STUDIES ON THE LOGIC OF AUTOMATIC COMPUTATION

(Incremental Data Assimilation in
San - Computer Systems)

by Liccello A. Lombardi

Working Paper 13-83
STUDIES ON THE LOGIC OF AUTOMATIC COMPUTATION

(Incremental Data Assimilation in Man - Computer Systems)

by Lionello A. Lombardi

Massachusetts Institute of Technology

SUMMARY

The main problem of modern computation theory and methodology arises from the fact that conventional digital computers, developed following the classical ideas of Turing and von Neumann, fail to meet many requirements as components or terminals of complex man-computer information system textures. Their main limitation in such context is sometimes identified as their exceedingly high needs regarding the specificity of both the algorithms that they can accept for execution, and their data, which makes them not primarily suitable as organs for incremental data assimilation through adaptively growing and incrementally modifiable algorithms. Accordingly, basic research is being carried out on designing new foundations for the logic of automatic digital computation.

This paper consists of two preliminary probes into the idea of the "incremental declarative computer" as basis for a possible solution to this problem. The first part is devoted to developing the syntax of a programming language for such computer based on a revision of Karl Megler's notation. Though it has been recently decided to discard this notation and replace it by a substantially new one, to which the human is more immediately responsive, still
this first part should be considered as an early version of the
first chapter of the book devoted to the development of a new
computation philosophy, that this author is writing. The second
part (corresponding to the second chapter of the book) presents
a new kind of memory organization based on the ideas of Shaw,
Shaw and Simon, but where such ideas are revised in a way to
enable the computer to scan symbolic expressions embedded in
lists right to left (i.e., in the extended linearization sense),
which would not be easy if the IFIV, LISP or FLPL list-structured
memory organization were adopted.

These are here no hints to the other chapters, devoted to
the design of the control unit of the new computer and recursive
functions of symbolic expressions, its input-output system,
algorithms to co-ordinate the information flow, random accessing
and information retrieval, and identification of parameters to
describe computer systems quantitatively, respectively.

Chambers, Massachusetts

January 30, 1963
In the present section, we shall consider a situation in which a family of \( n \) persons is given a set of tasks. The tasks are divided into two types: those that involve direct interaction with the environment and those that require processing information. The direct interaction tasks are characterized by a high degree of variability, while the processing tasks are more uniform in nature. The tasks are performed in a sequence, and the order of tasks can be changed. The family members are instructed to perform the tasks as efficiently as possible. The tasks are designed to test the ability of the family members to work together and to adapt to changing conditions. The tasks are designed to be challenging, but they are also meant to be enjoyable. The family members are encouraged to work together and to communicate effectively. The tasks are designed to be flexible, and the family members are allowed to modify the tasks as needed.
There are an infinite number of possible solutions to $x^2 + 1 = 0$.

To find the roots of such a polynomial, we need to consider complex numbers in addition to real numbers. This is because there are some quadratic equations that do not have real roots.

There are two roots for this equation: $x = i$ and $x = -i$, where $i$ is the imaginary unit defined as $i = \sqrt{-1}$. These roots satisfy the equation $x^2 + 1 = 0$.

Depending on the context, we may use these roots in various mathematical applications. In real-world problems, these roots can represent physical quantities or solutions that are not directly observable in the real world.
Any expression of the form $\sum_{i=1}^{n_1} a_i x_i$, where $a_i$ are coefficients and $x_i$ are variables, is a linear form. In order to ensure that the resulting equations are consistent, the following conditions must be satisfied:

1. The number of equations must be equal to the number of variables.
2. The coefficients of the variables must be such that the equations can be solved.

A particular case of this expression is a linear equation. When $n_1 = 1$, the expression simplifies to $a_1 x_1 = b$, which is a linear equation in one variable. The solutions to such equations can be found using various methods, such as substitution or elimination. The process of finding solutions involves manipulating the equations to isolate the variable of interest.

In the case of a system of linear equations, the solution process involves finding values for the variables that satisfy all the equations simultaneously. This can be achieved by using methods such as Gaussian elimination or Cramer's rule. The outcome of solving such a system will provide the values of the variables that make all the equations true.
For ex-sapientia, in each case an order, called order, as follows:

The level is the order.

The level is a form that is defined by construction by the order.

The level is a form that is defined by the order.

Definition b. (recursion clause) No aggregate is a form unless its being a form follows from a finite number of instances of the level.
only prove that no finite sequence of instances of definitions 1, 2, and 3 could prove them to be true.

Definition 1 says that, for each aggregate which is a form, there must be at least one proof, called a structural proof, of its being so, consisting of a finite sequence of instances of definitions 1, 2, and 3, such that each instance of definition 1 is always applied to a constant or an operative mark and each instance of definition 2 or 3 is applied to the concatenation, enclosure, or parenthesis, respectively, of aggregates which have been proved to be forms as result of instances of definitions 1, 2, or 3 preceding it in such proof.

We shall always assume that the last statement of any structural proof of an expression g is true if g is a form.

Consider subsequences of statements of the structural proof of a consisting of statements each of which, excluding the first, applies to at least one aggregate which is stated to be a form by the preceding statement of the subsequence, and those last statements of the last one of the subsequences. If a statement does not belong to any such subsequence, it is evidently irrelevant to the proof. We shall always assume that structural proofs do not contain such irrelevant statements.

