HENK BARENDREGT

DISCRIMINATING CODED LAMBDA TERMS

Abstract. A *coding* for a (type-free) lambda term M is a lambda term $\ulcorner M \urcorner$ in normal form such that M (and its parts) can be reconstructed from $\ulcorner M \urcorner$ in a lambda definable way. Kleene (*1936*) defined a coding $\ulcorner M \urcorner^K$ and a *self-interpreter* $\mathbf{E}^K \in \Lambda^\circ$ such that:

$$\forall M \in \Lambda^\circ \ \mathbf{E}^K \ulcorner M \urcorner^K = M. \tag{1}$$

(Kleene did it for the $\lambda\mathbf{I}$-calculus, but the result is valid also for the $\lambda\mathbf{K}$-calculus.) In this style one can construct a *discriminator* $\Delta^K \in \Lambda^\circ$ such that:

$$\forall M, N \in \Lambda \ \ \Delta^K \ulcorner M \urcorner^K \ulcorner N \urcorner^K = \begin{cases} \mathtt{true} & (\equiv \lambda xy.x) \quad \text{if } M \equiv N, \\ \mathtt{false} & (\equiv \lambda xy.y) \quad \text{else.} \end{cases} \tag{2}$$

The terms $\mathbf{E}^K$ and Δ^K are complicated. $\mathbf{E}^K$ appears also in *Church 1941* under the name `form`. They depend on the lambda definability of functions on the integers dealing with coded syntactic properties. Inspired by a construction of P. de Bruin (see *Barendregt 1991*) a different coding $\ulcorner M \urcorner$ and an efficient self-interpreter $\mathbf{E} \in \Lambda^\circ$ was constructed by Mogensen (*1992*) such that even

$$\forall M \in \Lambda \ \mathbf{E} \ulcorner M \urcorner = M. \tag{3}$$

This construction does not represent syntax via an encoding as numbers, but directly as lambda terms. This results in a much less complex $\mathbf{E}$. Mogensen's construction was simplified even further in *Böhm et al. 1994*. In this paper we construct a simple discriminator $\Delta \in \Lambda^\circ$ such that:

$$\forall M, N \in \Lambda^\circ \ \ \Delta \ulcorner M \urcorner \ulcorner N \urcorner = \begin{cases} \mathtt{true} & \text{if } M \equiv_\alpha N, \\ \mathtt{false} & \text{else.} \end{cases} \tag{4}$$

Note that in (1) and (4) the statement is only about closed lambda terms, while that in (2) and (3) is about all lambda terms. Moreover in (2) syntactic equality on terms is considered literary, while in (4) we can deal in an easy way with the better notion of equality modulo a change of names for bound variables. It will become clear why this is so.

This paper first appeared in *From Universal Morphisms to Megabytes—a Baayen Space Odyssey*, printed at CWI, Amsterdam.

1. INTRODUCTION

The most important notations for the type-free lambda calculus will be given here. Background can be found in *Barendregt 1984*.

DEFINITION 1.1. Variables and terms of the lambda calculus are defined by the following abstract syntax.

$$\begin{aligned} \mathtt{var} &= a \mid \mathtt{var}' \\ \mathtt{term} &= \mathtt{var} \mid \mathtt{term\ term} \mid \lambda\, \mathtt{var\ term} \end{aligned}$$

C. Anthony Anderson and M. Zelëny (eds.), Logic, Meaning and Computation, 275-285.

NOTATION.

(i) $M, N, \ldots, P, Q, \ldots$ range over λ-terms. The letters $x, y, z, \ldots$ range over variables. Note that the variables are $\{a, a', a'', \ldots, a^{(n)}, \ldots\}$.

(ii) Λ is the set of lambda terms. $\mathrm{FV}(M)$ is the set of free variables of M. The set of closed terms is $\Lambda^\circ = \{M \in \Lambda \mid \mathrm{FV}(M) = \emptyset\}$.

(iii) The relation $\equiv$ denotes syntactic equality; the relation $\equiv_\alpha$ denotes syntactic equality up to a change of names of the bound variables. For example

$$\lambda x.x \equiv_\alpha \lambda y.y \not\equiv \lambda x.x.$$

(iv) The relation $=$ denotes β-convertibility, axiomatized by

$$(\lambda x.M)N = M[x := N].$$

Here $[x := n]$ denotes substitution of N for the free occurrences of x. E.g.,

$$(x(\lambda x.x))[x := a] \equiv a(\lambda x.x).$$

(v) $\mathbb{N}$ is the set of natural numbers. For $n \in \mathbb{N}$ the terms $\mathbf{c}_n \equiv \lambda f x.f^n x$, where $f^0 x \equiv x$ and $f^{n+1}x \equiv f(f^n x)$, denote the so called Church numerals. Note that the $\mathbf{c}_n$ are distinct normal forms; hence

$$\mathbf{c}_n = \mathbf{c}_m \;\Rightarrow\; n = m$$

by the Church-Rosser theorem.

