Lambda Calculus

Week 3

Self-coding and evaluation

Henk Barendregt, Freek Wiedijk

assisted by Andrew Polonsky

Two examples: natural numbers and trees

Natural numbers:

$$\texttt{Nat} \quad := \quad \texttt{zero} \mid \texttt{suc Nat}$$

$$\texttt{Tree} \quad := \quad \texttt{leaf} \mid \texttt{pair Tree Tree}$$

Equivalently, as a context-free grammar

$$\texttt{Nat} \quad \to \quad \texttt{z} \mid \texttt{(s Nat)}$$

$$\texttt{Tree} \quad \to \quad \texttt{l} \mid \texttt{(p Tree Tree)}$$

We know what belongs to it

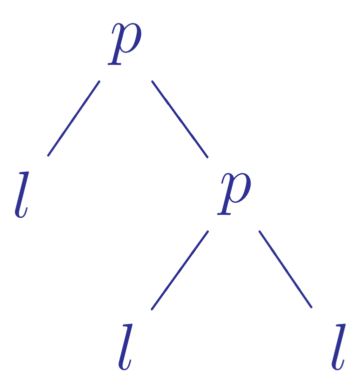
$$\texttt{Nat} = \{\texttt{z},\texttt{(sz)},\texttt{(s(sz))},\texttt{(s(s(sz)))},\ldots\} = \{\texttt{s}^n\texttt{z} \mid n \in \mathbb{N}\}$$

