CHAPTER 3

DENOTATIONAL SEMANTICS

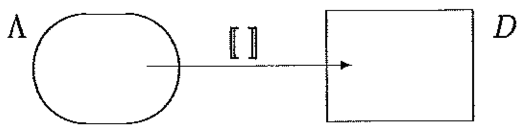
Semantics of lambda calculus: an introduction

In natural languages, one can explain the meaning of a particular word in two ways. One can *translate* the word into another language (of which the meaning is already known); the second way is to describe the *use* or *behaviour* of the word in the language itself.

Lambda calculus can be considered as a (formal) language. A λ-term (initially just a sequence of symbols) can be given a meaning in the abovementioned two ways. This leads to the notions of *denotational* and *operational* semantics respectively.

Denotational semantics

In the denotational approach, λ-terms are translated into another structure (usually some mathematical domain).



This semantics is usually given in a modular (or 'syntax-driven') way by equipping D with a binary application operation $\cdot$ and defining e.g.

$$[\![MN]\!] = [\![M]\!] \cdot [\![N]\!],$$

or, in a functional notation,

$$[\![MN]\!] = [\![M]\!]([\![N]\!])$$

since M is considered as a function and N as its argument. Because in λ-calculus the terms serve both as arguments and as functions applied to these arguments, one would like a domain D such that $D \to D$ (the space of functions from D to D) is isomorphic to D. For cardinality reasons this is impossible. The mathematician D.S. Scott solved this problem by restricting $D \to D$ to the set $[D \to D]$ of so-called *continuous* functions on D. He worked with complete lattices and constructed a D such that $[D \to D] \cong D$. It turned out that for a model of the λ-calculus it is sufficient to find a D such that $[D \to D]$ is a so-called *retract* of D.

The interpretation $[\![\cdot]\!]$ is *sound* if, roughly spoken,

$$\lambda \vdash M = N \;\Rightarrow\; [\![M]\!] = [\![N]\!],$$

so terms that are equal modulo λ-convertibility are given the same value in the model. This leads to the notion of λ-algebra.

This illustrates two motives for studying denotational semantics: firstly, by the translation one *identifies* certain distinct syntactical objects, e.g. **KIΩ** and **II**. Furthermore, by examining equality of terms in a given model $\langle D, [\![\cdot]\!]\rangle$ one obtains insight in possible extra identifications on the syntactical level. This has lead, e.g., to a good representation of the notion 'undefined' (known from recursion theory) in the lambda calculus.

Operational semantics

Operational semantics of λ-calculus is concerned with the *reduction behaviour* of λ-terms. This relates a λ-term M to the set of all possible 1-step reducts, and so on. Rather than studying the full reduction graph $G_\beta(M)$ one often considers one particular reduction path. Such a path is usually obtained from a *reduction strategy*, choosing in a term one or more redexes to be reduced.

This approach is common in the description of semantics of functional programming languages: the result of a functional program depends on the choice of a particular evaluation order. Therefore the often mentioned correspondence between functional programming languages and λ-calculus is preferably expressed by

$$\text{functional programming language} \approx \lambda\text{-calculus} + \text{reduction strategy}.$$

3.1. Complete lattices

3.1.1. Definition. Let D be a set, and let $\sqsubseteq\ \subseteq D \times D$ be an ordering. $(D, \sqsubseteq)$ is a partial ordering if for all $x, y, z \in D$ one has

$x \sqsubseteq x$ ($\sqsubseteq$ is reflexive);

$(x \sqsubseteq y\ \&\ y \sqsubseteq z) \Rightarrow x \sqsubseteq z$ ($\sqsubseteq$ is transitive);

$(x \sqsubseteq y\ \&\ y \sqsubseteq x) \Rightarrow x = y$ ($\sqsubseteq$ is antisymmetric).

3.1.2. Definition. Let $(D, \sqsubseteq)$ be a partial ordering, $a \in D$, and $X \subseteq D$.

(i) a is an upper bound of X (notation $X \sqsubseteq a$) if

$$\forall x \in X \quad x \sqsubseteq a.$$

a is a lower bound of X (notation $a \sqsubseteq X$) if

$$\forall x \in X \quad a \sqsubseteq x.$$

(ii) a is the supremum of X (notation $a = \sup X$, $a = \bigsqcup X$) if

(1) $X \sqsubseteq a$ ("a is an upper bound of X");

(2) for all $b \in D$: if $X \sqsubseteq b$, then $a \sqsubseteq b$ ("a is the least upper bound of X").

Note that this definition implies that suprema are unique.

3.1.3. Definition. Let D be a set, and $\sqsubseteq \in \wp(D \times D)$. $(D, \sqsubseteq)$ is a complete lattice if

(1) $(D, \sqsubseteq)$ is a partial ordering;

(2) for all $X \subseteq D$ there is $a \in D$ such that $a = \sup X$.

3.1.4. Proposition. Each complete lattice $(D, \sqsubseteq)$ has a largest element (top, $\top$) and a least element (bottom, $\bot$).

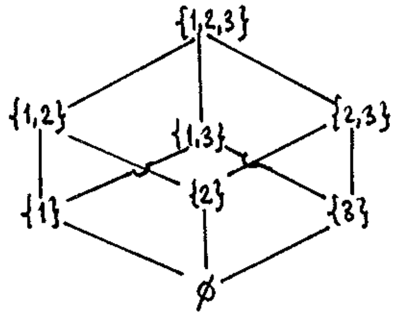
Proof. Take

$$\top = \sup D;$$
$$\bot = \sup \emptyset \quad (!). \quad \boxtimes$$

3.1.5. Examples. (i) Let A be a set. Then $(\wp(A), \subseteq)$ is a complete lattice with for $X \subseteq \wp(A)$

$$\sup X = \bigcup_{S \in X} S.$$

The following picture shows $(\wp(\{1,2,3\}), \subseteq)$.



(ii) $([0,1]_{\mathbb{R}}, \leq)$ is a complete lattice.

