M 12

N= AUT 84

Maryland?

A new definition of correctness of expressions in d-typed λ-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 downward, and the descendants of a point are drawn above the point. See figure 1 and 2.

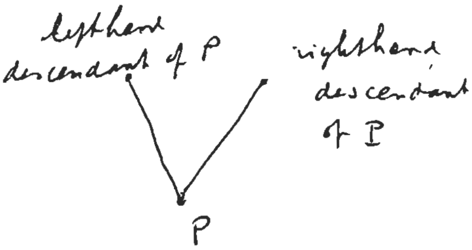


fig. 1

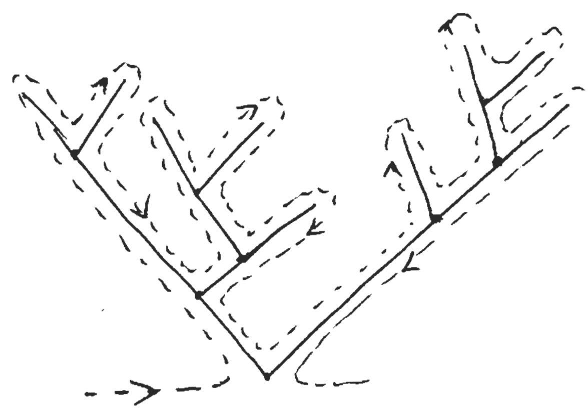


fig 2

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 <u>left order</u>, <u>middle order</u> and <u>right order</u>. 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.

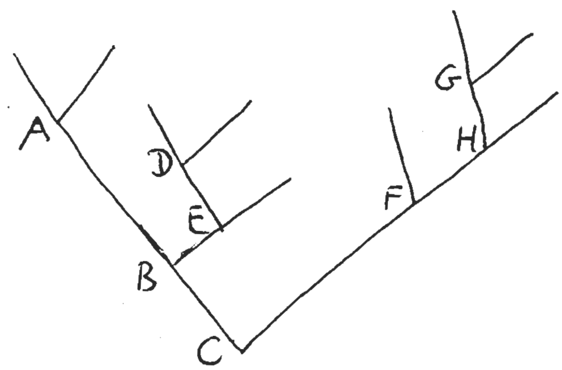


fig 3

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 aaa, aba, bba, etc. If P and Q are binary nodes then

$$P\,(abb)\,Q$$

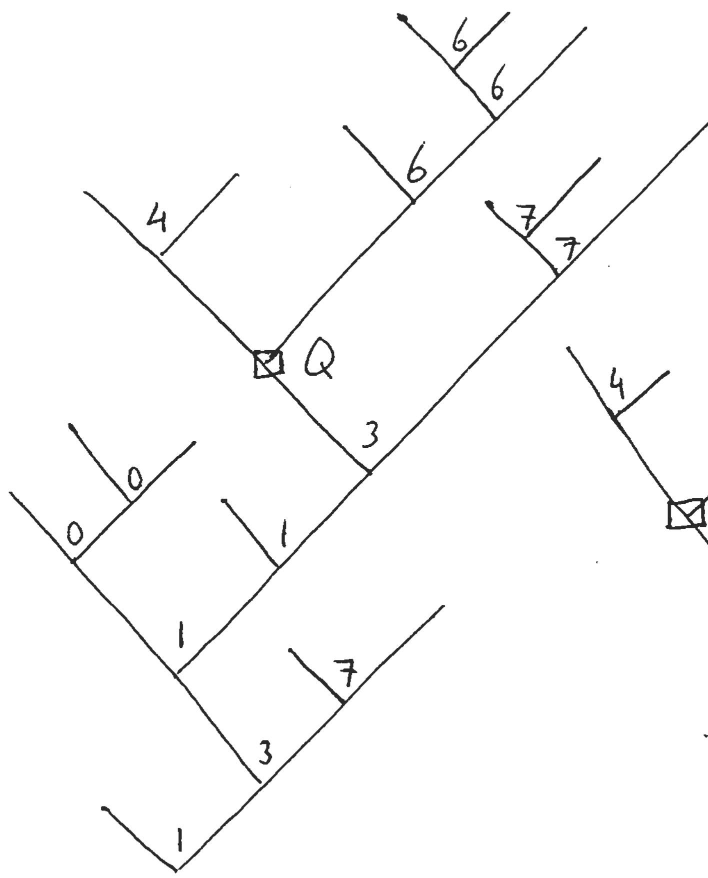
means that P comes after Q in the left order, before Q in the middle order, and before Q in the right order. Let us code the triples by

bbb	= 0		abb	= 4	
bba	= 1		aba	= 5	
bab	= 2		aab	= 6	
baa	= 3		aaa	= 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 $P\,(baa)\,Q$.

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

In figure 6 Q lies on the "right edge", and we do not get code numbers 3, 7. In figure 7 Q is the root, and we only get the code numbers 4 and 6.



