f = b\ list, e' = d'\ list, d' = int, d'\ list = b\ list, t_3 = int, a = a_1 + a_2, a_1 = b'\ list * c_2$$
$$, a_2 = f, t_2 = t_3, t_4 = t_2, a_1 = int\ list * b\ list, t_5 = t_4\})$$

$$\rightarrow([b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{d = int, t_1 = int, c_2 = d\ list,$$
$$t_2 = int, f = d\ list, e' = d'\ list, d' = int, d'\ list = d\ list, t_3 = int, a = a_1 + a_2, a_1 = d\ list * c_2$$
$$, a_2 = f, t_2 = t_3, t_4 = t_2, a_1 = int\ list * d\ list, t_5 = t_4\})$$

$$\rightarrow([d = int] \circ [b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{t_1 = int, c_2 = int\ list,$$
$$t_2 = int, f = int\ list, e' = d'\ list, d' = int, d'\ list = int\ list, t_3 = int, a = a_1 + a_2,$$
$$a_1 = int\ list * c_2, a_2 = f, t_2 = t_3, t_4 = t_2, a_1 = int\ list * int\ list, t_5 = t_4\})$$

Problem-1

$\rightarrow([c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{$
$t_2 = int, f = int\ list, e' = d'\ list, d' = int, d'\ list = int\ list, t_3 = int, a = a_1 + a_2,$
$a_1 = int\ list * int\ list, a_2 = f, t_2 = t_3, t_4 = t_2, t_5 = t_4\})$

$\rightarrow([t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ$
$[c = c_1 * c_2] \circ I, \{$
$f = int\ list, e' = d'\ list, d' = int, d'\ list = int\ list, t_3 = int, a = a_1 + a_2, a_1 = int\ list * int\ list$
$, a_2 = f, int = t_3, t_4 = int, t_5 = t_4\})$

$\rightarrow([f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ [e = d\ list] \circ$
$[c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{$
$e' = d'\ list, d' = int, d'\ list = int\ list, t_3 = int, a = a_1 + a_2, a_1 = int\ list * int\ list$
$, a_2 = int\ list, int = t_3, t_4 = int, t_5 = t_4\})$

$\rightarrow([e' = d'\ list] \circ [f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ$
$[e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{$
$d' = int, d'\ list = int\ list, t_3 = int, a = a_1 + a_2, a_1 = int\ list * int\ list$
$, a_2 = int\ list, int = t_3, t_4 = int, t_5 = t_4\})$

Problem-1

$$\rightarrow([d' = int] \circ [e' = d'\ list] \circ [f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ$$
$$[b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{$$
$$int\ list = int\ list, t_3 = int, a = a_1 + a_2, a_1 = int\ list * int\ list$$
$$, a_2 = int\ list, int = t_3, t_4 = intt_5 = t_4\})$$
$$\rightarrow([d' = int] \circ [e' = d'\ list] \circ [f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ$$
$$[d = int] \circ [b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{$$
$$t_3 = int, a = a_1 + a_2, a_1 = int\ list * int\ list$$
$$, a_2 = int\ list, int = t_3, t_4 = int, t_5 = t_4\})$$
$$\rightarrow([t_3 = int] \circ [d' = int] \circ [e' = d'\ list] \circ [f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ$$
$$[d = int] \circ [b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{$$
$$a = a_1 + a_2, a_1 = int\ list * int\ list$$
$$, a_2 = int\ list, int = int, t_4 = int, t_5 = t_4\})$$
$$\rightarrow([a = a_1 + a_2] \circ [t_3 = int] \circ [d' = int] \circ [e' = d'\ list] \circ [f = int\ list] \circ [t_2 = int] \circ$$
$$[c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{$$
$$a_1 = int\ list * int\ list$$
$$, a_2 = int\ list, int = int, t_4 = int, t_5 = t_4\})$$

