Description of the language Automath

de Bruijn, N.G.

Published: 01/01/1967

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's 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

Citation for published version (APA):

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?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 18. Dec. 2018
Description of the language AUTOMATH.

1. The grammar of the language consists of a set of rules according to which "books" can be written.

1.1 A book is a linearly ordered finite set of "lines". If we wish, these lines can be numbered: line 1, ... , line N. The rules of grammar will be rules for adding an (N+1)-st line to a book of N lines (and there is, of course, a rule for writing a first line). If these rules are obeyed, we say that the (N+1)-st line is acceptable.

The rules for acceptability of the (N+1)-st line make sense only if for each n (1 ≤ n < N) the (n+1)-st line is acceptable with respect to the book consisting of lines 1,...,n.

1.2 A line consists of five parts: an indicator, an identifier, a definition, a category. Occasionally we may add a fifth part, called a hint.

1.2.1 The hint is intended to assist the reader in his attempts to check whether the rules have been properly applied. The hints will be expressed by means of symbols different from those we are going to discuss next. The hint of a certain line plays its rôle only when checking that line; it is not to be consulted when checking later lines. For the time being we shall disregard the hints entirely.

1.2.2 The basic symbols of which the other parts are composed, are

(i) The seven separation marks, listed here:

., ( ) { } [ ]

(ii) Four basic symbols, listed here:

0 PN sort
Arbitrarily many other symbols to be called identifiers, mutually distinct, and distinct from the 11 symbols listed under (i) and (ii)

1.2.3 The identifier part of a line consists is a single identifier. It has to be different from the identifier part of any previous line. There would be no objection against systematic use of positive integers in such a way that the number \( n \) is the identifier part of the \( n \)-th line. However, in order to make books easier to read, and easier to compare with existing ways to express mathematics, one may prefer to choose more suggestive symbols like words, or words with numbers added to them. Note that an identifier is to be considered as a single symbol. It has already been stipulated that identifiers have to be distinct from the other basic symbols (see 1.2.2). In a printed text an identifier may be represented by a string of letters, digits or other signs, containing no separation marks.

1.2.4 The indicator part of a line is either the symbol 0 or the identifier of any previous line. The identifier of the \( k \)-th line may be used as indicator of a later line if and only if the definition part of the \( k \)-th line is the symbol — .

In the discussion of the language (not in the language itself) we often consider the "indicator string" of a line. We can describe it recursively. If the indicator is 0, then the indicator string is empty. If the indicator of the \( n \)-th line is the identifier part of the \( k \)-th line (whence \( k < n \)), then the indicator string of the \( n \)-th line is the indicator of the \( n \)-th line preceded by the indicator string of the \( k \)-th line.

Example: If the book consists of 8 lines, with identifiers 1, 2, 3, 4, 5, 6, 7, 8, and indicators 0, 1, 1, 0, 4, 5, 5, 4, then the indicator strings are

empty ; 1 ; 1 ; empty ; 4 ; 4, 5 ; 4, 5 ; 4.

If \( \sigma \) and \( \tau \) are strings, we write \( \sigma \subset \tau \) if either \( \sigma = \tau \) or \( \tau \) equals \( \sigma \) followed by some other string. If \( \sigma \) is any string (possibly empty), then a block is the set of all lines whose indicator string \( \tau \) satisfies \( \sigma \subset \tau \), plus the one line whose identifier equals the last entry in \( \sigma \).
In the above example the blocks are (2), (3), (6), (7), (2,3), (1,2,3), (5,6,7) (6,7) (4,5,6,7,8), (1,2,3,4,5,6,7,8), and they all represent sets of consecutive lines.

We do not always have this simple block structure. E.g., if the indicators are 0,0,0,3,3,2,2,5, then we have as blocks (1), (4), (6), (7), (8), (2,6,7), (5,8), (3,4,5,8), (1,2,3,4,5,6,7,8).

We can represent the indicators by means of an oriented rooted tree. The points of the tree are identifiers, apart from the point 0 that figures as the root of the tree. We draw a directed path from the identifier b to a point p if and only if p is the indicator of the line whose identifier is b.