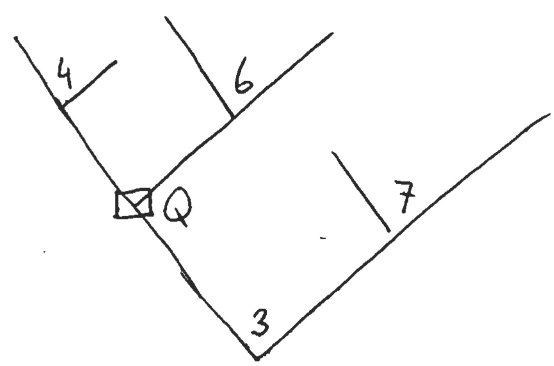
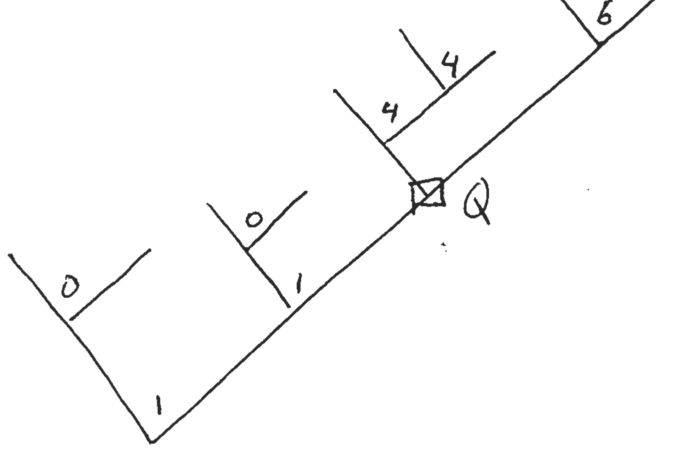


figure 5

figure 4



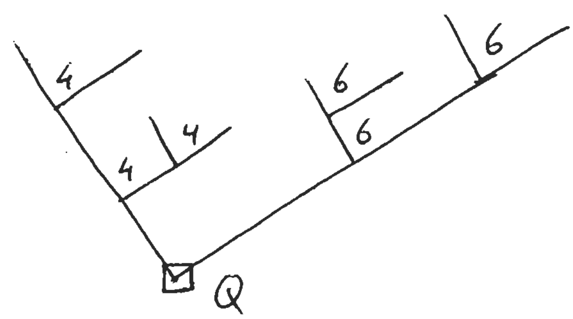


figure 6

figure 5

$AT\lambda\tau$-trees.

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

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

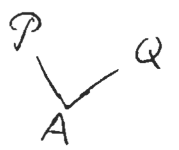
where $Z_0, Z_1, \ldots$ are defined recursively as follows:

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

<u>either</u> τ

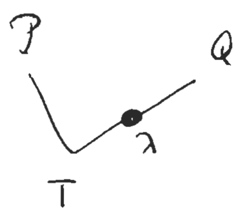
<u>or</u> an m where $m \in \mathbb{N}$, $1 \leq m \leq k$

<u>or</u>



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

<u>or</u>



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

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

A <u>centered $AT\lambda\tau$-tree</u> 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

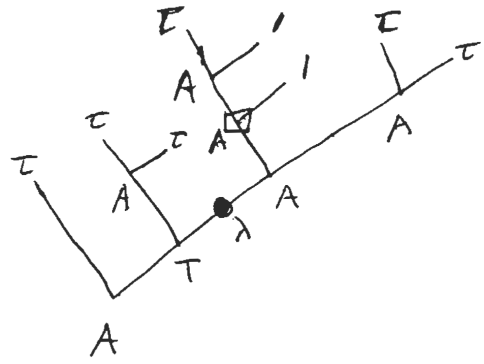


fig. 7.

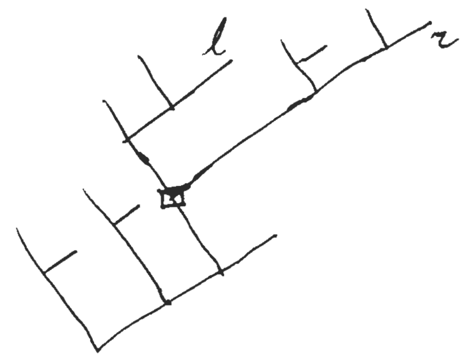


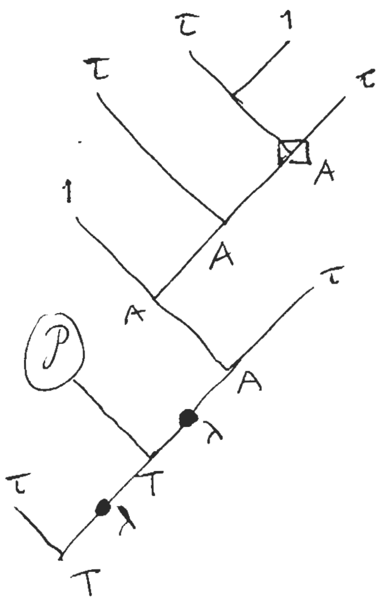
fig 8

The left index and right index of a centered ATλτ-tree.

