INSTITUTUL
DE
MATEMATICA

INSTITUTUL NATIONAL
PENTRU CREATIE
STIINTIFICA SI TEHNICA

ISSN 0250 3638

A FORMAL REPRESENTATION OF FLOWCHART SCHEMES

by

Virgil-Emil CAZANESCU and Gh. STEFANESCU

PREPRINT SERIES IN MATHEMATICS
No. 22/1987

BUCURESTI

A FORMAL REPRESENTATION OF FLOWCHART SCHEMES

by

Virgil-Emil CAZANESCU*) and Gh. STEFANESCU**)

June 1987

*) *Department of Mathematics, University of Bucharest, Bucharest, Romania.*

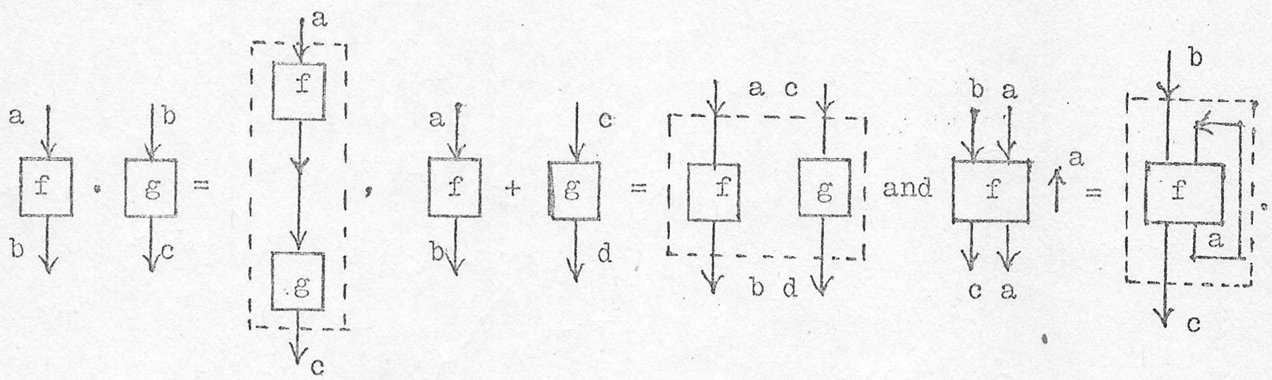
**) *Department of Mathematics, The National Institute for Scientific and Technical Creation, Bd. Pacii 220, 79622 Bucharest, Romania.*

A FORMAL REPRESENTATION OF FLOWCHART SCHEMES

by

Virgil-Emil CĂZĂNESCU, Gheorghe ŞTEFĂNESCU

This paper is included in a sequence of papers (beginning with [5,6]) where we intend to give a new foundation of the algebraic theory of multi-input/multi-exit flowchart schemes. The main new feature of this approach is the use of a new operation modelling cycles, called feedback [14, 15]; more precisely we believe that the basic operations on flowchart schemes are composition ·, (separated) sum + and feedback ↑, which pictorially look as follows



In the present paper the natural formal representation of flowchart schemes in terms of composition, sum and feedback is studied: we introduce an algebraic structure (called flow) such that the algebra of flowchart scheme representations, obtained using statements in a double ranked set and connections in a flow is a flow having a universal property similar to the universal property of polynomials.

In order to motivate our representation of flowchart schemes let us consider the following example. The flowchart scheme in Figure 1 (a) may be rearranged as in Figure 1 (b).

In order to obtain the scheme in Figure 1 (b) we use the following method:

a) we put all the statements in the first picture in an (arbitrary) linear order; in our case this linear order is x (top), y, z and x (bottom). (This linear order, included in an artificial way, generates the difference between a flowchart scheme and one of its representations, namely a flowchart scheme is the set of all the representations

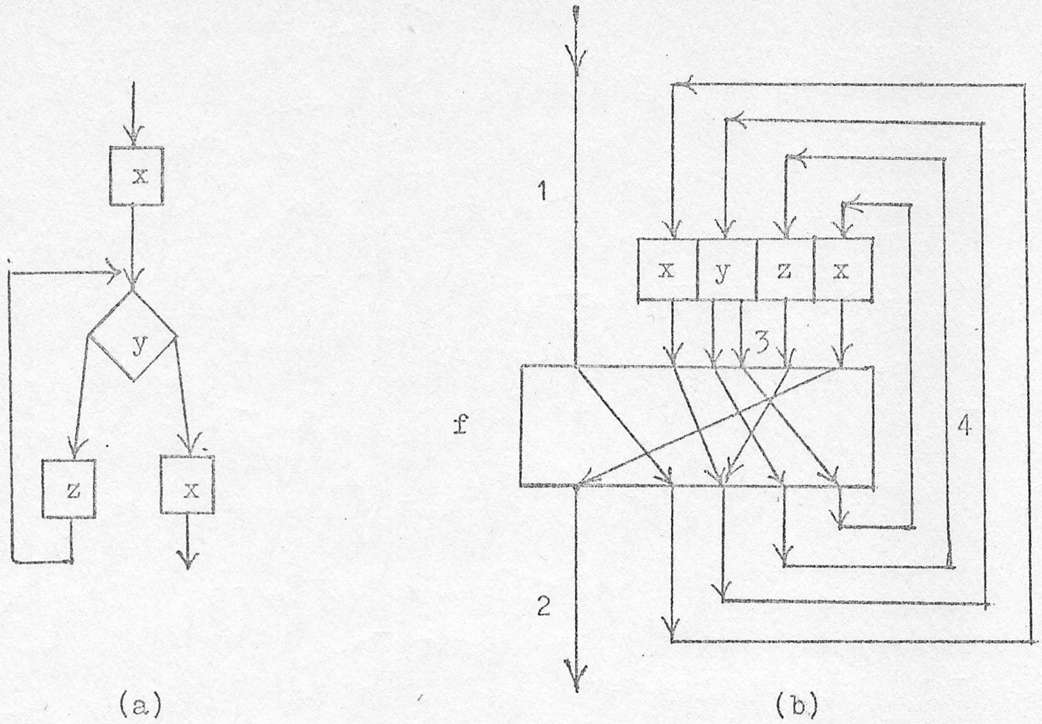


Figure 1. The standard forme of a flowchart scheme

obtained putting different linear orders on the statements of the scheme.);

b) we draw the rectangle f and its external connections as follows: the arrow 1, giving the input into the scheme; the arrow 2, giving the output from the scheme; the arrows 3 which connect all the exits of the statements to f; and the arrows 4 which connect f to all the inputs of the statements;

c) we draw the arrows inside the rectangle f is such a way that the connections are the same as in the first picture.

Putting $[n] = \{1,2,\ldots,n\}$ for every nonnegative integer n we remark the rectangle f contains a function

$$f : [n + o(xyzx)] \rightarrow [p + i(xyzx)]$$

where $n = 1$ is the number of the inputs into the scheme, $p = 1$ is the number of the exits from the scheme, $o(xyzx) = 5$ is the sum of the exit numbers of the statements and $i(xyzx) = 4$ is the sum of the input numbers of the statements.

Thus using Figure 1 (b) the flowchart scheme in Figure 1 (a) is represented as the ordered pair (xyzx, f) or as the formal expression $((1_1 + x + y + z + x) \cdot f)\uparrow^4$, where f is the function which maps 1,2,3,4,5,6 into 2,3,4,5,3,1, respectively. The type of the scheme is correlated with the type of f. For example, for a partial scheme the rectangle f is a partial function and for a nondeterministic scheme f is a relation.

Notice that our representation differs from all representations of flowchart schemes in [1,3,9].

In the sequel we use a more general framework. The monoid M is the monoid of sorts. Its neutral element is denoted by λ. The monoid X is the monoid of statements. Its neutral element is denoted by ε. We assume that $i : X \to M$ and $o : X \to M$ are two monoid morphisms.

The particular cases of interest are: the monoid M is freely generated by the set of sorts (in the above one-sorted example, M is the additive monoid of nonnegative integers) and the monoid X is freely generated by the set of statements. For every $x \in X$, $i(x)$ gives the number and the sorts of the inputs of x while $o(x)$ gives the number and the sorts of the outputs of x.

In this more general framework we have to work with representations of schemes as ordered pairs. The calculus using formal expressions (in the particular case when X is a free generated monoid) is briefly presented in [15].