The number of occurrences of marks in an aggregate g is denoted L(g).

An aggregate consisting of the total occurrences of marks of a form having at least n occurrences of marks is referred to as a partial form.
Let us first consider the following:

Then, by induction on the level of $A$,

Theorem 2: If $A$ is in the form $\mathcal{A}_2$, then

Proof:

Let us first consider the first case of $A$, the trivial case for $A$. Assume that the statement is true for all cases. Then, by induction on the level of $A$, we have
if definltioa 2 or .3. B-fuction
say e-
cccitaiaiag a
3s a
i'orsi of
l-svei
n<\;:
-•:
•:»«
eaclosirrf
In
psrv::m,.h'r^
ovei
tor
b'iC.suGe
«?=
it*
tie
ref<ui
f-
of
corjc-j-s^eTiciv::;
-.ffis
e.**
sad
ssj-,
one
of
wiilch^B&y
ej^,
has
velev
ri
m;
?;!-;^-,
oay
e,,
en-taioa
a
end
has
level
n.
The
rt-irtfiij
nder
ox"
cae
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
Theorem 4.1. If \( g \) is a form, then for all \( b, c \):

\[
0_f (g, b) \leq C_f (g, b, c)
\]

Proof. Again (3), the case if the level \( f \) in \( g \) is \( b \). Let us repeat the arguments and reasoning of the proof of Theorem 2 until the point of defining the forms (2), (3) and (4).

In both cases

\[
0_f (g, b) \leq C_f (g, b, c)
\]

by comparison. The contribution of \( g \) and of all \( g \) different from \( g \) in such a manner that the variance of \( f \) and \( f \) is zero. In the second case the occurrence of an expression limiting \( g \) to the left contributed a unit at the representation of (3a) and none to the right one. From these facts the above follows, like the previous one.

Analogously one can prove

Theorem 4.2. If \( g \) is a form, then for all \( c, d, e \):

\[
0_f (g, b) \leq C_f (g, b, c, d, e)
\]

Let \( g \) and \( b \) be aggregates, and let \( g \) be another aggregate consisting of a sequence of consecutive occurrences of marks in \( g \). Then the aggregate obtained by replacing \( b \) for \( a \) in \( g \) will be called substitute \( f \) of \( b \) for \( a \) in \( g \).
Theorem 4. If \( a \) and \( b \) are forms and \( c \) is an occurrence of an atom in \( a \), then \( S(b, a) \) is a form.

Proof. Let \( b \) be the atom occurring in \( a \). Consider in any one structural proof of \( b \), the instance of definition 1, section 1.2, stating that \( b \) is a form, and insert right after the above occurrence a structural proof of \( b \). Then replace graphically \( b \) for \( c \) in all statements of the structural proof following such insertion, and then remove all irrelevant statements from the result, if there are some. The resulting sequence of statements is a structural proof of \( S(b, a) \), which is therefore a form.

The reason for keeping the statement that \( c \) is a form instead of simply replacing \( c \) with a structural proof of \( b \) is that there might be occurrences of \( c \) in \( a \) other than \( b \).

So far the concept of level of a form has been envisaged as dependent on the proof that one can give for an aggregate in \( a \). This proof is not necessarily unique, and it is easy to give examples where more than one proof is possible. Consequently, the level is not a priori unique, but it is necessary for further development to give a proof of its uniqueness.

Before doing this, however, it is handy to introduce a new concept: we shall call left depth and right depth, respectively, of the occurrence of a mark different from a parenthesis in an
1.0

(by theorem 3, neither depth $\alpha$ nor negative weight $\mu$ is 0.)

Theorem 1: the last non-negligible occurrence of a relation $e$

The main consequence of this theorem is the fact that it provides a guarantee of the uniqueness of the level of any $e$.

Proof: Notice first that each occurrence of definition 1 in section 1.1 yields a form of relation to which therefore has no occurrence of root bases and whose only atom has consequence depth 0. Each occurrence of definition 2 in section 1.2, increases forms where the maximum level of occurrences of $e_1$ is the maximum of the one of the form those are concatenated as fact, by lemma 1 applied to any two forms $e_1$ and $e_2$, the concatenation of these does not yield an occurrence of any atom in either $e_1$ or $e_2$. Each occurrence of definition 7 in section 1.2, increases by one the depth of all affected occurrences of atoms. Then all three definitions yield the same generation or variation both of maximum depth and level.

This applies in particular to any sequence of occurrences of definitions 1, 2, and 3, section 1.2, constituting a structural proof of $e$ being the theorem.
If \( a \) denotes an aggregate, let \( a^{-1} \) denote the inverse aggregate of \( a \) consisting of the occurrences of \( a \) which compose \( b \), but in the reverse order and where open parentheses are replaced by closed ones in all of their occurrences, and vice versa. Then there subsists the following

**Theorem 6:** If \( a \) is a form, so is \( a^{-1} \)

**Proof:** In fact, a structural proof for \( a^{-1} \) can be constructed by simply inverting the order of all couples of forms which are concatenated by occurrences of definition 2, section 1.2., in the structural proof of \( a \).


