Lambda Calculus

Numerals

We introduce terms to represent arithmetic in λ-calculus

NOTATION For terms $F, A \in \Lambda$ and $n \in \mathbb{N}$ define $F^n A$ as follows

$$F^0 A \triangleq A$$

$$F^{k+1} A \triangleq F(F^k A)$$

Thus $F^2 A = F(FA),\ F^3 A = F(F(FA))$

DEFINITION (i) The *Church numerals* are $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \ldots$ with

$$\mathbf{c}_n \triangleq \lambda f x. f^n x$$

(ii) A function $f : \mathbb{N}^k \to \mathbb{N}$ is called λ-*definable* if there is a term $F \in \Lambda$ such that for all $\vec{n} \in \mathbb{N}^k$ one has

$$F \mathbf{c}_{n_1} \ldots \mathbf{c}_{n_k} = \mathbf{c}_{f(n_1, \ldots, n_k)}$$

Some representable functions

Define

$$
\begin{aligned}
F_+ &= \lambda nm\lambda fx.nf(mfx) \\
F_* &= \lambda nm\lambda fx.n(mf)x \\
F_{\exp} &= \lambda nm\lambda fx.mnfx
\end{aligned}
$$

These functions λ-define addition, multiplication, and exponentiation

$$
\begin{aligned}
F_+ \, \mathbf{c}_n \, \mathbf{c}_m &=_\beta \mathbf{c}_{n+m} \\
F_* \, \mathbf{c}_n \, \mathbf{c}_m &=_\beta \mathbf{c}_{n \cdot m} \\
F_{\mathsf{exp}} \, \mathbf{c}_n \, \mathbf{c}_m &=_\beta \mathbf{c}_{m^n}
\end{aligned}
$$

We verify this only for F_+

$$F_+\mathbf{c}_n\mathbf{c}_m =_\beta \lambda fx.\mathbf{c}_n f(\mathbf{c}_m fx) =_\beta \lambda fx.f^n(f^m x) \equiv \lambda fx.f^{n+m}x \equiv \mathbf{c}_{n+m}$$

Corollary. The function $f{:}\,\mathbb{N}\to\mathbb{N}$, with $f(n) = (n+2)^3$, is λ-definable

The predecessor function

This is $p\colon \mathbb{N} \to \mathbb{N}$ with $p(0) = 0,\ p(n+1) = n$

Define

$$\langle M, N\rangle \triangleq \lambda z.zMN$$

$$\begin{array}{rcl} \mathsf{true} & \triangleq & \mathsf{K} \triangleq \lambda x_1 x_2.x_1 \\ \mathsf{false} & \triangleq & \mathsf{K}_* \triangleq \lambda x_1 x_2.x_2 \\ \mathsf{P}_1 & \triangleq & \lambda a.a\mathsf{K} \\ \mathsf{P}_2 & \triangleq & \lambda a.a\mathsf{K}_* \end{array}$$

Then we can find the components from a pair:

$$\mathsf{P}_i\langle M_1, M_2\rangle = \langle M_1, M_2\rangle(\lambda x_1 x_2.x_i) = (\lambda x_1 x_2.x_i)M_1 M_2 = M_i$$

Construction

In order to λ-define the predecessor p we do it for "$n \leadsto \langle n, p(n)\rangle$"

I.e. $0 \leadsto \langle 0,0\rangle$, $1 \leadsto \langle 1,0\rangle$, $2 \leadsto \langle 2,1\rangle$, $3 \leadsto \langle 3,2\rangle$ ...

Let $T = \lambda p.\langle F_+ \,(\mathsf{P}_1 p)\,\mathbf{c}_1, \mathsf{P}_1 p\rangle$. Then $T\langle \mathbf{c}_n, \mathbf{c}_m\rangle = \langle \mathbf{c}_{n+1}, \mathbf{c}_n\rangle$

Note that

$$\langle \mathbf{c}_0, \mathbf{c}_0\rangle \leadsto^T \langle \mathbf{c}_1, \mathbf{c}_0\rangle \leadsto^T \langle \mathbf{c}_2, \mathbf{c}_1\rangle \leadsto^T \langle \mathbf{c}_3, \mathbf{c}_2\rangle \leadsto^T \ldots$$

