1.2 Bounded Minimalization

1.2.1 Case discrimination function is primitive recursive. The ternary case analysis function D satisfies

$$D(0,y,z) = z$$
$$D(x+1,y,z) = y.$$

We can take these identities as its primitive recursive definition, cf. the definition of the predecessor function.

1.2.2 Equality predicate is primitive recursive. The characteristic function of the equality predicate is primitive recursive by explicit definition:

$$(x =_* y) = D(x \dot{-} y + (y \dot{-} x), 0, 1).$$

This is because we have $x = y$ iff $x \dot{-} y + (y \dot{-} x) = 0$.

1.2.3 Bounded minimalization. For every $n \geq 1$, the operator of *bounded minimalization* takes an $(n{+}1)$-ary function g and yields an $(n{+}1)$-ary function f satisfying:

$$f(x,\vec{y}) = \begin{cases} \text{the least } z \leq x & \text{such that } g(z,\vec{y}) = 1 \text{ holds, if } \exists z\, g(z,\vec{y}) = 1; \\ 0 & \text{if there is no such number.} \end{cases}$$

This is usually abbreviated to

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

1.2.4 Theorem *Primitive recursive functions are closed under bounded minimalization.*

Proof. Suppose that f is obtained by the bounded minimalization

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

of a primitive recursive function g. Clearly we have

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

Thus we can derive f as a p.r. function by the primitive recursive definition:

$$f(0,\vec{y}) = 0$$
$$f(x+1,\vec{y}) = D\big(g(f(x,\vec{y}),\vec{y}) =_* 1, f(x,\vec{y}), D(g(x+1,\vec{y}) =_* 1, x+1, 0)\big).$$ ◻

1.2.5 Boolean functions are primitive recursive. The standard *boolean* functions are primitive recursive by the following explicit definitions:

$$
\begin{aligned}
(\neg_* x) &= D(x, 0, 1)\\
(x \wedge_* y) &= D(x, D(y, 1, 0), 0)\\
(x \vee_* y) &= \bigl(\neg_* (\neg_* x \wedge_* \neg_* y)\bigr)\\
(x \rightarrow_* y) &= (\neg_* x \vee_* y)\\
(x \leftrightarrow_* y) &= \bigl((x \rightarrow_* y) \wedge_* (y \rightarrow_* x)\bigr).
\end{aligned}
$$

Note that non-zero values are identified with truth and 0 with falsehood.

1.2.6 Explicit definitions of predicates with bounded formulas. *Bounded formulas* are formulas which are built from atomic formulas by propositional connectives and bounded quantifiers. *Explicit definitions* of predicates *with bounded formulas* are of a form

$$P(x_1, \ldots, x_n) \leftrightarrow \varphi[x_1, \ldots, x_n],$$

where φ is a bounded formula with at most the indicated n-tuple of variables free and without any application of the predicate symbol P.

Every such definition can be viewed as a function operator which takes all functions occurring in the formula φ (this also includes the characteristic functions of every predicate occurring in φ) and which yields as a result the characteristic function P_* of the predicate P.

1.2.7 Theorem *Primitive recursive predicates are closed under explicit definitions of predicates with bounded formulas.*

Proof. We show that the class of primitive recursive predicates is closed under explicit definitions $P(\vec{x}) \leftrightarrow \varphi[\vec{x}]$ of n-ary predicates by induction on the structure of bounded formulas φ.

If $\varphi \equiv \tau = \rho$ then the characteristic function P_* of P is primitive recursive by the following explicit definition: $P_*(\vec{x}) = (\tau[\vec{x}] =_* \rho[\vec{x}])$.

If $\varphi \equiv R(\vec{\tau})$ then, since R_* is primitive recursive, we define P_* as primitive recursive by explicit definition: $P_*(\vec{x}) = R_*(\vec{\tau}[\vec{x}])$.

