A FORMAL REPRESENTATION OF FLOWCHART SCHEMES II

by

V.-E.CAZANESCU[*] and Gh. STEFANESCU[**]

November 1988

[*] University of Bucharest, Faculty of Mathematics, Str.Academiei 14 70109 Bucharest Romania.

[**] Department of Mathematics, The National Institute for Scientific and Technical Creation, Bd.Păcii 220, 79622 Bucharest, Romania.

A Formal Representation of Flowchart Schemes II

Virgil-Emil Căzănescu and Gheorghe Ştefănescu

Abstract. We represent a flowchart scheme by a pair (x,f), where x and f give the statements and the arrows in the scheme, respectively. The representation is not unique and a flowchart scheme appears as a class of isomorphic pairs. This way we get a syntactic model for flowchart schemes.

The algebra of pairs was developed in the first part of this work, which appeared in An. Univ. Bucuresti, Mat.-Inf. XXXVII, 2(1988), 33-51. Here we develop the algebra of flowchart schemes (= isomorphic pairs). We introduce an algebraic structure, called biflow, which completely characterizes flowchart schemes from the algebraic point of view. Another feature of this paper is the use of abstract flowchart schemes, that is (statement abstraction:) the concrete statements are replaced by variables and (arrow abstraction:) the set of arrows which connect the statements is replaced by an element from an adequate algebraic structure.

A. ALGEBRA

> Motto: "It is not too much to hope that a class of these identities may be isolated as axioms of an algebraic ... theory analogous (say) to rings or vector spaces.
>
> Robin Milner

It is an increasing need to find some basic algebraic structures for theoretical computer science. The motto is taken from Milner [Mi83] and refers to the algebrization of a recent and important field of computer science, i.e. "concurrency"; here important results have been obtained by Milner's group (see [Mi88]), by the group of Bergstra and Kloop (see [BK84]) etc. But we think this motto may equally well be applied to other fields of computer science.

The algebrization of the classical field of automata and language theory was made in the setting of Kleene's operations [Kl56], i.e. constants union, product and repetition (star). To this end an algebraic structure, called "regular algebra" has been introduced by Conway [Co71]. (A complete axiomatization for regular algebra is still unknown, by authors' knowledge.)

The algebrization of (the) behaviour of Yanov deterministic program schemes was made in the setting of Elgot's operations [El75], i.e. constants, composition, tupling and iteration. In this context an algebraic structure called "iteration theory" has been single out in [BEW80]. Perhaps the most important result is Esik's Theorem [Es80] which give a complete set of equations for iteration theories.

The present paper deals with the algebrization of flowchart schemes. Our convinction is the most adequate algebraic structure to study acyclic flowchart schemes is a symmetric strict monoidal category, presented for example in [ML71,Ma76]. To study (cyclic) flowchart schemes we introduce an algebraic structure, called biflow, which is a symmetric strict monoidal category endowed with a looping operation, called feedbackation, axiomatized by a few simple equations. The feedback operation was introduced in [St86,86a]. This operation is more adequate to study cyclic flowchart schemes than iteration [St86a]; see also [CS88] for a detalied comparation of feedbackation, iteration and repetition.

This paper is included in a sequence of papers 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 the feedbackation as the looping operation. The motivation and the framework of this theory are given in [CS87]. The present paper together with [CS87a] and [CS89a] cover the results presented in Sections 1 and 4 in [CS87].

In the last years we have observed a tendency to find an algebraic formalism for graphs, digraphs or nets emerging from different fields: communication systems [Pa87], graph grammars [BC88], flowchart schemes [CU82 & CG84, BE85, St86, CS87 & 87a]. It seems for us that all these algebraic approaches lead to a common, and consequently basic, algebraic structure. For example, there is a characterization of biflow using only identities, summation and (an extended) feedbackation. Summation and this extended feedbackation are similar to parallel and linking operators in [Pa87], respectively and the resulted axioms for biflow are essentially the axioms in [Pa87], without ∅ and completed with some axioms for identities !!.

1. Flows and Biflows

The objects of the categories $\mathbb{B}$ we work with form a monoid denoted by $(Ob(\mathbb{B}),+,e)$. The operations we use are:

Composition (in diagramatic order) $\_\cdot\_ : \mathbb{B}(a,b)\times\mathbb{B}(b,c)\to\mathbb{B}(a,c)$;

Identity $I_a\in\mathbb{B}(a,a)$;

Summation $\_+\_ : \mathbb{B}(a,b)\times\mathbb{B}(c,d)\to\mathbb{B}(a+c,b+d)$;

Block Transposition $\gamma_{a,b} : \mathbb{B}(a+b,b+a)$;

(Right) Feedbackation $\_\uparrow^a : \mathbb{B}(b+a,c+a)\to\mathbb{B}(b,c)$.

The composition sign "$\cdot$" is usually omitted.
Some axioms for these operations are given in Table 1.

A <u>symmetric strict monoidal category</u> (<u>ssmc</u>, for short) cf. [ML71, Ma76] is a category endowed with summation and block transpositions and fulfilling the axioms B1-10 of Table 1. A <u>biflow</u> is an ssmc endowed with a (right) feedbackation and fulfilling the axioms B1-16 of Table 1. A <u>flow</u> is a structure which fulfills all the axioms in $\{B1\text{-}16\}\setminus\{B6,B10,B13\}$, the axioms B6, B10 whenever g or u is a block transposition and B13 whenever g is a block transposition. As we shall see below the only difference between a flow and a biflow is B10, i.e. biflow = flow fulfilling B10.

We say two morphisms u and g <u>permute</u>, and write u P g, if they fulfill the equality in B10. It is easy to see that in a flow if u P g then $fu+gv = (f+g)(u+v)$ holds for arbitrary f and v, hence a flow fulfilling B10 also fulfills B6.

<u>1.1 Proposition.</u> Suppose $\mathbb{B}$ is a flow and $f\in\mathbb{B}(c+a,d+b)$, $g\in\mathbb{B}(u+b,v+a)$ are such that f P g. Then

$$((I_u+f)(\gamma_{u,d}+I_b)(I_d+g))\uparrow^a = \gamma_{u,c}((I_c+g)(\gamma_{c,v}+I_a)(I_v+f))\uparrow^b\cdot\gamma_{v,d}\,.$$

TABLE 1. These axioms define a biflow (B1–10 define an ssmc).

