

Lambda Calculus – Notes

Lambda calculus is an abstract way of representing a programming language, commonly known as the *smallest universal programming language* there is. The lambda calculus is equivalent to a single-tape Turing machine, where all abstractions are composed of functions. The way function are representing are through the syntax:

$$\text{function} := \lambda < \text{argument} > . < \text{expression} >$$

where the argument and expression are composed of placeholder variables. For example, the identity function $f(x) = x$ can be represented in the lambda calculus as

$$(\lambda x.x)$$

We can also apply these functions to particular arguments, for example, the identity function evaluated at y would be

$$(\lambda x.x)y$$

Notice that when functions are being evaluated, the value gets “substituted” into the function. For this reason, we use the notation $[y/x]$ to denote that y is being substituted in for x :

$$(\lambda x.x)y \rightarrow [y/x]x \rightarrow y$$