1.2.5 The definition part of a line can be one of the following things:

(i) The symbol — (to be called "bar")
(ii) The symbol PN (to be called "primitive notion")
(iii) An expression. This is a certain string of symbols, consisting of separation marks and identifiers. If an identifier used in an expression occurring in the k-th line is not the identifier part of a previous line, then it is called a "bound variable". Although it is not strictly necessary, it is better to think of a bound variable as an identifier that does not occur as the identifier part of the k-th or any further line either.

We shall explain later how expressions should be built.

1.2.6 The category part of a line can be one of the following things:

(i) The symbol sort.
(ii) An expression. Everything said under 1.2.5 (iii) again applies here.
1.2.7 **Linguistic variables.** In our description of the language rules we shall use greek letters as linguistic variables. They denote identifiers, expressions or other parts of a book. They occur in general statements about the language, but do not appear literally in the books. In order to avoid confusion, we shall agree that identifiers (see 1.2.3) are composed entirely of symbols different from greek letters, so as to minimize confusion between the contents of the book on the one hand, and our discussion about the book on the other hand.

The following notation will be used in the sequel. If $j$ is a positive integer, we denote by

- $\lambda_j$ : the $j$-th line,
- $i_j$ : the indicator of $\lambda_j$,
- $\sigma_j$ : the indicator string of $\lambda_j$,
- $\beta_j$ : the identifier part of $\lambda_j$,
- $\delta_j$ : the definition part of $\lambda_j$,
- $\gamma_j$ : the category of $\lambda_j$.

1.3 **Primitive AUTOMATH language.** In order to facilitate the exposition we shall first describe a set of rules for a language that uses only a part of the complete AUTOMATH language. A book written in the primitive language (to be called PAL) will also be acceptable in the complete language (to be called AL). In PAL we use only

, ()

as separation marks, and we do not use bound variables.

PAL is an abbreviated form of a language LONGPAL. The latter has simpler rules, but has the practical disadvantage of very long expressions. We shall first define LONGPAL.

1.4 **Description of LONGPAL.** We start with an example of a book written in LONGPAL.
<table>
<thead>
<tr>
<th>line</th>
<th>indicator</th>
<th>indicator string</th>
<th>identifier</th>
<th>definition</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>0</td>
<td></td>
<td>elt</td>
<td>PN</td>
<td>sort</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>0</td>
<td></td>
<td>x</td>
<td>—</td>
<td>elt</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>x</td>
<td></td>
<td>y</td>
<td>—</td>
<td>elt</td>
</tr>
<tr>
<td>$\lambda_4$</td>
<td>y</td>
<td></td>
<td>x,y</td>
<td>PN</td>
<td>sort</td>
</tr>
<tr>
<td>$\lambda_5$</td>
<td>x</td>
<td></td>
<td>a</td>
<td>PN</td>
<td>is(x,x)</td>
</tr>
<tr>
<td>$\lambda_6$</td>
<td>y</td>
<td></td>
<td>y</td>
<td>—</td>
<td>is(x,y)</td>
</tr>
<tr>
<td>$\lambda_7$</td>
<td>u</td>
<td></td>
<td>x,y,u</td>
<td>—</td>
<td>elt</td>
</tr>
<tr>
<td>$\lambda_8$</td>
<td>z</td>
<td></td>
<td>x,y,u,z</td>
<td>—</td>
<td>is(z,y)</td>
</tr>
<tr>
<td>$\lambda_9$</td>
<td>v</td>
<td></td>
<td>x,y,u,z,v</td>
<td>—</td>
<td>is(z,x)</td>
</tr>
<tr>
<td>$\lambda_{10}$</td>
<td>u</td>
<td></td>
<td>x,y,u</td>
<td>b(x,y,u,y,a(y))</td>
<td>is(y,x)</td>
</tr>
<tr>
<td>$\lambda_{11}$</td>
<td>u</td>
<td></td>
<td>x,y,v</td>
<td>—</td>
<td>is(y,z)</td>
</tr>
<tr>
<td>$\lambda_{12}$</td>
<td>w</td>
<td></td>
<td>x,y,u,w</td>
<td>b(y,z,w,z,a(z))</td>
<td>is(z,y)</td>
</tr>
<tr>
<td>$\lambda_{13}$</td>
<td>w</td>
<td></td>
<td>x,y,u,w</td>
<td>b(x,y,u,z,b(y,z,w,z,a(z)))</td>
<td>is(z,x)</td>
</tr>
<tr>
<td>$\lambda_{14}$</td>
<td>w</td>
<td></td>
<td>x,y,u,w</td>
<td>b(z,x,b(x,y,u,z,b(y,z,w,z,a(z))),x,a(x))</td>
<td>is(x,z)</td>
</tr>
</tbody>
</table>

