CSE 3302/5307 Programming Language Concepts

Homework9 - Fall 2023

Due Date: Oct.28, 2023, 11:59p.m. Central Time

Problem1 - 60%

Given the following variant of untyped lambda calculus:

```
e::=
x (variables)
| c (constants)
| \x.e
| e1 e2
| e1 bop e2 (binary 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)
| 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)
```

(a) Inductively define the constraint generation judgement:

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

(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)
```

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

(3) ==BONUS POINT==

```
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)
```

Problem2 - 40%

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

Prove this lemma.