From Section 1 of the preceding text, we have:

\[ M(e) = 1 \]  

and

\[ \text{canonical decomposition of } e \text{ is } e. \]

Furthermore, for the sake of simplicity, we assume that:

\[ \text{there are no such occurrences.} \]

Now, let us consider the situation where there are no such occurrences, and we have:

\[ e \text{ is decomposed into canonical elements.} \]

We define:

\[ g(e) = \text{the decomposition of } e. \]

From the preceding text, we have:

\[ g(e) = e. \]  

and

\[ \text{canonical decomposition of } e \text{ is } e. \]
of such $p$ as to allow one as large of $g$ containing $g_j$. A more formal way is the occurrence of a form of depth $d_j$ occurs at the end of some right of $p$, in $g_j$ constant as called $d_j$. Let us denote $d_t$ or $d_r$, the sharpened, which would prove the theorem. If the supposed they were v, in the case it cannot be an occurrence of definition 2. However, let $g_j$ be a form because it is the concatenation of the some $g_j$ or $g_j$, not call of the occurrence of a form having each concatenation, which must be in $g_j$ otherwise $d_t$ or $d_r$ could not be the first after an inclusion $g_j$ concatenating $g_j$ in there $g_j$ that have depth 0 in $g_j$. In fact, $g_j$ would cause $g_j$ would a solution of a means of $d_j$. Would that $d_j$ and $g_j$ must consequently have the same depth of $g_j$. In $g_j$ many constant forms of depth 0 are constructed in the theorem.
Let us call $gB^r$ the $I$-element consisting of the enclosure $\alpha$ and present $\psi$ of the $c$-elements of taxation of the $\alpha$-elements.

For example, these are $\alpha$-elements containing any amount of elements, consequences the negative side to usage. Let $t$ be chosen.

A systematic approach should also be formulated; such methods are essentially forms in the sense of a system. An evaluation of such systems.

A strong variable in the context of $b$-elements may be, in general. In addition $1$-elements in the control of the $e$-elements, while computations progress.

and $e$-elements are for $g$-elements as is available on the $w$-side of the $b$-elements.
The proof can immediately be extended to the case where $y$ is an occurrence of any one or several forms in $\phi$ or $\psi$. If each form has several components, this proof would still hold. The theorem would still be true but should be proved in a different way.

If $\phi$ is a form, the largest of all values of the subscripts of the variable letters of $\phi$ will be denoted $B(\phi)$.

Let $\phi$ and $\psi$ be forms, and suppose that

$$B(\phi) = B(\psi) \quad (2)$$

Suppose further that $\psi$ has a total of $x$ ($=0$) occurrences of
partial derivatives of such functions left to rigor below. The \( x_j \)'s are to denote the partial derivatives of \( x_j \)'s. In a manner new form functions to denote \( x_j \)'s. In this way the new variable \( x_j \)'s \( (x, x_j) \) replace the \( x_j \)'s. In other words the called partial

\[(x, x_j)\]

By Theorem 1, the partial derivatives of this form is a form.

Furthermore

\[ V(\xi, \eta) = W_{\xi} \] (2)

and

\[ \phi(\xi, \eta) = \phi(\eta) - 1(\xi) \] (3)

For example, let \( \gamma \) be \( \gamma_1, \gamma_2 \) and \( \delta \) be \( \delta_1, \delta_2 \),

\[ (\xi, \eta) = (\beta_1, \beta_2) \text{ and } (\eta, \xi) = (\gamma_1, \gamma_2) \] (a)

\[ (\xi, \eta) = (\gamma_1, \beta_2), \eta_2, \xi_2, \beta_1, \gamma_1) \] (b)

Further, and considered as proof of form of this is another

For instance, this is also consistent form. that is, partial derivatives

Equation (3) yields the derivatives.

And for partially of partial derivatives of form there can be

two occurrences of a single variable letters. In this paper they
are both the result of replacing two occurrences of a single

variable letters which originally are contained under each of

or more in all equal terms, partially partial derivatives

The numbers of single and symbols or indicate matters.
Theorem 5. (First, and only, theorem) \( \text{If } y \neq x \),

Thus and the eventually terms \( y \neq x \), and \( \exists i \in \mathbb{N} \) \( P(x_i, y_i) \)

and \( y = y(x, i) \) as \( x \) and \( y \).

\[ P(y(x, i)) = 0 \quad (5) \]

when

\[ (x, y(x, i)) = (x, y(x, i)) \quad (6) \]

Proof. \( U_0 \) does not contain any other letters without reference to reference letters and the same as the ones above, and \( U_0 \) has a form \( x \rangle \langle y \rangle \langle z \rangle \).

We will then consider a form by considering all possible terms, and the difference among all by taking the form in which the terms are identical to them. Only be such a consequence, one of the variables which \( y \) is replaced by \( x \) the letter \( (x, y) \) after \( (x, y) \), which is obtained the form \( y \) is replaced by \( x \) the letter \( (x, y) \) after \( (x, y) \), but not \( (x, y) \) after \( (x, y) \).

In particular the right member of \( (x) \), \( y \) is first replaced by \( 1_0 \), and then each occurrence of \( y \) is replaced by \( (x, y) \) within this replacement is replaced by \( (x, y) \) a letter of \( 1_0 \), depending on \( (x, y) \) a letter of \( 1_0 \). In this case the results are obviously that each letter term from the same letter takes on the same
data, which are simply differently located when the action is taken.