Lines $\lambda_{10} - \lambda_{14}$ rewritten in PAL

<table>
<thead>
<tr>
<th>line</th>
<th>indicator</th>
<th>indicator string</th>
<th>identifier</th>
<th>definition</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda'_{10}$</td>
<td>u</td>
<td></td>
<td>x,y,u</td>
<td>b(y,a(y))</td>
<td>is(y,x)</td>
</tr>
<tr>
<td>$\lambda'_{11}$</td>
<td>u</td>
<td></td>
<td>x,y,u</td>
<td>—</td>
<td>is(y,z)</td>
</tr>
<tr>
<td>$\lambda'_{12}$</td>
<td>w</td>
<td></td>
<td>x,y,u,w</td>
<td>c(y,z,w)</td>
<td>is(z,y)</td>
</tr>
<tr>
<td>$\lambda'_{13}$</td>
<td>w</td>
<td></td>
<td>x,y,u,w</td>
<td>b(z,d)</td>
<td>is(z,x)</td>
</tr>
<tr>
<td>$\lambda'_{14}$</td>
<td>w</td>
<td></td>
<td>x,y,u,w</td>
<td>c(z,x,e)</td>
<td>is(x,z)</td>
</tr>
</tbody>
</table>
1.4.1 We shall describe the rules in the following way. We first say when a book is called partly correct, next we give an addition law, i.e., a set of rules for adding a line to a partly correct book, and we show that this leads again to a partly correct book. Finally we call a book correct if the first line is correct, and if it can be obtained step-by-step by repeated application of the addition law.

1.4.2 A partly correct book is a set of lines $\lambda_1, \ldots, \lambda_N$ satisfying the following conditions:

(i) For $1 \leq j \leq N$ the indicator $t_j$ either is equal to $0$ or to some $\beta_i$ with $1 \leq i < j$. In the latter case it is required that $\delta_i = \cdots$.

(ii) We have $\delta_1 = \cdots$ or $PN$; $\gamma_j = \text{sort}$.

As a preparation to condition (iii) we define the notion PAL-expression.

We define it recursively. If $\beta$ is an identifier then $\beta$ is a PAL-expression. Moreover, if $\Sigma_1, \ldots, \Sigma_t$ are PAL-expressions, then $\beta \left( \Sigma_1, \ldots, \Sigma_t \right)$ is a PAL-expression.

We now phrase the last condition for a book to be partly correct:

(iii) If $1 \leq j \leq N$, then $\delta_j$ is either $\cdots$ or $PN$ or a PAL-expression. Moreover, $\gamma_j$ is either $\text{sort}$ or a PAL-expression.

1.4.3 We define indicator strings as in 1.2.4. Given a partly correct book $\lambda_1, \ldots, \lambda_N$, the term "admissible string" will either denote the indicator string of one of the lines, or the indicator string that a partly correct book $\lambda_1, \ldots, \lambda_{N+1}$ might have at its last line. That is, it might be the string obtained by adding $\beta_i$ to its indicator string $\sigma_i$, for an $i$ with $\delta_i = \cdots$. This string does not necessarily occur as the indicator string of one of the lines in the book.

*) This notion does not depend on having any particular book, but only on the presence of a set of distinct symbols called identifiers.
1.4.4 By $S_N$ we denote the set of all admissible strings of the book $\lambda_1, \ldots, \lambda_N$. By $P_N$ we denote the set of all P/N-expressions, that can be found from the identifiers $\beta_1, \ldots, \beta_N$.

Let $a$ be a string consisting of $k - 1$ different identifiers, taken from $\beta_1, \ldots, \beta_N$, and let $E_1 \in P_N, \ldots, E_k \in P_N$.

This operator transforms $P_N$ into itself, according to the following rule.