(iii) $([0,1]_{\mathbb{Q}}, \leq)$ is not a complete lattice. For example, the set

$$\{x \in [0,1]_{\mathbb{Q}} \mid 2x^2 < 1\}$$

has no supremum in $[0,1]_{\mathbb{Q}}$.

3.1.6. Definition. Let $(D, \sqsubseteq)$ be a partial ordering, $a \in D$, and $X \subseteq D$. a is the infimum of X (notation $a = \inf X$, $a = \sqcap X$) if

(1) $a \sqsubseteq X$ ("a is a lower bound of X");

(2) for all $b \in D$: $b \sqsubseteq X \Rightarrow b \sqsubseteq a$ ("a is the greatest lower bound of X").

3.1.7. <u>Proposition</u>. Let $(D, \sqsubseteq)$ be a complete lattice. Then for all $X \subseteq D$ the infimum $\inf X$ exists.

<u>Proof</u>. One easily verifies that

$$\inf X = \sup \{ y \in D \mid y \sqsubseteq X \}. \quad \boxtimes$$

Below $(D, \sqsubseteq)$, $(D', \sqsubseteq')$, $(D'', \sqsubseteq'')$, ... range over complete lattices.

3.1.8. <u>Notation</u>. For $x, y \in D$ we write

$$x \sqcup y = \bigsqcup \{x, y\}$$

and

$$x \sqcap y = \bigsqcap \{x, y\}.$$

3.1.9. <u>Definition</u>. Let $X \subseteq D$. X is <u>directed</u> if

$$\forall x \in X \ \forall y \in X \ \exists z \in X \ [x \sqsubseteq z \ \& \ y \sqsubseteq z].$$

3.1.10. <u>Definition</u>. Let $f: D \to D'$ be a function.

(i) f is <u>monotonic</u> if for all $x, y \in D$

$$x \sqsubseteq y \Rightarrow f(x) \sqsubseteq' f(y).$$

(ii) f is <u>continuous</u> if for all directed $X \subseteq D$ one has

$$f(\sup X) = \sup f(X) \quad (= \sup \{ f(x) \mid x \in X \}).$$

3.1.11. <u>Proposition</u>. Let $f: D \to D'$ be a function. Then

$$f \text{ is continuous } \underset{\not\Leftarrow}{\Rightarrow} f \text{ is monotonic.}$$

<u>Proof</u>. ($\Rightarrow$) Suppose f is continuous. Note that for $x, y \in D$ with $x \sqsubseteq y$ one has

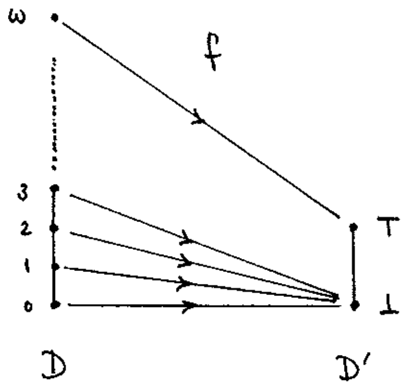
$$x \sqcup y = y.$$

Therefore

$$f(x) \sqcup f(y) = f(y).$$

Hence $f(x) \sqsubseteq' f(y)$ so f is monotonic.

($\not\Leftarrow$) A counterexample is suggested by the following picture.



Clearly f is monotonic, but

$$f(\sup\{0,1,2,\cdots\}) = f(\omega) = \top \neq \bot = \sup f(\{0,1,2,\cdots\}). \boxtimes$$

3.1.12. Proposition. Let $f: D \to D'$, $g: D' \to D''$. If f and g are monotonic (continuous), then $g \circ f: D \to D''$ is monotonic (continuous).

Proof. Straightforward; note that for directed $X \subseteq D$, $f(X)$ is also directed by monotonicity of f. $\boxtimes$

3.1.13. Definition. Let $X \subseteq D \times D'$. Then

$$(X)_0 = \{x \in D \mid \exists x' \in D' \ \ (x, x') \in X\};$$
$$(X)_1 = \{x' \in D' \mid \exists x \in D \ \ (x, x') \in X\}.$$

3.1.14. Definition. (i) Given D, D', let $D \times D'$ be the carthesian product partially ordered by

$$(x, x') \sqsubseteq (y, y') \ \text{ iff } \ x \sqsubseteq y \ \& \ x' \sqsubseteq' y'.$$

(ii) Let D, D' be given. Define

$$[D \to D'] = \{f: D \to D' \mid f \text{ is continuous}\}.$$

This set can be ordered pointwise:

$$f \sqsubseteq g \ \text{ iff } \ \forall x \in D \ \ f(x) \sqsubseteq' g(x).$$

3.1.15. Proposition. (i) $D \times D'$ is a complete lattice; for $X \subseteq D \times D'$ one has

$$\bigsqcup X = (\bigsqcup (X)_0, \bigsqcup (X)_1).$$

(ii) Let $\{f_i\}_i$ be a collection of continuous maps $f_i: D \to D'$.

Define $f: D \to D'$ by $f(x) = \sup_i (f_i(x))$. Then f is continuous and in $[D \to D']$ one has $f = \sup_i f_i$. Therefore $[D \to D']$ is a complete lattice.

<u>Proof</u>. (i) Easy.

(ii) Let $X \subseteq D$ be directed. Then

$$
\begin{aligned}
f(\sup X) &= \sup_i f_i(\sup X) \\
&= \sup_i \sup_{x \in X} f_i(x), \quad \text{by continuity of } f_i \\
&= \sup_{x \in X} \sup_i f_i(x) \qquad \text{(see practicum)} \\
&= \sup_{x \in X} f(x).
\end{aligned}
$$

Therefore f is continuous. Moreover $\{f_i\}_i \sqsubseteq f$, by definition of $\sqsubseteq$. Suppose $\{f_i\}_i \sqsubseteq g$, then $\forall i, x \; f_i(x) \sqsubseteq g(x)$ so $\forall x \quad \sup_i f_i(x) \sqsubseteq g(x)$. Hence $f \sqsubseteq g$. ⊠

3.1.16. <u>Remark</u>. If $\lambda\!\!\lambda x.$ denotes meta-λ-abstraction, then we have as a consequence of proposition 3.1.15 (ii)

$$\sup_i \lambda\!\!\lambda x. f_i(x) = \lambda\!\!\lambda x. \sup_i (f_i(x)),$$

i.e. sup commutes with $\lambda\!\!\lambda$.