Now, the algebra of formal representations of flowchart schemes can be defined as follows. Let B be an M-flow (cf. definition 1.4 below). For $a,b \in M$ we define

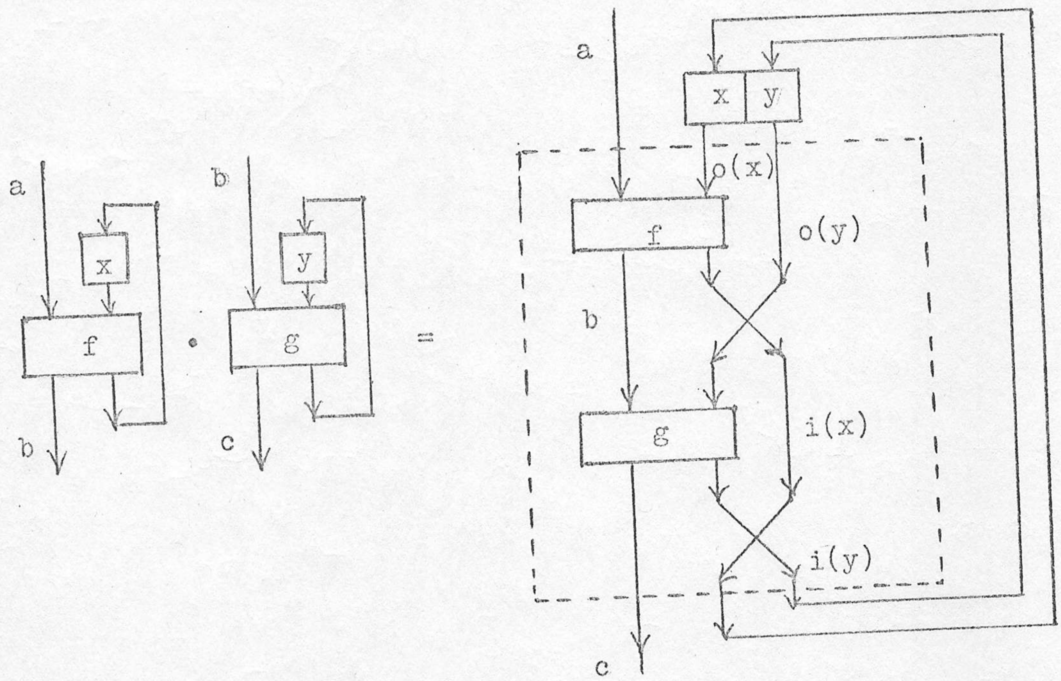
$$Fl_{X,B}(a,b) = \{(x,f) \mid x \in X,\ f \in B(ao(x),bi(x))\}.$$

An element $(x,f) \in Fl_{X,B}(a,b)$ is said to represent a flowchart scheme from a to b with statements in X and connections in B.

The basic operations on scheme representations together with their pictural motivations are:

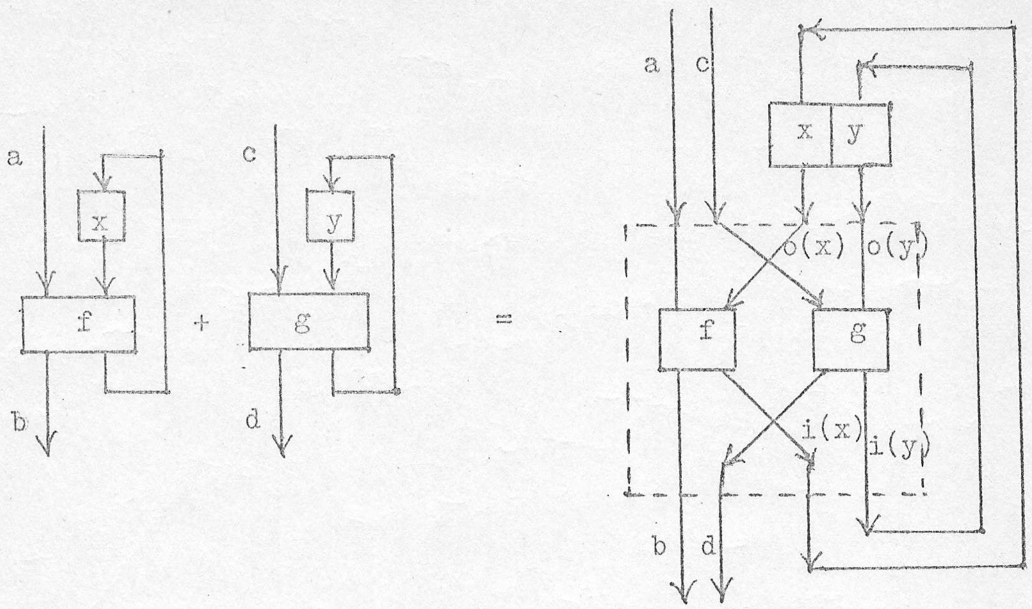
a) For $(x,f) \in Fl_{X,B}(a,b)$ and $(y,g) \in Fl_{X,B}(b,c)$ the composite is

$$(x,f)\cdot(y,g) = (xy,\ (f + 1_{o(y)})(1_b + i(x) \leftrightarrow o(y))(g + 1_{i(x)})(1_c + i(y) \leftrightarrow i(x))$$



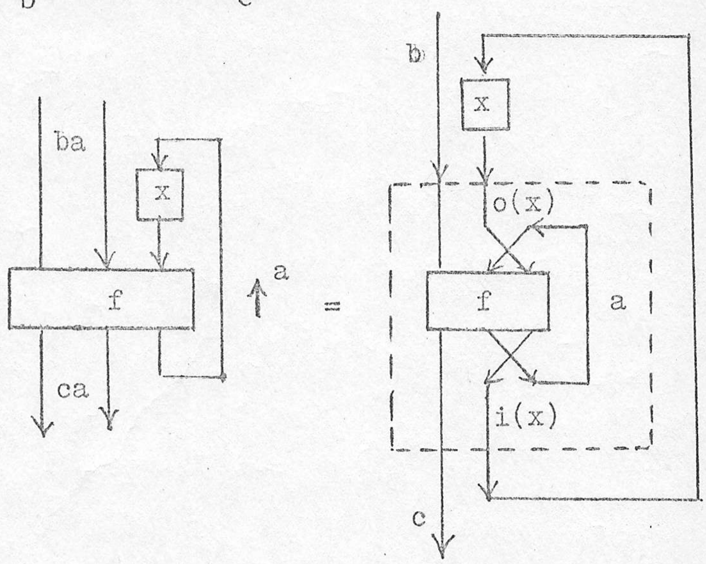
b) For $(x,f) \in Fl_{X,B}(a,b)$ and $(y,g) \in Fl_{X,B}(c,d)$ the sum is

$$(x,f)\cdot(y,g) = (xy,\ (1_a + c \leftrightarrow o(x) + 1_{o(y)})(f + g)(1_b + i(x) \leftrightarrow d + 1_{i(y)}))$$



c) For $(x,f) \in Fl_{X,B}(ba,ca)$ the <u>right feedback</u> is

$(x,f)\uparrow^a = (x, ((1_b + o(x)\leftrightarrow a)\, f\, (1_c + a\leftrightarrow i(x)))\uparrow^a)$



d) For every $f \in B(a,b)$ we define $E_B(f) \in Fl_{X,B}(a,b)$ by $E_B(f) = (\varepsilon, f)$. Hence in $Fl_{X,B}$ we have by definition $1_a = E_B(1_a)$ and $a\leftrightarrow b = E_B(a\leftrightarrow b)$.

Using these operations we prove that $Fl_{X,B}$ is an M-flow, preserves the M-flow structure of B and has a universal property (theorem 2.b.5 below) similar to the universal property of the polynomials ring R[X] over the ring R. We emphasize that the role played by (integer, rational, real, etc.) numbers in the case of polynomials is played in the case of representations by some classes of finite relations (bijections, injections, etc.).

1. FLOWS

In this section an algebraic structure (called flow) is introduced; its basic model is the algebra of flowchart scheme representations, presented in the introduction. In the acyclic case (i.e. the restriction of this algebraic structure to composition and sum) this algebra is weaker than a strict monoidal category which extends **Sur** (the theory of surjective functions) used in [9], in two respects: (i) an axiom used in a strict monoidal category [11] is weakened; (ii) the present algebra should extend only **Bi** (the theory of bijective functions). In the cyclic case, the axioms used for feedback are the translations of those used for iteration in an algebraic theory with iterate [4] - see the Appendix below.

Let M be a monoid and $\mathcal{C}$ a category having as objects all the elements of M. Suppose $\mathcal{C}$ in endowed with an operation +, called <u>sum</u>

$$+ : \mathcal{C}(a,b) \times \mathcal{C}(c,d) \longrightarrow (ac,bd)$$

defined for every a,b,c,d in M. Consider the following axiom used in a strict monoidal category which relates composition to sum

CS. $(x + f)(g + y) = xg + fy$ for $x \in \mathcal{C}(a,b)$, $g \in \mathcal{C}(b,c)$, $f \in \mathcal{C}(a',b')$ and $y \in \mathcal{C}(b',c')$.

Such an axiom is not valid in the algebra of scheme representations, mainly since in passing from left to right the linearly ordered vertices in f are permuted to those in g; however, the restriction of CS to the case when f or g has no internal vertices is valid. This comment may also be applied to the axiom B4 below.