B1 $(fg)h = f(gh)$	B2 $I_a f = f = f I_b$
B3 $(f+g)+h = f+(g+h)$	B5 $I_a + I_b = I_{a+b}$
B4 $I_e + f = f = f + I_e$	B6 $(f+g)(u+v) = fu+gv$
B7 $\chi_{a,b} \cdot \chi_{b,a} = I_{a+b}$	B9 $\chi_{a,b+c} = (\chi_{a,b}+I_c)(I_b+\chi_{a,c})$
B8 $\chi_{a,e} = I_a$	B10 $(u+g)\chi_{b,d} = \chi_{a,c}(g+u)$
	for $u:a \to b$, $g:c \to d$
B11 $f(g\uparrow^a)h = ((f+I_a)g(h+I_a))\uparrow^a$	B14 $f\uparrow^b\uparrow^a = f\uparrow^{a+b}$
B12 $f+g\uparrow^a = (f+g)\uparrow^a$	B15 $I_a\uparrow^a = I_e$
B13 $(f(I_d+g))\uparrow^a = ((I_c+g)f)\uparrow^b$	B16 $\chi_{a,a}\uparrow^a = I_a$
for $f:c+a \to d+b$, $g:b \to a$	

Proof. $((I_u+f)(\chi_{u,d}+I_b)(I_d+g))\uparrow^a$

$= ((I_u+f)(\chi_{u,d}+\chi_{b,b}\uparrow^b)(I_d+g))\uparrow^a$ by B6

$= ([I_u+(f+I_b)](\chi_{u,d}+\chi_{b,b})[(I_d+g)+I_b])\uparrow^b\uparrow^a$ by B11,12

$= ((I_u+\chi_{c+a,b})[(I_{u+b}+f)(I_u+\chi_{b,d+b})(\chi_{u,d}+\chi_{b,b})(\chi_{d,u+b}+I_b)(g+I_{d+b})]$

$\cdot(\chi_{v+a,d}+I_b))\uparrow^{a+b}$ by B6,10_{flow}

$= ((I_{u+c}+\chi_{a,b})(I_u+\chi_{c,b}+I_a)[(I_{u+b}+f)(g+I_{d+b})](I_v+\chi_{a,d}+I_b))\uparrow^{a+b}\cdot\chi_{v,d}$ B1–10 for χ

$= ((I_u+\chi_{c,b}+I_a)[(g+I_{c+a})(I_{v+a}+f)](I_v+\chi_{a,d+b}))\uparrow^{b+a}\cdot\chi_{v,d}$ by $B13_{flow}$, fPg

$= \chi_{u,c}((\chi_{c,u+b}+I_a)(g+I_{c+a})(I_v+\chi_{a,c+a})(I_v+f+I_a))\uparrow^a\uparrow^b\cdot\chi_{v,d}$ by B7–11_{flow}

$= \chi_{u,c}((I_c+g)\chi_{c,v+a}(I_v+\chi_{a,c})(I_{v+c}+\chi_{a,a}\uparrow^a)(I_v+f))\uparrow^b\cdot\chi_{v,d}$ by B10–12_{flow}

$= \chi_{u,c}((I_c+g)(\chi_{c,v}+I_a)(I_v+f))\uparrow^b\cdot\chi_{v,d}$ by B16. ∎

Applying this proposition for $u = v = e$ we find that in a flow B13 holds whenever $g \mathrel{P} f$.

1.2 Corollary. (i) A biflow is a flow over an ssmc.

(ii) B is a biflow iff B is a flow and all morphisms in B permute. ∎

$H:B \to B'$ is an <u>ssmc (resp. flow, biflow) morphism</u> if H is a functor which is a monoid morphism on objects and fulfills

1. $H(f \star g) = H(f)+H(g)$ for f,g morphisms in B and

2. $H(\gamma_{a,b}) = \gamma_{H(a),H(b)}$ for a,b objects in B

(resp. 1, 2 and

3. $H(f\uparrow^{a}) = (H(f))\uparrow^{H(a)}$ for f morphism in B and a object in B).

Let M be a monoid and <u>struc</u> $\in \{$ssmc, flow, biflow$\}$. We say B is an <u>M-struc</u> if B is a struc and its monoid of objects is M. Moreover, $H:B \to B'$ is an <u>M-struc morphism</u> if H is a struc morphism and $H(a) = a$ for every $a \in M$. Note that the M-struc structures form a variety in the sense of many-sorted universal algebra, hence there exists an initial M-struc structure.

A composite of an arbitrary number of morphisms of the type $I_a+\gamma_{b,c}+I_d$ is called an <u>aα-base morphism</u>. Note that in a flow if f or g is an aα-base morphism then $f\ P\ g$, therefore the subcategory of all the aα-base morphisms of a flow form an ssmc (in fact, a biflow) in which all morphisms are isomorphisms.

<u>1.3 Proposition.</u> Let S be a set and $\mathbb{B}i_S$ the biflow of the S-sorted bijections.[1] If B is a flow and $h:S^{*} \to Ob(B)$ a monoid morphism, then there is a unique flow morphism $H: \mathbb{B}i_S \to B$ such that $H(a) = h(a)$ for every $a \in S^{*}$.

<u>Proof.</u> The subcategory of all the aα-base morphisms in B form an ssmc, hence by Theorem . in [CS89] there is a unique ssmc morphism $H: \mathbb{B}i_S \to B$ such that $H(a) = h(a)$, for every $a \in S^{*}$. The proof that $H(f\uparrow^{a}) = H(f)\uparrow^{H(a)}$ is the same as the proof of the similar fact in Theorem 1.16 in [CS87a]. ☒

Sometimes we prefer to work with the <u>left feedbackation</u> $\uparrow^{a}\_ : B(a+b,a+c) \to B(b,c)$. Consequently, we recall the passing from one feedbackation to the other one:

$\uparrow^{a}f = (\gamma_{b,a}\, f\, \gamma_{a,c})\uparrow^{a}$, for $f \in B(a+b,a+c)$ and

1) A definition of $\mathbb{B}i_S$ may be found in Section 4.

$f\uparrow^{a} = \uparrow^{a}(\mathsf{X}_{a,b}\, f\, \mathsf{X}_{c,a})$, for $f \in B(b+a, c+a)$.

In the case the monoid of objects is a free monoid S^{*}, feedbackation is completly determinated by its restriction to letters, i.e. by $\uparrow^{s}$, for $s \in S$. We give here an equivalent axiom system based on this scalar feedbackation.

