

CS 320 Theory Assignment #3

Due: November 4th, 11:59 pm

Problem 1 : polymorphism and high-order function

```
let f x = [([x], x)]
```

```
let rec foo (g, n) =
```

```
  match n with
```

```
  | 0 -> [(g 1, g 0)]
```

```
  | n -> (g n, g (n - 1)) :: foo (g, n - 1)
```

```
let res = foo(f, 2)
```

What is the type of `f` ? Please, explain your reasoning.

What is the type of `foo` ? Please, explain your reasoning.

What is the value of `res` ? Please, explain your reasoning.

<https://powcoder.com>

Assignment Project Exam Help

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Problem 2 : data type

Consider the following program:

```
type fruit      = Apple | Pear | Grape | Banana
type veggie     = Carrot | Cucumber | Cabbage
type salad      = _____
```

<https://powcoder.com>

```
let rec foo (a:salad) c =
  match a with
  | [] -> c
  | s::ss -> let c1, c2 = c in
    (match s with
     | Apple -> foo ss (c1+1, c2)
     | _ , Cabbage -> foo ss (c1, c2+1)
     | _ -> foo ss c)
```

<https://powcoder.com>

What is the definition of the type salad? Please explain your reasoning.

Problem 3 : Data type

```
type symbol = A | B | C
type op = ____ ??? ____
```

```
let apply1 (operation: op) (ls: symbol list) =
  match operation with
  | Un a -> List.map a ls
  | Bi b ->
    ( match ls with
      | h1::h2::tl -> b h1 h2 :: tl
      | _ -> [A])
```

```
let rec apply2 (two_op : op * op ) (ls: symbol list) =
  match two_op with
  | S s , Out o -> s :: ls
  | Out o , S s -> apply2 (S s , Out o) ls
  | S s , Un a -> let _ = a s in 1
```

Give a definition for the type `op` that would not generate any type error in the two programs above (hint: this type has 4 constructors). Please, explain your reasoning.

Provide an argument `myarg` such that given `ls`, `apply2 myarg ls` would return an `int`, indicating the number of occurrences of `A` in `ls`.

<https://powcoder.com>

Assignment Project Exam Help

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Problem 4 : Grammar

Consider the following grammar:

$\langle \text{expr} \rangle ::= \langle \text{term} \rangle + \langle \text{expr} \rangle \mid \langle \text{expr} \rangle * \langle \text{expr} \rangle \mid \langle \text{term} \rangle$

$\langle \text{term} \rangle ::= - \langle \text{val} \rangle \mid \langle \text{val} \rangle$

$\langle \text{val} \rangle ::= 0 \mid 1 \mid 2$

<https://powcoder.com>

Draw all the possible parse trees generating the following sentence $-2 + 1 * 2 + 0$

Add WeChat powcoder

<https://powcoder.com>

Add WeChat powcoder

Is the grammar ambiguous or unambiguous? Please explain

Please design an equivalent grammar where * has precedence over +.

<https://powcoder.com>

Assignment Project Exam Help

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Problem 5 : Grammar

Consider the following grammar:

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle + \langle \text{exp} \rangle \mid \langle \text{ter} \rangle - \langle \text{exp} \rangle \mid \langle \text{ter} \rangle \mid \langle \text{val} \rangle$

$\langle \text{ter} \rangle ::= \langle \text{ter} \rangle / \langle \text{con} \rangle \mid \langle \text{con} \rangle * \langle \text{ter} \rangle \mid \langle \text{con} \rangle$

$\langle \text{val} \rangle ::= 0 \mid 1$

$\langle \text{con} \rangle ::= c \mid d$

Can the following sentences be generated by the grammar above? If they can, draw all the possible parse trees of the sentence

(1). $c * d / 1 - 1$

(2). $0 + d * c + c$

(3) . 0 + d * c - c

<https://powcoder.com>
Is the grammar ambiguous or unambiguous? Please explain.

Assignment Project Exam Help

[Add WeChat powcoder](https://powcoder.com)
Assignment Project Exam Help

<https://powcoder.com>

Please design an equivalent grammar where * has precedence over all the other operations, / have precedence over + and -, and + has precedence over -.