Hence $T^k\langle \mathbf{c}_0, \mathbf{c}_0\rangle = \langle \mathbf{c}_k, \mathbf{c}_{k-1}\rangle$ Taking $P \triangleq \lambda n.\mathsf{P}_2(nT\langle \mathbf{c}_0, \mathbf{c}_0\rangle)$ one has

$$\begin{aligned}
P\mathbf{c}_0 &\equiv (\lambda n.\mathsf{P}_2(nT\langle \mathbf{c}_0, \mathbf{c}_0\rangle))\mathbf{c}_0 \\
&= \mathsf{P}_2(\mathbf{c}_0 T\langle \mathbf{c}_0, \mathbf{c}_0\rangle) \\
&= \mathsf{P}_2\langle \mathbf{c}_0, \mathbf{c}_0\rangle \\
&= \mathbf{c}_0 \equiv \mathbf{c}_{p(0)} \\
P\mathbf{c}_{k+1} &\equiv (\lambda n.\mathsf{P}_2(nT\langle \mathbf{c}_0, \mathbf{c}_0\rangle))\mathbf{c}_{k+1} \\
&= \mathsf{P}_2(T^{k+1}\langle \mathbf{c}_0, \mathbf{c}_0\rangle) \\
&= \mathsf{P}_2\langle \mathbf{c}_{k+1}, \mathbf{c}_k\rangle \\
&= \mathbf{c}_k \equiv \mathbf{c}_{p(k+1)}
\end{aligned}$$

Therefore $P\mathbf{c}_k = \mathbf{c}_{p(k)}$ for all $k \in \mathbb{N}$. Hence p is λ-definable by P

Church's Thesis

Thesis: Let the function $f\colon \mathbb{N}^k \to \mathbb{N}$ be given.

Then f is computable by a human if and only if it is λ-definable

Then Church went on constructing a function that is not λ-defiable

hence by his thesis, not human computable

Data types

We give two examples: natural numbers and trees

Natural numbers:

```
 Nat  :=  zero | suc Nat
Tree  :=  leaf | pair Tree Tree
```

Equivalently, as a context-free grammar

$$\texttt{Nat} \rightarrow \texttt{z} \mid \texttt{(s Nat)}$$

$$\texttt{Tree} \rightarrow \texttt{l} \mid \texttt{(p Tree Tree)}$$

We know what belongs to it

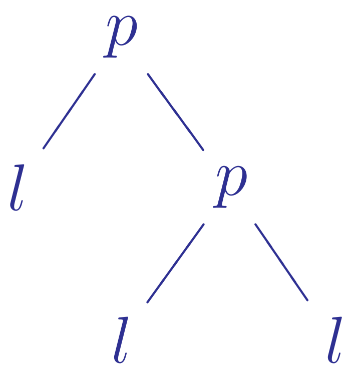
$$\texttt{Nat} = \{\texttt{z},\texttt{(sz)},\texttt{(s(sz))},\texttt{(s(s(sz)))},\ldots\} = \{\texttt{s}^n\texttt{z} \mid n \in \mathbb{N}\}$$

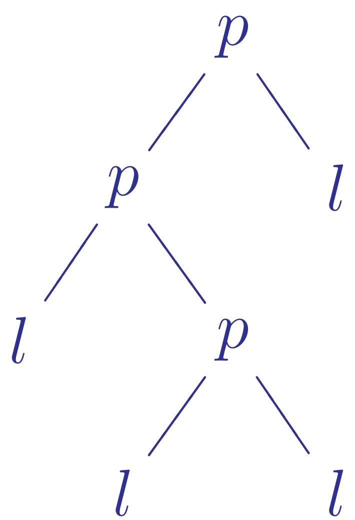
Trees

```
Tree  :=  l | (p Tree Tree)
```

Examples of elements of (language defined by) `Tree`

`(pl(pll))`   and   `(p(pl(pll))l)`





Translating data into lambda terms (Böhm-Berarducci)

$\texttt{Nat:} \quad \texttt{t} \leadsto \ulcorner \texttt{t} \urcorner := \lambda \texttt{sz.t}$

For example

$$\ulcorner (\texttt{s}(\texttt{s}(\texttt{sz}))) \urcorner = \lambda \texttt{sz}.(\texttt{s}(\texttt{s}(\texttt{sz}))) = \mathbf{c}_3$$

$\texttt{Tree:} \quad \texttt{t} \leadsto \ulcorner \texttt{t} \urcorner := \lambda \texttt{pl.t}$

For example

$$\ulcorner (\texttt{pl}(\texttt{pll})) \urcorner = \lambda \texttt{pl}.(\texttt{pl}(\texttt{pll}))$$

Operating on data after representing them

For `Nat` we could operate on the codes to represent wanted functions:

$$\mathtt{F}_{+} \ulcorner \mathtt{n} \urcorner \ulcorner \mathtt{m} \urcorner =_{\beta} \ulcorner \mathtt{n} + \mathtt{m} \urcorner$$

$$\mathtt{F}_{\times} \ulcorner \mathtt{n} \urcorner \ulcorner \mathtt{m} \urcorner =_{\beta} \ulcorner \mathtt{n} \times \mathtt{m} \urcorner$$

Define on `Trees` the operation of mirroring:

```
      Mirror (l)  =  l
Mirror (p t1 t2)  =  (p (Mirror t2) (Mirror t1))
```

We will construct a λ-term F_M such that

$$F_M \ulcorner t \urcorner = \ulcorner \mathtt{Mirror}(t) \urcorner$$

LEMMA. There exists a $P \in \Lambda$ such that

$$P \ulcorner \mathtt{t}_1 \urcorner \ulcorner \mathtt{t}_2 \urcorner = \ulcorner \mathtt{p} \mathtt{t}_1 \mathtt{t}_2 \urcorner \quad (1)$$

PROOF. Taking $P{:}= \lambda \mathtt{t}_1 \mathtt{t}_2 \mathtt{pl}.\mathtt{p}(\mathtt{t}_1 \mathtt{pl})(\mathtt{t}_2 \mathtt{pl})$ we claim that (1) holds.