3.1.17. <u>Theorem</u>. Let $f \in [D \to D]$. Then f has a least fixed point defined by

$$a = \mathrm{Fix}(f) = \sup_n f^n(\perp).$$

<u>Proof</u>. Note that the set $\{f^n(\perp) \mid n \in \mathbb{N}\}$ is directed: $\perp \sqsubseteq f(\perp)$ so by monotonicity $f(\perp) \sqsubseteq f^2(\perp)$, etcetera. Therefore $\perp \sqsubseteq f(\perp) \sqsubseteq f^2(\perp) \sqsubseteq \cdots$. Hence

$$
\begin{aligned}
f(a) &= \sup_n f(f^n(\perp)) \\
&= \sup_n f^{n+1}(\perp) \\
&= a.
\end{aligned}
$$

Suppose x is another fixedpoint of f. Then $f(x) = x$ and $\perp \sqsubseteq x$, so by monotonicity $f^n(\perp) \sqsubseteq f^n(x) = x$. Therefore $a \sqsubseteq x$. ⊠

3.1.18. Lemma. Let $f: D \times D' \longrightarrow D''$. Then f is continuous iff f is continuous in each of its variables seperately (i.e. $\lambda x.\, f(x, x_0')$ and $\lambda x'.\, f(x_0, x')$ are continuous for all x_0, x_0').

Proof. ($\Rightarrow$) Let f be continuous, and $x_0' \in D'$. In order to show that $h = \lambda x.\, f(x, x_0')$ is continuous, define $g: D \longrightarrow D \times D'$ by

$$g(x) = (x, x_0').$$

Clearly g is continuous. Moreover $h = f \circ g$. Hence, by proposition 3.1.12, h is continuous. Similarly one proves the continuity of $\lambda x'.\, f(x_0, x')$ for $x_0 \in D$.

($\Leftarrow$) Let $X \subseteq D \times D'$ be directed. Then

$$\begin{aligned} f(\sup X) &= f(\sup (X)_0, \sup (X)_1) \\ &= \sup_{x \in (X)_0} f(x, \sup (X)_1) \\ &= \sup_{x \in (X)_0} \sup_{x' \in (X)_1} f(x, x') \\ &= \sup_{(x,x') \in X} f(x, x'). \end{aligned}$$

The last equality holds because X is directed. Therefore f is continuous. ☒

3.2. Towards a λ-model

In order to turn a complete lattice into a model of the λ-calculus, we need the operations "application" and "abstraction".

3.2.1. PROPOSITION. *(Continuity of application).*

Define $Ap: [D \to D'] \times D \to D'$ *by*

$$Ap(f, x) = f(x).$$

Then Ap *is continuous.*

PROOF. Apply lemma 3.1.18. $\lambda x.\, Ap(f,x) = \lambda x.\, f(x) = f$ is continuous since $f \in [D \to D']$. Let $H = \lambda f.\, Ap(f, x_0) = \lambda f.\, f(x_0)$. Then for f_i, $i \in I$, directed

$$\begin{aligned} H(\sup_i f_i) &= (\sup_i f_i)(x_0) \\ &= \sup_i (f_i(x_0)), \text{ by proposition 3.1.15(ii)}, \\ &= \sup_i H(f_i). \;\square \end{aligned}$$

3.2.2. PROPOSITION. *(Continuity of abstraction). Let* $f \in [D\times D' \to D'']$. *Then* $\lambda y.\ f(x,y) \in [D' \to D'']$ *and depends continuously on* x.

PROOF. By lemma 3.1.18 it follows that $\lambda y.\ f(x,y) \in [D' \to D'']$. Moreover let $X \subseteq D$ be directed. Then

$$\begin{aligned}\lambda y.\ f(\sup X, y) &= \lambda y.\ \sup_x f(x,y)\\ &= \sup_x \lambda y.\ f(x,y)\end{aligned}$$

by continuity of f and the remark 3.1.16. □

It now follows that the category of complete lattices with continuous maps forms a cartesian closed category. We will not use this terminology however.

3.2.3. DEFINITION. (i) D is a *retract* of D' (notation $D < D'$) if there are continuous maps $F\colon D' \to D$, $G : D \to D'$ such that $F \circ G = id_D$.
(ii) D is called *reflexive* if $[D \to D] < D$.

REMARK. If $D < D'$ via the maps F,G, then F is surjective and G injective. We may identify D with its image $G(D) \subseteq D'$. Then F "retracts" the larger space D' to the subspace D.

Now it will be shown how a reflexive D can be turned into a model of the λ-calculus.

3.2.4. DEFINITION. Let D be reflexive via F,G.
(i) F retracts D to its function space $[D \to D] \subseteq D$. So for $x \in D$ one has $F(x) \in [D \to D]$. In this way elements of D become functions on D and one may write

$$x\cdot_F y = F(x)(y) (\in D).$$

(ii) Conversely, every continuous function on D becomes via G an element of D. Now one may write

$$\lambda^G x.\ f(x) = G(f) (\in D).$$

for f continuous.

A *valuation* in D is a map ρ: variables $\to D$.

3.2.5. DEFINITION. Let D be reflexive via F,G.
(i) Given a valuation ρ in D and $M \in \Lambda$ the interpretation of M *in* D *under the valuation* ρ (notation $[\![M]\!]^D_\rho$) is defined as follows.

