THEORY OF COMPUTATION
Primitive Recursive Predicates and Operations Defined on Predicates - 5

Prof. Dan A. Simovici

UMB

Predicates are total functions that range in the two-element set $\{0, 1\}$. Therefore, the notion of "primitive recursive" makes sense for predicates.

Example

The predicate $x = y$ corresponds to the function

$$f(x, y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

f is primitive recursive because

$$f(x, y) = \alpha(|x - y|).$$

The predicate $x \leqslant y$ is primitive recursive because it is just $\alpha(x \dot{-} y)$.

Theorem

Let $\mathcal{C}$ be a PRC class. If $P, Q \in \mathcal{C}$, then $\neg P$, $P \vee Q$ and $P \& Q$ all belong to $\mathcal{C}$.

Proof.

Note that $\neg P = \alpha(P)$, so $\neg P \in \mathcal{C}$.
We have $P \& Q = P \cdot Q$, so $P \& Q \in \mathcal{C}$.
Finally, $P \vee Q \in \mathcal{C}$ because

$$P \vee Q = \sim (\sim P \& \sim Q).$$

□

A similar result holds for computable predicates:

Corollary

If P and Q are computable predicates, then so are $\sim P$, $P \& Q$, and $P \vee Q$.

Example

$x < y$ is primitive recursive because

$$x < y \Leftrightarrow \sim (y \leqslant x).$$

Theorem

Definition by Cases:

Let $\mathcal{C}$ be a PRC class. If the functions g, h and the predicate P belongs to $\mathcal{C}$, then the function f defined as

$$f(x_1, \ldots, x_n) = \begin{cases} g(x_1, \ldots, x_n) & \text{if } P(x_1, \ldots, x_n) \\ h(x_1, \ldots, x_n) & \text{otherwise} \end{cases}$$

belongs to $\mathcal{C}$.

Proof.

The result follows from the equality

$$f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n) \cdot P(x_1, \ldots, x_n) + h(x_1, \ldots, x_n) \cdot \alpha(P(x_1, \ldots, x_n)).$$

Corollary

Let $\mathcal{C}$ be a PRC class. If the functions $g_1, \ldots, g_m, h$ and the predicates $P_1, \ldots, P_m$ belongs to $\mathcal{C}$ such that

$$P_i(x_1, \ldots, n) \& P_j(x_1, \ldots, n) = 0$$

for all $1 \leqslant i < j \leqslant m$ and $x_1, \ldots, x_n$.
If f defined as

$$f(x_1, \ldots, x_n) = \begin{cases} g_1(x_1, \ldots, x_n) & \text{if } P_1(x_1, \ldots, x_n) \\ g_2(x_1, \ldots, x_n) & \text{if } P_2(x_1, \ldots, x_n) \\ \vdots & \\ g_m(x_1, \ldots, x_n) & \text{if } P_m(x_1, \ldots, x_n) \\ h(x_1, \ldots, x_n) & \text{otherwise,} \end{cases}$$

then f belongs to $\mathcal{C}$.

Proof.

The proof is by induction on m. For $m = 1$ the statement holds by the previous theorem. Suppose that the statement is true and let h' be

$$h'(x_1, \ldots, x_n) = \begin{cases} g_{m+1}(x_1, \ldots, x_n) & \text{if } P_{m+1}(x_1, \ldots, x_n) \\ h(x_1, \ldots, x_n) & \text{otherwise} \end{cases}$$

Since $h' \in \mathcal{C}$ (by the theorem on Slide 8) and

$$f(x_1, \ldots, x_n) = \begin{cases} g_1(x_1, \ldots, x_n) & \text{if } P_1(x_1, \ldots, x_n) \\ g_2(x_1, \ldots, x_n) & \text{if } P_2(x_1, \ldots, x_n) \\ \vdots & \\ g_m(x_1, \ldots, x_n) & \text{if } P_m(x_1, \ldots, x_n) \\ h'(x_1, \ldots, x_n) & \text{otherwise,} \end{cases}$$

it follows that $f \in \mathcal{C}$.

Theorem

Let $\mathcal{C}$ be a PRC class. If $f(t, x_1, \ldots, x_n)$ belongs to $\mathcal{C}$, then so do the functions:

$$g(y, x_1, \ldots, x_n) = \sum_{t=0}^{y} f(t, x_1, \ldots, x_n),$$

and

$$h(y, x_1, \ldots, x_n) = \prod_{t=0}^{y} f(t, x_1, \ldots, x_n).$$

Proof.