A category $\mathcal{C}$ as above is said to be an M-wsmc (<u>weak strict monoidal category</u>) if + fulfils the following axioms:

W1. $(f + g) + h = f + (g + h)$

W2. $f + 1_\lambda = f = 1_\lambda + f$

W3. $1_a + 1_b = 1_{ab}$

W4. CS restricted to the case f <u>or</u> g is an identity.

A strict monoidal category (as defined in [11]) is equivalent to a weak strict monoidal category which fulfils CS.

In an M-wsmc only (a copy of the set of) identity functions can be obtained from the distinguished morphisms 1_a, but in order to define operations in $Fl_{X,\mathcal{C}}$ bijective functions must be embedded in $\mathcal{C}$. For this we add some distinguished morphisms $a \leftrightarrow b \in \mathcal{C}(ab,ba)$, called "block transpositions" and some equations:

B1. $a \leftrightarrow b \cdot b \leftrightarrow a = 1_{ab}$,

B2. $ab \leftrightarrow cd = (1_a + b \leftrightarrow c + 1_d)(a \leftrightarrow c + b \leftrightarrow d)(1_c + a \leftrightarrow d + 1_b)$,

B3. $a \leftrightarrow \lambda = 1_a = \lambda \leftrightarrow a$,

B4. $a \leftrightarrow b \cdot (f + g) = (g + f) \cdot c \leftrightarrow d$, for every $f \in \mathcal{C}(b,d)$ and $g \in \mathcal{C}(a,c)$.

An M-wsmc is said to be an <u>M-birel</u> (<u>bi</u>jective <u>rel</u>ation) [6] if the distinguished morphisms $a \leftrightarrow b$ fulfil B1-B3 and if CS and B4 hold when f <u>and</u> g are of the type $a \leftrightarrow b$. In [6] it was proved that bijective functions are embedded in an M-birel.

1.1. **DEFINITION.** A schematic M-birel is an M-birel B fulfilling for every $f \in B(a,b)$:

$c \Leftrightarrow d + f = (1_{cd} + f)(c \Leftrightarrow d + 1_b)$, a particular case of CS, and

$a \Leftrightarrow c \cdot (1_c + f) = (f + 1_c) \cdot b \Leftrightarrow c$, a particular case of B4. ∎

Proposition 1.3 below shows that a schematic M-birel is an M-birel in which CS and B4 hold with a weaker restriction: f or g is of type $a \Leftrightarrow b$.

Note that a schematic M-birel which fulfils CS is a strict monoidal category which extends **Bi**. Hence a schematic M-birel is more general than a strict monoidal category which extends **Sur**, as used in [9].

Let B be a schematic M-birel. A composite of morphisms of the type $1_a + b \Leftrightarrow c + 1_d$ is said to be a bimorfism. We define a relation C on morphisms of B (giving the pairs of morphisms which satisfy CS) as follows:

$$g \; C \; f \quad \text{iff} \quad (1_a + f)(g + 1_d) = g + f$$

where $f \in B(c,d)$ and $g \in B(a,b)$. Computing rules for C may be found in [6]. A similar relation may be introduced giving the pairs of morphisms which satisfy B4. Lemma 1.2 below shows that both ways give the same relation.

Note that for $f \in B(c,d)$ and $g \in B(a,b)$ we have

B5. $a \Leftrightarrow c \cdot (f + g) = (1_a + f)(g + 1_d) \cdot b \Leftrightarrow d$.

1.2. **LEMMA.** In a schematic M-birel B, for every $f \in B(c,d)$ and $g \in B(a,b)$ the following conditions are equivalent:

a) $g \; C \; f$,

b) $a \Leftrightarrow c \cdot (f + g) = (g + f) \cdot b \Leftrightarrow d$,

c) $c \Leftrightarrow a \cdot (g + f) = (f + g) \cdot d \Leftrightarrow b$ and

d) $f \; C \; g$.

Proof. Using B5 and B1 we deduce that a) is equivalent to b) and c) is equivalent to d). The equivalence of b) and c) follows from B1. ∎

1.3. **PROPOSITION.** In a schematic M-birel if $f' \in B(a,b)$ or $f \in B(c,d)$ is a bimorphism then $f' C f$. Hence, by Lemma 1.2 $a \Leftrightarrow c \cdot (f + f') = (f' + f) \cdot b \Leftrightarrow d$.

Proof. By Lemma 1.2 only one case, say f' is a bimorphism, is to be studied. The axiom CS restricted to the case $g = a \Leftrightarrow b$ shows that $a \Leftrightarrow b \; C \; 1_v + f$, therefore $1_u + a \Leftrightarrow b + 1_v \; C \; f$. Since $h \; C \; f$ and $h' \; C \; f$ imply $hh' C f$ and every bimorphism is a composite of morphisms of type $1_u + a \Leftrightarrow b + 1_v$ the result follows. ∎

1.4. DEFINITION. An M-flow B is a schematic M-birel in which a right feedback

$\_\uparrow^a : B(ba,ca) \to B(b,c)$ for $a,b,c \in M$

is given, fulfilling the following conditions.

- The feedback is context-free.

F1a. $(f\uparrow^a)g = (f(g+1_a))\uparrow^a$, for $f \in B(ba,ca)$ and $g \in B(c,d)$;

F1b. $g(f\uparrow^a) = ((g+1_a)f)\uparrow^a$, for $f \in B(ba,ca)$ and $g \in B(d,b)$;

F2a-weak. $1_d + f\uparrow^a = (1_d + f)\uparrow^a$, for $f \in B(ba,ca)$ and $d \in M$.

- The "vectorial" feedback can be expressed in terms of the "scalar" one.

F3a. $f\uparrow^{ab} = (f\uparrow^b)\uparrow^a$ for $f \in B(cab, dab)$.

- Particular blocks can be shifted on feedback.

F4weak. $(f(1_c + g))\uparrow^a = ((1_b + g)f)\uparrow^d$ for $f \in B(ba,cd)$ and $g \in B(d,a)$ restricted to blocks g of the type $u \leftrightarrow v$.

- The feedback acts on the distinguished morphisms as follows:

F5weak. $1_a\uparrow^a = 1_\lambda$;

F6. $a\leftrightarrow a\uparrow^a = 1_a$. ∎

The following proposition complets the formal motivation for the verbal conditions.

1.5. PROPOSITION (properties in an M-flow). The following hold in an M-flow B:

F2a. $g + f\uparrow^a = (g+f)\uparrow^a$, for $g \in B(e,d)$ and $f \in B(ba,ca)$;

F2b. $f\uparrow^a + g = ((1_b + d\leftrightarrow a)(f+g)(1_c + a\leftrightarrow e))\uparrow^a$, for $f \in B(ba,ca)$ and $g \in B(d,e)$;

F3b. $f\uparrow^\lambda = f$;

F4. if f C g then $(f(1_c+g))\uparrow^a = ((1_b+g)f)\uparrow^d$, for $f \in B(ba,cd)$ and $g \in B(d,a)$;

F5. $(f+1_a)\uparrow^a = f$.

Proof. F2a. $g + f\uparrow^a = (g+1_b)(1_d + f\uparrow^a) = (g+1_b)(1_d+f)\uparrow^a =$
$= ((g+1_{ba})(1_d+f))\uparrow^a = (g+f)\uparrow^a$.

F2b. $f\uparrow^a + g = (f\uparrow^a + 1_d)(1_c + g) = b\leftrightarrow d\,(1_d + f\uparrow^a)\,d\leftrightarrow c\,(1_c+g) =$
$= ((b\leftrightarrow d + 1_a)(1_d + f)(d\leftrightarrow c + 1_a)(1_c + g + 1_a))\uparrow^a =$
$= ((1_b + d\leftrightarrow a)\,ba\leftrightarrow d\,(1_d+f)\,d\leftrightarrow ca\,(1_c + a\leftrightarrow d\,(g+1_a)))\uparrow^a =$
$= ((1_b + d\leftrightarrow a)(f+1_d)(1_{ca}+g)(1_c + a\leftrightarrow e))\uparrow^a = ((1_b + d\leftrightarrow a)(f+g)(1_c + a\leftrightarrow e))\uparrow^a$

F3b. $f\uparrow^\lambda = (f+1_\lambda)\uparrow^\lambda = f + 1_\lambda\uparrow^\lambda = f + 1_\lambda = f$.

F4. $(f(1_c+g))\uparrow^a = (f(1_c + d\leftrightarrow d\uparrow^d)(1_c+g))\uparrow^a =$
$= ((f+1_d)(1_c + d\leftrightarrow d)(1_c + g + 1_d))\uparrow^d\uparrow^a =$
$= ((f+1_d)(1_{cd}+g)(1_c + d\leftrightarrow a))\uparrow^{ad} = ((1_b + d\leftrightarrow a)(1_{ba}+g)(f+1_a))\uparrow^{da} =$
$= ((1_b + g + 1_a)(1_b + a\leftrightarrow a)(f+1_a))\uparrow^a\uparrow^d = ((1_b+g)(1_b + a\leftrightarrow a\uparrow^a)f)\uparrow^d = ((1_b+g)f)\uparrow^d$.