M	$[\![M]\!]^D_\rho$
x	$\rho(x)$
PQ	$[\![P]\!]^D \cdot_F [\![Q]\!]^D_\rho$
$\lambda x.P$	$\lambda^G d.\ [\![P]\!]^D_{\rho(x:=d)}$

where $\rho(x:=d)$ is the valuation ρ' with

$$\rho'(y) = \rho(y) \text{ if } y \not\equiv x$$
$$= d \quad \text{ if } y \equiv x.$$

This definition is correct: by induction on P one can show the continuity of $\lambda d. [\![P]\!]_{\rho(x:=d)}$.

(ii) $M = N$ is *true in* D (notation $D \models M=N$) if for all ρ one has $[\![M]\!]^D_\rho = [\![N]\!]^D_\rho$.

Intuitively $[\![M]\!]^D_\rho$ is M interpreted in D where each λ-calculus application . is interpreted as $\cdot_F$ and each λ as λ^G. E.g.

$$[\![\lambda x.xy]\!]^D_\rho = \lambda^G d.d\ \rho(y)$$
$$= \lambda^G x.x\ \rho(y).$$

Informal notation. If a reflexive D is given and $\rho(y) = d$, then we will loosely write $\lambda x.xd$ to denote the more formal $[\![\lambda x.xy]\!]^D_\rho$.

Clearly $[\![M]\!]^D_\rho$ depends only on the values of ρ on FV(M). That is

$\rho \upharpoonright FV(M) = \rho' \upharpoonright FV(M) \Rightarrow [\![M]\!]^D_\rho = [\![M]\!]^D_{\rho'}$, where $\upharpoonright$ denotes function restriction. In particular for combinators $[\![M]\!]^D_\rho$ does not depend on ρ and may be written $[\![M]\!]^D$. If D is clear from the context we write $[\![M]\!]_\rho$ or $[\![M]\!]$.

3.2.7. THEOREM. *If* D *is reflexive, then* D *is a sound model for the* λ*-calculus, i.e.*

$$\lambda \vdash M = N \Rightarrow D \models M = N.$$

PROOF. Induction on the proof of $M = N$. The only two interesting cases are the axioms (β) and the rule (ξ).

As to (β). This was the scheme $(\lambda x.M)N = M[x:=N]$.

Now

$$\begin{aligned}
[\![(\lambda x.M)N]\!]_\rho &= (\lambda^G d.[\![M]\!]_{\rho(x:=d)}) \cdot_F [\![N]\!]_\rho \\
&= F(G(\lambda d.[\![M]\!]_{\rho(x:=d)}))([\![N]\!]_\rho) \\
&= (\lambda d.[\![M]\!]_{\rho(x:=d)})([\![N]\!]_\rho) \\
&\qquad \text{since } F\circ G = id, \\
&= [\![M]\!]_{\rho(x:=[\![N]\!]_\rho)}.
\end{aligned}$$

Sublemma. $[\![M[x:=N]]\!]_\rho = [\![M]\!]_{\rho(x:=[\![N]\!]_\rho)}$.

Subproof. Induction on the structure of M.

Write $P^* \equiv P[x:=N]$, $\rho^* = \rho(x:=[\![N]\!]_\rho)$.

M	$[\![M^*]\!]_\rho$	$[\![M]\!]_{\rho^*}$	comment
x	$[\![N]\!]_\rho$	$[\![N]\!]_\rho$	OK
y	$\rho(y)$	$\rho(y)$	OK
PQ	$[\![P^*]\!]_\rho \cdot_F [\![Q^*]\!]_\rho$	$[\![P]\!]_{\rho^*} \cdot_F [\![Q]\!]_{\rho^*}$	IH
$\lambda y.P$	$\lambda^G d.[\![P^*]\!]_{\rho(y:=d)}$	$\lambda^G d.[\![P]\!]_{\rho^*(y:=d)}$	$(\rho(y:=d))^* = \rho^*(y:=d)$

□ sub

By the sublemma the proof of the soundness of (β) is complete.

As to (ξ). This was $M = N \Rightarrow \lambda x.M = \lambda x.M$.

We have to show

$$D \models M = N \quad \Rightarrow \quad D \models \lambda x.M = \lambda x.M.$$

Now

$$\begin{aligned}
&D \models M = N \\
&\Rightarrow [\![M]\!]_\rho = [\![N]\!]_\rho \quad \text{for all } \rho \\
&\Rightarrow [\![M]\!]_{\rho(x:=d)} = [\![N]\!]_{\rho(x:=d)} \quad \text{for all } \rho, d \\
&\Rightarrow \lambda d.[\![M]\!]_{\rho(x:=d)} = \lambda d.[\![N]\!]_{\rho(x:=d)} \quad \text{for all } \rho \\
&\Rightarrow \lambda^G d.[\![M]\!]_{\rho(x:=d)} = \lambda^G d.[\![N]\!]_{\rho(x:=d)} \quad \text{for all } \rho \\
&\Rightarrow [\![\lambda x.M]\!]_\rho = [\![\lambda x.N]\!]_\rho \quad \text{for all } \rho \\
&\Rightarrow D \models \lambda x.M = \lambda x.N. \qquad \square
\end{aligned}$$

3.3. A concrete model: D_A

Now we will give an example of a reflexive complete lattice called D_A. The method is due to ENGELER [1981] and is a code free variant of the graph model $P\omega$ due to PLOTKIN [1972] and SCOTT [1973].

3.3.1. DEFINITION. (i) Let A be a set. Define

$$B_0 = A\ ,$$

$$B_{n+1} = B_n \cup \{(\beta,b) \mid b \in B_n \text{ and } \beta \subseteq B_n,\ \beta \text{ finite}\},$$

$$B = \bigcup_n B_n.$$

$D_A = P(B) = \{x \mid x \subseteq B\}$, considered as complete lattice under inclusion ($\subseteq$). The set B is just the closure of A under the operation of forming ordered pairs (β,b). It is assumed that A consists of urelements, that is, does not contain pairs $(\beta,b) \in B$.