Consider now the case $i > H(b)$. In this case, in order to compute the left member of (5), $y$ is replaced by $(x, i-M(b \mid a) + H(b \mid a))$, that is by $(x, i-M(b) + H(b) + H(a) - M(a))$, while in order to compute the right member of (5) it is first replaced by $(x, i-M(b) + H(b))$, to compute $(\frac{y}{p} \mid b)$; then, because of (4), it is finally replaced by $(x, i-M(b) + H(b) - M(a) + H(a))$. These replacements are identical, and the theorem is thus proved.
The increase of craters is enhanced by the property of the preserved material which goes beyond the property of other preserved materials. The assumption of a constant increase by a property that is always true (I) cannot be supported by the results I have obtained. The increased crater formation is not limited to a constant increase, but rather a decrease. In a series of experiments, the crater formation decreases with time. The results of these experiments indicate that the crater formation is not constant but decreases with time.

This suggests that additional factors have an influence on the crater formation that were not considered before. The observed decrease of crater formation could be influenced by the experimental conditions. Thus, the crater formation is a complex process, and to determine the factors involved, further experiments are needed.
For most practical purposes, in this particular theory, detachent is useful when (2), section 1.5, is satisfied, and its associativity is utilized only under the hypotheses of theorem 2, section 1.5, which also allow for a trivial proof. However, for the sake of completeness, this section is devoted to defining this basic detachent operation and to proving its associativity in the most general case, that is, to proving the following.

**Theorem 1 (Associativity Theorem)** If $a$, $b$ and $c$ are forms, then

$$ (c, (b, a)) = ((c, b), a) $$

(1)

**Proof:** Let $x$ be a non-negative integer, and let $G(x)$ denote the infinite sequence of variable letters, separated by commas

$$ (x_{1+1}, (x_{2+1}), (x_{3+1}), \ldots \ldots \ldots \ldots ) $$

(2)

Let $g$ be a form, and let $g'$ denote the sequence of forms, separated by commas, whose first $h(x)$ elements are the canonical components of $g$, and the following are the elements of the sequence $G(h(x))$. For $1 < h(x)$, consequently, the $j$-th element of $g'$ is $(x_{h(x)}, x_{h(x)+j})$. Sequences obtained from forms like $g'$ will be called infinite forms and $g'$ will be called the standard extension of $g$.

The formalization of this concept can immediately be obtained, for example, by embedding infinite forms into the class of infinite sequences of forms obtained by replacing "finite or infinite" and "finite" in definition 1, section 1.2. The concept of canonical...
Let us call a form or an infinite form and let \( \mathcal{R}_g \) be the smallest non-negative integer \( i \) such that

\[
\mathcal{R}(g, i) = \mathcal{R}(L_g, i)
\]

We will then call contraction of \( g \) and denote \( \overline{g} \) the form

\[
L(g, g(\mathcal{R}_g))
\]

If two forms \( g \) and \( b \) have identical infinite extensions, they have also identical contractions and will be called similar. The similarity relation of forms, denoted \( g \sim b \), is reflexive, commutative and transitive, and classes of equivalence under such relation can be represented by either the common infinite extension or the common contraction of their members. The components of a form \( g \) which are not also components of \( b \) will be called \( \text{trajectory components or dummy components of } g \). Two similar forms can only differ by the number of \( \text{trajectory components} \).

If \( g \) and \( b \) are contracted forms, that is, forms without \( \text{trajectory components} \), then \( b | (a) \) is also contracted.

If \( a, b \) and \( c \) are forms such that \( a \sim b \), then

\[
(a | c) \sim (b | c)
\]  \quad (3)

and

\[
(c | a) \sim (c | b)
\]  \quad (4)

Let \( c \) and \( d \) be infinite forms. The partial detachment

\[
(c | d)
\]

is defined like in section 1.5 for forms, with the only difference that the limitation (2), section 1.5, does not apply in this case.
The partial development of infinite forms is not only always assumed to be also evidently associative, as can be proved by examining the first part of Theorem 2, section 2. Furthermore, for any two forms \( a \) and \( b \),

\[
\left[ \frac{a}{b} \right] = \frac{|a|}{|b|}
\]

Then, for any three terms \( a, b \) and \( c \)

\[
\left[ \frac{a}{b} \right] = \left[ \frac{b}{c} \right] = \left[ \frac{c}{a} \right]
\]

\[
\left[ \frac{c}{b} \right] - \left[ \frac{b}{a} \right] = \left[ \frac{c}{b} \right] - \left[ \frac{a}{b} \right] = \left[ \frac{c}{b} \right] - \left[ \frac{b}{c} \right] = \left[ \frac{b}{c} \right] - \left[ \frac{c}{b} \right] = \left[ \frac{a}{b} \right] - \left[ \frac{b}{a} \right] = \left[ \frac{a}{b} \right] - \left[ \frac{b}{a} \right]
\]