Let $A \in P_N$. This expression $A$ may contain the identifier represented by $\beta_i$ a number of times. We replace this identifier, wherever it occurs, by the symbol $E_i$. We repeat this procedure with $\beta_i \ldots, \beta_k$. Next we replace the symbols $E_1, \ldots, E_k$ by the expressions they denote. This defines $(\xi_1(A))A$. Note that $E_1, \ldots, E_k$ themselves may contain $\beta_1, \ldots, \beta_k$, and that the simple order: "replace $\beta_i$ by $E_i$ everywhere, $\beta_i$ by $E_i$ everywhere ...," would be quite confusing.

1.4.5 Admissible triples. Let the partly correct book $\lambda_1, \ldots, \lambda_N$ be given. We shall define the notion "admissible triple". The admissible triples all have the form $(\sigma, \lambda, \Gamma)$ where $\sigma \in S_N, \lambda \in P_N, \Gamma = \text{sort}$ or $\Gamma \in P_N$. For the definition of $\xi_1, \xi_N$.

We define the set of admissible triples recursively.

(i) If $\sigma \in S_N, (i)$ is such that $\beta_i$ is one of the entries of $\sigma$, then $(\sigma, \beta_i, \xi_1)$ is an admissible triple.

(ii) Let $1 \leq j \leq N$, and $\xi_j = \text{PN}$. We form the expression $\beta_j$. This $\beta_j$ is just $\beta_j$ if the indicator string $\sigma_j$ is empty. If it is not, and $\sigma_j \beta_1 \beta_2 \ldots \beta_k$, then $\beta_j = \beta_j(\beta_1, \ldots, \beta_k)$.

(That is, the expression $\beta_j$ is formed by writing the identifier denoted by $\beta_j$, an opening parenthesis, the identifier denoted by $\beta_1$, a comma, etc.).

Now we require: if $1 \leq j \leq N$, $\sigma \in S_N$, and $\sigma_j \preceq \sigma$ then $(\sigma, \beta_j, \xi_j)$ is an admissible triple.
(iii) Let \( k > 0 \), and let \( (\beta_1, \ldots, \beta_k) = \sigma \in \mathcal{P}_N \). Assume that \( (\sigma, \Lambda, \Pi), (\tau, \Sigma_1, \Gamma_1), \ldots, (\tau, \Sigma_k, \Gamma_k) \) are admissible triples. Moreover, assume that
\[
P_j = (\sigma(\Sigma_1, \ldots, \Sigma_k), \Lambda_j, (\sigma(\Sigma_1, \ldots, \Sigma_k))\Pi)
\]
Then also the following triple is admissible
\[
(\tau, (\sigma(\Sigma_1, \ldots, \Sigma_k))\Lambda, (\sigma(\Sigma_1, \ldots, \Sigma_k))\Pi).
\]

Examples. Consider the book \( \lambda_1, \ldots, \lambda_4 \) of 1.4. Take \( \sigma = x, y, u \). Then the following are examples of acceptable triples: \( (\sigma, x, \text{elt}) \), \( (\sigma, u, \text{is}(x, y)) \), \( (\sigma, \text{elt}, \text{sort}) \), \( (\sigma, b(x, y, u, y, a(y)), \text{is}(y, x)) \).

1.4.6 Addition law. Let \( \lambda_1, \ldots, \lambda_N \) be a partly correct book. The indicator of \( \lambda_{N+1} \) should be either 0 or a \( \beta_j \) with \( 1 \leq j \leq N \), \( \delta_j = -1 \). This defines the indicator string \( \sigma_{N+1} \).

Now we admit two cases for \( \delta_{N+1} \) and \( \gamma_{N+1} \):

(i) \( \delta_{N+1} = -1 \) or \( \Pi_N \), and \( \gamma_{N+1} \in \mathcal{P}_N \) is an expression with the property that \( (\sigma_{N+1}, \gamma_{N+1}, \text{sort}) \) is an admissible triple.

(ii) \( \delta_{N+1} \in \mathcal{P}_N \), \( \gamma_{N+1} \in \mathcal{P}_N \), such that \( (\sigma_{N+1}, \delta_{N+1}, \gamma_{N+1}) \) is an admissible triple.

Finally, the identifier \( \beta_{N+1} \) should be different from \( \beta_1, \ldots, \beta_N \), and different from the other basic symbols (see 1.2.2).

1.4.7 A book \( \lambda_1, \ldots, \lambda_N \) is called a correct LONGPAL book if it is partly correct (see 1.4.2) and if, for \( j = 2, \ldots, N \), the j-th line has been added to the book \( \lambda_1, \ldots, \lambda_{j-1} \) according to the addition law (see 1.4.6).