Problem-1

$$
\begin{aligned}
\rightarrow(&[a_1 = int\ list * int\ list] \circ [a = a_1 + a_2] \circ [t_3 = int] \circ [d' = int] \circ [e' = d'\ list] \circ [f = int\ list] \circ \\
&[t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ \\
&[c = c_1 * c_2] \circ I, \{a_2 = int\ list, int = int, t_4 = int, t_5 = t_4\}) \\
\rightarrow(&[a_2 = int\ list] \circ [a_1 = int\ list * int\ list] \circ [a = a_1 + a_2] \circ [t_3 = int] \circ [d' = int] \circ [e' = d'\ list] \circ \\
&[f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ [e = d\ list] \circ \\
&[c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{int = int, t_4 = int, t_5 = t_4\}) \\
\rightarrow(&[a_2 = int\ list] \circ [a_1 = int\ list * int\ list] \circ [a = a_1 + a_2] \circ [t_3 = int] \circ [d' = int] \circ \\
&[e' = d'\ list] \circ [f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ \\
&[e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{t_4 = int, t_5 = t_4\}) \\
\rightarrow(&[t_4 = int][a_2 = int\ list] \circ [a_1 = int\ list * int\ list] \circ [a = a_1 + a_2] \circ [t_3 = int] \circ [d' = int] \circ \\
&[e' = d'\ list] \circ [f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ [d = int] \circ [b = d] \circ \\
&[e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{t_5 = int\}) \\
\rightarrow(&[t_5 = int] \circ [t_4 = int][a_2 = int\ list] \circ [a_1 = int\ list * int\ list] \circ [a = a_1 + a_2] \circ \\
&[t_3 = int] \circ [d' = int] \circ [e' = d'\ list] \circ [f = int\ list] \circ [t_2 = int] \circ [c_2 = int\ list] \circ [t_1 = int] \circ \\
&[d = int] \circ [b = d] \circ [e = d\ list] \circ [c_1 = b\ list] \circ [c = c_1 * c_2] \circ I, \{\})
\end{aligned}
$$

Problem-1

pincipal solution:
S(a)=int list*int list+int list
S(b)=S(d)=S(d') = int
S(f)=S(e)=S(e') = int list

S(c)= int list*int list

universal polymorphic types:

$$let\ x : int\ list * int\ list + int\ list = inl\ \{3 :: 2 :: 1, nil : int\ list\}\ in$$
$$case\ x\ of\ inl\ y : int\ list * int\ list => (if\ y.2 == nil$$
$$then\ case\ y.1\ of\ nil => 0 | h : int :: l : int\ list => h$$
$$else\ 0)$$
$$inr\ y' : int\ list => (case\ y\ of\ nil => 0 | h' : int :: l' : int\ list => h)$$
$$: int$$

Problem-2

Lemma 1. *If a set of constraints q has a solution, then it has a most general one.*

Prove this lemma.

Proof.

Lemma 2. *If a set of constraint q has a solution, the unification algorithm always return the principal solution*

The proof is in the reference book *Types and Programming Languages*, page 328, 22.4.5

Then lemma 1 is proved. $\square$

Problem-2

THEOREM: The algorithm *unify* always terminates, failing when given a non-unifiable constraint set as input and otherwise returning a principal unifier. More formally:

1. *unify*(C) halts, either by failing or by returning a substitution, for all C;
2. if *unify*$(C) = \sigma$, then σ is a unifier for C;
3. if δ is a unifier for C, then *unify*$(C) = \sigma$ with $\sigma \sqsubseteq \delta$. □

Proof: For part (1), define the *degree* of a constraint set C to be the pair (m, n), where m is the number of distinct type variables in C and n is the total size of the types in C. It is easy to check that each clause of the *unify* algorithm either terminates immediately (with success in the first case or failure in the last) or else makes a recursive call to *unify* with a constraint set of lexicographically smaller degree.

Part (2) is a straightforward induction on the number of recursive calls in the computation of *unify*(C). All the cases are trivial except for the two involving variables, which depend on the observation that, if σ unifies $[\mathsf{X} \mapsto \mathsf{T}]D$, then $\sigma \circ [\mathsf{X} \mapsto \mathsf{T}]$ unifies $\{\mathsf{X} = \mathsf{T}\} \cup D$ for any constraint set D.

