Example — exponential function

Taylor series:

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} \tag{40}$$

Computing finite approximation (n terms)

$$e^x\ n = \sum_{i=0}^{n} \frac{x^i}{i!} \tag{41}$$

takes quadratic time. Wishing to calculate a linear-time algorithm from this mathematical definition, we first head for an inductive definition:

$$\begin{aligned} e^x\ 0 &= 1 \\ e^x\ (n+1) &= \underbrace{\frac{x^{n+1}}{(n+1)!}}_{h_x\, n} + \underbrace{\sum_{i=0}^{n} \frac{x^i}{i!}}_{e^x\ n} \end{aligned}$$

Example — exponential function

We thus get primitive recursive definition

$$\begin{array}{rcl} e^x\ 0 & = & 1 \\ e^x\ (n+1) & = & h_x n + e^x\ n \end{array}$$

where $h_x n$ unfolds to $\frac{x^{n+1}}{(n+1)!} = \frac{x}{n+1}\frac{x^n}{n!}$. Therefore:

$$\begin{array}{rcl} h_x 0 & = & x \\ h_x(n+1) & = & \dfrac{x}{n+2}(h_x n) \end{array}$$

Introducing $s2\ n = n+2$, we derive:

$$\begin{array}{l} s2\ 0 = 2 \\ s2(n+1) = 1 + s2\ n \end{array}$$

Example — exponential function

We can thus put e^x, $s2$ and h_x together in a system of three mutually recursive functions e^x, $s2_x$ and h_x over the naturals, which PF-transform to

$$
\begin{aligned}
e^x \cdot in &= \underbrace{[\underline{1}\ , (+) \cdot \langle \pi_1, \pi_2 \cdot \pi_2 \rangle]}_{r} \cdot \mathsf{F} \langle e^x, \langle s2_x, h_x \rangle \rangle \\
s2_x \cdot in &= \underbrace{[\underline{2}\ , suc \cdot \pi_1 \cdot \pi_2]}_{s} \cdot \mathsf{F} \langle e^x, \langle s2_x, h_x \rangle \rangle \\
h_x \cdot in &= \underbrace{[\underline{x}\ , (*) \cdot ((x/) \times id) \cdot \pi_2]}_{t} \cdot \mathsf{F} \langle e^x, \langle s2_x, h_x \rangle \rangle
\end{aligned}
$$

respectively, for

$$
\begin{aligned}
in &= [\underline{0}\ , suc] \\
\mathsf{F}\, X &= id + X
\end{aligned}
$$

Example — exponential function

From this system we obtain, thanks to the mutual recursion law (39)

$$
\begin{array}{lcl}
aux_x & \triangleq & \langle e^x, \langle s2_x, h_x \rangle\rangle \\
& = & \{\ (39)\ \} \\
& & (\!|\langle r, \langle s, t \rangle\rangle|\!)
\end{array}
$$

for

$$
\begin{array}{lcl}
r & = & [\underline{1}\ , (+) \cdot \langle \pi_1, \pi_2 \cdot \pi_2 \rangle] \\
s & = & [\underline{2}\ , suc \cdot \pi_1 \cdot \pi_2] \\
t & = & [\underline{x}\ , \underbrace{(*) \cdot ((x/) \times id) \cdot \pi_2}_{u}]
\end{array}
$$

Example — exponential function

Next we apply the exchange law (30) to $\langle r, \langle s, t \rangle\rangle$ (twice):

$$\langle r, \langle s, t \rangle\rangle \quad = \quad [\langle \underline{1}, \langle \underline{2}, \underline{x} \rangle\rangle \ , \langle (+) \cdot \langle \pi_1, \pi_2 \cdot \pi_2 \rangle, \langle suc \cdot \pi_1 \cdot \pi_2, u \rangle\rangle]$$

Thanks to universal properties (32) and (23) [2] we obtain

$$\begin{array}{rcl} aux_x \cdot \underline{0} & = & \langle \underline{1}, \langle \underline{2}, \underline{x} \rangle\rangle \\ aux_x \cdot suc & = & \langle (+) \cdot \langle \pi_1, \pi_2 \cdot \pi_2 \rangle, \langle suc \cdot \pi_1 \cdot \pi_2, u \rangle\rangle \cdot aux_x \\ e^x & = & \pi_1 \cdot aux_x \end{array}$$

that is, we have calculated linear implementation

[2] For functions.

Example — exponential function

```
exp x n = let (e,b,c) = aux x n
          in e where
              aux x 0 = (1,2,x)
              aux x (i+1) = let (e,s,h) = aux x i
                            in  (e+h,s+1,(x/s)*h)
```

which can be identified as the denotational semantics of a while loop, encoded below in the C programming language:

```
float exp(float x, int n)
{
  float e=1; int s=2; float h=x; int i;
  for (i=0;i<n+1;i++) {e=e+h;h=(x/s)*h;s++;}
  return e;
};
```