If $\varphi \equiv \neg\psi$ then we use IH and define an n-ary p.r. predicate R by explicit definition: $R(\vec{x}) \leftrightarrow \psi[\vec{x}]$. Now we define P_* as primitive recursive by the following explicit definition: $P_*(\vec{x}) = (\neg_* R_*(\vec{x}))$.

If $\varphi \equiv \psi \wedge \chi$ then we obtain as primitive recursive two auxiliary n-ary predicates $R(\vec{x}) \leftrightarrow \psi[\vec{x}]$ and $Q(\vec{x}) \leftrightarrow \chi[\vec{x}]$ by IH. We define P_* as primitive recursive by explicit definition: $P_*(\vec{x}) = (R_*(\vec{x}) \wedge_* Q_*(\vec{x}))$.

If $\varphi \equiv \exists y \le \tau\, \psi[y, \vec{x}]$ then we use IH and define an auxiliary $(n+1)$-ary p.r. predicate R by explicit definition: $R(y, \vec{x}) \leftrightarrow \psi[y, \vec{x}]$. Then we define an auxiliary *witnessing* p.r. function f by bounded minimalization:

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

The characteristic function P_* of the predicate P has the following explicit definition: $P_*(\vec{x}) = R_*\big(f(\tau[\vec{x}],\vec{x}),\vec{x}\big)$ as a p.r. function.

The remaining cases are treated similarly. □

1.2.8 Comparision predicates are primitive recursive. The standard comparison predicates are primitive recursive by the next explicit definitions:

$$x \le y \leftrightarrow \exists z \le y\, x = z \qquad x \ge y \leftrightarrow y \le x$$
$$x < y \leftrightarrow y \nleq x \qquad x > y \leftrightarrow y < x.$$

1.2.9 Divisibility is primitive recursive. The binary divisibility predicate $x \mid y$ is a p.r. predicate by the following explicit definition:

$$x \mid y \leftrightarrow \exists z \le y\, y = xz.$$

1.2.10 Definitions by bounded minimalization. Definitions of functions by *bounded minimalization* are of the form

$$f(\vec{x}) = \begin{cases} \text{the least } y \le \tau[\vec{x}] & \text{such that } \varphi[\vec{x},y] \text{ holds, if } \exists y \le \tau[\vec{x}]\,\varphi[\vec{x},y]; \\ 0 & \text{if there is no such number.} \end{cases}$$

Here $\tau[\vec{x}]$ is a term and $\varphi[\vec{x},y]$ a bounded formula with at most the indicated variables free, both without any application of the symbol f. Every such definition can be viewed as a function operator taking all functions and the characteristic functions of all predicates occurring in either the term τ or formula φ and yielding the function f.

In the sequel we abbreviate the definition to

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

We permit also strict bounds in definitions by bounded minimalization; i.e. we allow definitions of the form

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

as abbreviation for $f(\vec{x}) = \mu y \le \tau[\vec{x}]\big[y < \tau[\vec{x}] \land \varphi[\vec{x},y]\big]$.

1.2.11 Theorem *Primitive recursive functions are closed under definitions of functions with bounded minimalization.*

Proof. Consider an n-ary function f defined by the bounded minimalization

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

from primitive recursive functions and predicates. We can define f by the following series of definitions:

$$P(y, \vec{x}) \leftrightarrow \varphi[\vec{x}, y]$$
$$g(z, \vec{x}) = \mu y \le z[P_*(y, \vec{x}) = 1]$$
$$f(\vec{x}) = g(\tau[\vec{x}], \vec{x}).$$

By Thm. 1.2.7 and Thm. 1.2.4, the characteristic function P_* of P and the auxiliary function g are primitive recursive, and so is the function f. □

1.2.12 Integer division is primitive recursive. The integer division function $x \div y$ is a p.r. function by the following bounded minimalization:

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

1.2.13 Remainder is primitive recursive. The binary remainder function $x \bmod y$ is a p.r. function by the following explicit definition:

$$x \bmod y = D(y, x \dot{-} (x \div y)y, 0).$$