Hence, by contracting,

\[
\left[ \frac{c}{b} \right] - \left[ \frac{b}{a} \right] = \left[ \frac{c}{b} \right] - \left[ \frac{a}{b} \right] = \left[ \frac{b}{c} \right] - \left[ \frac{a}{b} \right] = \left[ \frac{b}{c} \right] - \left[ \frac{b}{a} \right] = \left[ \frac{a}{b} \right] - \left[ \frac{b}{a} \right] = \left[ \frac{a}{b} \right] - \left[ \frac{b}{a} \right]
\]

We may now prove that the two members of (5) have the same number of trailed components, that is, that they are equal. To see this, let us denote \( \alpha \) a constant which never occurs in \( a, b, c \), and let us call \( a, b, c \) the new forms obtained by substituting for each trailing component, say \( (x, y) \), of \( a, b, c \) respectively, one form

\[
((x, y))
\]

The forms \( a, b \) and \( c \) are contracted, and, by substituting them for \( a, b \) and \( c \) respectively in (5), since the decomposition of contracted forms is contracted, we obtain

\[
\left[ \frac{a}{b} \right] = \left[ \frac{b}{a} \right] = \left[ \frac{c}{b} \right] = \left[ \frac{b}{c} \right] = \left[ \frac{a}{b} \right] = \left[ \frac{b}{a} \right]
\]

Let us now replace \( (x, y) \) for all corresponding forms of the type (6) contained in (7). Because of the way \( \alpha \) was chosen, this replacement transforms the left-hand right member of (7) into \( \left[ \frac{a}{b} \right] \) and \( \left[ \frac{b}{a} \right] \), respectively; hence (5), and the theorem is completely proved.
Remark. The proof of the associativity postulate given in this section, which is based on a combination of examples, has the disadvantage of formally requiring the postulate of the existence of Ors in all sets of demonstrably many marks, that is, it is based on the postulate that certain operations cannot be represented by any automaton, and consequently one could in principle that the associativity holds only provided that the space while space of forms is embedded in an appropriate space of power \( n \). In order to show that associativity holds necessarily of such embedding, one should prove independently of the above postulate, that is, one should give a proof in the finite of associativity. However this attempt does not falling within the scope of this work, it will not here be the object of further elaboration.

Remark 2. Let \( A \) be the empty expression which can be represented, for example, by \((x,1)\), which is similar to it. Then for all forms \( A \) 

\[
\left( (1,2) \right) \cdot 2
\]

and

\[
\left( (2,1) \right) \cdot 2
\]

In other words, if we consider the family of all classes of equivalence by similarity of forms, then under the binary operation induced by detachment, this family is a subgroup having the class of equivalence of the empty object both left and right. See Clifford and Preston.
Section 1.7 Parametric Detachment

Let us associate to each form \( a \) an integer parameter \( \bar{m}(a) \) such that

\[
\bar{m}(a) \geq \bar{m}(a)
\]  

(1)

Whenever a form \( a \) is newly introduced without specifying its parameter, this one is taken to be \( \bar{m}(a) \).

Resuming the notation used in section 1.5 to define partial detachment, let us consider the new form obtained from \( a \) by replacing, for \( y_{\alpha} \) \( (1 \leq \alpha \leq b) \), the new variable letter \( (y_{\alpha} - \bar{m}(a) + \bar{m}(a)) \) if

\[
i_{\alpha} = \bar{m}(a),
\]  

(2)

do also the \( i_{\alpha} \)-th canonical component \( x_{\alpha a} \) of \( a \). This new form differs from \( (a |_{\alpha} b) \) only by the variable letters modifying (2), which in this case have a subscript which exceeds the one they would have in \( (a |_{\alpha} b) \) by \( \bar{m}(a) - \bar{m}(a) \).

This new form is called parametric partial detachment of \( b \).

For \( a \) with parameter \( \bar{m}(a) \) and notation

\[
\left( \begin{array}{c} b \\ \bar{m}(a) \\ \end{array} \right)
\]

and \( \bar{m}(b) \) is defined as

\[
\bar{m}(b) = \bar{m}(a) + \bar{m}(a)
\]  

(3)

The associativity of partial detachment of forms can be extended to parametric partial detachment of forms, thus obtaining the following,
Theorem 1: If \( g, h \) and \( a \) are free in each then

\[
\begin{align*}
    h(b) &\geq h(a) \\
    h(g) &\geq h(b) \\
    h(b) - h(b) &\geq h(b)
\end{align*}
\]

then

\[
\left( \begin{array}{c}
    \frac{\partial^2 h(b)}{\partial H^2} \\
    \frac{\partial^2 h(g)}{\partial H^2}
\end{array} \right) = \left( \begin{array}{cc}
    \frac{\partial^2 h(b)}{\partial H^2} & \frac{\partial^2 h(g)}{\partial H^2}
\end{array} \right)
\]

Proof: Can be obtained by procedurally replacing, for each occurrence of \( H \) an occurrence of \( h \) in theorem 1, section 1.5.

An undecomposable form of level 1 is called normal. The first occurrence of \( a_1 \) of a part in a normal form must be the one of an open parenthesis, otherwise \( g_0 \), which must exist because \( a \) has level \( \geq 1 \), would have to be an occurrence of depth 0 of a form, and \( a \) would not be undecomposable. Similarly, the last occurrence of a part \( a_1 \) of a part in the case of a closed parenthesis. There would be an occurrence of \( a \) in form \( b \), because \( a \) would be undecomposable, and \( b \) would be expressed in depth 0 of course.