We mention, without proof, a few properties of a correct book.
1.5 Description of PAL. Having described LONGPAL completely, it is quite easy to say what PAL is. The difference lies only in the fact that the PAL-expression are abbreviated notations for the LONGPAL-expressions. Actually a book written in PAL can be translated into a book in LONGPAL by the simple procedure of replacing every $\delta_j$ and every $\gamma_j$ (if they are not --, PN or sort) by the LONGPAL-expressions they are abbreviations for. On the other hand, every book written in LONGPAL is also a correct book in PAL. It may be possible to abbreviate some of its expressions, but it is by no means an obligation to do so.

1.5.1 Our present description of PAL is given by means of LONGPAL. However, part of the practical value of PAL lies in the fact that it is possible to manipulate with the abbreviated expressions themselves, rather than translating into LONGPAL at every stage.

1.5.2 The expressions occurring in PAL are still of the form described in 1.4.2, and a book written in PAL still satisfies (i) of 1.4.7, but it does not necessarily satisfy (ii) of 1.4.

Let $\lambda_1, \ldots, \lambda_N$ be a book written in LONGPAL, and let $\sigma$ be an admissible string (see 1.4.3). We shall define (by recursive definition) an operator $T_\sigma$ that maps a certain subset of $P_N$ into $P_N$ (as in 1.4.4, this $P_N$ is the set of all PAL-expressions). $T_\sigma(\Sigma)$ will be called the normal form of $\Sigma$, and $\Sigma$ is called an abbreviation for $\sigma(\Sigma)$. 

(i) At the $j$-th line $\delta_j$ and $\gamma_j$ (if they are not --, PN or sort) are expressions containing only identifiers $\beta_1, \ldots, \beta_{j-1}$ (and of course parentheses and commas).

(ii) If $\beta_1$ occurs in an expression, then $\gamma_1$ is -- or PN.

(iii) If $(\sigma, \lambda, \Pi)$ is an admissible triple for the book $\lambda_1, \ldots, \lambda_N$, and if $\Gamma \neq \text{sort}$, then $(\sigma, \Gamma, \text{sort})$ is also an admissible triple for that book.
(i) Let $1 < j < N$, let $\delta_j = \ell$, and assume that $\beta_1$ is one of the entries of $\sigma$. Then $\beta_j$ is a PAL-expression, and we define
\[ T_\sigma(\beta_j) = \beta_j. \]

(ii) Let $1 < j < N$, let $\tau$ be a string such that both $\tau \in \sigma_j$ and $\tau \in \sigma$. Let $k$ be the length of $\tau$, and let $h + k$ be the length of $\sigma_j$. Let the string $\sigma_j$ consist of $\beta_1, \ldots, \beta_j, \beta_1, \ldots, \beta_j$. If $k > 0$, let $\Sigma_1, \ldots, \Sigma_k$ be expressions for which $T_\sigma(\Sigma_1), \ldots, T_\sigma(\Sigma_k)$ have already been defined. Let now $\Sigma$ be the expression $\beta_j(\Sigma_1, \ldots, \Sigma_i)$. Then we define
\[ T_\sigma(\Sigma) = (\Omega_{\sigma_j} (\beta_1, \ldots, \beta_j, \Sigma_1, \ldots, \Sigma_k)) \]
if $\delta_j$ is different from $-$ and $\sigma$, and
\[ T_\sigma(\Sigma) = (\Omega_{\sigma_j} (\beta_1, \ldots, \beta_j, \Sigma_1, \ldots, \Sigma_k)) \]
(\beta_j as defined in 1.45 (ii)) if $\delta_j = \sigma$.

In the case that $k = 0$, we define
\[ T_\sigma(\beta_j) = \delta_j \]
if $\delta_j$ is different from $-$ and $\sigma$, and
\[ T_\sigma(\beta_j) = \beta_j^* \]
if $\delta_j = \sigma$.