(ii) Define $F: D_A \to [D_A \to D_A]$, $G: [D_A \to D_A] \to D_A$

by

$$F(x)(y) = \{b \mid \exists\, \beta \subseteq y\ (\beta,b) \in x\},$$

$$G(f) = \{(\beta,b) \mid b \in f(\beta)\}.$$

3.3.2. THEOREM. D_A *is reflexive via the maps* F,G.

PROOF. F and G are clearly continuous (use that the β's are finite). Moreover for continuous f

$$\begin{aligned} F \circ G(f)(y) &= F(\{\beta,b) \mid b \in f(\beta)\})(y) \\ &= \{b \mid \exists\, \beta \subseteq y \quad b \in f(\beta)\} \\ &= \cup_{\beta\subseteq y}\ f(\beta) \\ &= f(y) \end{aligned}$$

since $\sup = \cup$ in D_A and $y = \cup_{\beta\subseteq y}\ \beta$ is a directed supremum. Therefore

$$F \circ G(f) = f$$

and hence $F \circ G = id_{[D_A \to D_A]}$. □

Now a semantic proof of the consistency of the λ-calculus can be given.

3.3.3. COROLLARY. *The λ-calculus is consistent, i.e.* $\lambda \nvdash true = false$.

PROOF. Otherwise $\lambda \vdash x = y$; but then $D_A \models x = y$. This is not so, take $\rho(x) \neq \rho(y)$, in a D_A with $A \neq \emptyset$. □

The following definition and lemma are useful for the determination of $[\![M]\!]$ in D_A, and is taken from LONGO [1983].

3.3.4. DEFINITION. (i) For $b \in B$ the norm $|b|$ is defined inductively.

$$|b| = 1 \quad \text{if } b \in A\ ,$$

$$|(\beta,b)| = \max\{|c| \mid c \in \beta\} + |b| + 1\ .$$

(ii) For $x \in D_A$ define $x_n = \{b \in x \mid |b| \leq n\}$.
Write $|\beta| = \max\{|c| \mid c \in \beta\}$.

3.3.5. LEMMA. *For* $x, y \in D_A$ *one has*

(i) $(x_n)_m = x_{\min(n,m)}$;

(ii) $x = \cup_n x_n$;

(iii) $x_0 = \emptyset$;

(iv) $x_{n+1}\, y \subseteq (xy_n)_n$.

PROOF. (i), (ii), (iii) trivial.

(iv) $$\begin{aligned} x_{n+1}\, y &= \{ b \mid \exists \beta \subseteq y\ (\beta,b) \in x_{n+1} \} \\ &\subseteq \{ b \mid \exists \beta \subseteq y\ (\beta,b) \in x \text{ and } |\beta| \le n,\ |b| \le n \} \\ &= \{ b \mid \exists \beta \subseteq y_n\ (\beta,b) \in x \}_n \\ &= (xy_n)_n. \ \square \end{aligned}$$

3.4. References

Engeler, E.

[1981] Algebras and combinators, Algebra Universalis 13 (3), pp. 389 - 392.

Longo, G.

[1983] Set-theoretical models of λ-calculus: theories, expansions, isomorphisms, Ann. Pure Appl. Logic 24, pp. 153-188.

Plotkin, G.D.

[1972] A set-theoretical definition of application, School of Artificial Intelligence, Memo MIP-R-95, University of Edinburgh.

Scott, D.S.

[1973] Models for various type-free calculi, in: Suppes et al. [1973], pp. 157 - 187.

Suppes, P. et al. (= Suppes, P; Henkin, L.; Joja, A. and Moisil, Gr. C. (eds.))

[1973] Logic, methodology and philosophy of science IV, Studies in Logic 74 (North-Holland, Amsterdam).

CHAPTER 4

THEORY OF THE MODEL D_A

4.1. Böhm trees

For each $M \in \Lambda$ we will define a certain tree $BT(M)$, the so-called Böhm tree of M. Böhm trees will play an important role in the analysis of the model D_A.

4.1.1. Lemma. Each $M \in \Lambda$ is either of the form

(1) $$M \equiv \lambda x_1 \cdots x_n . y P_1 \cdots P_m , \quad n \geq 0, \ m \geq 0;$$

or

(2) $$M \equiv \lambda x_1 \cdots x_n . (\lambda y . P_0) P_1 \cdots P_m , \quad n \geq 0, \ m \geq 1.$$

Proof. If M is a variable, then M is of the form (1) with $n = m = 0$.

If M is an application term, then $M \equiv P_0 P_1 \cdots P_m$ with P_0 not an application term. Hence M is of the form (1) or (2) depending on whether P_0 is a variable or an abstraction term (and $n=0$).

If M is an abstraction term, then a similar argument shows that M is of the right form. ⊠

4.1.2. Definition. (i) A term M is a head normal form (hnf) if M is of the form (1) in lemma 4.1.1. In that case y is called the head variable of M.

(ii) M has a hnf if $M =_\beta N$ with N a hnf.

(iii) M is solvable if M has a hnf; otherwise M is unsolvable.

(iv) If M is of the form (2) in lemma 4.1.1., then $(\lambda y.P_0)P_1$ is called the head redex of M.

4.1.3. Examples. (i) $\mathbf{S} \equiv \lambda xyz . xz(yz)$ is a hnf;

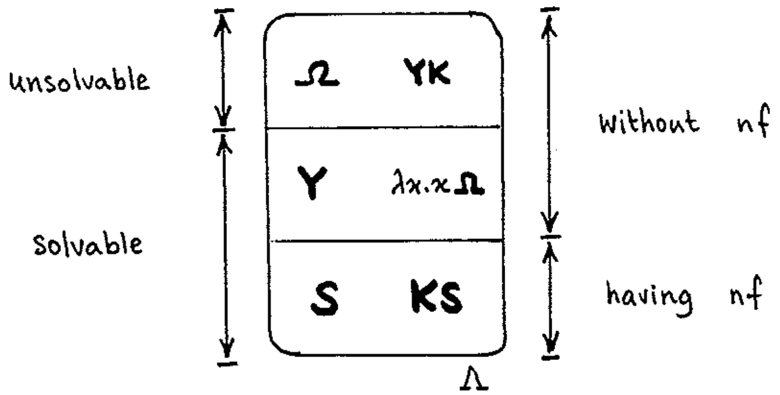
(ii) $\mathbf{Y} \equiv \lambda f . \omega_f \omega_f$ with $\omega_f \equiv \lambda x . f(xx)$ is not a hnf, but $\mathbf{Y}$ is solvable, since

$$\mathbf{Y} = \lambda f . f(\omega_f \omega_f).$$

Note that $\mathbf{Y}$ has no nf.

