R.A.I.R.O.
(9th year, December 1975, R–3, p. 25 to 49)

NOTE ON THE ALGEBRA OF FUNCTIONS

English translation

by F. H. RAYMOND*

Communicated by M. Nivat

Translator's note. This is an English translation of F. H. Raymond, "Note sur l'algèbre des fonctions," *Revue française d'automatique, informatique, recherche opérationnelle. Informatique théorique*, tome 9, n° R3 (1975), p. 25–49, originally published by AFCET and archived by Numdam (`http://www.numdam.org/item?id=ITA_1975__9_3_25_0`). The original source scan contains a number of OCR artifacts in its formal notation (subscripts, operator symbols); the translation reproduces the formalism as faithfully as could be reconstructed from context. Bibliographic references are reproduced as in the original (untranslated titles are kept in their original language where they were already in English).

Abstract. — The definition and the properties (those needed for the whole of the note) of a formal algebra, called the *algebra of functions*, whose objects are functions, are the subject of this article.

The application is illustrated in two sections, one dealing with the factorization of functions, the other with loop-functions, which form a subset of recursion functions, themselves generated within the algebra of functions by a single operator. The properties established directly encompass, for example, the approach of Böhm and Jacopini, and express the fact that every program can be constructed without GOTO instructions.

1. Introduction

1.1.

In an earlier work [9] [10] devoted to a formalization of the concept of computation, we were led to create an elementary tool that we called the *algebra of functions*. It is the subject of § 2 and § 4 of this note. The objects of this algebra are functions in the classical sense of the word, and naturally, following the approach of mathematics, we were led to consider a *formal algebra* of functions.

This algebra is characterized by a set of operators allowing new functions to be formed from given functions, in the first instance without any recursive scheme — this is the subject of § 2 — and then, in § 4, a single operator of the algebra of functions introduces recursivity.

Sections 2 and 4 are limited to what is essential for the exposition of § 3 and § 5 respectively, which illustrate the proposed approach.

Under the heading of *factorization* in § 3, we establish properties of functions (defined within the algebra of functions) whose practical interest is illustrated by the following observation. The product of functions f_1 and f_2, as classically defined, results within the algebra of functions from the application of the operator S^2 — a binary operator — to the functions f_1 and f_2, in the appropriate order; this leads to the function $f_1 f_2 S^2$ (we adopt post-fixed notation, which allows one to pass directly from function-objects to instruction-objects of a programming language, a

*Conservatoire National des Arts et Métiers, Paris.

problem not treated in this note). To factorize a function is to put it in the form of a product of functions.

The results of § 3 lead, in § 5, to the establishment of general properties of a class of recursion functions (defined in § 4) called "loop-functions," functions which are expressed, in a programming language, by sequences of instructions containing, in the general case, GOTO instructions, and which consequently contain labels. We show in § 5 that every loop-function is equivalent to a *primitive* loop-function, which is expressible by the WHILE-DO instruction.

It appears that approaches sharing the same concerns as the algebra of functions are emerging at present, and that computer scientists are seeking a tool meeting their needs, capable for instance of aiding programming and of expressing the semantics of a program. We cite among recent publications references [7] and [8]. The limited scope of this note has not allowed us to show that the algebra of functions can constitute such a tool; we believe its content suggests this to the reader.

1.2.

B is a set of atomic objects, F_0 is a set of executable functions; each member of F_0 is an atom, the proper name of an executable function. In the exposition, each member of F_0 will be denoted by f_i^0, $i \in N$ (N the set of natural numbers).

B contains four particular atoms, denoted V, F, Λ and ω.

The word "executable," used to define F_0, is related to the concept of an (mechanical) *executor* whose actions are evoked, at a first stage, by a function ν (which of course does not belong to F_0).

We say that the algebra $\mathcal{B} = (B, F_0)$ is a representation of the computation defined by the objects of the set X and a subset of the mappings $\{X^n \to X \mid n \in N\}$ (X^n being the n-fold Cartesian product of X), if, having defined

$$\nu_0 : B \to X, \qquad \mu_0 : F_0 \to \{X^n \to X \mid n \in N\},$$

the executor applies ν to every term t of $\mathcal{B}$ according to the relations

$$\begin{cases} [[t_1, \ldots, t_n]f_i^0]\nu = [[t_1]\nu, \ldots, [t_n]\nu][f_i^0]\mu_0 \\ \qquad\qquad [t]\nu = [t]\nu_0 \quad \text{if } t \in B. \end{cases} \tag{1}$$

We shall say that:

- $[f_i^0]\mu_0$ is the interpretation, in the computation on the objects of X, of the proper name f_i^0,
- $[t]\nu$ is the value (an object of X) of the term t of $\mathcal{B}$,
- $[b_i]\nu_0$ is the interpretation of any atomic object b_i, $b_i \in B$.

In every interpretation, the images of V, F, ω and Λ are the same symbols: they carry the meanings generally attributed to the words *true*, *false*, *undetermined*, and Λ is the atom representing an empty list.

$\mathcal{L}_B$ is the set of lists, classically formed with the atoms of B. In the exposition, x and x_i, $i \in N$, denote an object belonging to $B \cup \mathcal{L}_B$.

Remarks: 1) We shall systematically use post-fixed notation to construct the terms of $\mathcal{B}$, as indicated in (1) above, since it is closer to the means allowing a more precise definition of the concept of an executor, which we shall not do in this note.

2) For the same purpose, it is useful to distinguish two kinds of actions of the executor (or even two types of executors), for which $[t]\nu$ is an instruction, understood by it differently according to whether $t \in B$ or $t \notin B$. In the latter case, the term $[t_1, \ldots, t_n]f_i^0$ of B is subjected to these two actions in a fixed order:

- first, valuation by ν of the terms $t_1, \ldots, t_n$; let $b_1, \ldots, b_n$ be the objects obtained,
- second, effective construction, if possible, of the simplest representation of $[b_1, \ldots, b_n]f_i^0$ by an atom of B; if this is impossible, a decision must be made which there is no need to study here, the simplest of which is to assign to this term the value of the object ω.

Then, if **cal** is the operation performed in the second stage, returning to (1):

$$[[t_1, \ldots, t_n]f_i^0]\nu = [[[t_1]\nu, \ldots, [t_n]\nu][f_i^0]\mu_0]\,\mathbf{cal}.$$

The right-hand side brings out the principle of sequentiality if we are dealing with a single executor, or the simultaneous (parallel) "work" of n executors before a single executor obeys **cal**.

2. Algebra of functions

2.1. Definitions

1. $\mathcal{A}$ is an algebra with operators, represented by (F, S), whose terms form the set F; S is a finite set of operators:

$$S = \{\mathbf{cond}, S^0, S^1, S^2, \ldots\}$$

cond, S^0, $S^1, \ldots$ are atomic symbols.

2. To $\mathcal{A}$ is associated the formal algebra $\hat{\mathcal{A}} = (F, \hat{S})$ defined by:

— the symbols of the set $\hat{S} = \{\widehat{\mathbf{cond}}, \hat{S}^0, \hat{S}^1, \ldots\}$, which are its *functional* symbols, together with the mappings:

$$\hat{S}^1 : F \to F, \quad \hat{S}^1(f) = fS^1$$

$$\hat{S}^n : F \to F, \quad \hat{S}^n(f_1, \ldots, f_n) = f_1, \ldots, f_n S^n$$

$$\widehat{\mathbf{cond}} : F^3 \to F, \quad \widehat{\mathbf{cond}}(f_1, f_2, f_3) = f_1 f_2 f_3\,\mathbf{cond}$$

— the constant symbols forming the set $G_0 \cup \{b_i S^0 \mid b_i \in B\}$; G_0 will be defined below: it is the set of *primitive selection functions.*

3. Let F_0 be a nonempty subset of F, and let $\hat{\mathcal{A}}_{F_0} = (\bar{F}_0, \hat{S})$ be the functional subalgebra generated by F_0, $\bar{F}_0$ being the intersection of all the subdomains of $\hat{\mathcal{A}}_{F_0}$ that contain F_0.

Definition: Every member of $\bar{F}_0$ is called a *function*; it will be represented in the exposition by f, most often by a consonant with an integer index.

$f \in \bar{F}_0$ will more simply be written $f \in \mathcal{A}$, f evoking at once a function of $\mathcal{A}$ and a word of $\hat{\mathcal{A}}$.

Remark: Malcev [6] seems to have had a similar idea, but, to our knowledge, neither he nor his students developed it from a computer-science perspective, which is our purpose here.

2.2. Interpretation

The interpretation of $\mathcal{A}$ (and of $\hat{\mathcal{A}}$) postulates the existence of an executor, not otherwise defined.

Interpretation of fS^1, for every function f: the function fS^1 is understood by the executor as enjoining it to apply f to an object $x \in B \cup \mathcal{L}_B$; x is called the *context*, chosen by an unspecified third party. The executor will compute the value of the term $[x]f$.