**Examples.** With the PAL-book of 1.4 we have
\[ T_{x,y,u}(a) = a(x) \]
\[ T_{x,y,u}(a(x)) = a(x) \]
\[ T_{x,y,u}(c) = b(y,a(y)) \]
\[ T_{x,y,u}(c) = b(y,z,w,z,a(z)) \]
\[ T_{x,y,u}(b(z,d)) = \delta_{13} \]
\[ T_{x,y,u}(a(z,x,e)) = \delta_{14} \]
1.5.3 Correct PAL book. A book is called correct in PAL if it can be obtained from a LONGPAL book $\lambda_1, \ldots, \lambda_N$ in the following way: If $1 \leq j \leq n$, and if $\delta_j$ is not or PN, then replace $\delta_j$ by any expression $\delta'_j$ which is such that $T_{\delta'_j}(\delta'_j) = \delta_j$, and similarly replace $\gamma_j$, if it is not sort, by some $\gamma'_j$ with $T_{\gamma'_j}(\gamma'_j) = \gamma_j$. Notice that $\delta'_j$ and $\gamma'_j$ are not uniquely defined, and that in particular $T_{\delta'_j}(\delta'_j) = \delta_j$, $T_{\gamma'_j}(\gamma'_j) = \gamma_j$.

Examples. We refer to 1.4 for the PAL version of a LONGPAL book. (lines $\lambda'_1, \ldots, \lambda'_14$).

1.6 AL and LONGAL. These languages have the same structure as PAL and LONGPAL. They are richer, since they admit expressions that do not exist in PAL or LONGPAL. Every PAL-book is also an AL-book, but not the other way round. The relation between AL and LONGAL is similar to the relation between PAL and LONGPAL. Again, any LONGAL-book is also an AL-book. And every AL-book can be considered to arise from a LONGAL-book by abbreviation of the expressions given as definitions and categories.

1.6.1 In 1.4.1 we defined the notion of PAL-expression. The definition of AL-expression will be similar. As in 1.4.1, the definition will be chosen rather liberally, and it is not automatic that every AL-expression can actually occur in an AL-book.

Again we have a set of distinct symbols called identifiers. Moreover we now have an infinite set $K$ of symbols which are mutually different, and differ from the identifiers. The elements of $K$ are called indeterminates. We shall use the word letter to denote something that is either an indeterminate or an identifier. The set of all letters is denoted by $S$. 
1.6.2 The notion of "AL-expression" is defined by

(i) If $\beta$ is a letter, then $\beta$ is an AL-expression.

(ii) If $k \geq 1$, if $\beta$ is an identifier, and if $\Sigma_1, \ldots, \Sigma_k$ are AL-expressions, then $\beta(\Sigma_1, \ldots, \Sigma_k)$ is an AL-expression.

(iii) If $\Sigma_1$ and $\Sigma_2$ are AL-expressions, then $\{\Sigma_1\}\Sigma_2$ is an AL-expression.

(iv) If $\Sigma_1$ and $\Sigma_2$ are AL-expressions, and if $\lambda$ is an indeterminate, then

$$[\lambda, \Sigma_1] \Sigma_2$$

is an AL-expression.

1.6.3 Free variables. To every AL-expression $\Sigma$ we shall assign a subset $U_{\Sigma}$ of $S$ to be called the set of free variables of $\Sigma$. We define it recursively.

(i) If $\beta$ is an identifier, then $U_{\beta}$ is empty.

(ii) If $\lambda$ is an indeterminate, then $U_{\lambda} = \{\lambda\}$.

(iii) If $k \geq 1$, if $\beta$ is an identifier, and $\Sigma_1, \ldots, \Sigma_k$ are AL-expressions, then the set of free variables of $\beta(\Sigma_1, \ldots, \Sigma_k)$ is the union of those of $\Sigma_1, \ldots, \Sigma_k$.

(iv) If $\Sigma_1$ and $\Sigma_2$ are AL-expressions, the set of free variables of $\{\Sigma_1\}\Sigma_2$ is the union of those of $\Sigma_1$ and $\Sigma_2$.

(v) If $\Sigma_1$ and $\Sigma_2$ are AL-expressions, and if $\lambda \in S$, then the set of free variables of $[\lambda, \Sigma_1]\Sigma_2$ is defined as

$$(U_{\Sigma_2} \setminus \{\lambda\}) U U_{\Sigma_1}$$

1.6.4 We shall describe LONGAL in just the same way as outlined in 1.4.1 for LONGPAL.

1.6.5 A LONGAL-book is called partly correct if conditions (i), (ii), (iii) of 1.4.2 hold, provided that (iii) is phrased with AL-expressions instead of PAL-expressions.
The indicator strings and admissible strings are defined as in 1.2.4 and 1.4.3.

