Hindley–Milner type system

xhrbac12, xtothv02

A type system in a programming language assigns a type to each symbol in the program. This can allow for more elaborate compiler optimisations, avoiding runtime type checks, and helping the programmer keep their program correct. However, having to explicitly annotate each symbol with its type can get quite cumbersome, especially as types get more elaborate. This can be avoided using type inference - that is, automatically determining the type of a symbol based on the context of its usage.

While most modern programming languages do have some basic form of inference - auto in C++, var in Java, most variants of assignment in Rust, etc. the concept can be taken further, in this case, most notably to inferring the types of function arguments. We present the well-established Hindley–Milner type system, a simple type system for functional programming languages.

Using a number of examples, we clarify the role of each of the inference rules of the system and how they build upon each other to create a full type system capable of inferring entire codebases. We also explain the terminology and notation used in type system research.