1.4 Proposition. Suppose B is a category with $Ob(B) = S^{*}$ and fulfills all the axioms in $\{B1\text{–}10\} \setminus \{B6, B10\}$ and B6, B10 whenever g or u is a block transposition. If moreover, a scalar (right) feedbackation

$$\_\uparrow^{s} : B(a+s, b+s) \rightarrow B(a,b), \quad \text{for } a,b \in S \text{ and } s \in S$$

is given such that

SF1 $\quad g(f\uparrow^{s})h = ((g+I_{s})f(h+I_{s}))\uparrow^{s}$, for $s \in S$

SF2 $\quad I_{t}+f\uparrow^{s} = (I_{t}+f)\uparrow^{s}$, for $s, t \in S$

SF3 $\quad (f(I_{b}+\mathsf{X}_{s,t}))\uparrow^{s}\uparrow^{t} = ((I_{a}+\mathsf{X}_{s,t})f)\uparrow^{t}\uparrow^{s}$, for $s, t \in S$

SF4 $\quad I_{s}\uparrow^{s} = I_{\lambda}$, where λ is the empty word

SF5 $\quad \mathsf{X}_{s,s}\uparrow^{s} = I_{s}$

hold in B, then B becomes a flow using the feedbackation defined by

$$f\uparrow^{\lambda} = f \quad \text{and} \quad f\uparrow^{s+a} = f\uparrow^{a}\uparrow^{s} \ (s \in S).$$

Proof. The axioms of flow may be proved by induction on the lenght of the word a used in $\uparrow^{a}$. ▨

Sections 2,3 which follow relate our feedbackation to other well-known looping operations, i.e. to iteration and repetition. This sections may be skipped at a first reading and studied later.

2. Iteration and Feedbackation

A simple way to give examples of biflows is to use our knowledge about iteration and the first Theorem in [CS88]. This way we get examples of biflows over algebraic theories.

The ADJ-group has extended Lawvere's concept of algebraic theories (axiomatized by Elgot) to the case when the objects of the category form a free monoid S^*; they also has introduced some ordered variants of S^*-algebraic theories, for instance ω-continuous and rationally closed S^*-algebraic theories cf. [TWW79].

In [CS89] we have extended the concept of algebraic theory to the case when the objects of the category form an arbitrary monoid M, and called it <u>M-(algebraic) theory</u>; see also [CS88]. Although in this section we use ordered, ω-continuous and rationally closed theories over an arbitrary monoid M, the definitions and the proof are the same as in the case of S^*- algebraic theories.

In an M-theory T the source tupling of $f \in T(a,c)$ and $g \in T(b,c)$ is denoted by $\langle f,g \rangle$.

<u>2.1 Definition.</u> An <u>ordered M-theory</u> is an M-theory T fulfilling the following supplementary conditions:

- for $a,b \in M$ the set $T(a,b)$ is partially ordered with a least element $\perp_{a,b}$;
- composition is increasing and left strict, i.e. $\perp_{a,b} f = \perp_{a,c}$;
- tupling is increasing.

An <u>ω-continuous M-theory</u> is an ordered M-theory T fulfilling the following supplementary conditions:

- for $a,b \in M$ the ordered set $T(a,b)$ is ω-complete;
- composition is ω-continuous.

A <u>rationally closed M-theory</u> is an ordered M-theory T in which an iteration

$$^{\dagger} : T(a,a+b) \to T(a,b), \quad \text{for } a,b \in M$$

<u>is given</u> satisfying the following axioms:

RC1 $\quad (f(I_a+g))^{\dagger} = f^{\dagger}g, \quad$ for $f \in T(a,a+b), g \in T(b,c)$;

RC2 $f\langle f^{\dagger}, I_b\rangle = f^{\dagger}$, for $f \in T(a, a+b)$;

RC3 $f\langle g, I_b\rangle \leq g$ implies $f^{\dagger} \leq g$ for $f \in T(a, a+b)$, $g \in T(a, b)$.

[By applying the axioms RC2 and RC3 it is easy to prove that iteration in a rationally closed theory is increasing.]

An M-theory with iterate is an M-theory T in which an iteration is given fulfilling the axioms RC1, RC2 and

I1 $(f(\langle I_a, I_a\rangle + I_b))^{\dagger} = f^{\dagger\dagger}$, for $f \in T(a, a+a+b)$;

I2 $g(f(g+I_c))^{\dagger} = (gf)^{\dagger}$, for $f \in T(a, b+c)$, $g \in T(b, a)$. ▨

2.2 Proposition. (i) Every ω-continuous M-theory is rationally closed.
(ii) Every rationally closed M-theory is an M-theory with iterate.
(iii) Every M-theory with iterate is a biflow.

Proof. (i) The proof is well-known. We only mention that in an ω-continuous theory the iterate of $f: a \rightarrow a+b$ is by definition the least upper bound of the sequence defined by $f^{(0)} = \perp_{a,b}$ and $f^{(n+1)} = f\langle f^{(n)}, I_b\rangle$.

(ii) A proof may be obtained using the proof given for S^{*}-theories in [Ca85], Theorem 1.5 and Proposition B1 of Appendix B in [St87] to see that the axiomatic system used in [Ca85] is equivalent to the present one. An easier proof may be obtain by a direct verification of the axioms I1-2 in the context of rationally closed theories.

(iii) The first proof was given in [CS87a]; another proof may be found in [CS88]. We recall that in [CS88] we used the following passing between iteration and feedbackation

$\uparrow^{a}\langle f, g\rangle = g\langle f^{\dagger}, I_c\rangle$, for $f: a \rightarrow a+c$ and $g: b \rightarrow a+c$. ▨

In literature there are many examples of ω-continuous theories. Using the above proposition we get many examples of biflows. For instance, the basic semantic models $\mathbb{P}fn(S)$ for deterministic flowchart schemes and $\mathbb{R}el(S)$ for nondeterministic flowchart schemes are biflows. The construction of this models may be found in [CS87].

3. Repetition and Feedbackation

The most adecvate frame to study repetition is a matrix theory, introduced by [El76]. In [CS89] we have extended this concept to the case when the objects in the category form an arbitrary monoid, and called it M-matrix theory; see also [CS88].

