ldf.1 Primitive Recursive Functions are $\lambda$-Definable

Recall that the primitive recursive functions are those that can be defined from the basic functions zero, succ, and $P^n_i$ by composition and primitive recursion.

Lemma ldf.1. The basic primitive recursive functions zero, succ, and projections $P^n_i$ are $\lambda$-definable.

Proof. They are $\lambda$-defined by the following terms:

Zero $\equiv \lambda a. \lambda fx. x$
Succ $\equiv \lambda a. \lambda fx. f(ax)$
Proj$^n_i \equiv \lambda x_0 \ldots x_{n-1}. x_i$

Lemma ldf.2. Suppose the $k$-ary function $f$, and $n$-ary functions $g_0, \ldots, g_{k-1}$, are $\lambda$-definable by terms $F$, $G_0, \ldots, G_k$, and $h$ is defined from them by composition. Then $H$ is $\lambda$-definable.

Proof. $h$ can be $\lambda$-defined by the term

$$H \equiv \lambda x_0 \ldots x_{n-1}. F(G_0x_0 \ldots x_{n-1}) \ldots (G_{k-1}x_0 \ldots x_{n-1})$$

We leave verification of this fact as an exercise.

Problem ldf.1. Complete the proof of Lemma ldf.2 by showing that $H \map_{n_0 \ldots n_{n-1}} h(n_0, \ldots, n_{n-1})$.

Note that Lemma ldf.2 did not require that $f$ and $g_0, \ldots, g_{k-1}$ are primitive recursive; it is only required that they are total and $\lambda$-definable.

Lemma ldf.3. Suppose $f$ is an $n$-ary function and $g$ is an $n+2$-ary function, they are $\lambda$-definable by terms $F$ and $G$, and the function $h$ is defined from $f$ and $g$ by primitive recursion. Then $h$ is also $\lambda$-definable.

Proof. Recall that $h$ is defined by

$$h(x_1, \ldots, x_n, 0) = f(x_1, \ldots, x_n)$$
$$h(x_1, \ldots, x_n, y + 1) = h(x_1, \ldots, x_n, y, h(x_1, \ldots, x_n, y)).$$

Informally speaking, the primitive recursive definition iterates the application of the function $h$ $y$ times and applies it to $f(x_1, \ldots, x_n)$. This is reminiscent of the definition of Church numerals, which is also defined as an iterator.

For simplicity, we give the definition and proof for a single additional argument $x$. The function $h$ is $\lambda$-defined by:

$$H \equiv \lambda x. \lambda y. \text{Snd}(yD(\overline{0}, Fx))$$
where
\[ D \equiv \lambda p. \langle \text{Succ}(\text{Fst} p), (G x (\text{Fst} p)(\text{Snd} p)) \rangle \]

The iteration state we maintain is a pair, the first of which is the current \( y \) and the second is the corresponding value of \( h \). For every step of iteration we create a pair of new values of \( y \) and \( h \); after the iteration is done we return the second part of the pair and that’s the final \( h \) value. We now prove this is indeed a representation of primitive recursion.

We want to prove that for any \( n \) and \( m \), \( H n m \rightarrow h(n,m) \). To do this we first show that if \( D_n \equiv D[\pi/x] \), then \( D_n^m (\overline{0}, F \pi) \rightarrow \langle m, h(n,m) \rangle \). We proceed by induction on \( m \).

If \( m = 0 \), we want \( D_n^0 (\overline{0}, F \pi) \rightarrow \langle 0, h(n,0) \rangle \). But \( D_n^0 (\overline{0}, F \pi) \) just is \( \langle 0, F \pi \rangle \).

Since \( F \) \( \lambda \)-defines \( f \), this reduces to \( \langle 0, f(n) \rangle \), and since \( f(n) = h(n,0) \), this is \( \langle 0, h(n,0) \rangle \).

Now suppose that \( D_n^m (\overline{0}, F \pi) \rightarrow \langle m, h(n,m) \rangle \). We want to show that \( D_n^{m+1} (\overline{0}, F \pi) \rightarrow \langle m+1, h(n,m+1) \rangle \).

\[
D_n^{m+1} (\overline{0}, F \pi) \equiv D_n (D_n^m (\overline{0}, F \pi))
\]
\[
\rightarrow D_n (\langle m, h(n,m) \rangle) \quad \text{(by IH)}
\]
\[
\equiv \langle \lambda p. \langle \text{Succ}(\text{Fst} p), (G x (\text{Fst} p)(\text{Snd} p)) \rangle \rangle (\overline{m}, h(n,m))
\]
\[
\rightarrow \langle \text{Succ}(\text{Fst} (\overline{m}, h(n,m))),
\quad \langle G \pi (\text{Fst} (\overline{m}, h(n,m))) \rangle \rangle \langle \overline{m}, h(n,m) \rangle
\]
\[
\rightarrow \langle \text{Succ} (\overline{m}, (G \pi m h(n,m)))
\quad \rangle \langle m+1, g(n,m,h(n,m)) \rangle
\]

Since \( g(n,m,h(n,m)) = h(n,m+1) \), we are done.

Finally, consider
\[
H n m \equiv \lambda x. \lambda y. \text{Snd} (y (\lambda p. \langle \text{Succ}(\text{Fst} p), (G x (\text{Fst} p)(\text{Snd} p)) \rangle \langle 0, F x \rangle))
\]
\[
\rightarrow \text{Snd} (\underbrace{\langle \text{Succ}(\text{Fst} p), (G \pi (\text{Fst} p)(\text{Snd} p)) \rangle \langle 0, F \pi \rangle}_{D_n})
\]
\[
\equiv \text{Snd} (\overline{m} D_n (\overline{0}, F \pi))
\]
\[
\rightarrow \text{Snd} (D_n^m (\overline{0}, F \pi))
\]
\[
\rightarrow \text{Snd} (\overline{m}, h(n,m))
\]
\[
\rightarrow h(n,m).
\]

**Proposition ldf.4.** Every primitive recursive function is \( \lambda \)-definable.

**Proof.** By Lemma ldf.1, all basic functions are \( \lambda \)-definable, and by Lemma ldf.2 and Lemma ldf.3, the \( \lambda \)-definable functions are closed under composition and primitive recursion. □