F5. $(f + 1_a)\uparrow^a = f + 1_a\uparrow^a = f + 1_\lambda = f.$ ■

In an M-flow B we define the <u>left feedback</u>

$\uparrow^a\_ : B(ab,ac) \to B(b,c)$

for every $f \in B(ab,ac)$ by

F7. $\uparrow^a f = (b \leftrightarrow a \cdot f \cdot a \leftrightarrow c)\uparrow^a.$

1.6. PROPOSITION. (properties of the left feedback). In an M-flow B the following hold:

F1a'. $(\uparrow^a f)g = \uparrow^a(f(1_a + g))$, for $f \in B(ab,ac)$ and $g \in B(c,d)$;

F1b'. $g(\uparrow^a f) = \uparrow^a((1_a + g)f)$, for $f \in B(ab,ac)$ and $g \in B(d,b)$;

F2a'. $\uparrow^a f + g = \uparrow^a(f + g)$, for $f \in B(ab,ac)$ and $g \in B(d,e)$;

F2b'. $g + \uparrow^a f = \uparrow^a((a \leftrightarrow d + 1_b)(g + f)(e \leftrightarrow a + 1_c))$, for $f \in B(ab,ac)$, $g \in B(d,e)$;

F3a'. $\uparrow^{ab} f = \uparrow^b(\uparrow^a f)$, for $f \in B(abc,abd)$;

F3b'. $\uparrow^\lambda f = f$;

F4'. if $f \,C\, g$ then $\uparrow^a(f(g + 1_c)) = \uparrow^d((g + 1_b)f)$, for $f \in B(ab,dc)$, $g \in B(d,a)$;

F5'. $\uparrow^a(1_a + f) = f$;

F6'. $\uparrow^a a \leftrightarrow a = 1_a$;

F7'. $f\uparrow^a = \uparrow^a(a \leftrightarrow b \cdot f \cdot c \leftrightarrow a)$, for $f \in B(ba,ca)$;

F8. $(\uparrow^b f)\uparrow^a = \uparrow^b(f\uparrow^a)$, for $f \in B(bca,bda)$.

Proof. Easy, using the definition of the left feedback and the corresponding properties of the right feedback. We only prove F8:

$$(\uparrow^b f)\uparrow^a = (ca \leftrightarrow b \cdot f \cdot b \leftrightarrow da)\uparrow^b\uparrow^a = (ca \leftrightarrow b \cdot f \cdot b \leftrightarrow da)\uparrow^{ab} =$$
$$= ((1_c + b \leftrightarrow a)\, ca \leftrightarrow b \cdot f \cdot b \leftrightarrow da\, (1_d + a \leftrightarrow b))\uparrow^{ba} =$$
$$= ((c \leftrightarrow b + 1_a)f(b \leftrightarrow d + 1_a))\uparrow^a\uparrow^b = (c \leftrightarrow b\, (f\uparrow^a)\, b \leftrightarrow d)\uparrow^b = \uparrow^b(f\uparrow^a). \quad ■$$

The concept of M-flow may be introduced using a left feedback and taking F1a' - F6' in proposition 1.6 as axioms. Using F7' in proposition 1.6 as the definition of the corresponding right feedback and an easy computation one can prove that such a structure is an M-flow.

Also, the concept of M-flow may be introduced using both feedbacks. We mention the following equivalent axiomatic system {F1a, F1b, F2a-weak, F3a, F5-weak, F6, F7 and F8}.

1.7. DEFINITION. Let B and B' be two M-flows. A functor $H : B \to B'$ is said to be an <u>M-flow morphism</u> if

a) $H(a) = a$ for every $a \in M$,

b) $H(f + g) = H(f) + H(g)$ for every $f \in B(a,b)$ and $g \in B(c,d)$,

c) $H(a \leftrightarrow b) = a \leftrightarrow b$ for every $a,b \in M$ and

d) $H(f \Uparrow^a) = H(f) \Uparrow^a$ for every $f \in B(ba,ca)$.

When we work with the left feedback, the last condition is replaced by $H(\Uparrow^a f) = \Uparrow^a H(f)$ for every $f \in B(ab,ac)$. ■

In the sequel the category of M-flows will be denoted by Fl_M.

The concept of algebraic M-theory [6] is the extension of the concept of many-sorted algebraic theory to the case when the objects in the category form an arbitrary monoid (not a freely generated one as in the case of many sorted algebraic theories).

1.8. DEFINITION. An <u>algebraic M-theory</u> T is said to be <u>with iterate</u>, if an iteration

$$^{\dagger} : T(a,ab) \rightarrow T(a,b) \qquad \text{for } a,b \in M$$

is given fulfilling

I0. $(f(1_a + g))^{\dagger} = f^{\dagger}g$ for every $f \in T(a,ab)$ and $g \in T(b,c)$,

I1. $f \langle f^{\dagger}, 1_b \rangle = f^{\dagger}$ for every $f \in T(a,ab)$,

I2. $f^{\dagger\dagger} = (f(\langle 1_a, 1_a \rangle + 1_b))$ for every $f \in T(a,aab)$,

I3. $(gf)^{\dagger} = g(f(g + 1_c))^{\dagger}$ for every $g \in T(b,a)$ and $f \in T(a,bc)$. ■

For a free monoid this concept was introduced in [4] using another set of axioms (see [12] for the equivalence of that axiomatic system with this one - I4t used in [12] is a particular case of I3 above).

1.9. THEOREM. Every algebraic M-theory is a schematic M-birel. If T is an algebraic M-theory then T is with iterate if and only if T is an M-flow.

Hints for proof: Suppose T in an algebraic M-theory. We recall that + is the usual separated sum. For $a,b \in M$ we have by definition

$$a \leftrightarrow b = \langle 0_b + 1_a, 1_b + 0_a \rangle.$$

A routine computation shows that T is a schematic M-birel. The remainder of the proof may be found in the Appendix. We only mention the definitions

$\Uparrow^a f = (0_a + 1_b)(f(1_a + 0_b + 1_c))^{\dagger}$, for $f \in T(ab,ac)$ and

$f^{\dagger} = \Uparrow^a(aVa \cdot f)$, for $f \in T(a,ab)$. ■

As every ω-continuous S-sorted algebraic theory is an algebraic S^*-theory with iterate, theorem 1.9 gives a lot of examples of S^*-flows. We are only interested in two known examples of this type.

The first one is $\mathbf{Pfn}_A$ (the ADJ group uses the notation $\mathbf{Sum}_A$ [1]) where $A = \{A_s\}_{s \in S}$ is an S-sorted set. If $a \in S^*$ then $|a|$ denotes its length and for $i \in [|a|]$

the i^{th} letter of a is denoted by a_i . Let $A_a = \{(x,i) \mid i \in [\,|a|\,],\ x \in A_{a_i}\}$. For $a,b \in S^*$:

" the morphisms in $\mathbf{Pfn}_A(a,b)$ are those partial functions f from A_a to A_b which preserve sorts, i.e. $f(x,i) = (y,j)$ implies $a_i = b_j$. "

$\mathbf{Pfn}_A$ is an ω-continuous S-sorted algebraic theory, hence $\mathbf{Pfn}_A$ is an S^*-flow. This is the standard semantic model for the deterministic S-sorted flowchart schemes.

1.10. **PROPOSITION** (feedback in $\mathbf{Pfn}_A$). For $f \in \mathbf{Pfn}_A(ab,ac)$ and $(x,i) \in A_b$ we have

$(\uparrow^a f)(x,i) = (x',i')$ with $(x',i') \in A_c$

if and only if

there exist $n \geq 0$ and $(x_1,i_1), \ldots ,(x_n,i_n) \in A_a$ such that
$(x_k,i_k) = f(x_{k-1},i_{k-1})$ for every $k \in [n+1]$,
where $(x_o,i_o) = (x, |a| + i)$ and $(x_{n+1},i_{n+1}) = (x', |a| + i')$. ∎

The second example is $\mathbf{PStr}_S$. For every $a,b \in S^*$,

" $\mathbf{PStr}_S(a,b)$ is the set of all partial functions f from $[\,|a|\,]$ to $[\,|b|\,]$ such that $f(i) = j$ implies $a_i = b_j$. "

We may think of $\mathbf{PStr}_S$ as a particular case of $\mathbf{Pfn}_A$, i.e. A_s is a singleton for every $s \in S$. This viewpoint leads to the following corollary.