In an M-matrix theory T the cosource tupling of $f \in T(a,b)$ and $g \in T(a,c)$ is denoted by $[f,g]$. Note that in an M-matrix theory T one may define a union operation as follows

$$f \cup g = [I_a, I_a](f+g)\langle I_b, I_b\rangle, \text{ for } f,g \in T(a,b).$$

In [El76] it is shown that a matrix theory T is completely determinated by the semiring $(T(1,1), \cup, \cdot, \perp_{1,1}, I_1)$.[2] When this semiring is complete in the sense of [Ei74], the resulted matrix theory has some supplementary properties that may be axiomatized as follows.

3.1 Definition. A complete M-matrix theory is an M-matrix theory T in which for every $a,b \in M$ and for every family $\{f_i\}_{i\in I}$ of morphisms in $T(a,b)$ a morphism $\bigcup_{i\in I} f_i \in T(a,b)$ is given fulfilling the following axioms:

CMT1 $\bigcup_{i\in\{1,\ldots,n\}} f_i = f_1 \cup f_2 \cup \ldots \cup f_n$, for $n \geq 0$

CMT2 $\bigcup_{i\in I} f_i = \bigcup_{j\in J} \bigcup_{i \in I_j} f_i$, if $I = \bigcup_{j\in J} I_j$ and $I_j \cap I_k = \emptyset$ for $j \neq k$

CMT3 $g(\bigcup_{i\in I} f_i) = \bigcup_{i\in I} g f_i$, for $g \in T(c,a)$

CMT4 $(\bigcup_{i\in I} f_i) g = \bigcup_{i\in I} f_i g$, for $g \in T(b,c)$.

For $n = 0$ axiom CMT1 should be read as $\bigcup_{i\in\emptyset} f_i = \perp_{a,b}$. ■

In a complete M-matrix theory a repetition $^*: T(a,a) \to T(a,a)$ for $a \in M$ may be defined as follows

$$f^* = \bigcup_{n\geq 0} f^n, \text{ for } a \in M,\ f \in T(a,a).$$

3.2 Lemma. The following identities hold in a complete M-matrix theory.

R1 $f^* = I_a \cup ff^*$, for $f \in T(a,a)$

---

[2]) 0_a (resp. $\perp_a$) denotes the unique morphism in $T(e,a)$ (resp. in $T(a,e)$); $\perp_{a,b}$ denotes $\perp_a \cdot 0_b$.

Note that $\bigcup_{n\in N} \perp_{a,b} = \bigcup_{n \in N} \bigcup_{i\in\emptyset} f_i = \bigcup_{i \in \bigcup_{n\in N} \emptyset} f_i = \bigcup_{i\in\emptyset} f_i = \perp_{a,b}$.

R2 $(f \cup g)^* = (f^*g)^*f^*$, for $f,g \in T(a,a)$

R3 $f(gf)^* = (fg)^*f$, for $f \in T(a,b)$, $g \in T(b,a)$.

Proof. Since $T(a,a)$ is a complete semiring the identities R1 and R2 hold. The proof of R3 is obvious. ■

Let T be an M-matrix theory (in which a repetition is given) fulfilling the conditions R1-3 above. The second Theorem in [CS88] shows that this concept is equivalent to the concept of biflow over an M-matrix theory. So we get the following result.

3.3 Proposition. Every complete M-matrix theory is a biflow. ■

In an M-matrix theory T we denote by $\begin{pmatrix} f & g \\ h & i \end{pmatrix}$, for $f \in T(a,c)$, $g \in T(a,d)$, $h \in T(b,c)$ and $i \in T(b,d)$ the morphism $\langle [f,g],[h,i] \rangle$ $(= [\langle f,h \rangle, \langle g,i \rangle])$. Note that every morphism form $T(a+b,c+d)$ may be written in a unique way as above.

We recall that in a biflow over a matrix theory the left feedbackation is defined by

$$\uparrow^a \begin{pmatrix} f & g \\ h & i \end{pmatrix} = hf^*g \cup i \text{ , for } f:a \to a,\ g:a \to c,\ h:b \to a,\ i:b \to c.$$

Let T be an S^*-matrix theory. If $a \in S^*$ then $|a|$ is the length of a and for $i \in [|a|]$,[3] a_i denotes the i-th letter of a, therefore $a = a_1 + \ldots + a_{|a|}$. For $f \in T(a,b)$, $i \in [|a|]$ and $j \in [|b|]$ let $f_{ij} = (0_{a'} + I_{a_i} + 0_{a''}) f (\bot_{b'} + I_{b_j} + \bot_{b''})$, where $a' = a_1 + \ldots + a_{i-1}$, $a'' = a_{i+1} + \ldots + a_{|a|}$, $b' = b_1 + \ldots + b_{j-1}$ and $b'' = b_{j+1} + \ldots + b_{|b|}$.

We say $f \in T(a,b)$ preserves the sorts if for $i \in [|a|]$, $j \in [|b|]$

$$f_{ij} \neq \bot_{a_i,b_j} \text{ implies } a_i = b_j.$$

3.4 Proposition. In an S^*-matrix theory T the collection T_{sort} of all the morphisms which preserve sorts is a matrix subtheory. Moreover, if T is complete, then T_{sort} is closed under repetition.

---

[3]) $[n]$ denotes the set $\{1,2,\ldots,n\}$; particularly $[0] = \emptyset$.

Proof. It is easy to show that 0_a, $\perp_a$, $\langle I_a, I_a \rangle$ and $[I_a, I_a]$ are in T_{sort} for every $a \in S^*$.

T_{sort} is closed under composition. Indeed, suppose $f \in T_{sort}(a,b)$, $g \in T_{sort}(b,c)$, $i \in [|a|]$ and $k \in [|b|]$ are such that $(fg)_{ik} \neq \perp_{a_i, c_k}$. Since $(fg)_{ik} = \bigcup_{j \in [|b|]} f_{ij} g_{jk}$, there is $j \in [|b|]$ with $f_{ij} g_{jk} \neq \perp_{a_i, c_k}$. This show that $f_{ij} \neq \perp_{a_i, b_j}$ and $g_{jk} \neq \perp_{b_j, c_k}$, hence $a_i = b_j = c_k$.

T_{sort} is closed under summation. Indeed, suppose $f \in T_{sort}(a,b)$ and $g \in T_{sort}(c,d)$. Since $(f+g)_{ij} = f_{ij}$ for $i \in [|a|]$ and $j \in [|c|]$, and $(f+g)_{|a|+i, |c|+j} = g_{ij}$ for $i \in [|b|]$ and $j \in [|d|]$, we deduce that $f+g \in T_{sort}(a+c, b+d)$. Hence T_{sort} is a matrix subtheory of T.