1.6.7 Substitution operator. Let \( \sigma \) be a finite string of \( k \) different letters \((k > 1)\).
Let \( \Sigma_1, \ldots, \Sigma_k \) be AL-expressions. We shall define an operator \( \Omega_\sigma(\Sigma_1, \ldots, \Sigma_k) \) that maps a certain class of AL-expressions into AL-expressions. (For the special case that \( \sigma \) contains no indeterminates and that \( \Sigma_1, \ldots, \Sigma_k \) are PAL-expressions, the effect of the operator on PAL-expressions is the one described in 1.4.4).

We shall define \( \Omega_\sigma(\Sigma_1, \ldots, \Sigma_k) \) recursively.

(i) If \( \beta \) is a letter then \( \Omega_\sigma(\Sigma_1, \ldots, \Sigma_k) \beta \) equals \( \beta \) itself if \( \beta \) does not occur in the string \( \sigma \), and it equals \( \Sigma_i \) if \( \beta \) is the \( i \)-th element of the string.

(ii) If \( \beta \) is an identifier, if \( \Lambda_1, \ldots, \Lambda_l \) are AL-expressions \((l > 1)\), and \( \Lambda = \beta(\Lambda_1, \ldots, \Lambda_l) \), then

\[
\Omega_\sigma(\Sigma_1, \ldots, \Sigma_k) \Lambda = \beta(\tilde{\Lambda}_1, \ldots, \tilde{\Lambda}_l)
\]

where

\[
\tilde{\Lambda}_j = \Omega_\sigma(\Sigma_1, \ldots, \Sigma_k) \Lambda_j \quad (j = 1, \ldots, l).
\]

(It may arouse curiosity that \( \beta \) is transformed, but in our applications this \( \beta \) will not be a member of the string \( \sigma \).)

(iii) If \( \Lambda_1 \) and \( \Lambda_2 \) are AL-expressions, and \( \Lambda_2 = \{\Lambda_1\} \Lambda_2 \), then \( \tilde{\Lambda}_2 = \{\Lambda_1\} \Lambda_2 \), if

\[
\tilde{\Lambda}_j = \Omega_\sigma(\Sigma_1, \ldots, \Sigma_k) \Lambda_j \quad (j = 1, 2).
\]

(iv) Let \( \Lambda_1, \Lambda_2 \) be AL-expressions, let \( \lambda \) be an indeterminate, and
\( \Lambda_3 = [\lambda \Lambda_1] \Lambda_2 \). It is possible that \( \lambda \) occurs in the string \( \sigma \). We denote by \( \tau \) the string obtained from \( \sigma \) by deleting \( \lambda \) \((\text{so} \tau = \sigma \text{ if } \lambda \text{ does not occur})\) and by \( \Sigma_1, \ldots, \Sigma_k \) the string obtained from \( \Sigma_1, \ldots, \Sigma_k \) by deleting the \( \Sigma_j \) if \( \lambda \) happened to be the \( j \)-th entry of \( \sigma \). (Hence \( l = k \) or \( l = k - 1 \)).

We put
and we define

$$(\Omega_0 (\xi_1, \ldots, \xi_k)) \Lambda_3 = [\lambda_\Lambda_1] \Lambda_2^+.$$ 

1.6.8 **Admissible triples.** As in 1.4.5, we shall define a set of admissible triples with respect to a book $\lambda_1, \ldots, \lambda_N$ by recursion. The construction is more intricate in the present case, in particular since $N$ will be not constant during the recursion; the question whether some triples are admissible with respect to a given book will sometimes be answered by asking whether some slightly simpler triple will be admissible with respect to a slightly longer book.

We assume that $\lambda_1, \ldots, \lambda_N$ is a partly correct book. Let $P_N$ denote the set of all AL-expressions formed by means of $\beta_1, \ldots, \beta_N$ as identifiers. $S_N$ is the set of all admissible strings of $\lambda_1, \ldots, \lambda_N$. The admissible triples for the book $\lambda_1, \ldots, \lambda_N$ all have the form $(\sigma, \Lambda, \text{sort})$, where $\sigma \in S_N$, $\Lambda \in P_N$, $\Gamma = \text{sort}$ or $\Gamma \in P_N$. Actually it will turn out that $\Lambda$ has the empty set as their set of free variables. The same thing holds for $\Gamma$ if it is not $\text{sort}$.

