

AUT 47

62-7/5

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics

Memorandum 1977- 4

Issued May 1977.

On the use of identifiers in AUT-PI.

by

I.Zandleven.

University of Technology  
Department of Mathematics  
P.O.Box 513,  
Eindhoven  
The Netherlands.

BIBLIOTHEEK MATHEMATISCH CENTRUM  
AMSTERDAM

On the use of identifiers in AUT-PI.

(refers to: AUT-PI language reference manual).

1. The rules.

a : variables have only alphanumeric identifiers.

b : constants may have alphanumeric or fix identifiers.

c : all constants, defined in one paragraph, must have mutually distinct identifiers. ('AND' is not equal to AND).

d : The sets of identifiers used for variables and constants, defined in one paragraph, must be disjoint.

e : variables may be redefined; the old variable will be lost.  
(except in specific circumstances, see 3.2).

f : all variables, occurring in a momentary context, must have mutually distinct identifiers.

A momentary context is:

a (current) context, possibly extended with a contextextension, possibly extended with expression abstractors (=telescope segments, segments of abstraction expressions).

the context segments (used for abstraction) generated by ABSTR are not incorporated in this notion.

g : identifiers for dummy binders in expression abstractors may be chosen freely, as long as they are not conflicting with rule f.

h : variables, occurring in expressions, may not be followed by a § reference.

2. The algorithm.

a) Looking for an identifier in an expression:

1. if the identifier is followed by a §-reference:

look in the specified § for a constant with this identifier.

2. if no § reference follows:

2.1 if it is a fix identifier:

look for it, following automatic reference, in the current §, next in the mother of this paragraph, etc.

references to future constants will not be seen.

(might be in the case of inserting text, but would be an error then).

2.2 if it is an alphanumeric identifier:

2.2.1 look in the momentary context for a variable,  
if not found there, follow automatic reference as in 2.1,  
looking for a constant.

b) looking for an identifier in a context base:

1. if a § reference follows:

look in the specified § for a variable with this identifier.  
Only the last defined version will be returned, which may not be a  
future definition.

2. if no § reference follows:

follow autoreference as in a) 2.1, looking for a variable.

Provisional restriction:

Only the last variable of a binderstring may be used as a context base.

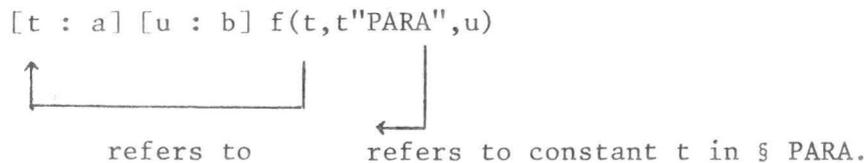
3. Special cases.

3.1 The forced § reference.

Suppose a momentary context contains the variable t, and we want to refer now  
to the constant t. Automatic reference would work in normal case, but now the  
system reads: variable t.

We may solve the problem by writing a § reference behind t, even if this  
constant is defined in the current §.

Example:



3.2 The regained variable.

Suppose we have the following book:

@ [a : x] [b : x] [c : x]	line 1
a@[b : y ] [d : y]	line 2
c@.... (f(a,b,c))...	line 3

The b, used in line 3 is a variable, occurs in its current context, and is thus: b of line 1.

Normal reference to this b is not possible, we are only able to regain it via c ( as long as c is not redefined).