A lambda term can be seen as an executable: the redexes want to be evaluated. In this sense a normal form is not executable anymore. For a lambda term M its *code* $\ulcorner M \urcorner$ is a normal form such that M is reconstructible from M. Kleene (*1936*) defined a code $\ulcorner M \urcorner^K$ essentially as follows.

DEFINITION 1.2.

(i) By induction on the structure of M we define $\#M$.

$$\begin{aligned} \#(a^{(n)}) &= <0, n>, \\ \#(PQ) &= <1, <\#(P), \#(Q)>>, \\ \#(\lambda x.P) &= <2, <\#(x), \#(P)>> . \end{aligned}$$

Here $<-,->$ denotes a recursive pairing function on $\mathbb{N}$ with the recursive projections $(-)_0, (-)_1$:

$$(<n_0, n_1>)_i = n_i.$$

(ii) The map $\ulcorner - \urcorner^K : \Lambda \to \Lambda$ is defined by

$$\ulcorner M \urcorner^K = \mathbf{c}_{\#M}.$$

Note that for all $M \in \Lambda$ the term $\ulcorner M \urcorner^K$ is in normal form. Moreover,

$$\ulcorner M \urcorner^K = \ulcorner N \urcorner^K \Rightarrow M \equiv N.$$

PROPOSITION 1.3. *There is no lambda term Q such that for all $M \in \Lambda^{(o)}$ one has*

$$QM = \ulcorner M \urcorner^K.$$

Proof. Suppose Q exists. Then for $\mathsf{I} \equiv \lambda x.x$ one has

$$Q(\mathsf{II}) = \ulcorner \mathsf{II} \urcorner^K = \mathbf{c}_{\#(\mathsf{II})} = \mathbf{c}_{<1,<\#\mathsf{I},\#\mathsf{I}>>}.$$

But also

$$Q(\mathsf{II}) = Q\mathsf{I} = \ulcorner \mathsf{I} \urcorner^K = \mathbf{c}_{\#\mathsf{I}} = \mathbf{c}_{<2,<\#(x),\#(x)>>}.$$

Hence $<1,<\#(\mathsf{I}),\#(\mathsf{I})>>=<2,<\#(x),\#(x)>>$, a contradiction. ⊣

In spite of this fact that the 'quote' Q does not exists, the inverse 'evaluation' E can be constructed.

THEOREM 1.4 (Kleene, *1936*). *There exists an $\mathsf{E}^K \in \Lambda^\circ$ such that for all $M \in \Lambda^\circ$ one has*

$$\mathsf{E}^K \ulcorner M \urcorner^K = M.$$

Proof. See *Kleene 1936* or *Barendregt 1984*, Theorem 8.1.16. ⊣

The self-interpreter E can work only for closed terms M (or terms having at most a fixed finite set of free variables). The reason is that if

$$\mathsf{E}^K \ulcorner M \urcorner^K = M,$$

then

$$\mathrm{FV}(M) \subseteq \mathrm{FV}(\mathsf{E}^K \ulcorner M \urcorner^K) = \mathrm{FV}(\mathsf{E}^K).$$

Therefore if E^K is closed, then the M have to be closed as well. This causes one difficulty in the construction of E^K. The closed terms do not form a context-free language. Kleene solved this problem by constructing E first for the set of combinatory terms $\mathcal{C}^\circ$ built from the basis $\{\mathsf{K}, \mathsf{S}\}$ using application only; then the real self-interpreter can be obtained by translations between Λ° and $\mathcal{C}^\circ$.

A different construction of a self-interpreter was given by a former student of mine, using ideas from denotational semantics. The equations for the self-interpreter are those of the semantic interpretation and the F plays the role of a valuation (environment).

THEOREM 1.5 (P. de Bruin). *There exists an $\mathsf{E}_0 \in \Lambda^\circ$ such that for all $M \in \Lambda$ and all $F \in \Lambda$ one has*

$$\mathsf{E}_0 \ulcorner M \urcorner F = M[x_1, \ldots, x_n := F\ulcorner x_1 \urcorner, \ldots, F\ulcorner x_n \urcorner] \tag{5}$$

(simultaneous substitution), where $\{x_1, \ldots, x_n\} = \mathrm{FV}(M)$.