Interpretation of $f_1 \ldots f_n S^n$, for all functions $f_1, \ldots, f_n$: considering the function $f_1 \ldots f_n S^n S^1$, the interpretation is expressed by:

$$[x]f_1 \ldots f_n S^n = [[x]f_1, \ldots, [x]f_{n-1}]f_n$$

Interpretation of $f_1 f_2 f_3\,\mathbf{cond}$, for all functions f_1, f_2, f_3: proceeding as above,

$$[x]f_1 f_2 f_3\,\mathbf{cond} = \begin{cases} [x]f_2 & \text{if } [[x]f_1]\nu = V \\ [x]f_3 & \text{if idem} = F \\ \omega & \text{if idem} \notin \{V, F\} \end{cases}$$

Interpretation of $b_i S^0$ for all $b_i \in B$: this is the function that selects the atomic object b_i; consequently, considering the function $b_i S^0 S^1$, we express the interpretation by the equality:

$$[x]b_i S^0 = b_i$$

2.3. Primitive selection functions

The set G_0 introduced in the definition of $\mathcal{A}$ can be defined in several ways; computing practice suggests three, denoted G_0^1, G_0^2, G_0^3.

Definition of G_0^1: $G_0^1 = \{\pi_0, \pi_1, \ldots, \pi_n, \ldots\}$, an infinite set of functions, practically limited to a function π_n. The interpretation, considering $\pi_0 S^1, \pi_1 S^1, \ldots$, is, for x_i belonging to $B \cup \mathcal{L}_B$:

$$[x_1, \ldots, x_n]\pi_0 = [(x_1, \ldots, x_n)]\pi_0 = (x_1, \ldots, x_n), \quad (x_1, \ldots, x_n) \in \mathcal{L}_B$$

$$\forall x \in \mathcal{L}_B,\ [x]\pi_0 = x, \qquad [()]\pi_0 = (),\ () \text{ is an empty list},\ \Lambda = ().$$

$$[x_1, \ldots, x_n]\pi_j = [(x_1, \ldots, x_n)]\pi_j \quad \text{for } j \geq 1.$$

$$[(x_1, \ldots, x_n)]\pi_j = \begin{cases} x_j & \text{if } 1 \leq j \leq n \\ \omega & \text{if } j > n \end{cases}$$

Definition of G_0^2: $G_0^2 = \{\mathbf{cons}, \pi_d, A_d\}$
Interpretation:

$$[(x_1, \ldots, x_n), x]\mathbf{cons} = (x_1, \ldots, x_n, x)$$

$$[(x_1, \ldots, x_n)]\pi_d = x_n, \qquad [(x_1, \ldots, x_n)]A_d = (x_1, \ldots, x_{n-1})$$

G_0^3 is defined symmetrically, $G_0^3 = \{\mathbf{edif}, \pi_g, A_g\}$, the index g recalling *gauche* (left), d *droite* (right), as in the case of G_0^2.

We shall limit the exposition to the case of G_0^1, for lack of space.

The definitions given justify the term *selection functions*, an abbreviation of "selection and construction," since π_0 and **cons** are interpreted as "constructors."

2.4. Isotony

Definition: a) $f_1 \preccurlyeq f_2$ if, in every interpretation of the atoms of F_0, the function f_1 is the restriction of the function f_2.

We shall say that f_1 is *less well defined* than f_2, and that the latter is *better defined* than the former.

b) If $f_1 \preccurlyeq f_2$ and $f_2 \preccurlyeq f_1$, we shall say that f_1 and f_2 are *equivalent*, which will be expressed by the relation $f_1 \equiv f_2$.

It is easy to prove the following lemmas [10]:

Lemmas:

1. $f \preccurlyeq f'$ implies $fS^1 \preccurlyeq f'S^1$
2. $f \preccurlyeq f'$ implies $f_1 \ldots f \ldots f_n S^n \preccurlyeq f_1 \ldots f' \ldots f_n S^n$, f and f' being at the same rank in $f_1 \ldots f_n S^n$,
3. $f \preccurlyeq f'$ implies $f_1 \ldots f_n f S^{n+1} \preccurlyeq f_1 \ldots f_n f' S^{n+1}$

4. $f \preccurlyeq f'$ implies

$$pfg\,\mathbf{cond} \preccurlyeq pf'g\,\mathbf{cond}$$

$$pgf\,\mathbf{cond} \preccurlyeq pgf'\,\mathbf{cond}$$

$p \preccurlyeq p'$ implies $pfg\,\mathbf{cond} \preccurlyeq p'fg\,\mathbf{cond}$

Let there be, in a function represented in the exposition by φ, a privileged function f having at least one occurrence, which is denoted by $\varphi(f)$. $\varphi(f)$ being obtained by the application, a finite number of times, of the operators $\hat{S}^n$, n an integer, and **cond**, we deduce from the lemmas:

Isotony theorem: $f \preccurlyeq f'$ implies $\varphi(f) \preccurlyeq \varphi(f')$

whence the corollary: $f \equiv f'$ implies $\varphi(f) \equiv \varphi(f')$

2.5. The function f_ω

Definition: f_ω is a function whose domain, in every interpretation, is the empty set.

We deduce, for every $x \in B \cup \mathcal{L}_B$, $[x]f_\omega = \omega$ and the property:

Property: $f_\omega f S^2 \equiv f f_\omega S^2 \equiv f_\omega$.

2.6. Axioms of $\mathcal{A}$

To ease reading, predicate functions are represented by p, q, possibly indexed.

$$A_0 : \quad f_1 \dots f_n g_1 \dots g_p g S^{p+1} S^{n+1} \equiv f_1 \dots f_n g_1 S^{n+1} \dots f_1 \dots f_n g_p S^{n+1} g S^{p+1}$$

if $n = p = 1$, it expresses the associativity of the product of functions;
if $n = 1$, it expresses the right-distributivity of the product of functions, except with respect to the last function suffixed by S^{p+1};
$g_1, \dots, g_p$ are said to be the *initial functions* of $g_1 \dots g_p g S^{p+1}$, see § 3.1;
if $p = 1$, it expresses the non-left-distributivity of the product of functions;
g_1 in $f_1 \dots f_n g_1 S^{n+1}$ is *terminal* there, see § 3.2

$$A_1 : \quad pff\,\mathbf{cond} \preccurlyeq f, \qquad VS^0 f_1 f_2\,\mathbf{cond} \equiv f_1$$

$A_2 :$ $f_\neg$ being, in every interpretation, the function defined by

$$VS^0 f_\neg S^2 \equiv FS^0, \qquad FS^0 f_\neg S^2 \equiv VS^0 :$$

$$pf_1 f_2\,\mathbf{cond} \equiv pf_\neg S^2 f_2 f_1\,\mathbf{cond}$$

$$A_3 : \quad ppf_1 f_2\,\mathbf{cond} f_3\,\mathbf{cond} \equiv pf_1 f_3\,\mathbf{cond}$$

$$A_4 : \quad p_1 p_2 p_1 f_1 f_2\,\mathbf{cond} f_3\,\mathbf{cond} f_4\,\mathbf{cond} \equiv p_1 p_2 f_1 f_3\,\mathbf{cond} f_4\,\mathbf{cond}$$

$$A_5 : \quad p_1 p_2 p_3\,\mathbf{cond} f_1 f_2\,\mathbf{cond} \equiv p_1 p_2 f_1 f_2\,\mathbf{cond} p_3 f_1 f_2\,\mathbf{cond}\,\mathbf{cond}$$

$$A_6 : \quad p_1 p_2 f_1 f_2\,\mathbf{cond} p_2 f_3 f_4\,\mathbf{cond}\,\mathbf{cond} \equiv p_2 p_1 f_1 f_3\,\mathbf{cond} p_1 f_2 f_4\,\mathbf{cond}\,\mathbf{cond}$$

$$A_7 : \quad f_1 p f f'\,\mathbf{cond} S^2 \equiv f_1 p S^2 f_1 f S^2 f_1 f' S^2\,\mathbf{cond}$$

$$A_8 : \quad f_1 \dots f_{i-1} p f_i f_i'\,\mathbf{cond} \dots f_n S^n \succcurlyeq p f_1 \dots f_i \dots f_n S^n f_1 \dots f_i' \dots f_n S^n\,\mathbf{cond}$$

In the case $n = 2$, one has an equivalence.

$$A_9: \quad f_1 \dots f_n S^n S^1 \equiv f_1 S^1 \dots f_{n-1} S^1 f_n S^n, \qquad p f_1 f_2\, \mathbf{cond} S^1 \equiv p S^1 f_1 S^1 f_2 S^1\, \mathbf{cond}$$

$$A_{10}: \quad f_1 f_2 S^1 S^2 \equiv f_2 S^1$$

$$A_{11}: \quad f b_i S^0 S^2 \equiv b_i S^0$$

$$A_{12}: \quad f \pi_0 S^2 \equiv f, \qquad \pi_0 f S^2 \equiv f$$

$$A_{13}: \quad f_1 \dots f_n \pi_j S^{n+1} \equiv \begin{cases} f_j & \text{if } 1 \le j \le n \\ f_\omega & \text{if } j > n \end{cases}$$

Finally, certain functions of F_0 may enjoy properties expressible by relations in $\mathcal{A}$; this is the case for f_ω; they will accordingly lead to supplementary axioms, to be made precise case by case.

