Summary 6 Jianfeng Jia

Type System

This paper present a full formalization of type systems.

The main purpose of a type system is to reduce bugs in computer programs by defining interfaces between different parts of a computer program. The issue of whether programming languages should have types is still subject to some debate. From view point of Economy of execution, type information was first introduced in programming to improve code generation and runtime efficiency for numerical computations.Accurate type information at compile time leads to the application of the appropriate operations at run time without the need of expensive tests. When a type system is well designed, type checking can capture a large fraction of routine programming errors, eliminating lengthy debugging sessions. Modules can then be compiled independently of each other, with each module depending only on the interfaces of the others. Type system should be decidably verifiable, transparent and enforceable. Before formalized a type system, we should first formalizing the syntax of language, then to define the scoping rules of the language (lexical scope or the dynamic scope). After those steps, we can define the has-type or subtype-of relation between terms and the types.

Imperative Functional Programming

In this paper the author present a new model, may be called Monads IO later. It’s based on monads but performing the IO in a non-strict, purely functional way. Input/output has always appeared to be one of the less satisfactory features of purely functional languages. They defined an expression in a functional language denotes a value while an I/O command perform an action, and then using the lazy functional way to execute the I/O command. We can think of a value of type IO as a function that takes as its argument the current state of the world, and will return a new world where the state has been changed according to the function's return value. The state created in this way can be passed to another function, thus defining a series of functions which will be applied in order as steps of state changes.