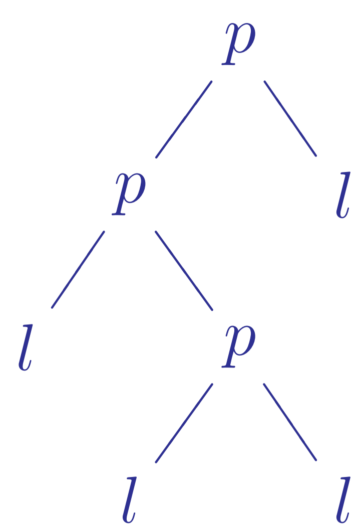
Trees

```
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 $\mathtt{Nat}$ we could operate on the codes to represent wanted functions:

$$\begin{aligned} \mathtt{A}_{+} \ulcorner \mathtt{n} \urcorner \ulcorner \mathtt{m} \urcorner &=_\beta \ulcorner \mathtt{n} + \mathtt{m} \urcorner \\ \mathtt{A}_{\times} \ulcorner \mathtt{n} \urcorner \ulcorner \mathtt{m} \urcorner &=_\beta \ulcorner \mathtt{n} \times \mathtt{m} \urcorner \end{aligned}$$

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 A_M such that

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

Representing the basic operation on Tree

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

$$P \ulcorner \mathtt{t}_1 \urcorner \ulcorner \mathtt{t}_2 \urcorner = \ulcorner \mathtt{p}\mathtt{t}_1\mathtt{t}_2 \urcorner \tag{1}$$

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

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

Representing mirroring in Λ

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

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

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

Case t=l. Then

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

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

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

More on data types

A data type D is a set with some operations (functions) on it.

An k-ary operation is a function $f : D^k \to D$.

Thereby a 0-ary operation $c : D^0 \to D$ is identified with a $c \in D$.

A datatype is determined by its operations on D:

$$\begin{array}{lcl} c_1, \ldots, c_{k_0} & : & D^0 \to D \ = \ D \\ f^1_1, \ldots, f^1_{k_1} & : & D^1 \to D \\ f^2_1, \ldots, f^2_{k_2} & : & D^2 \to D \\ \ldots & & \end{array}$$

$\texttt{Nat}$ has $\texttt{z} : \texttt{Nat}$, $\texttt{s} : \texttt{Nat} \to \texttt{Nat}$.

$\texttt{Tree}$ has $\texttt{l} : \texttt{Tree}, \texttt{p} : \texttt{Tree}^2 \to \texttt{Tree}$

Extra exercise

Let data type $\mathtt{D}$ be given as follows

$$\mathtt{D} ::= \mathtt{c} \mid \mathtt{fD} \mid \mathtt{gDD}$$

Let $A_0, A_1, A_2 \in \Lambda$. Show that there is an $F \in \Lambda$ such that for the Böhm-Berarducci coding $\ulcorner d \urcorner$ we have

$$\begin{array}{rcl} F\ulcorner \mathtt{c} \urcorner & = & A_0 \\ F\ulcorner \mathtt{fd} \urcorner & = & A_1(F\ulcorner \mathtt{d} \urcorner)\ulcorner \mathtt{d} \urcorner \\ F\ulcorner \mathtt{gd_1d_2} \urcorner & = & A_2(F\ulcorner \mathtt{d_1} \urcorner)(F\ulcorner \mathtt{d_2} \urcorner)\ulcorner \mathtt{d_1} \urcorner\ulcorner \mathtt{d_2} \urcorner \end{array}$$

The λ-definable functions over $\mathtt{D}$ are closed under primitive recursion.

Second translation (Böhm-Piperno-Guerini)

Consider the data type D with

$$c : D,\ f : D \to D,\ g : D^2 \to D$$

The second coding (also denoted by $\ulcorner t \urcorner$) is

$$\begin{array}{rcl} \ulcorner c \urcorner & = & \lambda e.eU_1^3 e \\ \ulcorner f(t) \urcorner & = & \lambda e.eU_2^3 \ulcorner t \urcorner e \\ \ulcorner g(t_1, t_2) \urcorner & = & \lambda e.eU_3^3 \ulcorner t_1 \urcorner \ulcorner t_2 \urcorner e \end{array}$$

PROPOSITION. There are lambda terms F, G such that

$$\begin{array}{rcl} F \ulcorner t \urcorner & = & \ulcorner f(t) \urcorner \\ G \ulcorner t_1 \urcorner \ulcorner t_2 \urcorner & = & \ulcorner g(t_1, t_2) \urcorner \end{array}$$

PROOF. Take

