1.3 Primitive Recursive Predicates and Bounded Minimalization

1.3.1 Case discrimination function is primitive recursive. The case discrimination function D is defined by

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

The function is primitive recursive by the following explicit definition which uses monadic discrimination on the first argument:

$$\begin{aligned} \mathrm{D}(0,y,z) &= z \\ \mathrm{D}(x+1,y,z) &= y. \end{aligned}$$

1.3.2 Equality predicate is primitive recursive. The characteristic function $x =_* y$ of the equality predicate $x = y$ is primitive recursive by the following explicit definition:

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

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

1.3.3 Boolean functions are primitive recursive. The *boolean* functions are defined by

$$\begin{aligned} (\neg_* x) &= y \leftrightarrow x \neq 0 \wedge y = 0 \vee x = 0 \wedge y = 1 \\ (x \wedge_* y) &= z \leftrightarrow x \neq 0 \wedge y \neq 0 \wedge z = 1 \vee (x = 0 \vee y = 0) \wedge z = 0 \\ (x \vee_* y) &= z \leftrightarrow (x \neq 0 \vee y \neq 0) \wedge z = 1 \vee x = 0 \wedge y = 0 \wedge z = 0 \\ (x \rightarrow_* y) &= z \leftrightarrow (x = 0 \vee y \neq 0) \wedge z = 1 \vee x \neq 0 \wedge y = 0 \wedge z = 0 \\ (x \leftrightarrow_* y) &= z \leftrightarrow x \neq 0 \wedge y \neq 0 \wedge z = 1 \vee x = 0 \wedge y = 0 \wedge z = 1 \vee \\ &\qquad x \neq 0 \wedge y = 0 \wedge z = 0 \vee x = 0 \wedge y \neq 0 \wedge z = 0. \end{aligned}$$

Note that we identify non-zero values with truth and 0 with falsehood.

The functions are primitive recursive by the following explicit definitions:

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

1.3.4 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:

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

This is usually abbreviated to

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

1.3.5 Theorem *Primitive recursive functions are closed under the operator of bounded minimalization.*

Proof. Suppose that f is obtained by the bounded minimalization

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

of a primitive recursive function g. Clearly we have

$$\begin{aligned} g\big(f(x,\vec{y}),\vec{y}\big) = 1 &\to f(x+1,\vec{y}) = f(x,\vec{y}) \\ g\big(f(x,\vec{y}),\vec{y}\big) \ne 1 \land g(x+1,\vec{y}) = 1 &\to f(x+1,\vec{y}) = x+1 \\ g\big(f(x,\vec{y}),\vec{y}\big) \ne 1 \land g(x+1,\vec{y}) \ne 1 &\to f(x+1,\vec{y}) = 0. \end{aligned}$$

We derive f as a p.r. function by the following primitive recursive definition:

$$\begin{aligned} f(0,\vec{y}) &= 0 \\ f(x+1,\vec{y}) &= \mathrm{D}\Big(\big(g(f(x,\vec{y}),\vec{y}) =_* 1\big), f(x,\vec{y}), \mathrm{D}\Big(\big(g(x+1,\vec{y}) =_* 1\big), x+1, 0\Big)\Big). \end{aligned}$$

□

1.3.6 Formulas with bounded quantifiers. *Bounded quantifiers* are formulas of the form $\forall x \le \tau\, \varphi$ and $\exists x \le \tau\, \varphi$, where the variable x is not free in τ. The bounded quantifiers abbreviate the formulas $\forall x(x \le \tau \to \varphi)$ and $\exists x(x \le \tau \land \varphi)$, respectively. *Strict* bounded quantifiers $\forall x < \tau\, \varphi$ and $\exists x < \tau\, \varphi$ are defined similarly.

Bounded formulas are formulas which are built from atomic formulas by propositional connectives and bounded quantifiers.

1.3.7 Explicit definitions of predicates with bounded formulas. *Explicit definitions* of predicates *with bounded formulas* are of a form

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

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.

1.3.8 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. □

1.3.9 Comparison predicates are primitive recursive. The standard comparison predicates are primitive recursive by explicit definitions:

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

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

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

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

1.3.11 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:

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

By Thm. 1.3.8 and Thm. 1.3.5, the characteristic function P_* of P and the auxiliary function g are primitive recursive, and so is the function f. ◻

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

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

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

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

Exercises

1.3.14 Exercise. Show that the predicate of divisibility

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

is primitive recursive.

Solution.

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

1.3.15 Exercise. Show that the predicate $Prime(x)$ holding of prime numbers is primitive recursive.

1.3.16 Exercise. Show that the integer square root function

$$\lfloor \sqrt{x} \rfloor = y \leftrightarrow y^2 \leq x < (y+1)^2$$

is primitive recursive.