

int.1 Computable Functions are λ -Definable

lam:int:lrp:
lam:int:lrp:
thm:computable-lambda

Theorem int.1. *Every computable partial function is λ -definable.*

Proof. We need to show that every partial computable function f is λ -defined by a lambda term F . By Kleene's normal form theorem, it suffices to show that every primitive recursive function is λ -defined by a lambda term, and then that the functions λ -definable are closed under suitable compositions and unbounded search. To show that every primitive recursive function is λ -defined by a lambda term, it suffices to show that the initial functions are λ -definable, and that the partial functions that are λ -definable are closed under composition, primitive recursion, and unbounded search. \square

We will use a more conventional notation to make the rest of the proof more readable. For example, we will write $M(x, y, z)$ instead of $Mxyz$. While this is suggestive, you should remember that terms in the untyped lambda calculus do not have associated arities; so, for the same term M , it makes just as much sense to write $M(x, y)$ and $M(x, y, z, w)$. But using this notation indicates that we are treating M as a function of three variables, and helps make the intentions behind the definitions clearer. In a similar way, we will say "define M by $M(x, y, z) = \dots$ " instead of "define M by $M = \lambda x. \lambda y. \lambda z. \dots$ "

Photo Credits

Bibliography