The axioms A_0, A_7, A_9, A_{10} and A_{11} follow from the interpretation of § 2.2, and they are independent, since they concern different terms of $\mathcal{A}$ and different interpretations.

Axioms A_1 through A_6 inclusive lead to those proposed by McCarthy for the calculus of conditional expressions: they express the properties of **cond**.

2.7. Calculus on functions

Replacement rule: it expresses the isotony theorem.

All the axioms of $\mathcal{A}$ express a relation $\varphi(f) \equiv \psi(f)$ or $\varphi(f) \preccurlyeq \psi(f)$, whatever the privileged function f chosen in the statement of the axiom.

Let us introduce into $\mathcal{A}$ the operator $S^f_{f'}$:

Definition: Let $\varphi(f) \in \mathcal{A}$; $\varphi(f) S^f_{f'}$ (or $\varphi S^f_{f'}$, if there is no possible ambiguity in the exposition) is the function obtained by substituting the function f' for all occurrences of f in $\varphi(f)$.

The *substitution rule* is then stated as follows:

$$\text{If } \varphi(f) \equiv \psi(f) \quad \text{or} \quad \varphi(f) \preccurlyeq \psi(f), \text{ then:}$$
$$\varphi(f) S^f_{f'} \equiv \psi(f) S^f_{f'} \quad \text{or} \quad \varphi(f) S^f_{f'} \preccurlyeq \psi(f) S^f_{f'}$$

The calculus we now have available will be extended by theorems and by the introduction of an extension of $\mathcal{A}$ (§ 4).

As an example:

Theorem: $f_1 \dots f_n f S^{n+1} \equiv f_1 \dots f_n \pi_0 S^{n+1} f S^2$

Proof: by A_{12}: $f \equiv \pi_0 f S^2$

by the replacement rule: $f_1 \dots f_n f S^{n+1} \equiv f_1 \dots f_n \pi_0 f S^2 S^{n+1}$

by axiom A_0: $f_1 \dots f_n f S^{n+1} \equiv f_1 \dots f_n \pi_0 S^{n+1} f S^2$

Remark: In the algebra $\mathcal{A}$ generated by G^1_0, this theorem allows the use of the operator S^n, $n > 2$, to be limited to being suffixed to the identity function π_0.

2.8. Remarks

The principle of sequentiality and the interpretation of § 2.2 lead to the following understanding of the executor for **cond**:

let $pf_1f_2\,\mathbf{cond}$: it computes the value of p, it computes that of f_1, whence an object x_1, then that of f_2, whence an object x_2, and finally **cond**, a ternary operator, enjoins it to choose between x_1, x_2 and ω according to the value of p.

It appears that here the prefixed notion is preferable[1], so we shall agree to represent $pf_1f_2\,\mathbf{cond}$ by $(p \to f_1, f_2)$, which naturally leads to McCarthy's conditional expressions; other forms have been proposed, $[p](f_1 \mid f_2)$ by Kott [5], $p(f_1 \mid f_2)$ by J. Arsac [1], for example; *if p then f_1 else* f_2 being too cumbersome.

The corresponding modification in the writing of the axioms is straightforward.

3. Factorization of a function

Two general factorization procedures are presented, allowing the study of recursion functions and, in particular, loop-functions. We then show how the algebra of functions allows us to describe a third procedure, none other than the one proposed by Böhm and Jacopini [2].

3.1. Initial functions

Definition: The *initial functions* of a function represented by φ are the functions suffixed by S^1 in one of the functions equivalent to φS^1, obtained by applying axioms A_9 and A_{10}.

Examples: 1. $f_1f_2S^2$ is an initial function in $f_1f_2S^2f_3S^2$, since $f_1f_2S^2f_3S^2S^1 \equiv f_1f_2S^2S^1f_3S^2$; f_1 is an initial function of the same function, applying axiom A_9 once more.

2. f_1 and f_2 are initial functions in $f_1f_2f_3S^3$.

3. p, f_1, f_2 are initial functions in $(p \to f_1, f_2)$.

Remark: The properties of the operator S^1, expressed by axioms A_9 and A_{10}, together with the above definition, lead to:

— a function is initial in a function φ if it is initial in an initial function of φ.

Property 1: *Let $f_1, \ldots, f_n$ be the initial functions of a function φ; then*

$$\varphi \equiv f_{\alpha_1} \ldots f_{\alpha_n} \varphi' S^{n+1} \tag{1}$$

where $\alpha_1, \ldots, \alpha_n \in \{1, 2, \ldots, n\}$, $\varphi' \equiv \varphi S^1 S_{\pi_1}^{f_{\alpha_1}S^1} \ldots S_{\pi_n}^{f_{\alpha_n}S^1}$.

We shall not give the proof in this note, see [10]; it results from the following property:

Property 2: *If $f_1 \ldots f_n$ are the initial functions of a function φ, then for every function h:*

$$h\varphi S^2 \equiv \varphi(f_1, \ldots, f_n) S_{hf_1S^2}^{f_1} \ldots S_{hf_nS^2}^{f_n} \tag{2}$$

where we denote by $\varphi(f_1, \ldots, f_n)$ the function φ while evoking its initial functions.

Remark: (2) is deduced from (1) by axiom A_0; indeed:

$$h\varphi S^2 \equiv hf_{\alpha_1} \ldots f_{\alpha_n} \varphi' S^{n+1} S^2 \equiv hf_{\alpha_1}S^2 \ldots hf_{\alpha_n}S^2 \varphi' S^{n+1}$$

Calculation rule: Property (2) is stated as a calculation rule in $\mathcal{A}$:

— to carry out the product $h\varphi S^2$, one distributes h as a left factor over all the initial functions of φ.

Remark: This rule leads to left-factorization by h if h is a left factor of all the initial functions of φ.

[1] Assuming that the executor "reads" $pf_1f_2\,\mathbf{cond}$ from left to right, as we ourselves do — the principle of sequentiality in computing.

Property 3: *Let one of the initial functions, represented by f_k, of a function φ be chosen arbitrarily; then:*

$$\varphi \equiv \pi_0 f_k \varphi' S^3 \qquad \text{or} \qquad \varphi \equiv f_k \pi_0 \varphi' S^3 \tag{3}$$

where $\varphi' \equiv \varphi S^1 S_{\pi_1}^{f_k S^1} S_{\pi_1}^{\pi_0}$ in the first case, $\varphi' \equiv \varphi S^1 S_{\pi_1}^{f_k S^2} S_{\pi_2}^{\pi_0}$ in the second, and φS^1 is constructed by replacing all the other initial functions by $\pi_0 f_i S^2$, $i \neq k$.

The proof follows directly from Property 2 by axioms A_9 and A_{12}.

3.2. Terminal functions

Definition: 1. — f is terminal in $f_1 f S^2$ and $f_1 \ldots f_n f S^{n+1}$,

2. f_1 and f_2 are terminal in $(p \to f_1, f_2)$,

3. f is terminal in a function φ if it is terminal in at least one terminal function of φ.

Example: $f_2 f_3 S^2$ is terminal in $f_1 f_2 f_3 S^2 S^2$ by 1., f_3 is terminal there by 1. and 3.

Property 4: *Let φ be a function and let $f_1 \ldots f_n$ be its terminal functions; then:*

$$\varphi f S^2 \equiv \varphi(f_1, \ldots, f_n) S_{f_1 f S^2}^{f_1} \ldots S_{f_n f S^2}^{f_n} \tag{4}$$

where the notation $\varphi(f_1, \ldots, f_n)$ evokes the terminal functions of φ.

Proof: If φ is $f_1 f_2 S^2$, the associativity of S^2 leads to $\varphi f S^2 \equiv f_1 f_2 S^2 f S^2$, whence (4).

If φ is $f_1, \ldots, f_n S^n$, axiom A_0 gives $\varphi f S^2 \equiv f_1 \ldots f_{n-1} f_n f S^2 S^n$, whence (4); finally, if φ is $(p \to f_1, f_2)$, axiom A_8 leads to (4).

Since every function of $\mathcal{A}$ is obtained by the repeated application, a finite number of times, of the three rules recalled by the three functions considered above, the property holds for every function of $\mathcal{A}$. ■

We deduce directly:

Property 5: *Let $f_1 \ldots f_n$ be the terminal functions of a function φ; then those of $\varphi f S^2$ are $f_1 f S^2, \ldots, f_n f S^2$,*

whence the calculation rule, symmetric to the preceding one:

Calculation rule: To carry out the product $\varphi f S^2$, f is distributed as a right factor over all the terminal functions of φ.

