2.4 Primitive Recursion by Regular Minimalization

2.4.1 Alternative definition of μ-recursive functions. The class of μ-*recursive functions* is generated from the identity functions $I_i^n(\vec{x}) = x_i$, the multiplication function xy, and from the characteristic function $x <_* y$ of the comparision predicate $x < y$ by composition and regular minimalization of functions.

2.4.2 Lemma μ-*Recursive functions are closed under explicit definitions of functions without constants.*

2.4.3 Lemma μ-*Recursive functions are closed under regular minimalization of the form $f(\vec{x}) = \mu y[\tau[\vec{x}, y] = 1]$, where the term τ is without constants.*

2.4.4 Successor function is μ-recursive. We have $\forall x \exists y\, x < y$ and the number $x + 1$ is the least such number y. Hence the successor function $S(x) = x + 1$ is μ-recursive by the following regular minimalization:

$$S(x) = \mu y[(x <_* y) = 1].$$

2.4.5 Unary constant functions are μ-recursive. Clearly $\forall x \exists y\, x < x + 1$ and 0 is the least such number y. Hence the zero function $Z(x) = 0$ is μ-recursive function by the following regular minimalization:

$$Z(x) = \mu y\big[\big(x <_* S(x)\big) = 1\big].$$

We can now define all unary constant functions $C_m(x) = m$ as μ-recursive functions by a series of explicit definitions ($C_0 = Z$):

$$C_{m+1}(x) = C_m(x) + 1.$$

2.4.6 Lemma μ-*Recursive functions are closed under explicit definitions of functions.*

2.4.7 Lemma μ-*Recursive functions are closed under regular minimalization of the form $f(\vec{x}) = \mu y[\tau[\vec{x}, y] = 1]$.*

2.4.8 Boolean functions are μ-recursive. The boolean functions $\neg_* x$ and $x \wedge_* y$ are μ-recursive by explicit definitions:

$$(\neg_* x) = (x <_* 1)$$
$$(x \wedge_* y) = (\neg_* \neg_* xy).$$

The remaining boolean functions are derived similarly as μ-recursive.

2.4.9 Comparision predicates are μ-recursive. The binary predicates $x \le y$ and $x = y$ are μ-recursive by explicit definitions of their characteristic functions:

$$(x \le_* y) = (\neg_* y <_* x)$$
$$(x =_* y) = (x \le_* y \wedge_* y \le_* x).$$

2.4.10 Case discrimination function is μ-recursive. The graph of the case discrimination function D satisfies the following obvious property:

$$D(x, y, z) = v \leftrightarrow x \neq 0 \wedge v = y \vee x = 0 \wedge v = z.$$

We define D as μ-recursive by regular minimalization:

$$D(x, y, z) = \mu v[(\neg_* x =_* 0 \wedge_* v =_* y \vee_* x =_* 0 \wedge_* v =_* z) = 1].$$

2.4.11 Lemma *μ-Recursive functions are closed under the operator of bounded minimalization.*

Proof. Let the $(n+1)$-ary function f be defined by bounded minimalization:

$$f(x, \vec{y}) = \mu z \le x[g(z, \vec{y}) = 1]$$

from a μ-recursive function g. We clearly have

$$\forall x \forall \vec{y} \exists z \big(z \le x \to g(z, \vec{y}) = 1\big)$$

since $x + 1$ is one of such numbers z. Hence the auxiliary $(n+1)$-ary function h is defined by regular minimalization as a μ-recursive function:

$$h(x, \vec{y}) = \mu z\big[\big(z \le_* x \to_* g(z, \vec{y}) =_* 1\big) = 1\big].$$

Note that $h(x, \vec{y})$ yields the smallest number $z \le x$ such that $g(z, \vec{y}) = 1$ holds or $x + 1$ if there is no such number. We now define f by explicit definition as a μ-recursive function:

$$f(x, \vec{y}) = D\Big(\big(h(x, \vec{y}) \le_* x\big), h(x, \vec{y}), 0\Big). \qquad \square$$

2.4.12 Lemma *μ-Recursive functions are closed under explicit definitions of predicates with bounded formulas.*

2.4.13 Lemma *μ-Recursive functions are closed under definitions of functions with bounded minimalization.*

2.4.14 Lemma *μ-Recursive functions are closed under definitions of functions with regular minimalization of bounded formulas.*

Proof. Consider a function f defined by regular minimalization

$$f(\vec{x}) = \mu y\big[\varphi[\vec{x}, y]\big]$$

from μ-recursive functions and predicates. Here φ is a bounded formula. We can define f by the following series of definitions:

$$\begin{aligned} P(y, \vec{x}) &\leftrightarrow \varphi[\vec{x}, y] \\ f(\vec{x}) &= \mu y[P_*(y, \vec{x}) = 1]. \end{aligned}$$