Part (3) again proceeds by induction on the number of recursive calls in the computation of *unify*(C). If C is empty, then *unify*(C) immediately returns the trivial substitution $[\,]$; since $\delta = \delta \circ [\,]$, we have $[\,] \sqsubseteq \delta$ as required. If C is non-empty, then *unify*(C) chooses some pair (S, T) from C and continues by cases on the shapes of S and T.

Problem-2

Case: $\quad \mathsf{S} = \mathsf{T}$

Since δ is a unifier for C, it also unifies C'. By the induction hypothesis, $\mathit{unify}(C) = \sigma$ with $\sigma \sqsubseteq \delta$, as required.

Case: $\quad \mathsf{S} = \mathsf{X}$ and $\mathsf{X} \notin FV(\mathsf{T})$

Since δ unifies S and T, we have $\delta(\mathsf{X}) = \delta(\mathsf{T})$. So, for any type U, we have $\delta(\mathsf{U}) = \delta([\mathsf{X} \mapsto \mathsf{T}]\mathsf{U})$; in particular, since δ unifies C', it must also unify $[\mathsf{X} \mapsto \mathsf{T}]C'$. The induction hypothesis then tells us that $\mathit{unify}([\mathsf{X} \mapsto \mathsf{T}]C') = \sigma'$, with $\delta = \gamma \circ \sigma'$ for some γ. Since $\mathit{unify}(C) = \sigma' \circ [\mathsf{X} \mapsto \mathsf{T}]$, showing that $\delta = \gamma \circ (\sigma' \circ [\mathsf{X} \mapsto \mathsf{T}])$ will complete the argument. So consider any type variable Y. If $\mathsf{Y} \neq \mathsf{X}$, then clearly $(\gamma \circ (\sigma' \circ [\mathsf{X} \mapsto \mathsf{T}]))\mathsf{Y} = (\gamma \circ \sigma')\mathsf{Y} = \delta\mathsf{Y}$. On the other hand, $(\gamma \circ (\sigma' \circ [\mathsf{X} \mapsto \mathsf{T}]))\mathsf{X} = (\gamma \circ \sigma')\mathsf{T} = \delta\mathsf{X}$, as we saw above. Combining these observations, we see that $\delta\mathsf{Y} = (\gamma \circ (\sigma' \circ [\mathsf{X} \mapsto \mathsf{T}]))\mathsf{Y}$ for all variables Y, that is, $\delta = (\gamma \circ (\sigma' \circ [\mathsf{X} \mapsto \mathsf{T}]))$.

Problem-2

Case: $\mathsf{T} = \mathsf{X}$ and $\mathsf{X} \notin FV(\mathsf{S})$

Similar.

Case: $\mathsf{S} = \mathsf{S}_1 \to \mathsf{S}_2$ and $\mathsf{T} = \mathsf{T}_1 \to \mathsf{T}_2$

Straightforward. Just note that δ is a unifier of $\{\mathsf{S}_1 \to \mathsf{S}_2 = \mathsf{T}_1 \to \mathsf{T}_2\} \cup C'$ iff it is a unifier of $C' \cup \{\mathsf{S}_1 = \mathsf{T}_1, \mathsf{S}_2 = \mathsf{T}_2\}$.

If none of the above cases apply to S and T, then *unify*(C) fails. But this can happen in only two ways: either S is `Nat` and T is an arrow type (or vice versa), or else $\mathsf{S} = \mathsf{X}$ and $\mathsf{X} \in \mathsf{T}$ (or vice versa). The first case obviously contradicts the assumption that C is unifiable. To see that the second does too, recall that, by assumption, $\delta\mathsf{S} = \delta\mathsf{T}$; if X occurred in T, then $\delta\mathsf{T}$ would always be strictly larger than $\delta\mathsf{S}$. Thus, if *unify*(C) fails, then C is not unifiable, contradicting our assumption that δ is a unifier for C; so this case cannot occur. □