Suppose moreover T is complete and $f \in T_{sort}(a,a)$, $i \in [|a|]$ and $j \in [|a|]$ are such that $f^*_{ij} \neq \perp_{a_i, a_j}$. Since $f^*_{ij} = \bigcup_{n \geq 0} f^n_{ij}$, there is $n \geq 0$ such that $f^n_{ij} \neq \perp_{a_i, a_j}$, hence $a_i = a_j$. ∎

4. Examples of Biflows

Let $(S^*, +, \lambda)$ be the free monoid generated by the set S. The theory $\mathbb{R}el_S$[4] of S-sorted relations is defined as follows:

$\mathbb{R}el_S(a,b) = \{ f \subseteq [|a|] \times [|b|] \mid (i,j) \in f \text{ implies } a_i = b_j \}$, for $a, b \in S^*$.[5]

Composition: for $f \in \mathbb{R}el_S(a,b)$ and $g \in \mathbb{R}el_S(b,c)$ the composite is the usual one defined by

$$fg = \{ (i,k) \mid \exists\, j \in [|b|] \text{ such that } (i,j) \in f \text{ and } (j,k) \in g \}.$$

Summation: for $f \in \mathbb{R}el_S(a,b)$ and $g \in \mathbb{R}el_S(c,d)$ the sum is defined by

$$f+g = f \cup \{ (|a|+i, |b|+j) \mid (i,j) \in g \}.$$

Constants: for $a, b \in S^*$

$$I_a = \{ (i,i) \mid i \in [|a|] \}$$

$$V_{a,b} = \{ (i, |b|+i) \mid i \in [|a|] \} \cup \{ (|a|+j, j) \mid j \in [|b|] \}. ∎$$

4) When S has exactly one element the index S in $\mathbb{R}el_S$ (and in its subtheories $\mathbb{B}i_S$, $\mathbb{P}fn_S$ etc. defined below) is dropped.

5) See the notations introduced before Proposition 3.4.

$\mathbb{R}el_S$ is a biflow. To see this one may use different methods. (i) One may try to prove directly that the axioms B11-16 hold in $\mathbb{R}el_S$. To this end it is useful to use Proposition 1.4. In this case the following definition has to be used.

Scalar (Right) Feedbackation: for $f \in \mathbb{R}el_S(a+s, b+s)$ with $s \in S$, the scalar feedback is defined by

$$f\uparrow^s = \{(i,j) \mid (i,j) \in f \ \text{ or } \ [(i, |b|+1) \in f \text{ and } (|a|+1) \in f]\}.$$

(ii) Since Rel_S is an ω-continuous theory one may use iteration and the results in Section 2. (iii) Since $\mathbb{R}el_S$ is a complete matrix theory one may use repetition and the results in Section 3.

From the sexteen types of relations studied in [CS89] only nine of them are closed under feedback and the partial order of their inclusions may be represented as the Hasse diagram in Figure 1.

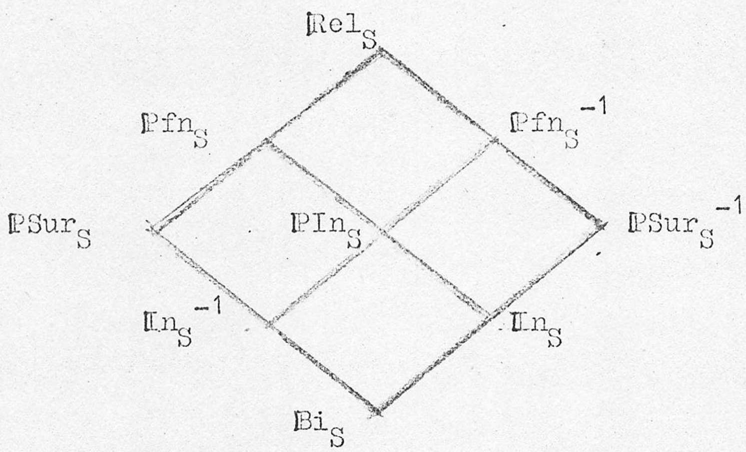

Figure 1. Relations closed under feedback.

In this figure $\mathbb{B}i_S$, $\mathbb{I}n_S$, $\mathbb{P}In_S$, $\mathbb{P}Sur_S$ and $\mathbb{P}fn_S$ are the biflows of the bijections, of the injections, of the partial injections, of the partial surjections and of the partial functions, respectively. In the same figure $\mathbb{I}n_S^{-1}$, $\mathbb{P}Sur_S^{-1}$ and $\mathbb{P}fn_S^{-1}$ are the biflows of the converses of injections, of the converses of partial surjections and of the converses of partial functions, respectively. In [CS89a] it is shown that all these ssmc-ies become biflows in a unique way.

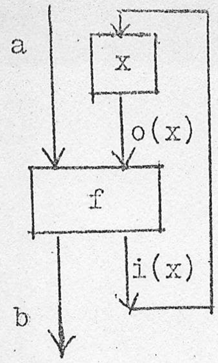

Figure 2.

B. FLOWCHART SCHEMES

5. The Flow of Flowchart Scheme Representations

Model $\mathbb{F}l_{X,B}$. Let B be a (bi)flow, $(X,+,\varepsilon)$ a monoid and $i,o:X \to Ob(B)$ two monoid morphisms. The algebra $\mathbb{F}l_{X,B}$ of pairs over X and B is defined as follows (see [CS87a] for motivation).

$$\mathbb{F}l_{X,B}(a,b) = \{(x,f) \mid x \in X \text{ and } f \in B(a+o(x), b+i(x))\}, \quad \text{for } a,b \in Ob(B).$$

Composition: for $(x,f):a \to b$ and $(y,g):b \to c$ the composite is

$$(x,f)(y,g) = (x+y, (f+I_{o(y)})(I_b + \mathsf{X}_{i(x),o(y)})(g+I_{i(x)})(I_c + \mathsf{X}_{i(y),i(x)}));$$

Summation: for $(x,f):a \to b$ and $(y,g):c \to d$ the sum is

$$(x,f)+(y,g) = (x+y, (I_a + \mathsf{X}_{c,o(x)} + I_{o(y)})(f+g)(I_b + \mathsf{X}_{i(x),d} + I_{o(y)}));$$

(Left)Feedbackation: for $(x,f):a+b \to a+c$ the left feedback is

$$\uparrow^a(x,f) = (x, \uparrow^a f);$$

Constants: $I_a = (\varepsilon, I_a)$ and $\mathsf{X}_{a,b} = (\varepsilon, \mathsf{X}_{a,b})$. ∎

It is known from [CS87a, Theorem 2.a.4] that $\mathbb{F}l_{X,B}$ is a flow, whenever B is a flow.