Consider a structural part of a normal form. The first statement, stating that \( a \) is a form which is an occurrence of either definition 1 or 2, section 1.2, because in the first case \( a \) would have level 0 and in the second \( a \) would be decomposable.

Therefore this statement must be an occurrence of definition 3, section 1.2, stating that \( a \) is a form because it results from the enclosures in parentheses of a form \( g \). Consequently, if one
because the initial open and the final closed parentheses from a normal form $g$ be will be another form $h$. This parentheses removal operation is called peeling, and $h$ is called interior of $g$.

The concept of permutation partial detachment needs some justification. Let $\{g_i\}$, $i=1,2,\ldots M$, be a sequence of forms, and let $g$ be another form (such at $M(g) = M$) and such that all canonical components of $g$ are normal. Consider the form

$$\left[ \begin{array}{c} (g_1)_{e_1} \\ (g_2)_{e_2} \\ \cdots \\ (g_M)_{e_M} \end{array} \right]$$

where $e_j$ denotes the interior of the $j$-th canonical component of $g_j$. In general, $e_j$ may be not empty, and it can happen for up to all values of $j$ but one, $\frac{1}{2} < M(g) < M$. If this happens $M(g_j) = M(e_j)$. Then the $j$-th component of the form $\ref{eq:interior}$ contains occurrences of a variable letter $(x_1, \tilde{H}(e_j))$ which is generated by changing the subscript of the variable letter $(x_1, M(e_j)-1)$ occurring in $e_j$. Let now $g_j$ be such that $\tilde{H}(e_j) = M(g_j)$, and assume that $e_j$ is noncompact. The $j$-th component of $\ref{eq:interior}$ still contains occurrences of the variable letter $(x_1, \tilde{H}(e_j))$ but, however, has a completely different origin because such occurrences pre-existed in $g_j$. This simple means that forms $g_j$ are detached from the interior of the corresponding components of another form $g$, when the global notational distinction of variable letters $e$ can be no longer preserved. However, if $\ref{eq:interior}$ is replaced by

$$\left[ \begin{array}{c} (g_1)_{e_1} \\ (g_2)_{e_2} \\ \cdots \\ (g_M)_{e_M} \end{array} \right]$$

then the global notational distinction of variable letters $e$ can be no longer preserved.
(the free storage list) as soon as their location becomes irrelevant to further processing. Mathematics, in other words, is carried out automatically by the computer. Thus, the percentage of memory space devoted to organizational overhead is thus constant with respect to the length of the aggregates stored there. As it will be seen in the sequel, also the percentage of space devoted to overhead operations, such as addressing, is kept constant with respect to the length of the aggregates operated upon. This constant ratio of overhead space and time is an important peculiar feature of this abstract computer, and certain aspects of its design have been devised in order to provide a maximum of this feature. The study of computation schemes where the ratio between average organizational overhead and time or space, respectively, increases and tends to increase with length or the complexity of the information to be processed increases, may have a certain mathematical interest but cannot possibly give good indications for the advancement of the computation methodology. In fact, while a constant overhead ratio, even very high, can potentially be reduced by skilled technology, this is not the case for overhead ratios not bounded above by a number $< 1$.

This is, in essence, the angle from which the design of the memory organization of this abstract computer should be viewed.
The term of the first and only word is 1,671,900 and 1,671,900, respectively. In this case, the term is referred to as the initial element of the word, respectively.

The term of the middle word is 2,100,000.

For 31 and $\phi_0$, we have the following

**Theorem**: If $\Sigma_1$ and $\Sigma_2$ are defined as

$$
\Sigma_1 = a + b + c
$$

and

$$
\Sigma_2 = \frac{\Sigma_1}{2}
$$

with $a$, $b$, and $c$ denoting the terms of the middle word, then $\Sigma_1$ and $\Sigma_2$ are uniquely defined.

The middle term is defined as the middle of the word.

For a word of 32 or 33 terms, the word will be shortened in the middle...
Let $S$ denote this $n_2$-section of $\mathbb{R}$. The $n_2$-consisting paper consists of the $n_2$th page, copy, and

