Hindley-Milner Type System and Modern Type Systems

David Kozák, Dominik Labáš

xkozak15@stud.fit.vutbr.cz
xlabas00@stud.fit.vutbr.cz

Type systems assign types to various elements of programs such as variables and functions. Over the time, the complexity of type systems has grown significantly. Current type systems provide much more than just type checking. Quite often type systems provide type inference, which can significantly reduce the boilerplate connected with strongly typed languages such as Java, while keeping the same level of security.

In this talk, we will first present Hindley-Milner Type System, a classical example of type system for lambda calculus. This system provides both type checking and type inference and also has a powerful feature called parametric polymorphism. We will discuss the formal definition as well as the algorithms behind it.

Then we will briefly mention directions into which classical type systems are being extended these days. For example, we will show how TypeScript can provide additional safety guarantees for JavaScript and how dependent types in languages like Idris can be used to annotate types with predicates, which significantly increases the expressive power of the type system.