Comment. A pair (x,f) in $\mathbb{F}l_{X,B}(a,b)$ represent a flowchart scheme with input a and output b, which may be illustrated as in Figure 2.

In the case of usual flowchart schemes X is the monoid freely generated by a set Σ of statements and B is $\mathbb{P}fn$, namely we work with partial flowchart schemes. The usual complete flowchart schemes are represented by pairs in $\mathbb{F}l_{X,\mathbb{F}n}$, but neither the theory of functions $\mathbb{F}n$, nor $\mathbb{F}l_{X,\mathbb{F}n}$ is closed under feedback. Consequently we prefer to use a greater set of schemes which has better properties.

In other papers, for example in [BE85] the authors prefer to use functions in $\mathbb{F}n$ and to add a distinguished statement $\perp$ with one input and without outputs. The flowchart schemes studied in this case is the same as in our case, but the mathematical representation slightly differs from ours. This is a matter of choice: What is better, to use partial functions or total functions with an additional undefined element, partial trees or total trees with an additional undefined label, etc? We think is technically better to use partial functions, partial trees or partial schemes. ■

The morphisms of a flow C form a monoid denoted by Mor(C). First, note that $E_X : X \to Mor(\mathbb{F}l_{X,B})$, defined by $E_X(x) = (x, \vee_{i(x),o(x)})$ for $x \in X$, is an injective monoid morphism. Hence we may identify x with $E_X(x)$. Second, note that $E_B : B \to \mathbb{F}l_{X,B}$, defined by $E_B(f) = (\varepsilon, f)$ for a morphism f in B, is an Ob(B)-flow morphism which is injective on every B(a,b). Hence we also may identify a morphism f in B with $E_B(f)$. Finally, note that if $(x,f) \in \mathbb{F}l_{X,B}(a,b)$, then $(x,f) = ((I_a + x)f)\uparrow^{i(x)}$.

<u>5.1 Definition.</u> An <u>interpretation</u> of X and B in a flow B' is a pair (I,H), where $H : B \to B'$ is a flow morphism, $I : X \to Mor(B')$ is a monoid morphism such that $I(x) \in B'(H(i(x)), H(o(x)))$ for $x \in X$, and $I(x)\ P\ H(f)$ for every x in X and f morphism in B. ■

The last condition is superflous when B' is a biflow. By the above observations (E_X, E_B) is an interpretation of X and B in $\mathbb{F}l_{X,B}$ that we call the <u>standard interpretation</u>.

<u>5.2 Theorem.</u> If (I,H) is an interpretation of X and B in a flow B', then there is a unique flow morphism

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

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

<u>Proof.</u> The proof is similar to that of Theorem 5.2.b in [CS87a]. We only mention that the extension is defined as follows $(I,H)^f(x,g) = ((I_{H(a)} + I(x))H(g))\uparrow^{H(i(x))}$ for $(x,g) \in \mathbb{F}l_{X,B}(a,b)$. ■

5.3 Example. In order to help the reader to have a better insight into the role of Definition 5.1 and Theorem 5.2 we come back to the usual case $X = \Sigma^*$ and $B = \mathbb{P}fn$. Let S be the set of the states of the computer device. As B' we take $\mathbb{P}fn(S)$, the theory of partial functions over S. (Recall that $\mathbb{P}fn(S)(m,n)$ is equal to the set of all the partial functions from $S \times [m]$ to $S \times [n]$, for $m,n \in \mathbb{N}$.)

As H we take the embedding of $\mathbb{P}fn$ into $\mathbb{P}fn(S)$ given by $g \in \mathbb{P}fn(m,n) \longmapsto H(g) \in \mathbb{P}fn(S)(m,n)$, where

$$H(g)(s,i) = (s,g(i)), \text{ for } (s,i) \in S \times [m].$$

Note that H is a biflow morphism, indeed. In fact H(g) gives the behaviour of g, when g is thought as a program without statements.

In order to get the semantics of flowchart schemes over Σ we still need to know the behaviour $I(\sigma) \in \mathbb{P}fn(S)(i(\sigma),o(\sigma))$ of every atomic statement $\sigma \in \Sigma$. The monoid morphism $I: \Sigma^* \to Mor(\mathbb{P}fn(S))$ is the unique extension of this function to a monoid morphism.

The most important fact is that the behaviour of a flowchart scheme represented by $(x,g) \in \mathbb{F}l_{\Sigma,\mathbb{P}fn}(m,n)$ is

$$(I,H)^f(x,g) = ((I_m + I(x))H(g))\uparrow^{i(x)} \in \mathbb{P}fn(S)(m,n).$$

Therefore: the monoid morphism I gives the behaviour of the statements, the flow morphism H gives the behaviour of the connection morphisms (seen as programs without statements, i.e. containing only gotos) and the unique extension $(I,H)^f$ in Theorem 5.2 gives the resulted behaviour of the flowchart schemes, represented by pairs. ∎

6. Simulation by Bijections

The graph isomorphism is usualy used to define the flowchart scheme isomorphism: two concrete flowchart schemes are isomorphic (equal) if they are given by isomorphic, labelled graphs. We shall see in this section that two pairs represent the same concrete scheme if and only if they are similar via a bijection. But every concrete flowchart scheme is represented by at least one pair. So we may identify a concrete flowchart scheme with the set of all the pairs which represent it. This way we get a <u>syntactic model</u> for flowchart schemes which consist of classes of isomorphic pairs.

At the beginning of this section we work with usual, concrete, nondeterministic schemes, i.e. $X = \Sigma^*$ and $B = \mathbb{R}el$, the biflow of relations.

A word $w \in \Sigma^*$ may be seen as a function $w: [|w|] \to \Sigma$ which maps k to the k-th letter of w. Hence a morphism $j \in \mathbb{B}i_\Sigma(x,y)$ is a bijection $j: [|x|] \to [|y|]$ such that $jy = x$.