We require

(i) (identical to (i) of 1.4.5)

(ii) (identical to (ii) of 1.4.5)

(iii) (identical to (iii) of 1.4.5)

(iv) Let $\Lambda \in P_N$, $\theta \in P_N$. Let $\lambda$ be an indeterminate. Then the triple $(\sigma, [\lambda, \Lambda], \theta, \text{sort})$ is admissible if the following conditions are both satisfied: (a) $(\sigma, \Lambda, \text{sort})$ is an admissible triple; (b) $(\sigma', [\lambda], (\beta_{N+1})\theta, \text{sort})$ is an admissible triple with respect to the book $\lambda_1, \ldots, \lambda_N, \lambda_{N+1}$. Here $\lambda_{N+1}$ is the line with identifier $\beta_{N+1}$, and with $\sigma_{N+1} = \sigma$, $\delta_{N+1} = \lambda$, $\gamma_{N+1} = \Lambda$, and $\sigma'$ is the string $\sigma + \{\beta\}$ (i.e. the string obtained by placing the extra element $\beta_{N+1}$ after the string $\sigma$). Finally, $\{\lambda\}$ stands for the string consisting of the element $\lambda$ only.
Let $A \in P_N$, $\theta \in P_N$, $\Sigma \in P_N$, and let $\lambda$ be an indeterminate. Then the triple $(\sigma, [\lambda_{\theta}], \Sigma, [\lambda_{\Sigma}])$ is admissible if both $(\sigma, A, \text{sort})$, $(\sigma, \Omega_{\lambda} (\beta_{N+1}^{\theta}), \Omega_{\lambda} (\beta_{N+1}^{\Sigma}))$ are admissible (for notation see under (iv)).

Let $A \in P_N$, $\theta \in P_N$, $\Sigma \in P_N$, let $\sigma$ be an admissible string, and let $\lambda, \mu$ be distinct variables. Assume that $(\sigma, \theta, \lambda)$ and $(\sigma, \Sigma, [\lambda_{\Sigma}])$ are admissible, we consider two cases:

(a) $\Sigma$ has the form $[\mu_{\Sigma}].$ Now we define that

$$(\sigma, \Omega_{\mu} (\theta))_{\Sigma}, (\Omega_{\lambda} (\theta))_{\Sigma}$$

is admissible.

(b) $\Sigma$ does not start with $[.$ Then we define that

$$(\sigma, \theta)_{\Sigma}, \Omega_{\lambda} (\theta)_{\Sigma}$$

is an admissible triple.

1.6.9 **Addition law.** This law is literally the same as the one of 1.4.6, although the meaning of the words and symbols is the one of sections 1.6.1-1.6.8 rather than the one of sections 1.4.1-1.4.5.

1.6.10 A book $\lambda_1, \ldots, \lambda_N$ is called a correct LONGAL-book if it is partly correct and if, for $j = 2, \ldots, N$, the $j$-th line has been added to the book $\lambda_1, \ldots, \lambda_{j-1}$ according to the addition law.

1.7. **Description of AL.** An AL-book is a LONGAL-book with abbreviated forms of the expressions, just like PAL abbreviates LONGPAL. The rules are those of 1.5 plus some extra rules:

(iii) If $\theta$ and $\Lambda$ are expressions for which $T_\sigma$ has been defined, and if $\Sigma = [\lambda_{\Theta}]_{\Sigma}$, then we define

$T_\sigma (\Sigma) = [\lambda, T_\sigma (\theta)] T_\sigma (E).$

(iv) If $\theta$ and $\Sigma$ are expressions for which $T_\sigma$ has been defined, if $\Gamma = \{\theta\}_{\Sigma}$, if $T_\sigma (\Sigma)$ has the form $[\mu, \Lambda]_{\Sigma}$, then we define
If $\theta$ and $\Sigma$ are expressions for which $T_\sigma$ has been defined, and if $T_\sigma(\Sigma)$ does not start with $[,$ then we define

$$T_\sigma(\{\theta\} \Sigma) = \{T_\sigma(\theta)\} T_\sigma(\Sigma).$$