(iii) $\mathbf{\Omega} \equiv (\lambda x.xx)(\lambda x.xx)$ has no hnf.

4.1.4. Remark. Λ can be divided into three parts, indicated by the following figure. In each part some characteristic terms are given.



4.1.5. Proposition. Identification of all λ-terms without a nf leads to an inconsistent theory.

Proof. Note that

$$\lambda x.x\,\underline{\text{true}}\,\mathbf{\Omega} = \lambda x.x\,\underline{\text{false}}\,\mathbf{\Omega} \;\vdash\; \underline{\text{true}} = \underline{\text{false}}. \quad \boxtimes$$

4.1.6. Fact. All unsolvable terms can be identified such that the resulting theory is consistent. For example,

$$\lambda + (\mathbf{\Omega} = (\lambda x.xxx)(\lambda x.xxx))$$

is a consistent theory.

4.1.7. Lemma. If $M =_\beta M'$ and

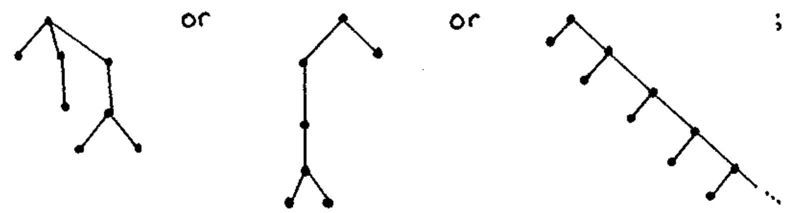
M has hnf $M_1 \equiv \lambda x_1 \cdots x_n.\, y\, P_1' \cdots P_m'$,

M' has hnf $M_1' \equiv \lambda x_1 \cdots x_{n'}.\, y'\, P_1 \cdots P_{m'}$,

then $n = n'$, $y \equiv y'$, $m = m'$ and $P_1 =_\beta P_1'$, $\cdots$, $P_m =_\beta P_m'$.

Proof. By the Church-Rosser theorem M_1 and M_1' have a common reduct L. But then $L \equiv \lambda x_1 \cdots x_{n''}.\, y''\, P_1'' \cdots P_{m''}''$ with $n = n'' = n'$, $y \equiv y'' \equiv y'$, $m = m'' = m'$ and $P_1 =_\beta P_1'' =_\beta P_1'$, $\cdots$. $\boxtimes$

4.1.8. Definition. (i) A tree is something like



that is, a partially ordered set with

(1) there is a root (least element);

(2) each node (point) has finitely many direct successors;

(3) the set of predecessors of a node is finite and linearly ordered.

Note that our trees grow upside-down!

(ii) A labelled tree is a tree with symbols at some of its nodes.

4.1.9. Definition. Let $M \in \Lambda$. The Böhm tree of M (notation $BT(M)$) is a labelled tree defined as follows.

$$BT(M) = \begin{array}{c} \lambda x_1 \cdots x_n .\, y \\ / \qquad \backslash \\ BT(M_1) \quad \cdots\cdots \quad BT(M_m) \end{array} \qquad \begin{array}{l} \text{if } M \text{ is solvable,} \\ M \text{ has as hnf} \\ \lambda x_1 \cdots x_n . y M_1 \cdots M_m \,; \end{array}$$

$$= \quad \cdot \quad \text{(just a root, no lable)} \qquad \text{if } M \text{ is unsolvable.}$$

4.1.10. Examples. (i) $BT(\mathbf{S}) = \begin{array}{c} \lambda xyz.\, x \\ / \quad \backslash \\ z \qquad y \\ \qquad\; | \\ \qquad\; z \end{array}$

(ii) $BT(\mathbf{\Omega}) = \cdot$

(iii) $BT(\mathbf{Y}) = \begin{array}{c} \lambda f. f \\ | \\ f \\ | \\ f \\ | \\ \vdots \end{array}$

This because $\mathbf{Y} = \lambda f.\, \omega_f \omega_f$ $\quad (\omega_f \equiv \lambda x. f(xx))$. But $\omega_f \omega_f = f(\omega_f \omega_f)$, so

$$BT(\mathbf{Y}) = \begin{array}{c} \lambda f. f \\ | \\ BT(\omega_f \omega_f) \end{array} = \begin{array}{c} \lambda f. f \\ | \\ f \\ | \\ BT(\omega_f \omega_f) \end{array} = \begin{array}{c} \lambda f. f \\ | \\ f \\ | \\ f \\ \vdots \end{array} .$$

4.1.11. Remark. Note that definition 4.1.9 is not an inductive definition of $BT(M)$. Indeed, $M_1 \cdots M_m$ may be more complicated than M itself, e.g. if $M =_\beta x(yM)$; in this case $BT(M)$ is

$$\begin{array}{c} x \\ | \\ y \\ | \\ x \\ | \\ y \\ | \\ \vdots \end{array}$$

4.1.12. Proposition. $BT(M)$ is well defined and if $M =_\beta N$ then $BT(M) = BT(N)$.

Proof. What is meant is that $BT(M)$ is independent of the choice of head normal forms. This and the second property follow from lemma 4.1.7. ☒

4.2. The approximation theorem

In this section we will show that for all $M, N \in \Lambda$

$$BT(M) = BT(N) \Rightarrow D_A \models M = N.$$

The main tool to show this is the so called approximation theorem, originally due to Hyland for the model $P\omega$. It tells us how the value $[\![M]\!]^{D_A}$ can be approximated from below by parts of the Böhm tree of M. We need some extra notation. Since the only model that is considered is D_A, we write $[\![-]\!]$ for $[\![\ ]\!]^{D_A}$.