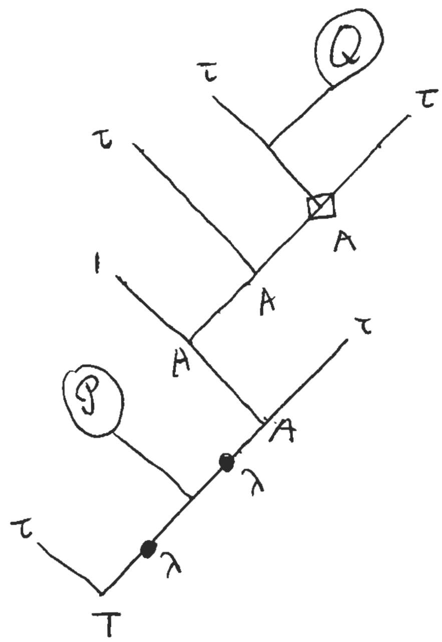
The right index is the endpoint we eventually reach if we move outward from the center, always taking the right-hand branch at every binary node. That endpoint is indicated by r in figure 8. The left index is the endpoint we get if we start at the center, take one step to the left and all further steps 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 τ we say that leftcat is undefined. If at the left index we have an integer k, then this k refers to some λ. The T just below that λ 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 leftcat is obtained by replacing the integer k at the left index by the tree Q. Example:



has leftcat



The definition of rightcat is obtained from the one of rightcat by replacing "leftindex" by "right index".

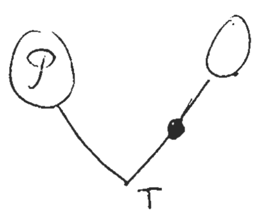
<u>Admissible β-reductions.</u>

A β-reduction is either a <u>shift</u> (of an A-T-λ triple) or a <u>long-distance reduction</u>. (see p. 10)

A β-reduction in a centered ATλτ-tree is called <u>admissible</u> if the A and the T of the A-T-λ triple have code numbers different from 3 or 7. The code number is taken in the sense of figure 4 if Q is taken to be the center. The center itself has no code number, so the A of the A-Tλ triple is not allowed to fall in the center.

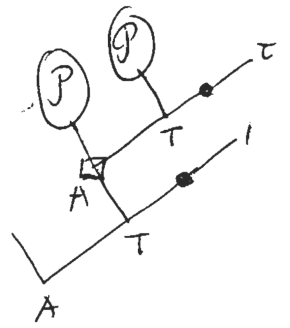
We note that an admissible β-reduction transforms a centered ATλτ-tree into a centered ATλτ-tree. (this statement would remain true if we would exclude as code numbers the 3's only).

A centered ATλτ-tree is called <u>balanced</u> if the righthand subtree of the center has the form



where P is identical to the lefthand subtree.

Example:



A centered ATλτ-tree W is called <u>correctly centered</u> if it has a leftcat, V say, and if there is a finite sequence $V = V_0, V_1, \ldots, V_n$ such that for $i = 1, \ldots, n$ we get V_i from V_{i-1} by applying to V_{i-1} either the rightcat operation or some admissible β-reduction, and such that V_n is balanced.

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

An $AT\lambda\tau$-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 $AT\lambda\tau$-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 $AT\lambda\tau$-trees: If in the set of all correct $AT\lambda\tau$-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 β-reduction to P), then equivalent trees have a common β-reduct.

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

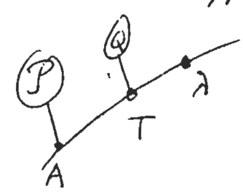
3. The strong normalization property: If $\mathcal{P}$ is a correct $AT\lambda\tau$-tree then every reduction sequence

$$\mathcal{P} = \mathcal{P}_0 \underset{\beta}{>} \mathcal{P}_1 \underset{\beta}{>} \mathcal{P}_2 > \dots$$

terminates.

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

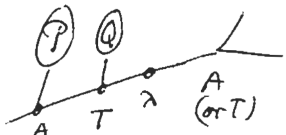
Note on β-reductions. The β-reductions used in the definition of correctness are not subject to any condition on type correctness: there is no condition that in



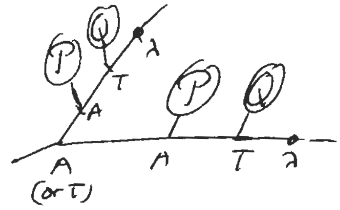
there should be any relation between $\mathcal{P}$ and Q.

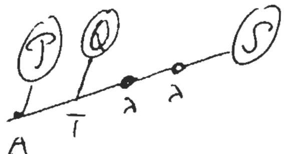
"Long distance β-reduction" is the replacement of an integer by "the $\mathcal{P}$ of the A of the λ it refers to".

Shifting means:

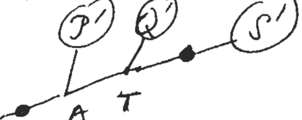


to





to



with adapted $\mathcal{P}$, Q and S, or

to

if $k = 1$

k if $k \neq$