

# A new definition of correctness of expressions in lambda typed lambda calculus

**Citation for published version (APA):**

Bruijn, de, N. G. (1982). *A new definition of correctness of expressions in lambda typed lambda calculus*. Technische Universiteit Eindhoven.

**Document status and date:**

Gepubliceerd: 01/01/1982

**Document Version:**

Uitgevers PDF, ook bekend als Version of Record

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

M12

N = AUT84

Mex, land?

A new definition of correctness of  
expressions in d-typed  $\lambda$ -calculus

by

N.G. de Bruijn

Department of Mathematics and Computer Science

Eindhoven University of Technology

PO Box 513 5600 MB - EINDHOVEN

The Netherlands -

## Classification of points with respect to a given point in a binary ordered tree.

We consider a binary ordered tree, i.e. a tree where every point which is not an endpoint has a lefthand descendant and a righthand descendant. We draw such trees with root downwards, and the descendants of a point are drawn above the point. See figure 1 and 2.

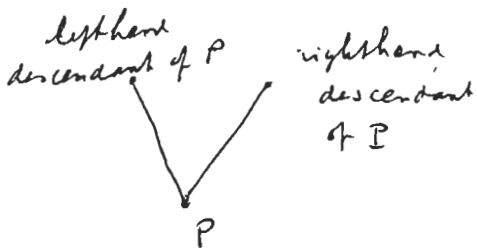


fig. 1

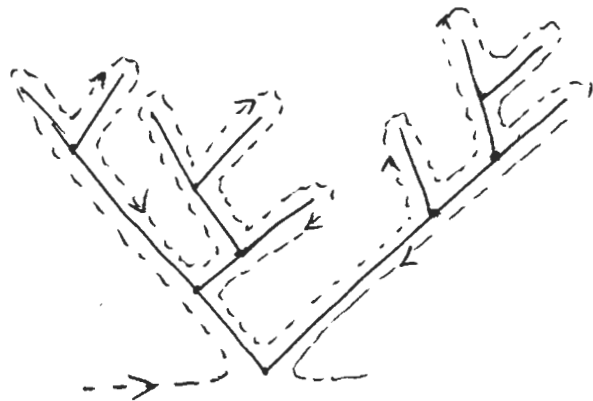


fig 2

In figure 2 we have also drawn (as a dotted line) the tree traversal from the left.

Such a tree traversal gives rise to three different linear orderings of the set of binary nodes. We call them left order, middle order and right order.

The basis of this distinction is that the tree traversal

gets to every binary node three times: the first time on the left, the second time in the middle (what botanists call the axil), the third time on the right.

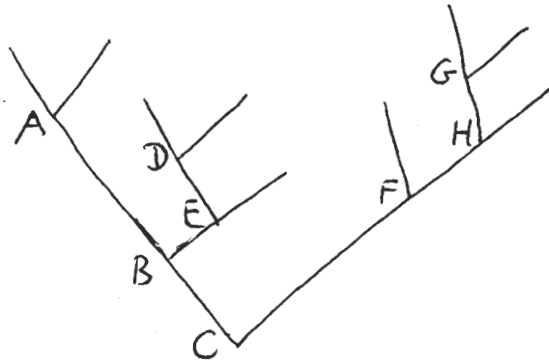


fig 3

In figure 3 the binary nodes are labelled with roman capitals. We can order them in the order in which the tree traversal meets them on the left:

C B A E D F H G (left order).

We can also put them in order by noting the moments when the tree traversal meets them in the middle:

A B D E C F G H (middle order).

Finally we can arrange them in the order the tree traversal meets them on the right:

A D E B G H F C (right order)

We can compose these three orderings in a single relation if we use composite symbols like

3  
a a a, a b a, b b a, etc. If P and Q are binary nodes then

$$P (a b b) Q$$

means that P comes after Q in the left order, before Q in the middle order, and before Q in the right order. Let us code the triples by

b b b = 0	a b b = 4
b b a = 1	a b a = 5
b a b = 2	a a b = 6
b a a = 3	a a a = 7

In figure 4 we have taken a point Q in a binary ordered tree, and at every other point we have given the code for its relation to Q. So if P bears the code number 3 then that means that  $P (b a a) Q$ .

We note that the code numbers 2 and 5 do not occur, and it is not hard to prove this in general. In figure 5 we have a tree where Q lies on the "left edge", and we do not get the code numbers 0 and 1 any more



AT $\lambda$  $\tau$ -trees.

The set  $Z$  of all AT $\lambda$  $\tau$ -trees, is defined by

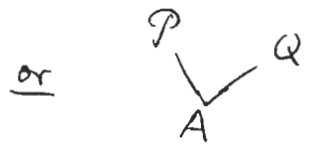
$$Z = Z_0 \cup Z_1 \cup Z_2 \cup \dots$$

where  $Z_0, Z_1, \dots$  are defined recursively as follows:

If  $k \in \{0, 1, 2, \dots\}$ , then an element of  $Z_k$  is:

either  $\tau$

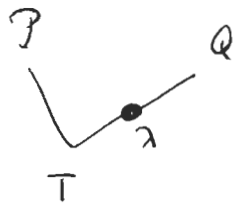
or an  $m$  where  $m \in \mathbb{N}, 1 \leq m \leq k$



or

where  $P \in Z_k, Q \in Z_k$

or



where  $P \in Z_k, Q \in Z_{k+1}$ .