$$\begin{array}{rcl} F & := & \lambda te.eU_2^3 te \\ G & := & \lambda t_1 t_2 e.eU_3^3 t_1 t_2 e. \blacksquare \end{array}$$

THEOREM. Given $A_1, A_2, A_3 \in \Lambda$ there is an $H \in \Lambda$ such that

$$
\begin{array}{rcccl}
H\ulcorner c\urcorner & = & & = & A_1 H \\
H(F\ulcorner t\urcorner) & = & H(\ulcorner f(t)\urcorner) & = & A_2\ulcorner t\urcorner H \\
H(G\ulcorner t_1\urcorner\ulcorner t2\urcorner) & = & H(\ulcorner g(t_1,t_2)\urcorner) & = & A_3\ulcorner t_1\urcorner\ulcorner t_2\urcorner H
\end{array}
$$

PROOF. Try $H = \langle\langle B_1, B_2, B_3\rangle\rangle$.

$$
\begin{array}{rcll}
H\ulcorner c\urcorner & = & \langle\langle B_1, B_2, B_3\rangle\rangle\ulcorner c\urcorner & \\
& = & \ulcorner c\urcorner\langle B_1, B_2, B_3\rangle & \\
& = & \langle B_1, B_2, B_3\rangle U_1^3\langle B_1, B_2, B_3\rangle & \\
& = & B_1\langle B_1, B_2, B_3\rangle & \\
& = & A_1\langle\langle B_1, B_2, B_3\rangle\rangle, & \text{if } B_1 := \lambda z.A_1\langle z\rangle. \\
& = & A_1 H & \\
H\ulcorner f(t)\urcorner & = & \langle B_1, B_2, B_3\rangle U_2^3\ulcorner t\urcorner\langle B_1, B_2, B_3\rangle & \\
& = & B_2\ulcorner t\urcorner\langle B_1, B_2, B_3\rangle & \\
& = & A_2\ulcorner t\urcorner H, & \text{if } B_2 := \lambda tz.A_2 t\langle z\rangle. \\
H\ulcorner g(t_1,t_2)\urcorner & = & A_3\ulcorner t_1\urcorner\ulcorner t_2\urcorner H, & \text{if } B_3 := \lambda t_1 t_2 z.A_3 t_1 t_2\langle z\rangle. \ \blacksquare
\end{array}
$$

Data type for coding lambda terms

Consider the data type

$$\begin{aligned} \texttt{var}: &\quad D \to D \\ \texttt{app}: &\quad D \to D \to D \\ \texttt{abs}: &\quad D \to D \end{aligned}$$

Define `Var`, `App`, `Abs` as follows

$$\begin{aligned} \texttt{Var} &:= \lambda xe.eU_1^3 xe \\ \texttt{App} &:= \lambda xye.eU_2^3 xye \\ \texttt{Abs} &:= \lambda xe.eU_3^3 xe \end{aligned}$$

Coding lambda terms $M \rightsquigarrow \ulcorner M \urcorner$ (Mogensen)

$$\begin{aligned} \ulcorner x \urcorner &:= \texttt{Var}\, x \\ \ulcorner MN \urcorner &:= \texttt{App} \ulcorner M \urcorner \ulcorner N \urcorner \\ \ulcorner \lambda x.M \urcorner &:= \texttt{Abs}\, (\lambda x. \ulcorner M \urcorner) \end{aligned}$$

Self-interpretation

THEOREM. There exists a λ-term E such that for all $M \in \Lambda$

$$\mathsf{E}\ulcorner M \urcorner = M$$

PROOF. By recursion we can find an E such that

$$\begin{aligned} \mathsf{E}(\mathtt{Var}\ x) &= x \\ \mathsf{E}(\mathtt{App}\ m\ n) &= \mathsf{E}m(\mathsf{E}n) \\ \mathsf{E}(\mathtt{Abs}\ m) &= \lambda x.\mathsf{E}(mx) \end{aligned}$$

Then

$$\begin{aligned} \mathsf{E}(\ulcorner x \urcorner) &= \mathsf{E}(\mathtt{Var}\ x) &&= x \\ \mathsf{E}(\ulcorner MN \urcorner) &= \mathsf{E}(\mathtt{App}\ulcorner M \urcorner \ulcorner N \urcorner) &&= \mathsf{E}\ulcorner M \urcorner(\mathsf{E}\ulcorner N \urcorner) &= MN \\ \mathsf{E}(\ulcorner \lambda x.M \urcorner) &= \mathsf{E}(\mathtt{Abs}(\lambda x.\ulcorner M \urcorner)) &&= \lambda x.\mathsf{E}\ulcorner M \urcorner &= \lambda x.M. \end{aligned}$$

Filling in the details of E one has (writing C $:= \lambda xyz.xzy$)

$$\mathsf{E} = \langle\langle \mathsf{K}, \mathsf{S}, \mathsf{C} \rangle\rangle. \qquad \blacksquare$$

Application 1

If you see someone coming out of 'arrivals' in an airport, you cannot determine where he or she comes from.

Similarly, there is no F such that for all $X, Y \in \Lambda$