The monoid morphisms $i,o: \Sigma \to \mathbb{N}$ may be extended in a unique way to ssmc morphisms $i,o: \mathbb{B}i_\Sigma \to \mathbb{R}el$, cf. [CS89]. For an $j \in \mathbb{B}i_\Sigma(x,y)$, the extension $i(j) \in \mathbb{B}i(i(x),i(y))$ fulfills

(*) $i(j)(i(x_1+\ldots+x_{m-1})+t) = i(y_1+\ldots+y_{j(m)-1})+t$

for every $m \in [|x|]$ and $t \in [i(x_m)]$. We mention this may be taken as the definition of $i(j)$. An analogous property holds for $o: \mathbb{B}i_\Sigma \to \mathbb{R}el$.

The scheme represented by (x,f) in $\mathbb{F}l_{\Sigma,\mathbb{R}el}(a,b)$ may be seen as a graph in the following way (see Figure 2).

(1) There are a vertices for inputs, b vertices for outputs and $|x|$ internal vertices labelled by $x: [|x|] \to \Sigma$.

(2) The relation $f \in \mathbb{R}el(a+o(x), b+i(x))$ gives all the arrows of the scheme in the following way:

– $(p,q) \in f$ for $p \in [a]$ and $q \in [b]$ iff there is an arrow from input p to output q ;

– $(p, b+i(x_1+\ldots+x_{m-1})+t) \in f$ for $p \in [a]$, $m \in [|x|]$ and $t \in [i(x_m)]$ iff there is an arrow from input p to entry t of the statement which labels vertex m ;

- $(a+o(x_1+\ldots+x_{n-1})+s,q)\in f$ for $n\in[|x|]$, $s\in[o(x_n)]$ and $q\in[b]$ iff there is an arrow from exit s of the statement which labels vertex n to output q ;

- $(a+o(x_1+\ldots+x_{n-1})+s,\ b+i(x_1+\ldots+x_{m-1})+t)\in f$ for $n,m\in[|x|]$, $s\in[o(x_n)]$ and $t\in[i(x_m)]$ iff there is an arrow from exit s of the statement which labels vertex n to entry t of the statement which labels vertex m.

Suppose we have two pairs (x,f) and (y,g) in $\mathbb{Fl}_{\Sigma,\mathbb{Rel}}(a,b)$, representing the schemes F and G, respectively, and a bijection $j\in\mathbb{Bi}_{\Sigma}(x,y)$. This bijection j gives a bijection between the labelled, internal vertices of F and the labelled, internal vertices of G such that the corresponding vertices have the same label.

Let see the meaning of the bijections $i(j)\in\mathbb{Bi}_{\smile}(i(x),i(y))$ and $o(j)\in\mathbb{Bi}_{\smile}(o(x),o(y))$. In (✱) $i(x_1+\ldots+x_{m-1})+t$ represent entry t of statement x_m which labels vertex m of F and $i(y_1+\ldots+y_{j(m)-1})+t$ represents entry t of statement $y_{j(m)}$ $(= x_m)$ which labels the corresponding vertex j(m) of G. Therefore i(j) is a bijection between all the entries of the statements from F and all the entries of the statements from G such that two entries correspond by i(j) iff they are the entries with the same number of a statement which labels two vertices that correspond by j. Analogously, o(j) is a bijection between all the exits of the statements from F and all the exits of the statements from G such that two exits correspond by o(j) iff they are the exits with the same number of a statement which labels two vertices that correspond by j.

We show that $j\in\mathbb{Bi}_{\Sigma}(x,y)$ gives an isomorphism of F and G iff $f(I_b+i(j)) = (I_a+o(j))g$. This equality is equivalent with the following four conditions a, b, c and d.

a) $(p,q)\in f$ iff $(p,q)\in g$, for $p\in[a]$ and $q\in[b]$,

i.e. in F there is an arrow from input p to output q iff in G there is an arrow from input p to output q ;

b) $(p,b+i(x_1+\ldots+x_{m-1})+t)\in f$ iff $(p,b+i(y_1+\ldots+y_{j(m)-1})+t)\in g$ for $p\in[a]$, $m\in[|x|]$ and $t\in[i(x_m)]$,

i.e. in F there ia an arrow from input p to entry t of the statement which labels vertex m iff in G there is an arrow from input p to entry t of the statement which labels the corresponding vertex j(m) ;

c) $(a+o(x_1+\ldots+x_{n-1})+s,q)\in f$ iff $(a+o(y_1+\ldots+y_{j(n)-1})+s,q)\in g$ for $n\in[|x|]$, $s\in[o(x_n)]$ and $q\in[b]$,

i.e. in F there is an arrow from exit s of the statement which labels vertex n to output q iff in G there is an arrow from exit s of the statement which labels the corresponding vertex j(n) to output q ;

d) $(a+o(x_1+\ldots+x_{n-1})+s,\ b+i(x_1+\ldots+x_{m-1})+t)\in f$ iff $(a+o(y_1+\ldots+y_{j(n)-1})+s,\ b+i(y_1+\ldots+y_{j(m)-1})+t)\in g$, for $n,m\in[|x|]$, $s\in[o(x_n)]$ and $t\in[i(x_m)]$,

i.e. in F there is an arrow from exit s of the statement which labels vertex n to entry t of the statement which labels vertex m iff in G there is an arrow from exit s of the statement which labels vertex j(n) to entry t of the statement which labels vertex j(m).

These facts lead to the following result.

<u>6.1 Proposition.</u> The schemes represented by (x,f) and (y,g) in $\mathbb{F}l_{\Sigma,\mathbb{R}el}(a,b)$ are isomorphic if and only if there is a Σ-sorted bijection $j\in\mathbb{B}i_{\Sigma}(x,y)$ such that $f(I_b+i(j)) = (I_a+o(j))g$. ∎

In this case we say that (x,f) and (y,g) are similar via the bijection j. The resulted concept of simulation by bijections may be lifted to the general frame used in Section 5 as simulation via α-base morphisms.

For a monoid M we denote by $\mathbb{B}i_M$ the initial M-ssmc. When M is a free monoid S^* we have a clean model for $\mathbb{B}i_M$, namely the ssmc of all S-sorted bijections. We do not know such a model for an arbitrary monoid M. For a free monoid the following result is covered by Theorem . in [CS89].

6.2 Proposition. If B is an ssmc and $h:X \to Ob(B)$ is a monoid morphism, then there is a unique ssmc morphism $H: \mathbb{B}i_X \to B$ such that $H(x) = h(x)$ for every $x \in X$.