...
The control panel of a vessel contained 2-inch slide-in...
Let $S$ be a set, let $f: S \to \mathbb{R}$ be a function, and let $\mathcal{L}$ be a linear operator. Let $S^\mathcal{L}$ denote the set of all functions $f: S \to \mathbb{R}$ such that $\mathcal{L}(f) = \mathcal{L}(g)$ for all $g \in S^\mathcal{L}$. Let $\mathcal{L}^\mathcal{L}$ denote the set of all linear operators $\mathcal{L}: S^\mathcal{L} \to \mathbb{R}$ such that $\mathcal{L}(f) = \mathcal{L}(g)$ for all $f, g \in S^\mathcal{L}$. Let $\mathcal{L}^\mathcal{L}(\mathcal{L})$ denote the set of all elements $x \in S^\mathcal{L}$ such that $\mathcal{L}(x) = \mathcal{L}(y)$ for all $y \in S^\mathcal{L}$. Let $\mathcal{L}^\mathcal{L}(\mathcal{L}(x))$ denote the set of all elements $y \in S^\mathcal{L}$ such that $\mathcal{L}(y) = \mathcal{L}(z)$ for all $z \in S^\mathcal{L}$. Let $\mathcal{L}^\mathcal{L}(\mathcal{L}(\mathcal{L}(x)))$ denote the set of all elements $z \in S^\mathcal{L}$ such that $\mathcal{L}(z) = \mathcal{L}(w)$ for all $w \in S^\mathcal{L}$. Let $\mathcal{L}^\mathcal{L}(\mathcal{L}(\mathcal{L}(\mathcal{L}(x))))$ denote the set of all elements $w \in S^\mathcal{L}$ such that $\mathcal{L}(w) = \mathcal{L}(v)$ for all $v \in S^\mathcal{L}$.
After finding that the right depth of each parenthesis match in
and notation, one must be aware of this. If we are interested in
depth $d$ of a tree, sometimes a right parenthesis is then there may
be a occurrence of right and left brackets or braces of depth $d$ or
sign of $a$, $b$, $c$, and even negative. If the parenthesis rem-
not a complete list in the case of $b$, the right depth is given at $d$ and of a case, in $a = b$ or $b = a$. If $g$, $w$ expression of trees in string representation of another line
the occurrence of brackets or braces of depth $d$. Correctly, a line that is

The section $q$ is of $c$ at $q$ is of $q$ at $a$. To make $a$ a
letter, we will only take $g$ of $b$ and $b$ of $a$ at $b$. To assume,
$g$ is said to be labeled because $b$ is labeled by $g$. If $g$ is a
sequence of depth $d$ is a label $q$ such that $q$, to the right
in $a$ in the left of $g$ occurrence is of other occurrence in $b$ depth,
then rightness of $g$ is in the case of $b$, but we find the situation
of the case of $g$ at the occurrence of $g$,
the expression, if we have

the table of
$g$ be taken into the right column.


...
It is clear that, while they have supporting data, the conclusions drawn make it apparent that the data presented do not fully support the hypothesis. It is important to note that the conclusions drawn are based on the data presented and do not necessarily reflect the broader context.

In summary, the conclusions drawn are supported by the data presented, but it is important to consider the broader context when evaluating the results.
the cidrccs of the ii-al list; which ua3 subilut or S constistne? re p tr.; icg.l':;-3i libt.

:£ bot?:i defined to consist

rse pny^lcal threadlag, is that j'der, and toa end
The purpose of physical threading is to achieve maximum utilization of physical storage, and the reason why it goes backwards with respect to logical threading may depend on the mode of operation of the abstract computer which, as it will be explained in chapter 3, must read the forms backwards in order to evaluate them.

In the metalanguage the contents of the top element of logical lists will always be displayed in a line at the end, separated from the contents of the words of the list by a horizontal rule.

In the sequel a list other than a logical list will be referred to as physical list. Physical lists are not suitable to carry structured information such as partial forms, and their main usage is connected with storing and addressing in sequence marks, mainly constants.
Under the assumption that a tape is readable for self-contained units as originally in the sense, only lists or problems should arise as far as the component of level of elements concerned. In fact, additions, removals and replacements described above are all operations which preserve physical threading, but this is not the case of logical threading, unless something is done about it. The scheme discussed in this section solves the solve task problem by associating to each logical item another list, called sequencing list of L, such that the logical items allocated in L are not aggregates, but simply marks, so that logical threading can be exceptionally used also for addressing purposes. An algorithm to produce the required threading of virtual forms embedded in logical lists is thus defined in terms of the allocation of marks into a physical list.

The scheme of the simple logical list related to all U in this section is at all similar to the one of devices, sometimes called push-down stacks or stacks, which are utilized in programmed forms in most program compilers for conventional computers and in wired form in certain semi-conventional computers. Such lists are lists of marks, not of structured information, and therefore can be physically addressed. On the other hand, the logical lists used by this abstract computer are a more general concept, for which there is no intuitive equivalent in conventional or semi-conventional machines.
Let $a$ be a partial form and $b$, the $i$-th occurrence, from the right of a word in $a$. The first thing to do is to devise a formula for

$$a^t,$$

the $i$-th occurrence from the right of a word in $a$. For this purpose, one utilizes a sequence $b_0 = 0, b_1, \ldots, b_i$ of numbers and a sequence $c_0$ of truth values defined as follows:

$$b_0 = 0$$

$$b_i = \begin{cases} b_i - 1, & \text{whenever } a_i \text{ is the occurrence of a closed parenthesis} \\ b_i - 1, & \text{whenever } c_{i-1} \text{ has the value } 1 \\ b_{i-1} \text{ otherwise} \end{cases}$$

while

$$i > 0.$$  

$$b_i = \begin{cases} 0, & \text{whenever } a_i \text{ is the occurrence of an open parenthesis} \\ \infty, & \text{otherwise} \end{cases}$$

The purpose of the definitions is to compute, for all $i$, the maximum integer $D_i = \max_i b_i$, which equals $b_i$ of the value $c_i$ containing by $a_i$ and $a_{i-1}$ alone.