By Thm. 2.4.12 the characteristic function P_* of the predicate P is μ-recursive and so is the function f. □

2.4.15 Addition is μ-recursive. First note that if $z \neq 0$ then we have

$$\begin{aligned} x + y = z \Leftrightarrow (x + y)z = z^2 \Leftrightarrow (x + y)z + xyz^2 + 1 = z^2 + xyz^2 + 1 \Leftrightarrow \\ \Leftrightarrow (xz + 1)(yz + 1) = (xy + 1)z^2 + 1. \end{aligned}$$

Addition can be thus derived as a μ-recursive function by regular minimalization of its graph:

$$x + y = \mu z\big[z = 0 \wedge x = 0 \wedge y = 0 \vee z \neq 0 \wedge S(xz)S(yz) = S\big(S(xy)zz\big)\big].$$

2.4.16 Modified subtraction is μ-recursive. The binary modified subtraction function $x \dot{-} y$ is μ-recursive by bounded minimalization:

$$x \dot{-} y = \mu z \leq x[x = y + z].$$

2.4.17 Integer division is μ-recursive. We define the integer division function $x \div y$ as μ-recursive by bounded minimalization:

$$x \div y = \mu q \leq x[x < (q + 1)y].$$

2.4.18 Pairing function is μ-recursive. The modified Cantor pairing function $\langle x, y\rangle$ is μ-recursive by explicit definition:

$$\langle x, y\rangle = (x + y)(x + y + 1) \div 2 + x + 1.$$

2.4.19 Projection functions are μ-recursive. Both projection functions of the pairing function are μ-recursive by bounded minimalization:

$$\begin{aligned} \pi_1(x) &= \mu y < x[\exists z < x\, x = \langle y, z\rangle] \\ \pi_2(x) &= \mu z < x[\exists y < x\, x = \langle y, z\rangle]. \end{aligned}$$

2.4.20 Lemma *The unary iteration* $\pi_2^n(x)$ *of the second projection:*

$$\pi_2^0(x) = x$$
$$\pi_2^{n+1}(x) = \pi_2\,\pi_2^n(x)$$

is a μ*-recursive function.*

Proof. Very hard. It will be supplied later. □

2.4.21 Sequence length is μ-recursive. We clearly have $\pi_2^x(x) = 0$ and thus $\forall x \exists n\, \pi_2^n(x) = 0$. Hence, the function $L(x)$ yielding the length of finite sequences is μ-recursive by regular minimalixzation:

$$L(x) = \mu n[\pi_2^n(x) = 0].$$

2.4.22 Indexing function is μ-recursive. The binary sequence indexing function $(x)_i$ yielding the $(i+1)$-st element of the sequence x is a μ-recursive function by explicit definition

$$(x)_i = \pi_1 \pi_2^i(x).$$

2.4.23 Lemma μ*-Recursive functions are closed under primitive recursion.*

Proof. Let the $(n+1)$-ary function f be defined by primitive recursion from μ-recursive functions g and h:

$$f(0, \vec{y}) = g(\vec{y})$$
$$f(x+1, \vec{y}) = h(x, f(x, \vec{y}), \vec{y}).$$

We will derive f as μ-recursive with the help of its course of values function:

$$\overline{f}(x, \vec{y}) = \big\langle f(x, \vec{y}), f(x-1, \vec{y}), \ldots, f(2, \vec{y}), f(1, \vec{y}), f(0, \vec{y}), 0\big\rangle.$$

The graph of the course of values function is μ-recursive by explicit definition:

$$\overline{f}(x, \vec{y}) = s \leftrightarrow L(s) = x + 1 \land (s)_{x-0} = g(\vec{y}) \land$$
$$\forall u < x\ (s)_{x \dot{-} (u+1)} = h\big(u, (s)_{x \dot{-} u}, \vec{y}\big).$$

The function $\overline{f}$ is μ-recursive by regular minimalization of its graph and thus the following explicit definition derives f as a μ-recursive function:

$$f(x, \vec{y}) = \big(\overline{f}(x, \vec{y})\big)_0 .$$ □

2.4.24 Theorem μ*-Recursive functions are primitively recursively closed.*

Proof. The class of μ-recursive functions contains the successor function $S(x) = x + 1$ and the zero function $Z(x) = 0$ by Par. 2.4.4 and Par. 2.4.5, respectively, and it is closed under primitive recursion by Thm. 2.4.23. □