

# Translation

March 12, 2023

As is well known, in the type-free  $\lambda$ -calculus exactly the recursive functions are *definable*<sup>1</sup>. The concept used here of the definability of a function is also useful for the  $\lambda$ -calculus with types. We answer the question *here*<sup>2</sup> according to the then definable functions.

Types are 0 and with  $\sigma, \tau$  also  $(\sigma \rightarrow \tau)$ . Terms (denoted by  $r^\tau, s^\tau, t^\tau$ ) formed from variables with types by application and  $\lambda$ -abstraction. Type indices that result from the context or that are immaterial, we often leave it out. Terms that differ only by bound renaming divorce, are identified. The relation  $t \models t'$  ( $t$  is reducible to  $t'$ ) becomes inductively defined by

1.  $(\lambda x.t)s \models t_x[s]$
2. When  $t \models t'$  and  $s \models s'$ , so  $ts \models t's'$
3. When  $t \models t'$ , so  $\lambda x.t \models \lambda x.t'$
4.  $t \models t$
5. When  $t \models t'$  and  $t' \models t''$ , so  $t \models t''$

A term is called in normal form if it has no subterms of the form  $(\lambda x.t)s$ . As is well known, for every term  $t$  there is a uniquely determined normal form  $t'$  with  $t \models t'$ . Two terms are equal if they have the same normal form. One can now introduce natural numbers as terms  $\bar{n} \equiv \lambda \alpha. \alpha^n$  vom type  $\nu \equiv (0 \rightarrow 0) \rightarrow (0 \rightarrow 0)$ ; where  $\alpha^n$  is the n-th iterate of  $\alpha$ , i.e.  $\equiv \lambda x. \alpha(\alpha \dots (\alpha x) \dots)$ . Every closed term of type  $\nu$  in normal form is an  $\bar{n}$ . So defined every closed term  $t$  of the type  $\nu \rightarrow (\nu \rightarrow \dots (\nu \rightarrow \nu) \dots)$  is a number-theoretic one Function  $f$  determined by  $t\bar{n}_1, \dots, \bar{n}_k = \overline{f(n_1, \dots, n_k)}$ . For example *will*<sup>3</sup>

1.  $n + m$  define as  $\lambda F G \alpha. (F \alpha) \circ (G \alpha)$
2.  $n \cdot m$  define as  $\lambda F G \alpha. F \circ G$
3.  $k$  (constant function) define as  $\lambda F \alpha. \alpha^k$
- 4.

$$d(n, m, i) = \begin{cases} n & i = 0 \\ m & i \neq 0 \end{cases}$$

define as  $\lambda F G H \alpha x. H(\lambda y. G \alpha x)(F \alpha x)$

[F, G, H variables of type  $\nu, t^{\tau \rightarrow \tau} \circ s^{\tau \rightarrow \tau} \equiv \lambda x^\tau. t(sx)$ ]. The set of bare functions is apparently closed against insertions. So are all Polynomials definable, and more generally all functions resulting from polynomials Case distinction according to disappearance or non-disappearance of arguments are definable; in the 2-digit case, these are the functions

$$f(n, m) = \begin{cases} k & n = 0 \text{ and } m = 0 \\ P_1(m) & n = 0 \text{ and } m \neq 0 \\ P_2(m) & n \neq 0 \text{ and } m = 0 \\ P_3(n, m) & n \neq 0 \text{ and } m \neq 0 \end{cases}$$

with polynomials  $P_1, P_2, P_3$ . In the following it will be shown that this is all definable functions are.

So  $t$  defines a roughly 2-place function. Bring  $tFG\alpha$  to normal form.

Each subterm (of the normal form of  $tFG\alpha$ ) has one of the types  $0, 0 \rightarrow 0$ , or  $\nu$ . Each subterm of type  $\nu$  is identical to  $F$  or  $G$ . Possible subterms from Type  $0 \rightarrow 0$  are:

1.  $\alpha$
2. With  $s$  also  $Fs, Gs$ .
3. With  $s_1, \dots, s_q$ , formed according to (1), (2) also  $\lambda y.s_1(\dots s_q(z)\dots)$ , where  $z$  also identical with  $y$  can be.

In the following,  $s$  stands for subterms of the type  $0 \rightarrow 0$ . — For  $F, G$  set  $\bar{n}, \bar{m}$ , where initially  $n, m \geq 1$  is assumed. By induction over  $s$  one shows: Each  $s' \equiv s_{F,G}[\bar{n}, \bar{m}]$  is equal to  $\alpha^{P(n,m)}$  or equal to a constant function  $\lambda y.\alpha^{P(n,m)}z$  ( $P(n, m)$  polynom). Proof: To (2)

$$\begin{aligned} (\lambda\beta.\beta^n)\alpha^{P(n,m)} &= \alpha^{P(n,m) \cdot n} \\ (\lambda\beta.\beta^n)(\lambda y.\alpha^{P(n,m)}z) &= \lambda y.\alpha^{P(n,m)}z \end{aligned} \quad (n \geq 1)$$

To (3). Case 1: None of the  $s'_i$  is constant.

$$\lambda y.s'_1(\dots s'_q(z)\dots) = \lambda y.\alpha^{P_1(n,m) + \dots + P_q(n,m)}z$$

Case 2: There is a first constant  $s'_i$ , say  $s'_i = \lambda\tilde{y}.\alpha^{P_i(n,m)}\tilde{z}$

$$\lambda y.s'_1(\dots s'_q(z)\dots) = \lambda y.\alpha^{P_1(n,m) + \dots + P_q(n,m)}z$$

Since the normal form of  $tFG\alpha$  does not contain a free variable of type  $0$ , it is nach replacing  $F, G$  with  $\bar{n}, \bar{m}$  ( $n, m \geq 1$ ) equal to an  $\alpha^{P(n,m)}$ .

If about  $n = 0, m \geq 1$ , then one can replace all outermost subterms of the form  $Fs$  by  $\lambda x.x$  and gets  $\alpha^{P(m)}$  like before.