Previously we used the expressions "right factor" and "terminal" interchangeably. Compare with J. Arsac [1].

Like the preceding rule, this one allows right-factorization by f if f is a right factor (or terminal) in all the terminal functions of φ.

To obtain a property analogous to Property 3 of the previous section, we are led to take one further step and to encompass the approach of Böhm and Jacopini [2] within the algebra of functions, in a very simple way, which moreover leads to a natural generalization.

3.3. Factorization by adjunction of variables

Definitions: Let h_V and h_F be the functions defined by:

$$\begin{cases} h_V \equiv \pi_0 V S^0 \pi_0 S^3, \qquad h_F \equiv \pi_0 F S^0 \pi_0 S^3 \\ \quad \text{and } p^* \text{ the predicate } p^* \equiv \pi_2 V S^0 \text{ EGAL } S^3 \end{cases} \tag{5}$$

Remark: Other definitions are equivalent, either by changing the order of π_0 and the functions selecting the objects V and F, or by using one or the other of the functions **cons** or **edif**.

Properties: These follow directly from the definitions,

$$h_V \pi_1 S^2 \equiv h_F \pi_1 S^2 \equiv \pi_0, \qquad h_V p^* S^2 \equiv V S^0, \qquad h_F p^* S^2 \equiv F S^0$$

Property 6: *Let $f_1 \dots f_n$ be the terminal functions of a function denoted $\varphi(f_1, \dots, f_n)$, and let us single out one of the terminal functions, say f_1; then:*

$$\begin{cases} \varphi \equiv \varphi'(p^* \to \pi_1 f_1 S^2, \pi_1) S^2 \\ \text{where} \quad \varphi' \equiv \varphi(f_1, \dots, f_n) S^{f_1}_{h_V} S^{f_2}_{f_2 h_F S^2} \dots S^{f_n}_{f_n h_F S^2} \end{cases} \tag{6}$$

φ' will be denoted $\varphi'(h_V, h_F)$ since its terminal functions are h_V and h_F.

Proof: apply the calculation rule expressing Property 5 to the rightmost product of functions in (6).

The result is the function obtained by substituting, for every occurrence of h_V in φ', the product:

$$h_V(p^* \to \pi_1 f_1 S^2, \pi_1) S^2$$

which, by properties (5) of the function introduced above, is equivalent to f_1; and, for every occurrence of h_F, the product $h_F(p^* \to \pi_1 f_1 S^2, \pi_1) S^2$, which for the same reasons is equivalent to π_0.

We finally obtain $\varphi(f_1, \dots, f_n)$, expressed in short by:

$$\varphi'(p^* \to \pi_1 f_1 S^2, \pi_1) S^2 \equiv \varphi(h_V, f_2 h_F S^2, \dots, f_n h_F S^2)(p^* \to \pi_1 f_1 S^2, \pi_1) S^2$$

$$\equiv \varphi(f_1, \dots, f_n).$$

Examples:

$$(p \to f_1 g_1 S^2, f_2 g_2 S^2) \equiv (p \to f_1 h_V S^2, f_2 g_2 S^2 h_F S^2)(p^* \to \pi_1 g_1 S^2, \pi_1) S^2$$

$$(p \to f_1(p_1 \to g_1, g_2) S^2, f_2)$$
$$\equiv (p \to f_1(p_1 \to h_V, g_2 h_F S^2) S^2, f_2 h_F S^2)(p^* \to \pi_1 g_1 S^2, \pi_1) S^2$$

Generalization: instead of the atoms V and F used in the definition of the functions h_V and h_F, we may take the numbers 0 and 1; then p^* reduces to π_2 NUL S^2, NUL being defined on the set N of integers. This leads to the generalization:

Definitions: $i \in N$, $h^*_i \equiv \pi_0 i S^0 \pi_0 S^3$, $p^*_i \equiv \pi_2 i S^0$ EGAL S^3

Properties:

$$\forall i, j \in N, \qquad h^*_i \pi_1 S^2 \equiv \pi_0, \qquad h^*_i p^*_j S^2 \equiv \begin{cases} V S^0 & \text{if } i = j \\ F S^0 & \text{if } i \neq j \end{cases}$$

These follow directly from the definitions.

Property 7: *Let $f_1 \dots f_n$ be the terminal functions of a function represented by φ; then:*

$$\begin{aligned} \varphi \equiv \varphi'(h^*_0, \dots, h^*_{n-1})(p^*_0 \to \pi_1 f_1 S^2,\ &(p^*_1 \to \pi_1 f_2 S^2, \\ &(\dots, (p^*_{n-1} \to \pi_1 f_{n-1} S^2, \pi_1 f_n S^2) \dots)) \end{aligned} \tag{7}$$

where $\varphi' \equiv \varphi(f_1, \dots, f_n) S^{f_1}_{h^*_0} \dots S^{f_n}_{h^*_{n-1}}$

The proof is analogous to that of the preceding Property 6.

Remark: the conditional function on the right-hand side of (7) expresses the semantics of a "*choice instruction*" proposed, under closely related forms, by various programming languages.

4. Recursion functions

4.1. Simple recursion functions

Definition: 1. Let $\varphi(\chi)$ denote a function of $\mathcal{A}$ in which we single out a function χ, arbitrary in $\mathcal{A}$, having at least one occurrence there; $\varphi(\chi)$ is a *functional* of χ.

2. Let **rec** be a new operator of $\mathcal{A}$, by definition:

$$\chi\varphi(\chi)\,\mathbf{rec} \in \mathcal{A} \tag{1}$$

We shall call every function constructed in this way a *simple recursion function*; $\varphi(\chi)$ is the *body* of this function.

Interpretation: proceeding as in § 2.2, we consider $\chi\varphi(\chi)\,\mathbf{rec}S^1$; the executor's instruction is then:

$$[x]\chi\varphi(\chi)\,\mathbf{rec} = \begin{cases} [x]\varphi(\chi) & \text{if } [[x]\varphi(\chi)]\nu \text{ does not depend on } \chi \\ [x]\varphi(\chi)S^{\chi}_{\chi\varphi(\chi)\mathbf{rec}} & \text{if } [[x]\varphi(\chi)]\nu \text{ depends on } \chi. \end{cases}$$

This interpretation is expressed by the following axiom:

Axiom A_{14}: $\chi\varphi(\chi)\,\mathbf{rec} \equiv \varphi(\chi)S^{\chi}_{\chi\varphi(\chi)\mathbf{rec}}$

Extension of the algebra of functions: **rec** is introduced into S, and $\widehat{\mathbf{rec}}$ into $\hat{S}$, with:

$$\widehat{\mathbf{rec}} : F \times F \to F, \qquad \widehat{\mathbf{rec}}(f, \varphi(f)) = f\varphi(f)\,\mathbf{rec}$$

It has appeared to us more convenient, and closer to computing practice, to form a recursion function in a slightly different manner, by introducing the concept of a *label* (though we shall subsequently show how it can be eliminated).

Definition: 1. A *label* is represented by the symbol I (or evoked by I), possibly indexed by an integer.

The *scope* of a label is defined by a pair of parentheses (or brackets, if there is a risk of confusion in the exposition).

2. Let $f \in \mathcal{A}$; $I(f)$ is the function f labeled by I (note that I is neither the name given to f nor that of the object (f)); f is the *body* of the labeled function.

Remark: If we adopt the representation $(p \to f, f')$ for $pff'\,\mathbf{cond}$, we shall agree to abbreviate $I((p \to f, f'))$ as $I(p \to f, f')$.

Definition: The recursion function defined by (1) is represented by:

$$I(\varphi(\chi)S^{\chi}_{I}) \quad \text{or} \quad I(\varphi(I)) \text{ if no confusion can arise.} \tag{2}$$

Axiom A_{14} is then stated:

Axiom A_{14}:

$$I(\varphi(I)) \equiv \varphi(\chi)S^{\chi}_{I(\varphi(I))} \tag{3}$$

Remark: In the representation $I(\varphi(I))$, within the body $\varphi(\chi)$, the label plays the role of a function; this is justified by the extension of the calculation rules within $\mathcal{A}$ (see § 4.3 and § 5.2).

Kleene functions of the body $\varphi(\chi)$ of a recursion function:

The proof of Kleene's recursion theorem [4] and the interpretation given above of a recursion function suggest the recurrence of the following functions:

$$h^0 \equiv \varphi(f_\omega), \qquad h^1 \equiv \varphi(h^0), \; \ldots, \; h^i \equiv \varphi(h^{i-1}) \tag{4}$$

From $f_\omega \preccurlyeq h^0$ we deduce $\varphi(f_\omega) \preccurlyeq \varphi(h^0)$, hence $h^0 \preccurlyeq h^1$, and step by step:

$$h^0 \preccurlyeq h^1 \preccurlyeq h^2 \preccurlyeq \cdots \preccurlyeq h^{i-1} \preccurlyeq h^i \preccurlyeq \ldots \tag{5}$$

We shall call the functions so defined *Kleene functions.*

We note that the interpretation given of $\chi\varphi(\chi)\,\mathbf{rec}$, and hence of $I(\varphi(I))$, is equivalent to: *the steps of the interpretation of $I(\varphi(I))$ are described by the ordered sequence of Kleene functions of $\varphi(\chi)$.*

This sequence will be represented by $\{h^i\}$ and its limit by $\lim_i\{h^i\}$, as i is allowed to increase indefinitely.