Note that  $Z_0 \subset Z_1 \subset Z_2 \subset \dots$

A centered AT $\lambda$  $\tau$ -tree is a pair  $(P, M)$  where

$P \in Z_0$ , and  $M$  is a point of  $P$  with label  $A$ .

In figures this "center"  $M$  will be indicated by a square. See figure 7

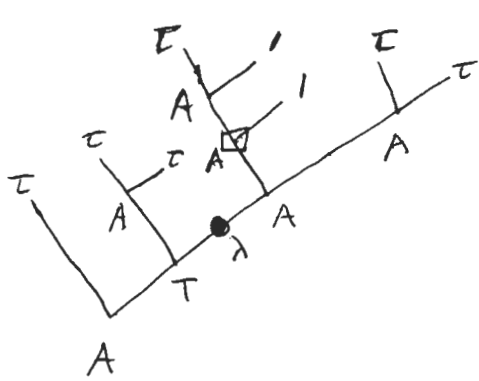


fig. 7.

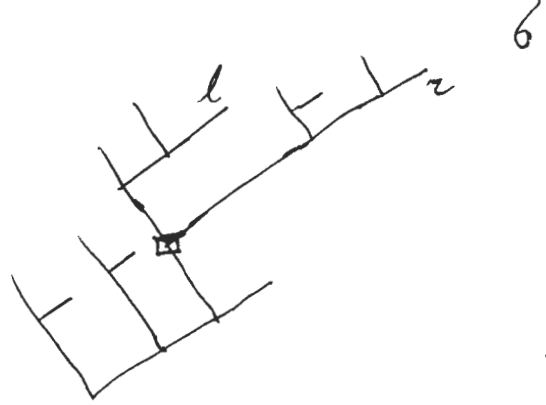


fig 8

The left index and right index of a centered AT $\tau$ -tree.

The right index is the endpoint we eventually reach if we move outward from the center, always taking the right-hand branch at every binary node. That endpoint is indicated by  $r$  in figure 8. The left index is the endpoint we get if we start at the center, take one step to the left and all further steps to the right. The left index is indicated by  $l$  in figure 8.

The operations leftcat and rightcat.

If at the left index we have the label  $\tau$  we say that leftcat is undefined. If at the left index we have an integer  $k$ , then this  $k$  refers to some  $\mathcal{A}$ . The  $T$  just below that  $\mathcal{A}$  has a subtree  $P$  on its left branch. Let  $Q$  be the tree we get from  $P$  if we add  $k$  to every integer occurring in  $P$ . Now the

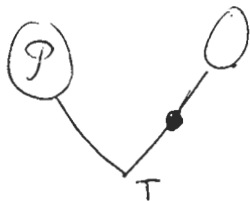




be the center. The center itself has no code number, so the A of the A-T $\lambda$  triple is not allowed to fall in the center.

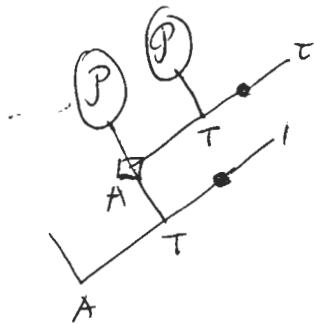
We note that an admissible  $\beta$ -reduction transforms a centered AT $\lambda$  $\tau$ -tree into a centered AT $\lambda$  $\tau$ -tree. (this statement would remain true if we would exclude as code numbers the 3's only).

A centered AT $\lambda$  $\tau$ -tree is called balanced if the righthand subtree of the center has the form



where P is identical to the lefthand subtree.

Example:



A centered AT $\lambda$  $\tau$ -tree  $W$  is called correctly centered if it has a left cat,  $V$  say, and if there is a finite sequence  $V = V_0, V_1, \dots, V_n$  such that for  $i = 1, \dots, n$  we get  $V_i$  from  $V_{i-1}$  by applying to  $V_{i-1}$  either the

rightmost operation or some admissible  $\beta$ -reduction,  
and such that  $V_n$  is balanced.

Definition of "correct AT $\lambda$ -tree".

A AT $\lambda$ -tree is called correct if it is an element  
of  $Z_0$ , and if for every point with label  $A$   
we get a balanced centered AT $\lambda$ -tree if we  
take that point as the center.

Language Theory.

The following items seem to be achievable by the  
methods thus far applied to related lambda calculi  
in particular the closure property seems to be quite  
accessible.

1. The Church-Rosser property for correct AT $\lambda$ -trees:

If in the set of all correct AT $\lambda$ -trees a relation of  
equivalence is defined as the reflexive, symmetric and transitive  
closure of the relation  $\geq_{\beta}$  ( $P \geq_{\beta} Q$  if we get  $Q$  by  
applying any  $\beta$ -reduction to  $P$ ), then equivalent trees  
have a common  $\beta$ -reduct.

2. The closure property: if  $P \geq_{\beta} Q$  and  $P$  is a correct AT $\lambda$ -tree, then  $Q$  is correct too.

3. The strong normalization property: If  $P$  is a correct AT $\lambda$ -tree then every reduction sequence

$$P = P_0 \geq_{\beta} P_1 \geq_{\beta} P_2 \geq_{\beta} \dots$$

terminates.

A consequence of 1. and 3. is that every correct AT $\lambda$ -tree has a unique normal form.

Note on  $\beta$ -reductions. The  $\beta$ -reductions used in the definition of correctness are not subject to any condition on type correctness: there is no condition that in



then should be any relation between  $P$  and  $Q$ .

"Long distance  $\beta$ -reduction" is the replacement of an integer by "the  $P$  of the  $A$  of the  $\lambda$  it refers to".