1.11. **COROLLARY** (feedback in $\mathbf{PStr}_S$). For $f \in \mathbf{PStr}_S(ab,ac)$ and $i \in [\,|b|\,]$ we have

$(\uparrow^a f)(i) = j$ where $j \in [\,|c|\,]$

if and only if

there exist $n \geq 0$ and $i_1,i_2, \ldots ,i_n \in [\,|a|\,]$ such that
$x_k = f(i_{k-1})$ for every $k \in [n+1]$,
where $i_o = |a| + i$ and $i_{n+1} = |a| + j$. ∎

1.12. **COROLLARY** (scalar feedback in $\mathbf{PStr}_S$). Let $f \in \mathbf{PStr}_S(sb,sc)$ where $s \in S$. For every $i \in [\,|b|\,]$ and $j \in [\,|c|\,]$

$$(\uparrow^s f)(i) = j \text{ if and only if } \begin{cases} f(1+i) = 1+j \text{ or} \\ f(1+i) = 1 \text{ and } f(1) = 1+j. \end{cases}$$ ∎

Let $\mathbf{In}_S$ be the subcategory of $\mathbf{PStr}_S$ which consists of all injective functions in $\mathbf{PStr}_S$.

1.13. **PROPOSITION.** If $f \in \mathbf{In}_S(ab,ac)$ then $\uparrow^a f \in \mathbf{In}_S(b,c)$. ∎

1.14. COROLLARY. In_S is an S^*-flow. ■

Let Bi_S be the subcategory of In_S which consists of all bijective functions in In_S.

1.15. COROLLARY. If $f \in Bi_S(ab,ac)$ then $\uparrow^a f \in Bi_S(b,c)$. ■

1.16. THEOREM. Bi_S is the initial S^*-flow.

Proof. By corollary 1.15 Bi_S is an S^*-flow.

Let B be an S^*-flow. In [6] we proved that there exists a unique functor $H : Bi_S \to B$ such that H satisfies the conditions a,b and c from definition 1.7.

We still have to prove that $H(\uparrow^s f) = \uparrow^s H(f)$ for every $f \in Bi_S(sa,sb)$ where $s \in S$.

If $f(1) = 1$ then there exists $g \in Bi_S(a,b)$ such that $f = 1_s + g$, therefore $\uparrow^s f = g$. In conclusion $\uparrow^s H(f) = \uparrow^s(1_s + H(g)) = H(g) = H(\uparrow^s f)$.

If $f(1) \neq 1$ we begin with some notation: $f(1) = 1 + j$, $f(1 + i) = 1$, $a = a'sa''$ where $|a'| = i - 1$ and $b = b'sb''$ where $|b'| = j - 1$. Since there exists $h \in Bi_S(a'a'',b'b'')$ such that

$$f = (1_s + a'\leftrightarrow s + 1_{a''})(s\leftrightarrow s + h)(1_s + s\leftrightarrow b' + 1_{b''})$$

we deduce that

$$\uparrow^s f = (a'\leftrightarrow s + 1_{a''})(1_s + h)(s\leftrightarrow b' + 1_{b''})$$

therefore

$$\uparrow^s H(f) = (a'\leftrightarrow s + 1_{a''})(\uparrow^s s\leftrightarrow s + H(h))(s\leftrightarrow b' + 1_{b''}) = H(\uparrow^s f).$$ ■

1.17. COROLLARY. Let B be an S^*-flow. If $f \in B(ab,ac)$ is a bimorphism then $\uparrow^a f$ is a bimorphism. ■

2. THE ALGEBRA OF FLOWCHART SCHEME REPRESENTATIONS

In this section we prove our main technical result. This result shows in particular that the axioms used for feedback in a flow (or equivalently, for iteration in a theory with iterate) capture precisely the facts needed in order that the natural interpretation of flowchart scheme representations preserve the operations. Note that from this theorem large parts of the proofs of the main theorems in [7, 12, 13] regarding the flow structure of flowchart schemes and the preserving of the operations follows directly.

2. a. SYNTAX. We have seen in the introduction that a flowchart scheme representation is an ordered pair. We split these pairs and we obtain two functors useful for our goal. The category of monoids is denoted by **Mon**; recall that Fl_M denotes the category of M-flows.

2. a. 1. The functor Q : Mon $\rightarrow Fl_M$

Let X be a monoid and let ε be its neutral element. The M-flow Q(X) is defined by

$Q(a,b) = \{(a,x,b) \mid x \in X\}$ for every $a,b \in M$,

composition: $(a,x,b)\,(b,y,c) = (a,xy,c)$,

identity: $1_a = (a, \varepsilon, a)$,

sum: $(a,x,b) + (c,y,d) = (ac,xy,bd)$,

block transposition: $a \leftrightarrow b = (ab, \varepsilon, ba)$,

left feedback: $\uparrow^a(ab,x,ac) = (b,x,c)$,

right feedback: $(ba,x,ca)\uparrow^a = (b,x,c)$.

If $h : X \rightarrow Y$ is a monoid morphism then the M-flow morphism $Q(h) : Q(X) \rightarrow Q(Y)$ is defined by $Q(h)(a,x,b) = (a,h(x),b)$.

2. a. 2. The functor K : $Fl_M \rightarrow Fl_M$

For a schematic M-birel B we define a schematic M-birel K(B). The morphism of K(B) are defined for $a,b \in M$ by

$K(B)(a,b) = \{(f,i,o) \mid i,o \in M,\ f \in B(ao,bi)\}$.

For $(f,i,o) \in K(B)(a,b)$ and $(f',i',o') \in K(B)(b,c)$ we define the composition by

$$(f,i,o)\,(f',i',o') = ((f + 1_{o'})(1_b + i \leftrightarrow o')(f' + 1_i)(1_c + i' \leftrightarrow i),\ ii', oo').$$

A routine computation shows that composition is associative.

For $f \in B(a,b)$ we define $I_B(f) \in K(B)(a,b)$ by

$$I_B(f) = (f, \lambda, \lambda).$$

We mention the following computation rules:

$(f,i,o)I_B(f') = (f(f' + 1_i),i,o)$

$I_B(f)(f',i',o') = ((f + 1_{o'})f',i',o')$

It follows that $1_a = I_B(1_a)$ is an identity morphism, hence K(B) is a category.

For $(f,i,o) \in K(B)(a,b)$ and $(f',i',o') \in K(B)(c,d)$ we define the sum by

$$(f,i,o) + (f',i',o') = ((1_a + c \leftrightarrow o + 1_{o'})(f + f')(1_b + i \leftrightarrow d + 1_{i'}),ii',oo').$$

We mention the particular case

$I_B(f) + (f',i',o') = (f + f',i',o')$.

A routine computation shows that K(B) is an M-wsmc.

For every $a,b \in M$ we define

$$a \Leftrightarrow b = I_B(a \leftrightarrow b).$$

It is easy to show that K(B) is a schematic M-birel.

Let B be an M-flow. We define in K(B) the feedbacks by

$$\uparrow^a(f,i,o) = (\uparrow^a f,i,o) \quad \text{for } (f,i,o) \in K(B)(ab,ac)$$

$$(f,i,o)\uparrow^a = (((1_b + o \leftrightarrow a)f(1_c + a \leftrightarrow i))\uparrow^a,i,o) \quad \text{for } (f,i,o) \in K(B)(ba,ca).$$

A routine computation using the left feedback shows that K(B) is an M-flow and $I_B : B \to K(B)$ is an M-flow morphism.

If $H : B \to B'$ is an M-flow morphism we define the M-flow morphism $K(H) : K(B) \to K(B')$ for (f,i,o) in K(B) by

$K(H)(f,i,o) = (H(f),i,o)$.

It is easy to show that $K : Fl_M \to Fl_M$ is a functor and that $I : 1_{Fl_M} \to K$ is a natural transformation.

2. a. 3. The M-flow of flowchart scheme representations

Let B be an M-flow and X a monoid. Suppose we are given two monoid morphisms $i : X \to M$ and $o : X \to M$.

The cartesian product $Q(X) \times K(B)$ is an M-flow. The operations are performed componentwise. For $a,b \in M$ we define

$P(a,b) = \{((a,x,b), (f,i(x),o(x)) \mid x \in X, (f,i(x),o(x)) \in K(B)(a,b)\}$.

Since P is a subset of $Q(X) \times K(B)$ which is closed under composition, sum, feedbacks and contains 1_a and $a \Leftrightarrow b$ for every $a,b \in M$, it is an M-flow.

The function that maps $((a,x,b), (f,i(x),o(x))) \in P(a,b)$ into $(x,f) \in Fl_{X,B}(a,b)$ is an isomorphism with respect to composition, sum, feedback and constants 1_a and $a \Leftrightarrow b$, therefore $Fl_{X,B}$ is an M-flow.

We recall that $E_B(f) = (\varepsilon,f) \in Fl_{X,B}(a,b)$ for $f \in B(a,b)$. We mention the following computation rules

$(x,f)E_B(g) = (x,f(g + 1_{i(x)}))$ for $(x,f) \in Fl_{X,B}(a,b)$ and $g \in B(b,c)$,

$E_B(f)(x,g) = (x,(f + 1_{o(x)})g)$ for $f \in B(a,b)$ and $(x,g) \in Fl_{X,B}(b,c)$ and

$E_B(f) + (x,g) = (x,f + g)$ for $f \in B(a,b)$ and $(x,g) \in Fl_{X,B}(c,d)$.