We shall simplify the exposition by postulating the continuity of $\varphi(\chi)$, that is, the equivalence:

$$\varphi\Big(\lim_i\{h^i\}\Big) \equiv \lim_i\{\varphi(h^i)\}$$

It follows directly[2] that:

Property 1:

$$\lim_i\{h^i\} \text{ is a fixed point of } \varphi(\chi), \text{ that is, a solution of } \chi \equiv \varphi(\chi).$$

Axiom A_{14} states, for its part: $\chi\varphi(\chi)\,\mathbf{rec}$ is a fixed point of $\varphi(\chi)$.

Suppose there exists a function f satisfying $f \equiv \varphi(f)$.

Considering the recurrence:

$$f^0 \equiv \varphi(f), \qquad f^1 \equiv \varphi(f^0), \ \ldots, \ f^i \equiv \varphi(f^{i-1})$$

from $f_\omega \preccurlyeq f$ we deduce $h^i \preccurlyeq f^i$; moreover, from $f \equiv \varphi(f)$ we deduce $f^i \equiv f$, whence:

Property 2: *the sequence $\{h^i\}$ is bounded above by every fixed point of $\varphi(f)$*, and finally:

Property 3: *$I(\varphi(I))$ is the smallest fixed point of $\varphi(\chi)$ and $I(\varphi(I)) \equiv \lim_i\{h^i\}$.*

Remarks: 1. Defining the *length* of a function as the number of symbols it contains, it is immediate that $\mathrm{len}(h^{i-1}) < \mathrm{len}(h^i)$, so in general $\mathrm{len}(\lim_i\{h^i\})$ is unbounded. This is the case, with exceptions, for the loop-functions studied further on (§ 5).

2. $\varphi(\chi)$ may possess other fixed points, among which the relation $\preccurlyeq$ does not necessarily hold.

Property 4:

$$I(f\varphi(I)S^2) \equiv fI(\varphi(fIS^2))S^2 \qquad \text{or} \qquad \chi f\varphi(\chi)S^2\,\mathbf{rec} \equiv f\chi\varphi(f\chi S^2)\,\mathbf{rec}.$$

Proof: Let f_1 be the smallest fixed point of $f\varphi(\chi)S^2$ and f_2 that of $\varphi(f\chi S^2)$.

$$f_1 \equiv I(f\varphi(I)S^2) \quad \text{and} \quad f_1 \equiv f\varphi(f_1)S^2$$

$$f_2 \equiv I(\varphi(fIS^2)) \quad \text{and} \quad f_2 \equiv \varphi(ff_2S^2)$$

whence $fI(\varphi(fIS^2))S^2 \equiv ff_2S^2 \equiv f\varphi(ff_2S^2)S^2$; f_1 and f_2 being smallest fixed points, we deduce $f_1 \equiv ff_2S^2$. ■

4.2. General recursion functions

Suppose the translation of the statement of a problem into the following statement has been accomplished:

Construct a function f such that $f \equiv \varphi(f)$, $\varphi(f)$ being thus constructed from the statement of the problem. The solution as apprehended by computer science is then:

$$f \equiv I(\varphi(f)S_I^f)$$

This leads to a natural generalization:

[2]We restrict ourselves to the properties needed for what follows; the reader may, as an exercise, establish with the help of $\mathcal{A}$ the result of Cadiou [3]: "every interpretation of a recursion function, different from the one given in the exposition, leads to a restriction of the smallest fixed point of $\varphi(\chi)$."

Construct a function f_1 satisfying $f_1 \equiv \varphi_1(f_1, \dots, f_n)$ where $f_2, \dots, f_n$ satisfy:

$$f_i \equiv \varphi_i(f_1, \dots, f_n), \qquad i = 2, \dots, n.$$

Let us restrict this exposition to the case $n = 2$.

The roles of f_1 and f_2 are symmetric, but it is essential that the function submitted to the executor respect the role of one relative to the other. If it is f_1 that solves the problem, then $f_2 \equiv \varphi_2(f_1, f_2)$ leads to $f_2 \equiv I_2(\varphi_2(f_1, I_2))$, and the problem becomes:

Construct f_1 such that

$$f_1 \equiv \varphi_1(f_1, I_2(\varphi_2(f_1, I_2)))$$

The previous approach furnishes the solution:

$$f_1 \equiv I_1(\varphi_1(I_1, I_2(\varphi_2(I_1, I_2)))) \tag{1}$$

and symmetrically:

$$f_2 \equiv I_2(\varphi_2(I_1(\varphi_1(I_1, I_2)), I_2)) \tag{2}$$

or:

$$f_1 \equiv \chi_1 \varphi_1(\chi_1, \chi_2 \varphi_2(\chi_1, \chi_2)\,\mathbf{rec})\,\mathbf{rec}$$

$$f_2 \equiv \chi_2 \varphi_2(\chi_1 \varphi_1(\chi_1, \chi_2)\,\mathbf{rec}, \chi_2)\,\mathbf{rec}$$

The generalization to the case $n > 2$ presents no difficulty.

The interpretation of $I_1(\varphi_1(I_1, I_2(\varphi_2(I_1, I_2))))$ in terms of Kleene functions is immediate:

$$\begin{cases} h_1^0 \equiv \varphi_1(f_\omega, \varphi_2(f_\omega, f_\omega)) \\ h_1^i \equiv \varphi_1(h_1^{i-1}, \varphi_2(h_1^{i-1}, h_2^{i-1})) \end{cases} \quad \text{where} \quad \begin{cases} h_2^0 \equiv \varphi_2(f_\omega, f_\omega) \\ h_2^i \equiv \varphi_2(h_1^{i-1}, h_2^{i-1}) \end{cases} \tag{3}$$

We deduce that:

$$f_1 \equiv \lim_i \{h_1^i\}$$

and if $f_2 \equiv \lim_i \{h_2^i\}$:

$$f_1 \equiv \varphi_1(f_1, f_2), \qquad f_2 \equiv \varphi_2(f_1, f_2) \tag{4}$$

The pair $\langle f_1, f_2 \rangle$ is the smallest fixed point of the pair of functionals $\langle \varphi_1(\chi_1, \chi_2), \varphi_2(\chi_1, \chi_2) \rangle$, the approach being analogous to that of simple recursion, under the hypothesis of continuity.[3]

It is clear that case (2) leads to the same result.

Property 5:

$$\chi \varphi(\chi)\,\mathbf{rec} \equiv \chi_1 \chi_2 \varphi(\chi_1, \chi_2)\,\mathbf{rec\,rec} \qquad \text{or} \qquad I(\varphi(I)) \equiv I_1(I_2(\varphi(I_1, I_2)))$$

The body of $I(\varphi(I))$ being $\varphi(\chi)$, let us form two subsets of the occurrences of χ, and denote by χ_1 the function present in the occurrences of the first subset, and by χ_2 those of the second subset: this leads to the function on the right-hand side above.

Let f_2 be the smallest fixed point of $\varphi(\chi_1, \chi_2)$ considered as a functional of χ_2, whence:

$$f_2(\chi_1) \equiv I_2(\varphi(\chi_1, I_2)) \qquad \text{and} \qquad f_2(\chi_1) \equiv \varphi(\chi_1, f_2(\chi_1))$$

$$I_1(I_2(\varphi(I_1, I_2))) \equiv I_1(f_2(I_1))$$

Let f' be the smallest fixed point of $f_2(\chi_1)$:

$$f' \equiv I_1(I_2(\varphi(I_1, I_2))) \qquad \text{and} \qquad f' \equiv f_2(f')$$

then $f_2(f') \equiv \varphi(f', f_2(f'))$, whence $f' \equiv \varphi(f', f')$, from which we conclude that f' is the smallest fixed point of $\varphi(\chi, \chi)$, that is, of $\varphi(\chi)$. ■

[3] Let h denote the function $f_1 f_2 \pi_0 S^3$, $f_1 \equiv h\pi_1 S^2$, $f_2 \equiv h\pi_2 S^2$; then system (4) becomes: $h \equiv \varphi_1(h\pi_1 S^2, h\pi_2 S^2)\varphi_2(h\pi_1 S^2, h\pi_2 S^2)\pi_0 S^3$, of the form $h \equiv \varphi(h)$, whence the possibility of reducing to the case of a simple recursion.