The recursion equations can be written as

$$\begin{aligned} g(0, x_1, \ldots, x_n) &= f(0, x_1, \ldots, x_n), \\ g(t+1, x_1, \ldots, x_n) &= g(t, x_1, \ldots, x_n) + f(t+1, x_1, \ldots, x_n). \end{aligned}$$

Since addition belongs to $\mathcal{C}$, $g \in \mathcal{C}$.
Similarly, since

$$\begin{aligned} h(0, x_1, \ldots, x_n) &= f(0, x_1, \ldots, x_n), \\ h(t+1, x_1, \ldots, x_n) &= g(t, x_1, \ldots, x_n) \cdot f(t+1, x_1, \ldots, x_n), \end{aligned}$$

it follows that $h \in \mathcal{C}$. □

Question:

Can we prove by induction on y that $g(y, x_1, \ldots, x_n) \in \mathcal{C}$?

NO! because such a proof would show only that the functions

$$g(0, x_1, \ldots, x_n), g(1, x_1, \ldots, x_n), \ldots$$

belong to $\mathcal{C}$ and not that $g(y, x_1, \ldots, x_n) \in \mathcal{C}$!

A variant of Theorem from Slide 11

Theorem

Let $\mathcal{C}$ be a PRC class. If $f(t, x_1, \ldots, x_n)$ belongs to $\mathcal{C}$, then so do the functions:

$$g(y, x_1, \ldots, x_n) = \sum_{t=1}^{y} f(t, x_1, \ldots, x_n),$$

and

$$h(y, x_1, \ldots, x_n) = \prod_{t=1}^{y} f(t, x_1, \ldots, x_n).$$

Proof.

For this variant take the initial recursion equations

$$g(0, x_1, \ldots, x_n) = 0,$$
$$h(0, x_1, \ldots, x_n) = 1.$$

with the remaining equations as in the previous proof. This defines a vacuous sum as 0 and a vacuous product to be 1. □

Theorem

If the predicate $P(t, x_1, \ldots, x_n)$ belongs to some PRC $\mathcal{C}$ then so do the predicates

$$(\forall t)_{\leqslant y} P(t, x_1, \ldots, x_n) \text{ and } (\exists t)_{\leqslant y} P(t, x_1, \ldots, x_n).$$

The defined predicates are obtained through bounded quantification.

Proof.

Note that

$$
\begin{aligned}
(\forall t)_{\leqslant y} P(t, x_1, \ldots, x_n) &= \left(\prod_{t=0}^{y} P(t, x_1, \ldots, x_n) \right) = 1 \\
(\exists t)_{\leqslant y} P(t, x_1, \ldots, x_n) &= \left(\sum_{t=0}^{y} P(t, x_1, \ldots, x_n) \right) \neq 0.
\end{aligned}
$$

□

Alternatively, we could have written

$$(\forall t)_{\leqslant y} P(t, x_1, \ldots, x_n) = \prod_{t=0}^{y} P(t, x_1, \ldots, x_n).$$

Another mode for using quantifiers is $(\forall t)_{t<y}$ and $(\exists t)_{t<y}$.
The result follows from the recursion equations

$$\begin{aligned}(\exists t)_{t<y}P(t,x_1,\ldots,c_n) &= (\exists t)_{\leqslant y}[t \neq y \& P(t,x_1,\ldots,x_n)]\\ (\forall t)_{t<y}P(t,x_1,\ldots,c_n) &= (\forall t)_{\leqslant y}[t = y \vee P(t,x_1,\ldots,x_n)].\end{aligned}$$

Example

$y|x$ (which stands for "y divides x"); for example, $3|12$ is TRUE while $5|12$ is FALSE.
The predicate is primitive recursive because

$$y|x \Leftrightarrow (\exists t)_{\leqslant x}(y \cdot t = x).$$

Example

Prime(x) which is TRUE when x is a prime number is primitive recursive because

$$\text{Prime}(x) \Leftrightarrow x > 1\&(\forall t)_{\leqslant x}\{t = 1 \vee t = x \vee \sim (t|x)\},$$

which expresses that a number is prime if it is greater than 1 and has no divisors other than 1 and itself.