We deduce that $E_B : B \to Fl_{X,B}$ is an M-flow morphism. We collect these facts as the following theorem.

2. a. 4. **THEOREM.** If B is an M-flow, then $Fl_{X,B}$ is an M-flow which preserves the flow structure of B. ■

2. b. **SEMANTICS.** Note that for an M-flow B the structure $(B,+,1_\lambda)$ is a monoid.

2. b. 1. **DEFINITION.** Let B be an M-flow. A monoid morhism $I : X \to B$ (B having the above monoid structure) such that $I(x) \in B(i(x),o(x))$ for every $x \in X$ is said to be an interpretation of X in B. ■

Suppose the monoid M is freely generated by S. Consider an S-sorted set A giving the sorted set of memory states in the underlying computing device and an interpretation $I : X \to \mathbf{Pfn}_A$.

Let $H : \mathbf{PStr}_S \to \mathbf{Pfn}_A$ be the S^*-flow morphism defined for $f \in \mathbf{PStr}_S(a,b)$ by

$$H(f)(x,i) = (x,f(i))$$

where $i \in [\,|a|\,]$ and $x \in A_{a_i}$.

Let F be a flowchart scheme represented by $(x,f) \in Fl_{X,\mathbf{PStr}_S}(a,b)$ (or by the formal expression $((1_a + x_1 + \ldots + x_{|x|})\, f) \uparrow^{i(x)}$). The program (F,I) obtained by interpreting via I the statements of F computes

$$(1_a + I(x))H(f)) \uparrow^{i(x)}$$

which is equal to the C.C. Elgot semanties [8]

$$H(j) < (I(x)H(t))^{\dagger}, 1_b >$$

where $j = (1_a + 0_{o(x)}) \cdot f \cdot b \leftrightarrow i(x)$ and $t = (0_a + 1_{o(x)}) \cdot f \cdot b \leftrightarrow i(x)$.

2. b. 2. **The standard interpretation.** For every $x \in X$ we define the interpretation $E_X(x) \in Fl_{X,B}(i(x),o(x))$ by

$$E_X(x) = (x, i(x) \leftrightarrow o(x)).$$

It is easy to show that E_X is an interpretation in the sense of 2.b.1.

A routine computation proves the following two propositions.

2. b. 3. **PROPOSITION.** $E_B(f) \subset E_X(x)$ for every $f \in B(a,b)$ and $x \in X$. ■

2. b. 4. **PROPOSITION.** If $(x,f) \in Fl_{X,B}(a,b)$ then

$$(x,f) = ((1_a + E_X(x))E_B(f)) \uparrow^{i(x)}.$$ ■

Now our main technical result is the following

2. b. 5. THEOREM. If $H : B \to B'$ is an M-flow morphism and $I : X \to B'$ is an interpretation such that

$$H(f) \subset I(x)$$

for every morphism f in B and every x in X, then there is a unique M-flow morphism

$$(I,H)^f : Fl_{X,B} \to B'$$

such that $E_X(I,H)^f = I$ and $E_B(I,H)^f = H$.

Proof. If $(I,H)^f$ exists and has the above properties, we deduce from proposition 2.b.4 that

$$(*) \qquad (I,H)^f(x,f) = ((1_a + I(x))\, H(f)) \uparrow^{i(x)}$$

for every $(x,f) \in Fl_{X,B}(a,b)$. This proves the uniqueness of $(I,H)^f$. Therefore we define $(I,H)^f(x,f)$ by $(*)$.

If $(x,f) \in Fl_{X,B}(a,b)$ and $(y,g) \in Fl_{X,B}(b,c)$ then

$(I,H)^f((x,f)(y,g)) =$

$$= ((1_a + I(xy))\ \ H((f + 1_{o(y)})\, (1_b + i(x) \leftrightarrow o(y))\, (g + 1_{i(x)})\, (1_c + i(y) \leftrightarrow i(x))))\uparrow^{i(xy)} =$$
$$= (((1_a + I(x))H(f) + I(y))\, (1_b + i(x) \leftrightarrow o(y))\, (H(g) + 1_{i(x)})\, (1_c + i(y) \leftrightarrow i(x)))\uparrow^{i(x)i(y)} =$$
$$= (((1_a + I(x))H(f) + 1_{i(y)})\, (1_b + i(x) \leftrightarrow i(y))\, ((1_b + I(y))H(g) + 1_{i(x)})\, (1_c + i(y) \leftrightarrow i(x)))\uparrow^{i(y)}\uparrow^{i(x)} =$$
$$= ((1_a + I(x))H(f)\ \ (((1_b + i(x) \leftrightarrow i(y))\, ((1_b + I(y))H(g) + 1_{i(x)})\, (1_c + i(y) \leftrightarrow i(x)))\uparrow^{i(y)}))\uparrow^{i(x)} =$$
$$= ((1_a + I(x))H(f)\ \ (((1_b + I(y))H(g))\uparrow^{i(y)} + 1_{i(x)}))\uparrow^{i(x)} =$$
$$= ((1_a + I(x))H(f))\uparrow^{i(x)} \cdot ((1_b + I(y))H(g))\uparrow^{i(y)} = (I,H)^f(x,f) \cdot (I,H)^f(y,g).$$

If $(x,f) \in Fl_{X,B}(a,b)$ and $(y,g) \in Fl_{X,B}(c,d)$ then

$(I,H)^f((x,f) + (y,g)) =$

$$= ((1_{ac} + I(xy))\ \ H((1_a + c \leftrightarrow o(x) + 1_{o(y)})\, (f + g)\, (1_b + i(x) \leftrightarrow d + 1_{i(y)})))\uparrow^{i(xy)} =$$
$$= ((1_a + c \leftrightarrow i(x) + 1_{i(y)})\, ((1_a + I(x))H(f) + (1_c + I(y))H(g))\, (1_b + i(x) \leftrightarrow d + 1_{i(y)}))\uparrow^{i(y)}\uparrow^{i(x)} =$$
$$= ((1_a + c \leftrightarrow i(x))\ \ ((1_a + I(x))H(f) + ((1_c + I(y))H(g))\uparrow^{i(y)})\ \ (1_b + i(x) \leftrightarrow d))\uparrow^{i(x)} =$$
$$= ((1_a + I(x))H(f))\uparrow^{i(x)} + ((1_c + I(y))H(g))\uparrow^{i(y)} = (I,H)^f(x,f) + (I,H)^f(y,g).$$

If $(x,f) \in Fl_{X,B}(ab,ac)$ then

$$(I,H)^f(\uparrow^a(x,f)) = ((1_b + I(x))H(\uparrow^a f))\uparrow^{i(x)} = \uparrow^a((1_{ab} + I(x))H(f))\uparrow^{i(x)} = \uparrow^a(I,H)^f(x,f).$$

If $f \in B(a,b)$ then

$$(I,H)^f(E_B(f)) = ((1_a + I(\varepsilon))H(f))\uparrow^{i(\varepsilon)} = H(f)\ ,$$

therefore $(I,H)^f(1_a) = 1_a$ and $(I,H)^f(a\leftrightarrow b) = a\leftrightarrow b$.

If $x \in X$ then

$$(I,H)^f(E_X(x)) = ((1_{i(x)} + I(x))\ i(x)\leftrightarrow o(x))\uparrow^{i(x)} = (i(x)\leftrightarrow i(x)(I(x) + 1_{i(x)}))\uparrow^{i(x)} = I(x). \quad \blacksquare$$

2.c. THE FUNCTOR Fl. The pair of monoid morphisms $i : X \to M$ and $o : X \to M$ may be replaced by only one monoid morphism $h : X \to M \times M$ defined by $h(x) = (i(x),o(x))$ for every $x \in X$.

The usual definition of the category **Mon**/$M\times M$ is:

a) the objects are all the pairs (X,h) where X is a monoid and $h : X \to M\times M$ is a monoid morphism;

b) $m : (X,h) \to (Y,g)$ is a morphism in **Mon**/$M\times M$ if and only if $m : X \to Y$ is a monoid morphism such that $mg = h$;

c) the composition is performed as in **Mon**.

The functor

$$Fl : (\mathbf{Mon}/M\times M) \times Fl_M \to Fl_M$$

is defined for every morphism $m : (X,h) \to (Y,g)$ in **Mon**/$M\times M$ and for every M-flow morpism $H : B \to B'$ by

$$Fl_{m,H} = (mE_Y, HE_{B'})^f\ .$$

Notice that if $(x,f) \in Fl_{X,B}(a,b)$ then, using Proposition 2.b.4, we deduce that

$$Fl_{m,H}(x,f) = (m(x),H(f)).$$

APPENDIX. SOME AXIOMATIC QUESTIONS

In this appendix we prove that, via a natural bijection between feedbacks and iterations, the axioms used for feedback in a flow are the translations of those used for iteration in an algebraic theory with iterate. This proves in particular Theorem 1.9.