4.3. Calculation rules in $\mathcal{A}$

Property 6: *If* $\forall \chi \in \mathcal{A}$, $\varphi_1(\chi) \preccurlyeq \varphi_2(\chi)$, *then* $I(\varphi_1(I)) \preccurlyeq I(\varphi_2(I))$

If $\forall \chi \in \mathcal{A}$, $\varphi_1(\chi) \equiv \varphi_2(\chi)$, *then* $I(\varphi_1(I)) \equiv I(\varphi_2(I))$

The proof is immediate from § 4. If $\varphi_1(\chi) \preccurlyeq \varphi_2(\chi)$ then $\varphi_1(f_\omega) \preccurlyeq \varphi_2(f_\omega)$, whence $h_1^0 \preccurlyeq h_2^0$, and step by step $h_1^i \preccurlyeq h_2^i$, whence $\lim_i\{h_1^i\} \preccurlyeq \lim_i\{h_2^i\}$.

We deduce the *calculation rule*:

If, by the calculus already defined within $\mathcal{A}$, one establishes that $\varphi_1(I) \preccurlyeq \varphi_2(I)$, treating I as a function, one obtains the same relation between the corresponding recursion functions (and likewise for equivalence).

This extends to general recursion functions.

5. Loop-functions

5.1. Primitive loop-function

Definition: $I(p \to fIS^2, \pi_0)$ is a *primitive loop-function.*

Property 1: $I(p \to fIS^2, f') \equiv I(p \to fIS^2, \pi_0)f'S^2$

This property reduces to a primitive loop-function (as a left factor) every loop-function with body $\varphi(\chi) \equiv (p \to f\chi S^2, f')$; we shall say that such a function is a *simple* loop-function.

We shall not give a proof of this property here, since we shall demonstrate further on a general property of which it is a particular case (Property 4, § 5.2).

Property 2: $I(p \to fIS^2, \pi_0) \equiv (p \to I(f(p \to I, \pi_0)S^2), \pi_0)$

Proof: By axiom A_{14}:

$$I(p \to fIS^2, \pi_0) \equiv (p \to fI(p \to fIS^2, \pi_0)S^2, \pi_0)$$

and by Property 4 of § 4, the right-hand side becomes:

$$(p \to I(f(p \to I, \pi_0)S^2), \pi_0)$$

which completes the proof.

Remark: $I(p \to fIS^2, \pi_0)$ expresses the semantics of the instruction:

$$\textbf{while } p(a) \textbf{ do } a := f(a)$$

a being an evocation of the name given to the object submitted to the function. Likewise, $I(f(p \to I, \pi_0)S^2)$ expresses the semantics of the instruction:

$$\textbf{repeat } a := f(a) \textbf{ until } \neg\, p(a)$$

We shall agree on the abbreviation $\neg p$ for $pf_\neg S^2$, where $f_\neg$ was defined in axiom A_2.

Property 3: *The Kleene functions of* $(p \to f\chi S^2, \pi_0)$ *are given by:*

$$\forall i \in N, \qquad h^i \equiv i\,.\,(p \to f, \pi_0)(p \to f_\omega, \pi_0)S^2$$

Let $i \in N$; $i\,.\,f$ denotes the i-fold iterate of f; $i\,.\,f$ is thus an abbreviation of $\underbrace{f\ldots f}_{i \text{ times}}\underbrace{S^2\ldots S^2}_{i-1 \text{ times}}$, or better of $\pi_0\underbrace{f\ldots f}_{i \text{ times}}\underbrace{S^2\ldots S^2}_{i \text{ times}}$, whence $0\,.\,f \equiv \pi_0$.

Proof: For $i = 0$ the property is obviously true.

For $i = 1$:

$$h^1 \equiv (p \to f(p \to f_\omega, \pi_0)S^2, \pi_0) \equiv (p \to f, \pi_0)(p \to f_\omega, \pi_0)S^2 \qquad \text{(axiom } A_3\text{)}$$

Suppose the property true for $i-1$; we have:

$$h^i \equiv (p \to f\ (i-1)\,.\,(p \to f, \pi_0)(p \to f_\omega, \pi_0)S^2 S^2, \pi_0)$$

$$h^i \equiv (p \to f, \pi_0)\ (i-1)\,.\,(p \to f, \pi_0)S^2(p \to f_\omega, \pi_0)S^2$$

by the repeated application, i times, of axiom A_3; hence the property holds for every i.

Consequence, by Property 1: the Kleene functions h^i of $(p \to f\chi S^2, f')$ are:

$$h^i \equiv i\,.\,(p \to f, \pi_0)(p \to f_\omega, \pi_0)S^2 f' S^2$$

5.2. General (simple) loop-function

Definition 1: $\varphi(\chi)$ is the body of a *general loop-function* if all occurrences of χ are terminal in $\varphi(\chi)$.

We shall denote by $\varphi(\chi, f_1, \ldots, f_n)$ the body of a general loop-function in which, besides χ, $f_1 \ldots f_n$ are its terminal functions, none of these latter functions containing χ.

Property 4: $\varphi(\chi)$ *being the body of a general loop-function:*

$$I(\varphi(I))fS^2 \equiv I(\varphi(I, f_1, \ldots, f_n)S^{f_1}_{f_1 f S^2} \cdots S^{f_n}_{f_n f S^2}) \tag{1}$$

Proof: let h_1^i be the Kleene functions of the body of the function on the left-hand side of (1), and h_2^i those of the body of the function on the right-hand side.

$$h_1^0 \equiv \varphi(f_\omega, f_1, \ldots, f_n)$$

$$h_2^0 \equiv \varphi(f_\omega, f_1 f S^2, \ldots, f_n f S^2)$$

f_ω is terminal in both functions above; now $f_\omega \equiv f_\omega f S^2$, whence, by Property 4 of § 3.2, $h_2^0 \equiv \varphi(f_\omega, f_1, \ldots, f_n)fS^2 \equiv h_1^0 f S^2$.

Suppose that $h_1^i f S^2 \equiv h_2^i$ for every integer i; then:

$$\begin{aligned} h_1^{i+1} f S^2 &\equiv \varphi(h_1^i, f_1, \ldots, f_n) f S^2 \\ &\equiv \varphi(h_1^i f S^2, f_1 f S^2, \ldots, f_n f S^2) \quad \text{(Property 4, § 3.2)} \\ &\equiv \varphi(h_2^i, f_1 f S^2, \ldots, f_n f S^2) \quad \text{(hypothesis)} \\ &\equiv h_2^{i+1} \quad \text{(definition of } h_2^{i+1}\text{)} \end{aligned}$$

Finally, for every i, $h_1^i f S^2 \equiv h_2^i$, whence the conclusion by Property 3, § 4. ■

Calculation rule: Property 4 allows us to state a calculation rule in $\mathcal{A}$ that includes general loop-functions:

— Let $I(\varphi(I))$ be a general loop-function; the product $I(\varphi(I))fS^2$ is carried out by applying the rule established in § 3.2, supplemented by the rule: $IfS^2 \equiv I$.

Conversely, this allows factorization by f under the same conditions as the rule of § 3.2, disregarding the labels I, if and only if all occurrences of the label I are terminal.

The above rule justifies our treating the label I as a function within the body of a loop-function.

Lemma 1: *Every conditional function can be put into the equivalent form:*

$$(q_1 \to f_1, (q_2 \to f_2, (\ldots, (q_n \to f_n, f'_n) \ldots))) \tag{2}$$

where the functions $f_1, \ldots, f_n, f'_n$ *are not conditional functions.*

Proof: We argue by induction on the number n of predicate functions contained in (1).

For $n = 1$ the property is evident.

For $n = 2$, only the case $(p_1 \to (p_2 \to f_1, f_2), f_2')$ is not of form (1); but by the axioms of $\mathcal{A}$ it is immediate that:

$$(p_1 \to (p_2 \to f_1, f_2), f_2') \equiv ((p_1 \to p_2, FS^0) \to f_1, (p_1 \to f_2, f_2'))$$

so the lemma is satisfied with $q_1 \equiv (p_1 \to p_2, FS^0)$, $q_2 \equiv p_1$.

Suppose the property true for $n \in N$; we may write it:

$$\varphi_n \equiv (q_1 \to f_1, (q_2 \to f_2, (\dots, (q_j \to f_j, \psi_j) \dots)))$$

where $\psi_j \equiv (q_{j+1} \to f_{j+1}, (q_{j+2} \to \dots, (q_n \to f_n, f_n') \dots))$

To construct, from this, an arbitrary conditional function, it suffices:

a) to choose j arbitrarily between 1 and n and to substitute for the function f_j a conditional function such as $(p \to g, g')$, g and g' not being conditional, since

$$(q_j \to (p \to g, g'), \psi_j) \equiv ((q_j \to p, FS^0) \to g, (q_j \to g', \psi_j)).$$

This establishes that the function φ_{n+1} so obtained takes form (1).

b) to replace, in φ_n, ψ_j either by $h\psi_j S^2$ or by $\psi_j h S^2$, and finally to replace h by a conditional function as in the preceding case; we are then reduced to case a), as is easily verified.

Finally, whatever the manner in which φ_{n+1} is obtained from form (1) of a conditional function φ_n, the function obtained can be put into form (1).