$$F(XY) = X$$

Proposition. There exists an $F_i \in \Lambda$, $i \in \{1, 2\}$ such that

$$F_i \ulcorner X_1 X_2 \urcorner = \ulcorner X_i \urcorner.$$

Proof. We do this for $i = 1$. By recursion there exists F_1 s.t.

$$F_1(\texttt{App}\ x_1\ x_2) = A_2 x_1 x_2 F_1 = x_1, \text{ taking } A_2 = \mathsf{U}_1^3.$$

This suffices. ∎

Second fixed point theorem*

LEMMA. There exists a term $\texttt{Num}\in\Lambda$ such that for all $M\in\Lambda$

$$\texttt{Num}\ulcorner M\urcorner =_\beta \ulcorner\ulcorner M\urcorner\urcorner$$

PROOF. Use recursion for the lambda calculus data type with

$$\begin{array}{rcl} A_1 xN & = & \texttt{App}\ulcorner\texttt{Var}\urcorner(\texttt{Var}\ x) \\ A_2 mnN & = & \texttt{App}(\texttt{App}\ulcorner\texttt{App}\urcorner(Nm))(Nn) \\ A_3 mN & = & \texttt{App}\ulcorner\texttt{Abs}\urcorner(\texttt{Abs}(\lambda x.N(mx))) \end{array}$$

SECOND FIXED POINT THEOREM. For all F there is an X with

$$F\ulcorner X\urcorner =_\beta X$$

PROOF. Let $W := \lambda z.F(\texttt{App}\ z(\texttt{Num}\ z))$ and $X := W\ulcorner W\urcorner$. Then

$$\begin{aligned} X &= W\ulcorner W\urcorner \\ &= F(\texttt{App}\ulcorner W\urcorner(\texttt{Num}\ulcorner W\urcorner)) = F(\texttt{App}\ulcorner W\urcorner\ulcorner\ulcorner W\urcorner\urcorner) \\ &= F\ulcorner W\ulcorner W\urcorner\urcorner = F\ulcorner X\urcorner. \ \blacksquare \end{aligned}$$

Decidability of sets of terms

A set $\mathcal{C} \subseteq \Lambda$ is *decidable* if there exists an $F \in \Lambda$ such that for all $M \in \Lambda$

$$\begin{array}{lcll} F\ulcorner M\urcorner & = & \text{true}, & \text{if } M \in \mathcal{C}; \\ F\ulcorner M\urcorner & = & \text{false}, & \text{otherwise.} \end{array}$$

Problem. Is $\Lambda^{\varnothing} \subseteq \Lambda$ decidable?

A theorem by Dana Scott

Theorem. Let $\mathcal{C} \subseteq \Lambda$ be a non-trivial set ($\emptyset \neq \mathcal{C}, \Lambda \neq \mathcal{C}$). Suppose that $\mathcal{C}$ is closed under $=_\beta$. Then $\mathcal{C}$ is undecidable.

Proof. Suppose $\mathcal{C}$ is decidable. Let $M_0 \notin \mathcal{C}, M_1 \in \mathcal{C}$. Then there exists a term $F \in \Lambda$ such that

$$\begin{array}{rcll} F\ulcorner M \urcorner & = & \text{true}, & \text{if } M \in \mathcal{C}; \\ F\ulcorner M \urcorner & = & \text{false}, & \text{otherwise.} \end{array}$$

Taking $G \triangleq \lambda m.F M M_1 M_0$ one gets

$$\begin{array}{rcll} G\ulcorner M \urcorner & = & M_0, & \text{if } M \in \mathcal{C}; \\ G\ulcorner M \urcorner & = & M_1, & \text{otherwise.} \end{array}$$

There exists a term $M \in \Lambda$ such that $G\ulcorner M \urcorner = M$. But then

$$\begin{array}{rcccl} M \in \mathcal{C} & \Rightarrow & M = G\ulcorner M \urcorner = M_0 & \Rightarrow & M \notin \mathcal{C}; \\ M \notin \mathcal{C} & \Rightarrow & M = G\ulcorner M \urcorner = M_1 & \Rightarrow & M \in \mathcal{C}. \end{array}$$

a contradiction. ■

Classical theory

Def. For $M \in \Lambda$, define $\#(M)$ as the Gödel number of M. Write

$$\mathsf{c}_M \triangleq \mathsf{c}_{\#(M)}.$$

Prop. There exists an $\mathsf{E}_c \in \Lambda^{\varnothing}$ such that for all $M \in \Lambda^{\varnothing}$

$$\mathsf{E}_c \mathsf{c}_M = M$$

Prop. For all $F \in \Lambda$ there exists an $M \in \Lambda$ such that

$$M = F \mathsf{c}_M.$$

Prop. Using this notion the set $\{M \in \Lambda \mid \mathrm{FV}(M) = \emptyset\}$ is decidable.