Let T be an algebraic M-theory considered as a schematic M-birel as in the comments following theorem 1.9. We recall the notation: 0_a is the unique morpism in $T(\lambda,a)$ and $a\nabla a = \langle 1_a,1_a\rangle$ for $a \in M$ (computing rules for ava may be found in [6]).

We now define to applications **it**: **Fd**$\to$**It** and **fd** : **It**$\to$**Fd**, where **Fd** is the set of left feedbacks defined in T and **It** is the set of iterations defined in T. The iteration † = **it**($\uparrow$) is given by

$$f^\dagger = \uparrow^a(a\nabla a \cdot f),\ \text{for}\ f \in T(a,ab);$$

the left feedback $\uparrow$ = **fd**(†) is given by

$\uparrow^a f = (0_a + 1_b)(f \cdot (1_a + 0_b + 1_c))^\dagger$, for $f \in T(ab,ac)$.

Before the promised translation we repeat the axioms. The axioms required for iteration in a theory with iterate are:

I0. $(f(1_a + g))^\dagger = f^\dagger g$, for $f \in T(a,ab)$ and $g \in T(b,c)$;

I1. $f < f^\dagger, 1_b > = f^\dagger$, for $f \quad T(a,ab)$;

I2. $(f(<1_a, 1_a> + 1_b))^\dagger = f^{\dagger\dagger}$, for $f \in T(a,aab)$;

I3. $g(f(g + 1_c))^\dagger = (gf)^\dagger$, for $f \in T(a,bc)$ and $g \in T(b,a)$.

The axioms required for feedback in a flow (over an algebraic theory) are:

A0. $\uparrow^a(f(1_a + g)) = (\uparrow^a f) \cdot g$, for $f \in B(ab,ac)$ and $g \in B(c,d)$,

A1a. $\uparrow^a(f + 1_d) = \uparrow^a f + 1_d$, for $f \in B(ab,ac)$,

A1b. $\uparrow^a a \leftrightarrow a = 1_a$,

A2. $\uparrow^{ab} f = \uparrow^b(\uparrow^a f)$, for $f \in B(abc,abd)$,

A3a. $\uparrow^a((1_a + g)f) = g \cdot \uparrow^a f$, for $f \in B(ab,ac)$ and $g \in B(d,b)$,

A3b. $\uparrow^a((g + 1_c)f) = \uparrow^b(f(g + 1_d))$, for $f \in B(bc,ad)$ and $g \in B(a,b)$.

Note the change of notation (A0 = F1a', A1a = F2a-weak', A1b = F6', A2 = F3a', A3a = F1b' and A3b = F4' in Proposition 1.6), the absence of F5 which is superfluous if T is an algebraic theory ($\uparrow^a 1_a = 1_\lambda$, the unique morphism of $T(\lambda, \lambda)$) and the use of F4' (we have f C g for every f and g in an algebraic theory).

THEOREM (equivalence of axiomatic systems for iteration in theories with iterate and feedback in flows). Let T be an algebraic theory. The above correspondence **(fd,it)** satisfies:

Part 1. If $\uparrow$ fulfils A3a and A3b then **fd(it(**$\uparrow$**))** $= \uparrow$ and if $\dagger$ fulfils I3 then **it(fd(**$\dagger$**))**$= \dagger$.

Part 2. a) I3 $\Leftrightarrow$ A3a + A3b;

b) I3 + I0 $\Leftrightarrow$ A3a + A3b + A0;

c) I3 + I0 + I2 $\Leftrightarrow$ A3a + A3b + A0 + A2;

d) I3 + I0 + I2 + I1 $\Leftrightarrow$ A3a + A3b + A0 + A2 + A1a + A1b,

where X$\Leftrightarrow$Y means: "if $\dagger$ fulfils X then **fd**($\dagger$) fulfils Y and if $\uparrow$ fulfils Y then **it**($\uparrow$) fulfils X".

COMMENT. This theorem reinforce our belief that <u>feedback is more natural than iteration</u>. More precisely, the passing from scalar to vectorial iteration is expressed in terms of iteration by the "pairing" axiom [7,10],

$$\mathbf{P}.\ \langle f,g\rangle^{\dagger} = \langle f^{\dagger}\langle (g\langle f^{\dagger},1_{bc}\rangle)^{\dagger},\ 1_{c}\rangle,(g\langle f^{\dagger},1_{bc}\rangle)^{\dagger}\rangle \text{ for } f\in T(a,abc), \text{ and } g\in T(b,abc),$$

simplified in [12] to I2 and I3. The corresponding axiom written in terms of feedback is

$$\uparrow^{ab} f = \uparrow^{b}(\uparrow^{a} f) \quad \text{for } f\in T(abc,abd)$$

which clearly is much more readable and usable then the above one. In addition, the axioms corresponding to the Elgot fixpoint equation

$$f^{\dagger} = f.\langle f^{\dagger},1_{b}\rangle$$

are

$$\uparrow^{a}(f + 1_{d}) = \uparrow^{a} f + 1_{d} \quad \text{and} \quad \uparrow^{a} a{\leftrightarrow}a = 1_{a}$$

also much easier to verify and use. This seems to be the first significant simplification of the Elgot fixpoint equation (in Ésik's axiomatization of iteration theories [10] this equation appears in an involved way in the pairing axiom, while in Arbib and Manes [2] it is unnecessary since the iteration is computed by an infinite sum). ∎

Proof of Theorem. Part 1. Using axioms I3, A3a and A3b we prove that the correspondence between left feedbacks and iterations is bijective.

Suppose the iteration fulfils I3. The new iteration $\uparrow^{a}(aVaf)$ of $f\in T(a,ab)$ is equal to the old one $f^{\dagger}$. Indeed

$$\uparrow^{a}(aVaf) = (0_{a} + 1_{a})\,(aVa \cdot f(1_{a} + 0_{a} + 1_{b}))^{\dagger} =$$

$$= (0_{a} + 1_{a})\ aVa\ (f(1_{a} + 0_{a} + 1_{b})(aVa + 1_{b}))^{\dagger} = f^{\dagger}.$$

Suppose the feedback fulfils axioms A3a and A3b. The new feedback $(0_{a} + 1_{b})(f(1_{a} + 0_{b} + 1_{c}))^{\dagger}$ of $f\in T(ab,ac)$ is equal to the old one $\uparrow^{a} f$. Indeed

$$(0_{a} + 1_{b})(f(1_{a} + 0_{b} + 1_{c}))^{\dagger} = (0_{a} + 1_{b})(\uparrow^{ab}(abVab \cdot f(1_{a} + 0_{b} + 1_{c}))) =$$

$$= \uparrow^{ab}((1_{ab} + 0_{a} + 1_{b}) \cdot abVab \cdot f(1_{a} + 0_{b} + 1_{c})) =$$

$$= \uparrow^{a}((1_{a} + 0_{b} + 1_{b})(1_{ab} + 0_{a} + 1_{b})\ abVab \cdot f) = \uparrow^{a} f.$$

Part 2. a) Suppose the iteration fulfils I3. The proof of A3a is

$$\uparrow^{a}((1_{a} + g)f) = (0_{a} + 1_{d})\,((1_{a} + g)f(1_{a} + 0_{d} + 1_{c}))^{\dagger} =$$

$$= (0_{a} + 1_{d})(1_{a} + g)\,(f(1_{a} + 0_{d} + 1_{c})(1_{a} + g + 1_{c}))^{\dagger} =$$

$$= g(0_{a} + 1_{b})\,(f(1_{a} + 0_{b} + 1_{c}))^{\dagger} = g \cdot \uparrow^{a} f.$$

The proof of A3b is

$$\uparrow^{a}((g + 1_{c})f) = (0_{a} + 1_{c})\,((g + 1_{c})f(1_{a} + 0_{c} + 1_{d}))^{\dagger} =$$

$$= (0_{a} + 1_{c})(g + 1_{c})\,(f(1_{a} + 0_{c} + 1_{d})(g + 1_{c} + 1_{d}))^{\dagger} =$$

$= (0_b + 1_c)\,(f(g + 1_d)(1_b + 0_c + 1_d))^\dagger = \uparrow^a(f(g + 1_d))$.

Suppose the feedback fulfils A3a and A3b. The proof of I3 is

$$(gf)^\dagger = \uparrow^b(bVb \cdot gf) = \uparrow^b((g + g) \cdot aVa \cdot f) =$$

$$= g \cdot \uparrow^b((g + 1_a) \cdot aVa \cdot f) = g \cdot \uparrow^a(aVa \cdot f(g + 1_c)) = g(f(g + 1_c))^\dagger.$$

b) If I0 holds, we prove A0:

$$\uparrow^a(f(1_a + g)) = (0_a + 1_b)(f(1_a + 0_b + g))^\dagger =$$

$$= (0_a + 1_b)(f(1_a + 0_b + 1_c))^\dagger g = (\uparrow^a f)g.$$

If A0 holds we prove I0:

$$(f(1_a + g))^\dagger = \uparrow^a(aVa \cdot f(1_a + g)) = \uparrow^a(aVa \cdot f)g = f^\dagger g.$$

Hence we have the equivalence I3 + I0 $\Leftrightarrow$ A3a + A3b + A0.

c) We prove that the adding of I2 is equivalent to the adding of A2.