Proof. By the representability of computable functions and the fixedpoint theorem there is a term $\mathbf{E}_0 \in \Lambda^\circ$ such that

$$\begin{aligned} \mathbf{E}_0 \ulcorner x \urcorner^K F &= F \ulcorner x \urcorner^K, \\ \mathbf{E}_0 \ulcorner PQ \urcorner^K F &= (\mathbf{E}_0 \ulcorner P \urcorner^K F)(\mathbf{E}_0 \ulcorner Q \urcorner^K F), \\ \mathbf{E}_0 \ulcorner \lambda x.P \urcorner^K F &= \lambda x.(\mathbf{E}_0 \ulcorner P \urcorner^K F_{[\ulcorner x \urcorner \mapsto x]}), \end{aligned}$$

where $F_{[\ulcorner x \urcorner \mapsto x]} = F'_x$ with

$$\begin{aligned} F'_x \ulcorner x \urcorner &= x, \\ F'_x \ulcorner y \urcorner &= F \ulcorner y \urcorner, \text{ if } y \not\equiv x. \end{aligned}$$

Note that F'_x can be written as GFx, with G closed. By induction on the structure of $M \in \Lambda$ one can show that the statement holds. ⊣

COROLLARY 1.6. *There exists an* $\mathbf{E}^{dB} \in \Lambda^\circ$ *such that for all* $M \in \Lambda^\circ$ *one has*

$$\mathbf{E} \ulcorner M \urcorner^K = M.$$

Proof (P. de Bruin). We can take

$$\mathbf{E}^{dB} \equiv \lambda m.\mathbf{E}_0 m \mathbf{I}.$$

Indeed, for closed terms M it follows from (5) that

$$\mathbf{E}^{dB} \ulcorner M \urcorner = \mathbf{E}_0 \ulcorner M \urcorner \mathbf{I} = M. \quad ⊣$$

2. REPRESENTING DATA TYPES

After seeing the method of P. de Bruin, an improved version of it was given by Mogensen (*1992*) by representing data types directly (i.e., not using the natural numbers) in the lambda calculus as done in, e.g., *Böhm and Berarducci 1985*. This approach was improved later by Böhm *et al.* (*1994*) by constructing a new representation of data types into the type-free lambda calculus. This new representation will be treated in a slightly modified form in this section.

DEFINITION 2.1. Write

$$\begin{aligned} \langle M_1, \ldots, M_n \rangle &= \lambda z.zM_1 \ldots M_n, \\ \mathbf{U}^n_i &= \lambda x_1 \ldots x_n.x_i, \\ \mathtt{true} &= \mathbf{U}^2_1, \\ \mathtt{false} &= \mathbf{U}^2_2. \end{aligned}$$

Note that

$$\begin{aligned} \langle M_1, \ldots, M_n \rangle \mathbf{U}_i^n &= M_i, \\ \mathtt{true}\, PQ &= P, \\ \mathtt{false}\, PQ &= Q. \end{aligned}$$

In particular we have $\langle M \rangle = \lambda z.zM$ and $\langle\,\rangle = \lambda x.x = \mathbf{I}$. Now we define the notion of lists inspired by the language LISP (*McCarthy et al. 1961*).

DEFINITION 2.2.

(i) Write

$$\begin{aligned} \mathtt{nil} &= \langle\,\rangle, \\ \mathtt{cons} &= \lambda xy.\langle x, y \rangle, \\ \mathtt{car} &= \langle \mathbf{U}_1^2 \rangle, \\ \mathtt{cdr} &= \langle \mathbf{U}_2^2 \rangle, \\ \mathtt{null?} &= \langle \mathbf{U}_3^3, \mathbf{U}_1^2, \mathtt{false}, \mathtt{true} \rangle. \end{aligned}$$

(ii) Define

$$\begin{aligned} [\,] &= \langle\,\rangle, \\ [M_1, \ldots, M_{n+1}] &= \mathtt{cons}\, M_1 [M_2, \ldots, M_{n+1}]. \end{aligned}$$

So for example

$$[M_1, M_2, M_3] = \langle M_1, \langle M_2, \langle M_3, \langle\,\rangle \rangle \rangle \rangle.$$

(In *Barendregt 1984* this term is written as $[M_1, M_2, M_3, \mathbf{I}]$. At the time of writing that book we did not yet see the usefulness of terminating a list with a special constructor.) Note that

$$\begin{aligned} \mathtt{car}\,(\mathtt{cons}\, PQ) &= P, \\ \mathtt{cdr}\,(\mathtt{cons}\, PQ) &= Q, \\ \mathtt{null?}\,\mathtt{nil} &= \mathtt{true}, \\ \mathtt{null?}(\mathtt{cons}\,\mathtt{PQ}) &= \mathtt{false}. \end{aligned}$$

PROPOSITION 2.3. *There exist lambda definable functions* $(\ )_i$ *such that for* $1 \leq i \leq n$ *one has*

$$([M_1, \ldots, M_n])_i = M_i.$$

Proof. Take

$$\begin{aligned} (l)_1 &= \mathtt{car}\, l, \\ (l)_{i+1} &= (\mathtt{cdr}\, l)_i. \end{aligned}$$

⊣

http://www.springer.com/978-1-4020-0141-3

Logic, Meaning and Computation
Essays in Memory of Alonzo Church
Anderson, C.A.; Zelëny, M. (Eds.)
2001, XIII, 627 p., Hardcover
ISBN: 978-1-4020-0141-3