For us consider the problem of proceeding from the partial form $a$ to generating a logical list $S$ with top pointer and control word $L(a)$, such that $H(S) = \text{fig}$, in which a is 1-embedded. The problem is then to find for this operation is a list $U$, which initially consists of just one word, containing in its center field the address $L(a)$, such that the list $S$ is initially a $Q$-list, consequently being $L(a)$ and

$$L_U = L_{L(a)} = \text{address of } L_{L(a)}.$$ The algorithm consists of log steps each
Let \( g_j \) denote the \( j \)-th word of \( S \). Then, if \( g_j \) is a closed parenthesis, a new word, carrying \( g_j \) in its centre field and the address \( \alpha_2 \) of the \( j \)-th word of \( S \) in its right field is added to the beginning of \( U \).

If \( g_j \) is a comma, \( \alpha_2 \) is placed into the right field of the \( j \)-th word of \( S \), whose address is \( \alpha_1 \): then, if \( \alpha_1 \) is a comma, \( \alpha_2 \) is replaced by \( \alpha_1 \) while, if \( \alpha_1 \) is a closed parenthesis, a new word, carrying \( g_j \) in its centre field and \( \alpha_1 \) in its right field is added to the beginning of \( U \). If \( g_j \) is an open parenthesis, \( \alpha_2 \) is placed into the right field of the \( j \)-th word of \( S \); then, if \( j \) is the smallest positive integer such that \( \beta u_j \) is a closed parenthesis, \\
(\text{or} \, \alpha_2 \text{ is placed into the right field of the word of a preceding \( u_j \) in address: finally the first \( j \) words of \( S \) are erased and given back to free storage.) In all other cases no further action is taken. At the end of the performance of the algorithm, a content of free words as the difference between the number of occurrences of open and closed parentheses in \( \alpha_1 \) plus \( j \).

Consider now the problem of proceeding from the above logical list with a 1-imbedded in \( U \) to another logical list \( S \) requiring from adding a partial expression \( \alpha' \) to the end of \( S \). This operation can be
easily performed provided that the symbol _i_ is the end of the generation of _S_, the internal _L(0)_, which is the length of _U_, minus one, and the truth value _L(n)_, which is the length of _E_(a) in its closed parentheses, are available. In this case, _S_ can be generated just by continuing the operation of forcing _S_ before the _L(a)-th step, considering it an extension of _L_.

This procedure shows the relevance of the iterations of any logical list _S_ of three items, namely the lists _U_ and _L_ which of each step contain the right subscript _b_ and the constant _t_, respectively. Let _U_ be called associated list of _S_, while the compound of _b_ and _t_, which, for all purposes, can be placed in the same word of reality, is called the associated depth register of _S_ and denoted _R_2.

The above basic procedure can be easily extended to generating partial forces _S_ into logical lists _S_ such that if _S_ is a list, then _S_ is extended even then on. If a capital _R_ is an inner value without subscript or a list, the corresponding value or list will have the same superscripts and so in each subscript, _i_ will be denoted the _i-th_ word of that list if _i_ 0, or its control word if _i_ = 0.

The symbolic subscript is used to identify the top pointer. For example, if _S_ is a list, then _S_ is its control word, _S_ is its fifth word and _S_ is its top pointer.
Matlog

I have included a new feature to the code to allow for the addition of new symbols and operations. This has been done to improve the flexibility and robustness of the system. The new symbols can be added to the code by following the instructions in the documentation. The new features have been tested and found to be compatible with the existing code.

For example, the new feature allows for the inclusion of new variables and constants. The code can now be modified to include these new elements, and the new feature has been tested to ensure that it works correctly with the existing system.
contains in its right field the address of the \( i \)-th word of the list \( S \), while the first one contains in its right field the right depth \( b_{eff} = b_i \) and in its center field \( g_1 \). If \( g_1 = 1 \) comes, then, if both \( \sigma_{iL} = b_i \) and \( \sigma_{iR} = b_i \), no action is taken.

1. If \( \sigma_{iL} = b_i \), \( \sigma_{iR} = z_i \) and \( \sigma_{iU} = y_i \), then \( g_1 \) is the mate of the occurrence of \( y_i \) whose address is \( \sigma_{iU} \). The action to be taken consists of placing the address of the \( i \)-th word of \( S \) in the left field of the word of \( S \) whose address is \( \sigma_{iU} \), and then removing the first two words of \( U \) (that is, replacing \( U \) by the 2-remainder of \( U \)).

Another possibility is that \( b_i = \sigma_{iL} \), \( z_i = \gamma_i \) and \( \sigma_{iU} = \beta_i \). This case is analogous to the first one, with the only difference that \( g_1 \) is the mate of an occurrence of \( \gamma_i \) instead of \( \beta_i \). The action to be taken is exactly the one of the first case, with the only difference that \( \eta_0 \) and \( \eta^0 \) replace \( \theta_0 \) and \( \theta^1 \), respectively.

Finally, if neither of these two combinations of conditions is satisfied, \( g_1 \) is not the mate of anything and no action is taken.

The basic difference between the operations of rating occurrences of open parentheses or commas on one side and occurrences of closed on the other parentheses, \( \theta_0 \) and \( \theta_1 \), for imbedding purposes is that in the first case one should memorize all addresses of possible mates and place them when the occurrence to be rated is found, while in the second case, since the occurrence of closed parentheses and condition marks