Lemma 2: *Within the set $\mathcal{F}_n = \{f_1, \dots, f_n\}$ of terminal functions of a function φ_n, consider an arbitrary proper subset $\mathcal{F}_n'$, of which $\mathcal{F}_n''$ is the complementary set with respect to $\mathcal{F}_n$; then every conditional function having $\mathcal{F}_n$ as its set of terminal functions is equivalent to:*

$$(q \to \theta_n, \theta_n') \tag{3}$$

θ_n having $\mathcal{F}_n'$ as its set of terminal functions, and θ_n', $\mathcal{F}_n''$.

Proof: For $n = 1$, by axiom A_2 every function $(p \to f, f')$ satisfies (3) with $\mathcal{F}_1' = \{f\}$ or $\mathcal{F}_1' = \{f'\}$.

For $n = 2$, suppose Lemma 1 applied; then:

$$\varphi_2 \equiv (q_1 \to f_1, (q_2 \to f_2, f_2')) \tag{4}$$

it satisfies (3) for $\mathcal{F}_2' = \{f_1\}$, and by axiom A_2 for $\mathcal{F}_2' = \{f_2, f_2'\}$.

Now, by calculation in $\mathcal{A}$, (4) leads to:

$$\varphi_2 \equiv ((\neg q_1 \to q_2, FS^0) \to f_2, (q_1 \to f_1, f_2'))$$

which satisfies (3) for $\mathcal{F}_2' = \{f_2\}$ and, by axiom A_2, for $\mathcal{F}_2' = \{f_1, f_2'\}$.

Finally, considering:

$$\varphi_2 \equiv ((\neg q_1 \to \neg q_2, FS^0) \to f_2', (q_1 \to f_1, f_2))$$

completes the inventory of possible cases.

The lemma is thus verified for $n = 1$ and $n = 2$.

Suppose it true for arbitrary n:

$$\varphi_n \equiv (q \to \theta_n, \theta_n') \tag{5}$$

By Lemma 1, every function φ_{n+1} takes the form:

$$\varphi_{n+1} \equiv (p \to f, \varphi_n), \qquad q_1 \equiv p, \ f_1 \equiv f, \ \varphi_n \equiv \psi_1 \tag{6}$$

$\mathcal{F}_n$ being the set of terminal functions of φ_n, two cases arise:

a) $\mathcal{F}'_{n+1} = \{f\} \cup \mathcal{F}'_n$: from (5) and (6) we deduce

$$\varphi_{n+1} \equiv ((p \to VS^0, q) \to (p \to f, \theta_n), \theta'_n)$$

the lemma is thus verified with $\theta_{n+1} \equiv (p \to f, \theta_n)$.

b) $\mathcal{F}'_{n+1} = \mathcal{F}'_n$: we deduce from (5) and (6)

$$\varphi_{n+1} \equiv ((\neg p \to q, FS^0) \to \theta_n, (p \to f, \theta'_n))$$

and the lemma is verified.

The lemma thus holds for every conditional function.

Property 5: *Every general loop-function is equivalent to a simple loop-function.*

Proof: we must show that every general loop-function is equivalent to a function $I(p \to fIS^2, f')$ where p, f, f' do not contain I.

Let $\varphi(\chi)$ be the body of the proposed loop-function; there is no loss of generality in taking χ terminal in a finite set of terminal functions $g_1\chi S^2, \dots, g_k\chi S^2$; the body of the loop-function is thus represented by:

$$\varphi(g_1\chi S^2, \dots, g_k\chi S^2, f_1, \dots, f_n) \tag{7}$$

$f_1, \dots, f_n$ being the other terminal functions of φ.

Lemma 2 allows us to write:

$$\varphi \equiv (q \to \theta_n, \theta'_n)$$

θ_n containing, as terminal functions, only the functions $g_1\chi S^2, \dots, g_p\chi S^2$.

The calculation rule of § 3.2 allows us to put χ as a right factor in θ_n, whence $\theta_n \equiv f\chi S^2$, f containing, as terminal functions, only the functions $g_1, \dots, g_k$.

This completes the proof.

Remarks: 1) The proof shows that the predicate function p in $I(p \to fIS^2, f')$ is a conditional function formed from φ in such a way that it is most often impossible to express it by an instruction permitted by a programming language.

As an example, let the function be:

$$\varphi(\chi) \equiv (p_1 \to f_1\chi S^2, (p_2 \to f_2\chi S^2, f_3))$$

$I(\varphi(I))$ is expressed by a labeled loop, two conditional instructions, and two GOTOs. Property 5, applied to this case, gives:

$$\varphi(\chi) \equiv ((p_1 \to VS^0, p_2) \to (p_1 \to f_1, f_2)\chi S^2, f_3)$$

Now consider the example:

$$\varphi(\chi) \equiv (p_1 \to q(p_2 \to f_1\chi S^2, f_2)S^2, f_3)$$

One arrives at:

$$\varphi(\chi) \equiv ((p_1 \to gp_2S^2, FS^0) \to f_1\chi S^2, (p_1 \to gf_2S^2, f_3))$$

not directly translatable by a **while–do** instruction.

2) We observe, on the contrary, that factorization by adjunction of variables (§ 3.3) does not modify the structure of the body $\varphi(\chi)$, at least not as radically as in the previous case; this justifies the interest of [2] and of the generalization that follows from it.

The path by which a general loop-function is reduced to a simple loop-function, hence to a primitive loop-function (which is expressed in a programming language without labels and without `goto`), is therefore not unique.

Applying Property 6 of § 3.3 to the body (7) of a general loop-function gives, in effect:

$$\varphi(g_1\chi S^2, \dots, g_k\chi S^2, f_1, \dots, f_n) \equiv \varphi'(p^* \to \pi_1\chi S^2, \pi_1)S^2 \tag{8}$$

where $\varphi' \equiv \varphi(g_1 h_V S^2, \ldots, g_k h_V S^2, f_1 h_F S^2, \ldots, f_n h_F S^2)$

The properties established by Böhm and Jacopini are obtained easily.

By (8) and Property 6 of § 4:

$$I(\varphi(I)) \equiv I(\varphi'(p^* \to \pi_1 I S^2, \pi_1) S^2) \tag{9}$$

Now $h_V \pi_1 S^2 \equiv \pi_0$, so the right-hand side of (9) becomes:

$$h_V \pi_1 I(\varphi'(p^* \to \pi_1 I S^2, \pi_1) S^2) S^2 S^2$$

whence, by Property 4 of § 4:

$$h_V I(\pi_1 \varphi' S^2 (p^* \to I, \pi_1) S^2) S^2$$

Now (Property 2, § 5.1):

$$\begin{aligned} h_V I(p^* \to \pi_1 \varphi' S^2 I S^2, \pi_0) S^2 &\equiv h_V (p^* \to I(\pi_1 \varphi' S^2 (p^* \to I, \pi_0) S^2), \pi_0) S^2, \\ &\equiv h_V I(\pi_1 \varphi' S^2 (p^* \to I, \pi_1) S^2) S^2, \qquad (h_V p^* S^2 \equiv V S^0) \end{aligned}$$

finally, returning to (9):

$$I(\varphi(I)) \equiv h_V I(p^* \to \pi_1 \varphi' S^2 I S^2, \pi_0) S^2 \pi_1 S^2 \tag{10}$$

The structure of φ' respects that of φ, and if the latter is constructed so as to be translatable into instructions of a programming language, or expresses the semantics of such instructions (according to the approach chosen), the primitive loop-function contained in (10) enjoys these properties as well.

3) The foregoing certainly does not exhaust the properties of the loop-functions studied; thus it can be shown that every loop-function is equivalent to a function formed with primitive loop-functions (expressed by **while–do** instructions).

The following example illustrates the approach.

Let the function be

$$\chi(p \to f\chi S^2, \varphi(\chi))\,\mathbf{rec}$$

By Property 5, § 4:

$$\chi(p \to f\chi S^2, \varphi(\chi))\,\mathbf{rec} \equiv \chi_1 \chi_2 (p \to f\chi_2 S^2, \varphi(\chi_1))\,\mathbf{rec}\,\mathbf{rec}$$

Now, for every function $\varphi(\chi_1)$, by Property 1, § 5.1:

$$\chi_2(p \to f\chi_2 S^2, \varphi(\chi_1))\,\mathbf{rec} \equiv \chi_2(p \to f\chi_2 S^2, \pi_0)\,\mathbf{rec}\,\varphi(\chi_1) S^2$$

From this, we deduce:

Property 6: $I(p \to f I S^2, \varphi(I)) \equiv I_1(I_2(p \to f I_2 S^2, \pi_0)\varphi(I_1) S^2)$

By Property 4, § 4, we finally obtain:

Property 7:

$$I(p \to f I S^2, \varphi(I)) \equiv I_2(p \to f I_2 S^2, \pi_0) I_1(\varphi(I_2(p \to f I_2 S^2, \pi_0) I_1 S^2)) S^2$$

It is immediate that, if χ is terminal in $\varphi(\chi)$, (1) is then a general loop-function, and the property leads to the product of two primitive loop-functions, one of which contains a primitive loop-function.