4.2.1. Definition. (i) $\Lambda\perp$ is an extension of the set Λ by adding a constant $\perp$ to the formation rules:

$$\begin{array}{l} \perp \in \Lambda\perp \\ x \in Var \Rightarrow x \in \Lambda\perp \\ M, N \in \Lambda\perp \Rightarrow (MN) \in \Lambda\perp. \\ M \in \Lambda\perp,\ x \in Var \Rightarrow (\lambda x.M) \in \Lambda\perp. \end{array}$$

(ii) The term $\perp$ serves as a constant for ϕ: we extend $[\![\ ]\!]$ and set $[\![\perp]\!] = \phi$.

(iii) Reduction for terms in $\Lambda\perp$ is ordinary β-reduction extended with the contraction rules

$$\lambda x.\perp \ \rightarrow\ \perp,$$
$$\perp M \ \rightarrow\ \perp.$$

The resulting reduction relation is called *$\beta\perp$-reduction* (notation $\rightarrow_{\beta\perp}$, $\twoheadrightarrow_{\beta\perp}$ as usual).

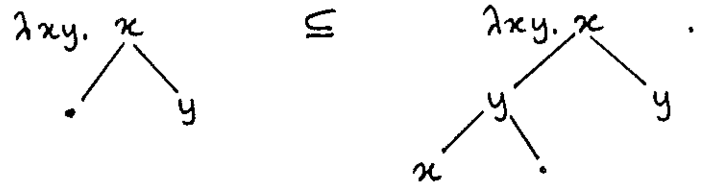
(iv) A term $P \in \Lambda\perp$ is in *$\beta\perp$-normal form* if P does not have a subexpression of the form $(\lambda x.R)S$, $\lambda x.\perp$ or $\perp Q$. P *has a $\beta\perp$-nf* if $P \twoheadrightarrow_{\beta\perp} P'$ for some P' in $\beta\perp$-nf.

(v) Böhm trees of $\Lambda\perp$-terms are defined by letting $BT(\perp) = \cdot$

4.2.2. Remarks. (i) Note that since $D_A \models \lambda x.\phi = \phi$ and $D_A \models \phi y = \phi$, $\beta\perp$-reduction preserves the value of a $\lambda\perp$-term in D_A.

(ii) If P has a β-nf, then P has also a $\beta\perp$-nf. This is because replacements of the form $\lambda x.\perp \rightarrow \perp$ and $\perp M \rightarrow \perp$ decrease the length of a term and do not create new β-redexes.

4.2.3. Definition. (i) Let A and B be Böhm trees of some terms. Then A is *included in* B (notation $A \subseteq B$) if A results from B by cutting of some subtrees, leaving an empty node. For example,



(ii) Let $P, Q \in \Lambda\perp$. P *approximates* Q (notation $P \sqsubseteq Q$) if $BT(P) \subseteq BT(Q)$.

(iii) Let $P \in \Lambda\perp$, and $M \in \Lambda(\perp)$. P is an *approximate normal form* (anf) of M if P is a $\beta\perp$-nf and $P \sqsubseteq M$.

(iv) $\mathcal{A}(M) = \{P \in \Lambda\bot \mid P$ is an anf of $M\}$.

4.2.4. Example. Consider the fixed point combinator

$$\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)).$$

Then $\lambda f.f\bot \sqsubseteq \mathbf{Y}$ (see example 4.1.10 (iii)), so $\lambda f.f\bot \in \mathcal{A}(\mathbf{Y})$. In fact

$$\mathcal{A}(\mathbf{Y}) = \{\bot, \lambda f.f\bot, \lambda f.f^2\bot, \dots\}.$$

We now state the

4.2.5. Approximation theorem. For $M \in \Lambda(\bot)$ one has

$$[\![M]\!]_\rho = \sup\{[\![P]\!]_\rho \mid P \in \mathcal{A}(M)\}.$$

The proof is postponed until 4.2.7.

4.2.6. Corollary. For all $M, N \in \Lambda$

$$BT(M) = BT(N) \Rightarrow D_A \models M = N.$$

Proof. By the approximation theorem,

$$\begin{aligned} BT(M) = BT(N) &\Rightarrow \mathcal{A}(M) = \mathcal{A}(N) \\ &\Rightarrow [\![M]\!]_\rho = [\![N]\!]_\rho, \end{aligned}$$

for all valuations ρ. ⊠

Longo [1983] has shown also the converse of corollary 4.2.6, so one has in fact

$$BT(M) = BT(N) \Leftrightarrow D_A \models M = N,$$

but this requires more work.

We now establish the proof of the approximation theorem 4.2.5. This occupies 4.2.7 - 4.2.15.

4.2.7. Lemma. Let $P, M \in \Lambda\bot$. Then

$$P \in \mathcal{A}(M) \Rightarrow [\![P]\!]_\rho \subseteq [\![M]\!]_\rho.$$

Proof. Note that M results (up to $=_\beta$) from P by replacing some $\bot$'s by other terms. Now the result follows by monotonicity of the "λ-calculus operations" in D_A. Example. Let $M =_\beta \lambda x.xM$. Then $\lambda x.x\bot \in \mathcal{A}(M)$ and

$[\![\lambda x.x\bot]\!] \subseteq [\![\lambda x.xM]\!]$. ⊠

The following "indexed λ-calculus" was introduced by Hyland and Wadsworth in order to prove the approximation theorem.

