SEMIPAL 2, an extension of the mathematical nottional language SEMIPAL

Notitie Nr 43 (1968-1969) 11 maart 1969.
Technological University Eindhoven
Department of Mathematics
Internal Report

AUT3

SEMIPAL 2, an extension of the mathematical notational language SEMIPAL.

by N.G. de Bruijn.

1. SEMIPAL is a simple language, or rather a notational system for the abbreviation of expressions. It was described in [2], sec. 4.3 - 4.8. In this note we shall extend that language by extending the notion of the indicator strings used in SEMIPAL.

In SEMIPAL every line represents the definition of a new identifier in terms of previously introduced ones, as far as the new identifier was not introduced as a variable or as a primitive notion. And every line has a context indicator, which is either 0 or a previous block opener. If the indicator of a line is x_n, and if, for every j $(1 \leq j \leq n)$, x_{j-1} is the indicator of the line whose identifier is x_j, and if 0 is the indicator of the line whose identifier is x_1, then $(x_1,\ldots,x_n)$ is called the indicator string of the given line.

With this definition of the indicator strings, the following property holds. If $(x_1,\ldots,x_n)$ and $(y_1,\ldots,y_m)$ are the indicator strings of two different lines, then there is an integer k $(0 \leq k \leq \min(n,m))$ such that $x_1 = y_1,\ldots,x_k = y_k$, and such that there is no further case where an x_i equals an y_j. In the extended form of SEMIPAL, to be discussed presently, this property no longer holds, and the indicator strings can no longer be obtained from single indicators.

In SEMIPAL 2, as we shall call this version, we shall write an indicator string in front of every line, and no longer a single indicator.

2. In order to facilitate our discussion, we number the lines by integers 1,2,3,...; the m-th line is called λ_m, the indicator string of λ_m is μ_m, the identifier part of λ_m is a_m, the definition part of λ_m is δ_m. The order in the indicator strings is, as is usually done in strings, referred to as

left to right, with left < right.

The following rules define SEMIPAL 2 uniquely.

(i) δ_m is either the symbol —, or the symbol PN, or an expression (see sec. 3).

(ii) a_m is called a block opener if δ_k = —.

(iii) μ_m is either 0 or a linearly ordered set of block openers taken from the sequence $(a_1,\ldots,a_{m-1})$. The order in μ_m need not be the same as in $(a_1,\ldots,a_{m-1})$.

(iv) If a_j occurs in μ_m, then μ_j is a subset of μ_m. Moreover, the order in $\mu_j \cup \{a_j\}$ is preserved in μ_m (the order in $\mu_j \cup \{a_j\}$ is defined by taking the existing order in μ_j, and requiring $b < a_j$ for all $b \in \mu_j$).

(v) If δ_m is an expression, then the pair (m,δ_m) is a "valid pair" (see sec. 3).

3.1 The notion "expression" (or "parentheses expression") is defined by recursion: if p is an identifier then p is an expression; if p is an identifier and if $\Sigma_1,\ldots,\Sigma_k$ are expressions, then $p(\Sigma_1,\ldots,\Sigma_k)$ is an expression.

3.2 We shall also define the notion of "valid pair" by recursion. The letters m,j,n,k will stand for integers.

(i) If $a_j \in \mu_m$ (whence a_j is a block opener), then (m,a_j) is a valid pair.

(ii) Assume that $1 \leqslant j < m$, and that a_j is not a block opener. Let n be the number of elements of μ_j, let k be an integer satisfying $0 \leqslant k \leqslant n$. Let $\Sigma_1,\ldots,\Sigma_k$ be such that $(m,\Sigma_1),\ldots,(m,\Sigma_k)$ are valid pairs. And assume that the first n - k entries of μ_j occur in μ_m. Under these circumstances (m,Σ) is also a valid pair, where

$$\Sigma = a_j(\Sigma_1,\ldots,\Sigma_k) .$$

These descriptions are to be modified in the obvious way if k = n, or k = 0, or both.

4. Examples. The SEMIPAL book in [2], sec. 4.1 (categories to be omitted) can be transformed at once into the following SEMIPAL 2 book:

0	nat	:=	PN
0	real	:=	PN
0	a	:=	—
a	b	:=	—
a,b	prod	:=	PN
0	n	:=	—
n	x	:=	—
n,x	power	:=	—
n	y	:=	—
n,y	d	:=	power(m,y)
n,y	e	:=	prod(d,y)
n,y	f	:=	prod(d,d)
n,y	g	:=	e(d)

The following example is not a line-by-line translation of a SEMIPAL book:

0	x	:=	—
0	y	:=	—
x,y	f	:=	PN
y	g	:=	f(y,y)
0	z	:=	—
z,y	k	:=	PN
z,x	l	:=	f(k(y))
x	w	:=	—
w,x	q	:=	f(w)

Remark 1.

In some cases (see f := PN) we define a thing in terms of what seems to be independent variables, in other cases (like q := f(w)) the variables seem to be less independent. There does not seem to be much of a reason to admit the latter possibility, but it will become significant if we wish to attach a category to each variable, like we do in PAL. In that case, the category of a variable w may depend on the variables occurring in the indicator string of the line where w is introduced.

Remark 2.

