VE477 Lab1 Report

Author: Zhou Haoquan

ID number: 519370910059

Functional Programming

What are imperative and object oriented languages? Give examples.

The imperative and object oriented languages focus on the class, methods and objects. In these languages, data and methods to manipulate it are kept as one unit called an object. Programmer will only focus on "what I want to do", as for "how to do", the program should be able to handle it. The imperative and object oriented languages include `Java, C++, C#`

What does it mean for a function to be a first class citizen?

The "first class citizen" means that the function should be treated the same as the other variables. They can be treated as value and given to other variables, they can be used as parameter of a function, they can also be the return value of a function.

What is a higher order function?

A higher order function takes other functions as parameters or return a function, or both.

Using basic mathematics show that integration defines is a linear map. How does it relate to higher order functions?

As we already know that integration has following two rules:

$$\int (f+g) = \int f + \int g$$

$$\int kf = k \int f$$

The relation between integration and higher order function is that, integration itself is a higher order function. It takes a function and return a new function. Also, this "integration function" can be treated as first class citizen and perform some linear calculations. This is corresponding to the linear map property of integration.

What does it mean for a variable to be immutable?

The immutable variable is the variable such that once it is created and given a value, they cannot be changed.

What are the pros and cons of dealing with immutable variables?

The pros are:

1. The data are safe from unconscious change.
2. It ensures that if the input is the same, the output must be the same. Which is the spirit of functional programming.
3. It is fast and memory efficient to do deep copy. Because and change will return a new immutable variable.

4. The data in different state are all stored (in different immutable variables).

The cons are:

1. It may be inconvenient to change the value.
2. To realize some function of the program, a lot of immutable variables may be declared.

What is a pure function or a function with no side-effects?

The pure function requires that all the function does is just do some calculation and return a value. It should not change the value of input variables or global variables. In another word, the function should not do anything other than calculation. Also, if you give a pure function the same input, the output should also be the same.

Getting started with OCaml

What is the meaning of REPL?

The REPL (Read-Eval-Print Loop) means that the compiler will read one sentence or one line of command. In `OCaml` we use `;;` to call an end of command. The complier will then evaluate the value of the expression and then print the result on the screen.

Is OCaml an interpreted or a compiled language?

It is an interpreted language, since we can run it immediately without compiling it before.

Which is “faster”, C or OCaml?

It seems that `OCaml` is the second fastest language in the world. It is faster than `C++` but slower than `C`.

What is the let keyword used for?

The `let` keyword is used for declare a variable or function. If we write an equation without `let`, the complier will think that it should do a comparison.

What are the basic types in OCaml?

The basic types in `OCaml` are: `int`, `float`, `bool`, `char`, `string`.

What is type inference?

When some expression does not state the type of variable explicitly, the compiler will automatically give it a suitable type, this is called `type inference`. For example `let x = 3` does not specifies the type of `x`, but `OCaml` will assign `x` as `int`.

What is the meaning of f : int -> int -> float ?

It means that the function `f` takes two input parameters in type `int` and return a value in type `float`.

What is the “ NULL reference problem”? How does the keyword option help in OCaml?

The "NULL reference problem" is that, if some variables are set to be `NULL` and then be used as if it is a real instance. Then problem will come up, since you cannot use a `NULL` as a real instance. The `option` is a built-in polymorphic type in `OCaml`, the `option` instance can any type as well as `None`. So that we can do things like return a desired-type variable when the expression is reasonable while a `None` when the expression meets some error and want to throw an exception.

What is the difference between statically and dynamically typed languages? Which category does OCaml belong to?

The statically typed language decides the type of variable when the program is compiled, like `C, Java`, while the dynamically typed language decides the type of variable when the variable is given a value. `OCaml` is obviously a dynamically typed language.

What is pattern matching. Provide a simple OCaml example different from the one in the tutorial.

The pattern matching is a good feature to replace a lot of `if...then...else` sentences. It will judge whether the variable satisfies some pattern (some specific values), then operate the corresponding instructions. For example,

```
# let check_1 n =
    match n with
    | 1 -> true
    | -> false ;;
val check_1 : int -> bool = <fun>
```

How does pattern matching increase type safety?

You can use `| _ -> ...` in `OCaml` as a default condition so that the function will return error or exception when the result is not desired.