4.2.8. Definition. (i) The set of indexed $\lambda\bot$-terms (notation $\Lambda\bot^{\mathbb{N}}$) is defined by adding to the formation rules of $\Lambda\bot$

$$M \in \Lambda\bot^{\mathbb{N}},\ n \in \mathbb{N} \Rightarrow M^n \in \Lambda\bot^{\mathbb{N}}.$$

(ii) $[\![\ ]\!]$ is extended to $\Lambda\bot^{\mathbb{N}}$ by adding

$$[\![M^n]\!]_\rho = ([\![M]\!]_\rho)_n.$$

(For a definition of $(.)_n$ see definition 3.3.4.)

(iii) If $M \in \Lambda\bot^{\mathbb{N}}$, then $M^* \in \Lambda\bot$ is obtained from M by leaving out all indices.

(iv) Let $M \in \Lambda\bot^{\mathbb{N}}$. M is completely indexed if every subterm N of M^* has an index in M (i.e. occurs as part of N^n in M).

4.2.9. Example. $(\lambda x.(x^2x^3)^4)^3 \in \Lambda\bot^{\mathbb{N}}$ is completely indexed, but $(\lambda x.x^2x^3)$ and $((\lambda x.x^2x)^4)^5$ are not.

4.2.10. Definition. Indexed $\beta\bot$-reduction (notation $\to_i$, $\twoheadrightarrow_i$) is defined by the following contraction rules.

$$\begin{array}{lll}
(\lambda x.M)^0 N & \to_i & \bot; \\
(\lambda x.M)^{n+1} N & \to_i & (M[x:=N^n])^n; \\
\bot^n & \to_i & \bot; \\
\bot M & \to_i & \bot; \\
\lambda x.\bot & \to_i & \bot; \\
(M^n)^m & \to_i & M^{\min(n,m)}.
\end{array}$$

4.2.11. Lemma. Let $M, N \in \Lambda\bot^{\mathbb{N}}$ and $M \twoheadrightarrow_i N$. Then

(i) $N^* \sqsubseteq M^*$.

(ii) $[\![M]\!]_\rho \subseteq [\![N]\!]_\rho$.

(Note the difference in order.)

Proof. (i) Induction. The approximation appears because of the contractions $(\lambda x.M)^0 N \to_i \bot$.

(ii) By lemma 3.3.5 it follows that $\to_i$ preserves or increases the value of a term in D_A. ☒

4.2.12. Lemma. Each completely indexed term $M \in \Lambda\perp^{\mathbb{N}}$ $\twoheadrightarrow_i$-reduces to some $N \in \Lambda\perp^{\mathbb{N}}$ such that N^* is a $\beta\perp$-nf.

Proof. It is assumed that M is "minimally indexed", i.e. M does not contain subterms of the form $(P^m)^n$. (This can be achieved by contractions $(P^m)^n \to P^{\min(m,n)}$.)
M has a p-redex if $(\lambda x.R)^p S$ occurs in M. The order of M is the maximal p such that M has a p-redex; if M only contains redexes of the form $\perp^n$, $\perp M$, $\lambda x.\perp$, the order of M is 0. Now by induction on the order p of M the term N will be constructed.

Case $p=0$. Contractions of the form

$$\begin{aligned}
(\lambda x.R)^0 S &\to \perp,\\
\perp^n &\to \perp,\\
\perp M &\to \perp,\\
\lambda x.\perp &\to \perp
\end{aligned}$$

all decrease the length of a term. Hence after finitely many steps N can be found.

Case $p=n+1$. Replacing the rightmost p-redex $(\lambda x.R)^{n+1} S$ by $(R[x:=S^n])^n$ and then replacing terms $(S^n)^m$ by $S^{\min(n,m)}$ results in a term with one less occurence of a p-redex. (Typical example (some indices are left out):

$$\begin{aligned}
&(\lambda ab.baa)^{n+1}((\lambda x.x^{n+1}R)^{n+1}(\lambda z.z)^{n+2}) \to_i\\
&(\lambda ab.baa)^{n+1}(((\lambda z.z)^n)^{n+1}R) \to_i\\
&(\lambda ab.baa)^{n+1}((\lambda z.z)^n R).
\end{aligned}$$)

After a finite number of such steps the term is reduced to a minimally indexed one of order n. Now apply the induction hypothesis. ☒

4.2.13. Lemma. Let $M \in \Lambda\perp^{\mathbb{N}}$ be completely indexed. Then there exists an $N \in \Lambda\perp^{\mathbb{N}}$ such that

(1) $N^* \in \mathcal{A}(M^*)$;

(2) $[\![M]\!]_\rho \subseteq [\![N]\!]_\rho$.

Proof. By lemmas 4.2.12 and 4.2.11. ⊠

4.2.14. Definition. Let $M \in \Lambda\perp$. An indexing for M is a map I that assigns an element of $\mathbb{N}$ to each subterm of M. M^I is the resulting completely indexed term.

4.2.15. Lemma. Let $M \in \Lambda\perp$. Then

$$[\![M]\!]_\rho = \sup \{ [\![M^I]\!]_\rho \mid I \text{ indexing for } M \}.$$

Proof. Induction on the structure of M, using $x = \sup_n x_n$. ⊠

Now we can give the

Proof of the approximation theorem. Let $M \in \Lambda(\perp)$. In D_A we have

$$\begin{aligned} M &= \sup \{M^I \mid I \text{ indexing for } M\}, \text{ by lemma 4.2.15} \\ &\subseteq \sup \{ N \mid N^* \in \mathcal{A}(M)\}, \text{ by lemma 4.2.13} \\ &\subseteq \sup \{ N^* \mid N^* \in \mathcal{A}(M)\}, \text{ since clearly } N \subseteq N^* \\ &= \sup \{ N \mid N \in \mathcal{A}(M) \} \\ &\subseteq M, \text{ by lemma 4.2.7.} \end{aligned}$$ ⊠

4.3. Reference

Longo, G.

[1983] Set-theoretical models of λ-calculus: theories, expansions, isomorphisms, Ann. Pure Appl. Logic 24, pp. 153-188.