Note that $\mathtt{t} \in \mathtt{Tree}$ can be considered as a λ-term: $\mathtt{Tree} \subseteq \Lambda$

Since $\ulcorner \mathtt{t} \urcorner = \lambda \mathtt{pl}.\mathtt{t}$ one has $\ulcorner \mathtt{t} \urcorner \mathtt{pl} =_\beta \mathtt{t}$. Hence

$$\begin{aligned} P \ulcorner \mathtt{t}_1 \urcorner \ulcorner \mathtt{t}_2 \urcorner &= (\lambda \mathtt{t}_1 \mathtt{t}_2 \mathtt{pl}.\mathtt{p}(\mathtt{t}_1 \mathtt{pl})(\mathtt{t}_2 \mathtt{pl})) \ulcorner \mathtt{t}_1 \urcorner \ulcorner \mathtt{t}_2 \urcorner \\ &= \lambda \mathtt{pl}.\mathtt{p}(\ulcorner \mathtt{t}_1 \urcorner \mathtt{pl})(\ulcorner \mathtt{t}_2 \urcorner \mathtt{pl}) \\ &= \lambda \mathtt{pl}.\mathtt{p} \mathtt{t}_1 \mathtt{t}_2 \\ &= \ulcorner \mathtt{p} \mathtt{t}_1 \mathtt{t}_2 \urcorner. \blacksquare \end{aligned}$$

PROPOSITION. There exists an $F_M \in \Lambda$ such that for all $\mathtt{t} \in \mathtt{Tree}$

$$F_M \ulcorner \mathtt{t} \urcorner =_\beta \ulcorner \mathtt{Mirror(t)} \urcorner \qquad (2)$$

PROOF. Take $F_M \triangleq \lambda \mathtt{tpl.tp'l}$, where $\mathrm{p}' \triangleq \lambda \mathrm{ab.pba}$.
We claim by induction that (2) holds. Note that $F_M \ulcorner \mathtt{t} \urcorner \mathtt{pl} = \ulcorner \mathtt{t} \urcorner \mathtt{p'l}$.

Case $\mathtt{t=l}$. Then

$$F_M \ulcorner \mathtt{l} \urcorner = \lambda \mathtt{pl}.(\lambda \mathtt{pl.l})\mathtt{p'l} = \lambda \mathtt{pl.l} = \ulcorner \mathtt{l} \urcorner = \mathtt{Mirror}(\ulcorner \mathtt{l} \urcorner).$$

Case $\mathtt{t} = \mathtt{pt_1t_2}$. Then

$$\begin{aligned}
F_M \ulcorner \mathtt{pt_1t_2} \urcorner &= \lambda \mathtt{pl}.\ulcorner \mathtt{pt_1t_2} \urcorner \mathtt{p'l} \\
&= \lambda \mathtt{pl}.\mathrm{P} \ulcorner \mathtt{t_1} \urcorner \ulcorner \mathtt{t_2} \urcorner \mathtt{p'l} \\
&= \lambda \mathtt{pl.p'}(\ulcorner \mathtt{t_1} \urcorner \mathtt{p'l})(\ulcorner \mathtt{t_2} \urcorner \mathtt{p'l}) \\
&= \lambda \mathtt{pl.p}(\ulcorner \mathtt{t_2} \urcorner \mathtt{p'l})(\ulcorner \mathtt{t_1} \urcorner \mathtt{p'l}) \\
&= \lambda \mathtt{pl.p}(F_M \ulcorner \mathtt{t_2} \urcorner \mathtt{pl})(F_M \ulcorner \mathtt{t_1} \urcorner \mathtt{pl}) \\
&= \lambda \mathtt{pl.p}(\ulcorner \mathtt{Mirror(t_2)} \urcorner \mathtt{pl})(\ulcorner \mathtt{Mirror(t_1)} \urcorner \mathtt{pl}), \qquad \text{by the IH,} \\
&= \ulcorner \mathtt{p(Mirror(t_2))(Mirror(t_1))} \urcorner \\
&= \ulcorner \mathtt{Mirror(pt_1t_2)} \urcorner . \ \blacksquare
\end{aligned}$$