Using in turn I3, I0 and I2 we prove A2:

$$\uparrow^b \uparrow^a f = (0_b + 1_c)\;((0_a + 1_{bc})[f(1_a + 0_{bc} + 1_{bd})]^\dagger(1_b + 0_c + 1_d))^\dagger =$$

$$= (0_{ab} + 1_c)\,((f(1_a + 0_{bc} + 1_{bd}))^\dagger\,(0_a + 1_b + 0_c + 1_d))^\dagger =$$

$$= (0_{ab} + 1_c)\,(f(1_a + 0_{bca} + 1_b + 0_c + 1_d))^{\dagger\dagger} =$$

$$= (0_{ab} + 1_c)\;(f(1_a + 0_{bca} + 1_b + 0_c + 1_d)(<1_{abc}, 1_{abc}> + 1_d))^\dagger =$$

$$= (0_{ab} + 1_c)\,(f(1_{ab} + 0_c + 1_d))^\dagger = \uparrow^{ab} f\,.$$

Using in turn A3b, A2 and A3a we prove I2:

$$(f(<1_a, 1_a> + 1_b))^\dagger = \uparrow^a(aVa \cdot f \cdot (aVa + 1_b)) =$$

$$= \uparrow^{aa}((aVa + 1_a) \cdot aVa \cdot f) = \uparrow^a \uparrow^a((1_a + aVa) \cdot aVa \cdot f) =$$

$$= \uparrow^a(aVa \cdot \uparrow^a(aVa \cdot f)) = f^{\dagger\dagger}.$$

Hence we have the equivalence I3 + I0 + I2 $\Leftrightarrow$ A3a + A3b+ A0 + A2.

d) We prove that the adding of I1 is equivalent to the adding of A1a and A1b.

Using in turn I1, I3, again I1 and I0 we prove A1a:

$$\uparrow^a(f + 1_d) = (0_a + 1_{bd})((f + 1_d)(1_a + 0_{bd} + 1_{cd}))^\dagger =$$

$$= (0_a + 1_{bd})(f + 1_d)(1_a + 0_{bd} + 1_{cd}) < ((f + 1_d)(1_a + 0_{bd} + 1_{cd}))^\dagger, 1_{cd}> =$$

$$= (0_a + 1_{bd})(f(1_a + 0_b + 1_c) + 1_d) < (1_{ab} + 0_d)((f + 1_d)(1_a + 0_{bd} + 1_{cd}))^\dagger, 1_{cd}> =$$

$$= (0_a + 1_{bd}) <f(1_a + 0_b + 1_c) + 0_d, 0_{abc} + 1_d> <((1_{ab} + 0_d)(f + 1_d)(1_a + 0_b + 1_{cd}))^\dagger, 1_{cd}> =$$

$$= (0_a + 1_{bd}) <(f(1_a + 0_b + 1_c) + 0_d)<(f(1_a + 0_b + 1_c) + 0_d)^\dagger, 1_{cd}>,\; 0_c + 1_d> =$$

$= (0_a + 1_{bd}) <(f(1_a + 0_b + 1_c) + 0_d)^\dagger, 0_c + 1_d> =$

$= (0_a + 1_{bd}) <(f(1_a + 0_b + 1_c))^\dagger + 0_d, 0_c + 1_d> =$

$= (0_a + 1_b)(f(1_a + 0_b + 1_c))^\dagger + 1_d = \uparrow^a f + 1_d .$

Using in turn I1, I3 and aain I1 we prove A1b:

$\uparrow^a a\leftrightarrow a = (0_a + 1_a)(a\leftrightarrow a\,(1_a + 0_a + 1_a))^\dagger =$

$= (0_a + 1_a)\, a\leftrightarrow a\,(1_a + 0_a + 1_a) <(a\leftrightarrow a\,(1_a + 0_a + 1_a))^\dagger, 1_a> =$

$= (1_a + 0_a) <(1_a + 0_a)(a\leftrightarrow a\,(1_a + 0_a + 1_a))^\dagger, 1_a> =$

$= ((1_a + 0_a)a\leftrightarrow a)^\dagger = (0_a + 1_a)^\dagger = (0_a + 1_a) <(0_a + 1_a)^\dagger, 1_a> = 1_a .$

For the converse, using in turn A3b, A2, A0, A1a, again A0 and A1b we prove

A4. $\uparrow^a(aVa \cdot f) = \uparrow^a((f + 1_a)(1_a + b\leftrightarrow a)(aVa + 1_b))$, for $f \in T(a,ab)$.

Indeed,

$\uparrow^a((f + 1_a)(1_a + b\leftrightarrow a)(aVa + 1_b)) = \uparrow^{aa}((aVa + 1_a)(f + 1_a)(1_a + b\leftrightarrow a)) =$

$= \uparrow^a \uparrow^a((aVa \cdot f + 1_a)(1_a + b\leftrightarrow a)) = \uparrow^a((\uparrow^a(aVa \cdot f) + 1_a)\, b\leftrightarrow a) =$

$= \uparrow^a(a\leftrightarrow a\,(1_a + \uparrow^a(aVa \cdot f))) = (\uparrow^a a\leftrightarrow a) \cdot \uparrow^a(aVa \cdot f) = \uparrow^a(aVa \cdot f).$

Finally, using in turn A4, A1a, A3a and A0 we prove I1:

$f < f^\dagger, 1_b> = f\,(\uparrow^a(aVa \cdot f) + 1_b)\, bVb =$

$= f\,(\uparrow^a((f + 1_a)(1_a + b\leftrightarrow a)(aVa + 1_b)) + 1_b)\, bVb =$

$= \uparrow^a((1_a + f)(f + 1_{ab})(1_a + b\leftrightarrow a + 1_b)(aVa + 1_{bb})(1_a + bVb)) =$

$= \uparrow^a((f + f)\; abVab) = \uparrow^a(aVa \cdot f) = f^\dagger.$

REFERENCES

1. ADJ (J.W. Thatcher, E.G. Wagner and J.B. Wright), Notes on algebraic fundamentals for theoretical computer science, in "Foundations of computer science III, Part 2: Language, logic, semantics" (J.W. de Bakker and J. van Leeuwen, Eds.), pp.83-164, Mathematical Centre Tracts 109, Amsterdam, 1979.
2. M.A. Arbib and E.G. Manes, Partially additive categories and flowdiagram semantics, J. Algebra 62(1980), 203-227.
3. S.L. Bloom and Z. Ésik, Axiomatizing schemes and their behaviours, J. Comput. System Sci. 31 (1985), 375-393.
4. V.E. Căzănescu, On context-free trees, Theor. Comput. Sci. 41 (1985), 33-50.

6. V.E. Căzănescu and Gh. Ştefănescu, Finite relations as initial abstract data types (to appear).

5. V.E. Căzănescu and Gh. Ştefănescu, Towards a new algebraic foundation of flowchart scheme theory (to appear).

7. V.E. Căzănescu and C. Ungureanu, "Again on Advice on structuring compilers and proving them correct", INCREST Preprint Series in Mathematics, No. 75/1982.

8. C.C. Elgot, Monadic computation and iterative algebraic theories, in "Logic Colloquium" (H.E. Rose and J.C. Shepherdson, Eds.), pp. 175-230, North-Holland, Amsterdam, 1975.

9. C.C. Elgot and J.C. Shepherdson, An equational axiomatization of the algebra of reducible flowchart schemes, IBM Research Report, RC-8221 (April, 1980).

10. Z. Ésik, Algebras of iteration theories, J. Comput. System Sci. 27 (1983), 291-303.

11. S. MacLane, Categories for the Working Mathematician, Springer-Verlag, 1971.

12. Gh. Ştefănescu, On flowchart theories I, INCREST Preprint Series in Mathematics, No. 39/1984 and 7/1985; revised version to appear in J. Comput. System Sci., December 1987.

13. Gh. Ştefănescu, On flowchart theories II (nondeterminisitic case)", INCREST Preprint Series in Mathematics, No. 32/1985; revised version to appear in Theor. Comput. Sci., 1987.

14. Gh. Ştefănescu, An algebraic theory of flowchart schemes (extended abstract), Proceedings CAAP'86, LNCS 214, Springer-Verlag 1986, pp. 60-73.

15. Gh. Ştefănescu, Feedback theories (a calculus for isomorphism classes of flowchart schemes), INCREST Preprint Series in Mathematics, No. 24/1986.