It may help the reader to write the identifiers (apart from the block openers) as functions of the variables in the indicator string:

0	x	:=	—
0	y	:=	—
	f(x,y)	:=	PN
	g(y)	:=	f(y,y)
0	z	:=	—
	k(z,y)	:=	PN
	l(z,x)	:=	f(k(y))
x	w	:=	—
	q(w,x)	:=	f(w)

Remark 3.

Note that the variable w (see above) has x in its indicator string, and that this has the effect that x has to occur in every indicator string containing w.

5. Completion.

If (k,Σ) is a valid pair, then we define the "completion" Σ' of Σ. (cf. [2], see 4.6).

(1) If Σ is a single block opener, then $\Sigma' = \Sigma$.

(2) Let $\Sigma = a_j(\Sigma_1,\ldots,\Sigma_k)$ (for notation see sec. 3.1), and let $b_1,\ldots,b_{n-k}$ be the first $n-k$ entries of μ_j. Then

$$\Sigma' = a_j(b_1,\ldots,b_{n-k}, \Sigma_1,\ldots,\Sigma_k),$$

with obvious modifications if $k = n$, or $k = 0$, or both.

6. Normal form.

Let (k,Σ) be a valid pair. We shall give a recursive definition of the "normal form" of Σ. It will become an expression Σ^* that again has the property that (k,Σ^*) is a valid pair.

(i) If Σ is a single block opener then $\Sigma^* = \Sigma$.

(ii) If Σ is not a single block opener, we first form the completion Σ' of Σ. Let

$$\Sigma' = a_j(\Sigma_1,\ldots,\Sigma_n)$$

(possibly with $n = 0$).

If $\delta_j = PN$, we define Σ^* by

$$\Sigma^* = a_j(\Sigma_1^*,\ldots,\Sigma_n^*),$$

where Σ_i^* is the normal form of Σ_i $(i = 1,\ldots,n)$.

If δ_j is an expression, then we first produce the normal form δ_j^* of δ_j. Let $\mu_j = (u_1,\ldots,u_n)$ be the indicator string of the j-th line. The expression δ_j^* may contain u_1 at various places, it may contain u_2 at various places, etc. We now replace in δ_j^* each of these u_1's by Σ_1^*, each of the u_2's by Σ_2^*, etc. (This substituion refers only to the u_1's, u_2's,... that were originally present in δ_j^*, and not to those u_i's that enter into the formula since they occurred in $\Sigma_1^*,\Sigma_2^*,\ldots$). This substitution procedure leads to what we call Σ^*.

Remark.

If both (k,Σ) and (m,Σ) are valid, then the normal form evaluated with reference to k is identical to the one evaluated with reference to m.

7. Every SEMIPAL book Λ turns into a SEMIPAL 2 book Λ' if we attach to every line an indicator string obtained from the indicators in the way described in sec. 1. The normal form of an expression in Λ (normal form as defined for SEMIPAL in [2, sec. 4.7]) will be identical to the normal form of the same expression as defined in SEMIPAL 2 by means of sec. 6 above.

8. Let Λ be a correct SEMIPAL 2 book. An expression Σ is called a Λ-expression if there is a k such that (k,Σ) is a valid pair (with respect to Λ).

Every Λ-expression has a uniquely defined normal form. Two Λ-expressions are called definitionally equivalent if they have the same normal form.

In order to test definitional equivalence it is not always necessary to evaluate the normal forms. One of the devices that will enable us to reduce the amount of work is the "reduced indicator string", to be discussed presently.

9. Reduced indicator strings.

If δ_j is an expression, we define the reduced indicator string $\tilde{\mu_j}$ as the string obtained from μ_j by deleting all variables that do not occur in the normal form of δ_j.

A correct book can sometimes be made incorrect if we replace all μ_j's by $\tilde{\mu_j}$'s. In the first place we might have to omit subexpressions whose position in an expression refers to a variable omitted from the corresponding indicator string. Secondly we should bear in mind that replacing

μ_m by $\tilde{\mu_m}$ might endanger the validity of rule (iv) of sec. 2.

10. We can extend PAL and AUTOMATH (described in [1], [2]) to PAL 2 and AUTOMATH 2 by means of the same rules on indicator strings as discussed for the case of SEMIPAL in this note.

AUTOMATH 2 has some obvious advantages over AUTOMATH. It can be easier to write, and may require a smaller number of lines. On the other hand, disadvantages of AUTOMATH 2 are:

(i) Writing indicator strings (and reduced indicator strings) can give more trouble than writing single indicators.

(ii) We loose the easy structure of a book, as illustrated by the "tree of knowledge".

Even if we write AUTOMATH instead of AUTOMATH 2, it may be very useful to evaluate the reduced indicator string for every line. This can of course be done by an AUTOMATH processor. By means of these reduced indicator strings, the processor can economize considerably on the amount of checking it has to do.

References:

1. N.G. de Bruijn : AUTOMATH, a language for mathematics. Report 68-WSK-05 (1968) Technological University Eindhoven, The Netherlands.

2. N.G. de Bruijn : The mathematical language AUTOMATH, its usage, and some of its extensions. To be published in the Proceedings of the Symposium on Automatic Demonstration (IRIA, Versailles, December 1968).