Once again, Properties 6 and 7 indicate that the calculus in $\mathcal{A}$ is extended by treating labels as functions within the body of a loop-function.

Example: $\varphi(\chi) \equiv (p_1 \to f_1\chi S^2, \pi_0)$; Property 7 gives the equivalence:

$$I(p \to fIS^2, (p_1 \to f_1IS^2, \pi_0))$$

$$\equiv I_2(p \to fI_2S^2, \pi_0)I_1(p_1 \to f_1I_2(p \to fI_2S^2, \pi_0)S^2I_1S^2, \pi_0)S^2$$

it expresses the semantics of the equivalence (of which it is the proof):

$$I: \ \textbf{if } p(a) \textbf{ then } a := f(a); \ \textbf{goto } I \textbf{ else if } p_1(a) \textbf{ do begin } a := f_1(a);$$

$$\textbf{goto } I \textbf{ end} \equiv \textbf{while } p(a) \textbf{ do } a := f(a); \ \textbf{while } p_1(a) \textbf{ do begin}$$

$$a := f_1(a); \ \textbf{while } p(a) \textbf{ do } a := f(a); \ \textbf{end}$$

5.3. Second generalization

Definition 2: We shall call a *general loop-function* a function f_1 defined by the system: $f_1 \equiv \varphi_1(f_1, \ldots, f_n)$, $\ldots$, $f_n \equiv \varphi_n(f_1, \ldots, f_n)$, where all the occurrences of $f_1, \ldots, f_n$ are terminal in the functions $\varphi_1, \ldots, \varphi_n$.

It encompasses the previous one (§ 5.2) if we take $\varphi_2 \equiv f_2, \ldots, \varphi_n \equiv f_n$. To illustrate this definition without overburdening this note, let us take the case $n = 2$.

Then:

$$f_1 \equiv I_1(\varphi_1(I_1, I_2(\varphi_2(I_1, I_2)))) \tag{1}$$

all occurrences of I_1, I_2 being terminal. (1) expresses the relation:

$$f_1 \equiv \chi_1\varphi_1(\chi_1, \chi_2\varphi_2(\chi_1, \chi_2) \,\textbf{rec}) \,\textbf{rec} \tag{2}$$

χ_1, χ_2 being terminal in φ_2; by Lemma 2, § 5.2, and the calculation rule of § 3.2:

$$\varphi_2(\chi_1, \chi_2) \equiv (q \to f\chi_2S^2, \theta'_n) \tag{3}$$

χ_2 having no occurrence in θ'_n.

In θ'_n, all occurrences of χ_1 are terminal by hypothesis, so, in the same manner:

$$\theta'_n \equiv (q_1 \to f_1\chi_1S^2, f) \tag{4}$$

χ_1 having no occurrence in f.

It follows that:

$$\chi_2\varphi_2(\chi_1, \chi_2) \,\textbf{rec} \equiv \chi_2(q \to f\chi_2S^2, (q_1 \to f_1\chi_1S^2, f)) \,\textbf{rec}$$

$$\equiv \chi_2(q \to f\chi_2S^2, \pi_0) \,\textbf{rec}\, (q_1 \to f_1\chi_1S^2, f)S^2$$

Returning to (2):

$$f_1 \equiv \chi_1\varphi_1(\chi_1, \chi_2(q \to f\chi_2S^2, \pi_0) \,\textbf{rec}\, (q_1 \to f_1\chi_1S^2, f)S^2) \,\textbf{rec}$$

whence

$$f_1 \equiv I_1(\varphi_1(I_1, I_2(q \to fI_2S^2, \pi_0)(q_1 \to f_1I_1S^2, f)S^2)) \tag{5}$$

which satisfies the definition of § 5.2, the body of the general loop-function containing at least one occurrence of a primitive loop-function.

We leave it to the reader to verify that this property holds in general.

It is clear that this section does not exhaust the subject; it illustrates the use of the algebra of functions.

6. Equivalence problems

We shall conclude this note with two properties of some practical interest.

Property 1: *Let* h_1^i *and* h_2^i *be the Kleene functions of* $\varphi_1(\chi)$ *and* $\varphi_2(\chi)$ *respectively;* $I(\varphi_1(I)) \preccurlyeq I(\varphi_2(I))$ *if and only if* $h_1^i \preccurlyeq h_2^i$ *for every* i *(likewise for equivalence).*

The proof is immediate from § 4.

Example: $\varphi_1(\chi) \equiv (p \to f\chi S^2 \chi S^2, \pi_0)$, $\varphi_2(\chi) \equiv (p \to f\chi S^2, \pi_0)$.

For φ_2, by Property 3, § 5.1, h_2^i is known.

It is clear that $h_1^0 \equiv h_2^0$.

Suppose that $h_1^i \equiv h_2^i$:

$$h_1^{i+1} \equiv (p \to f h_1^i S^2 h_1^i S^2, \pi_0) \equiv (p \to f h_2^i S^2 h_2^i S^2, \pi_0)$$

Now, by an elementary calculation in $\mathcal{A}$, one verifies[4] that $h_2^i h_2^i S^2 \equiv h_2^i$, whence $h_1^{i+1} \equiv h_2^{i+1}$.

Property 2: *If* $\varphi_1(f_\omega) \equiv g\varphi_2(f_\omega)S^2$ *and* $\forall i \in N$, $\varphi_1(gh_2^i S^2) \equiv gh_2^{i+1}S^2$, *then* $I(\varphi_1(I)) \equiv gI(\varphi_2(I))S^2$.

This property allows the study of the relations between recursion and iteration [10], and more generally in every case where the computation of the Kleene functions of $\varphi_2(\chi)$ is possible (which generally occurs for loop-functions).

7. Final note

The introduction of names designating objects of $B \cup \mathcal{L}_B$, and of names designating functions, presents no difficulty and completes the constitution of the algebra of functions as a computing tool; we refer the reader to [10], notwithstanding the imperfections of that text.

The direct link between our subject and computer science is established through the concept of the *executor*; see the first remark on this subject, § 1.2. If a single executor, or such an executor, is made active by a microprogram, for example, it is clear that our approach applies at that level. The connection with programming languages is less direct (except toward an assembly-type language, for the reasons mentioned above); it seems to us possible to establish it in every case, without difficulty of principle.

I am grateful to Maurice Nivat for the criticisms he made of the first manuscript submitted to the RIRO; the present note has benefited from the discussions we have had.

References

[1] J. ARSAC, *Les langages sans étiquettes.* Publication 73/13. Institut de Programmation, Université de Paris VI.

[2] C. BÖHM et G. JACOPINI, *Flow diagrams, Turing machines and languages with only two formation rules*, Comm. ACM, May 1966, vol. 9, n° 5, 366–371.

[3] J. CADIOU, *Recursive definitions of partial functions and their conditions.* Ph.D. thesis, Computer Science Department, Stanford University, June 1972.

[4] More generally, $h_2^i h_2^j S^2 \equiv h_2^i$, which suggests the property:

$$2 . I(p \to fIS^2, \pi_0) \equiv I(p \to fIS^2, \pi_0)$$

(idempotence of the primitive loop-function). Indeed, the left-hand side gives successively:

$$2 . I(\dots) \equiv \lim_i \{h_2^i\} I(\dots) S^2 \equiv \lim_i \{h_2^i I(\dots) S^2\} \equiv \lim_i \left\{ h_2^i \lim_j \{h_2^j\} S^2 \right\}$$

$$\equiv \lim_{i,j} \{h_2^i h_2^j S^2\} \equiv \lim_i \{h_2^i\}.$$

[4] S. C. KLEENE, *Introduction to Metamathematics* (the recursion theorem, p. 348). Van Nostrand, 1952.

[5] J. KOTT, *Remarques sur la structure des schémas de programme*, in *Théorie des automates, des langages et de la programmation*, Colloques IRIA, 1972, pp. 191–194.

[6] A. L. MALCEV, *Algorithms and recursive functions*, Wolters–Noordhoff Publishing Corp., Netherlands, 1970.

[7] H. D. MILLS, *The new Math of Computer Programming*, Comm. ACM, January 1975, vol. 18, n° 1, pp. 43–48.

[8] R. E. PRATHER, *A convenient Cryptomorphic Version of Recursive Function Theory*, in *Information and Control*, vol. 27, n° 2, February 1975, pp. 178–195.

[9] F. H. RAYMOND, *Tentative de formalisation en informatique*, in *Structure et programmation des calculateurs*, Séminaires IRIA, 1972, pp. 201–248. And: *Formalisation du concept de calcul.* Publications 25 et 32, 1972–1973, Institut de Programmation, Université Paris VI.

[10] F. H. RAYMOND, *Algèbre des fonctions.* Cours du CNAM. Éditions Scientifiques Riber, Paris, 1974.