Proof. Let $h^{\Box}(B)$ be the X-ssmc defined by:

$h^{\Box}(B)(x,y) := B(h(x),h(y))$; $fg := fg$; $f+g := f+g$;

for $x,y \in X$ $I_x := I_{h(x)}$ and $\gamma_{x,y} := \gamma_{h(x),h(y)}$.

Let $\varepsilon_h : h^{\Box}(B) \to B$ be the ssmc morphism defined by $\varepsilon_h(x) = h(x)$ for $x \in X$, and $\varepsilon_h(f) = f$ for f morphism in $h^{\Box}(B)$. Since $\mathbb{B}i_X$ is the initial X-ssmc there is a unique X-ssmc morphism $H': \mathbb{B}i_X \to h^{\Box}(B)$. $H := H'\varepsilon_h$ is an ssmc morphism and $H(x) = h(x)$ for every $x \in X$. ∎

Applying this proposition to the monoid morphisms $i,o: X \to Ob(B)$ we obtain two ssmc morphisms $i^b, o^b: \mathbb{B}i_X \to B$ such that $i^b(x) = i(x)$ for $x \in X$, and $o^b(x) = o(x)$ for $x \in X$. Now we may say two pairs (x,f) and (y,g) in $\mathbb{F}l_{X,B}(a,b)$ are similar via an $a\alpha$-base morphism $j \in \mathbb{B}i_X(x,y)$ if $f(I_b + i^b(j)) = (I_a + o^b(j))g$.

Having in mind other examples of simulations we prefer to work even more abstract: We replace $\mathbb{B}i_X$ by an arbitrary X-ssmc Y and the ssmc morphisms i^b, o^b by two arbitrary ssmc morphisms $i,o: Y \to B$. (In this case $\mathbb{F}l_{X,B}$ is built up using the restrictions of i and o to the objects of Y.) We do not give a deep study of this abstract simulation in this paper. We only give the definition and some properties we need. All the hypotheses on Y we shall use in the next Section are valid when $Y = \mathbb{B}i_X$.

7. Abstract Simulation

Throughout this Section X is a monoid, Y is an X-ssmc, B is a biflow and $i,o: Y \to B$ are two ssmc morphisms.

<u>7.1 Definition.</u> Suppose we are given two pairs (x,f) and (y,g) in $\mathbb{F}l_{X,B}(a,b)$. We say (x,f) and (y,g) are <u>similar via $j\in Y(x,y)$</u>, and write $(x,f) \rightarrow_j (y,g)$, if $f(I_b+i(j)) = (I_a+o(j))g$. $(x,f) \rightarrow_Y (y,g)$ means $(x,f) \rightarrow_j (y,g)$ for some $j\in Y(x,y)$. The relation $\rightarrow_Y$ is called <u>simulation (via Y-morphisms)</u>. ∎

An easy computation shows that

<u>7.2 Lemma.</u> For every $x,y\in X$

$$(x+y)o(\gamma_{x,y}) \longrightarrow_{\gamma_{x,y}} i(\gamma_{x,y})(y+x).$$

<u>7.3 Lemma.</u> The simulation relation $\rightarrow_Y$ is a preorder, compatible to the flow operations.

<u>Proof.</u> Clearly $(x,f) \rightarrow_{I_x} (x,f)$. If $(x,f) \rightarrow_j (y,g)$ and $(y,g) \rightarrow_k (z,h)$, then $(x,f) \rightarrow_{jk} (z,h)$. Hence simulation is a preorder. The compatibility is proved by the following easy checked fact.

If $(x,f) \rightarrow_j (y,g)$ and $(x',f') \rightarrow_k (y',g')$, then:

- $(x,f)(x',f') \longrightarrow_{j+k} (y,g)(y',g')$
- $(x,f) + (x',f') \longrightarrow_{j+k} (y,g) + (y',g')$
- $\uparrow^a(x,f) \rightarrow_j \uparrow^a(y,g)$

whenever the operations make sense. ∎

<u>7.4 Lemma.</u> If every morphism in Y is an isomorphism, then the simulation relation is a flow congruence relation.

<u>Proof.</u> We have to show that $\rightarrow_Y$ is symmetric. If $(x,f) \rightarrow_Y (y,g)$ in $\mathbb{F}l_{X,B}(a,b)$, then there exists $j\in Y(x,y)$ such that $f(I_b+i(j)) = (I_a+o(j))g$. Since j is an isomorphism there is j^{-1} in Y such that $g(I_b+i(j^{-1})) = (I_a+o(j^{-1}))f$, hence $(y,g) \rightarrow_Y (x,f)$. ∎

<u>7.5 Lemma.</u> Suppose $\sim$ is a flow congruence relation in $\mathbb{F}l_{X,B}$ such that $(x+y)o(\gamma_{x,y}) \sim i(\gamma_{x,y})(y+x)$ for $x,y\in X$. Then

(i) If $(x,f) \to_j (y,g)$ for an aα-base morphism j, then $(x,f) \sim (y,g)$.

(ii) The quotient of $\mathbb{F}l_{X,B}$ by $\sim$ is a biflow.

Proof. The relation $x\, o(j) \sim i(j)\, y$ holds for every aα-base morphism $j \in Y(x,y)$. Indeed, if $j = I_x + \gamma_{y,z} + I_v$ then $(x+y+z+v)o(j) = x+(y+z)o(\gamma_{y,z})+v \sim x+i(\gamma_{y,z})(z+y)+v = i(j)(x+z+y+v)$ and a simple induction finishes the proof.

(i) Suppose $(x,f) \to_j (y,g)$ in $\mathbb{F}l_{X,B}(a,b)$. This shows that $f = (I_a+o(j))g(I_b+o(j^{-1}))$. Hence $(x,f) = ((I_a+x)f)\uparrow^{i(x)}$
$= ((I_a+xo(j))g(I_b+o(j^{-1})))\uparrow^{i(x)} = ((I_a+o(j^{-1}))(I_a+xo(j))g)\uparrow^{i(y)}$
$\sim ((I_a+o(j^{-1})o(j)y)g)\uparrow^{i(y)} = (y,g)$.

(ii) By Theorem 2.a.4 in [CS87a] $\mathbb{F}l_{X,B}$ is a flow. Suppose $(x,f):a \to b$ and $(y,g):c \to d$. Since

$$\gamma_{c,a}((x,f)+(y,g)) \to_{\gamma_{x,y}} ((y,g)+(x,f))\gamma_{d,b}$$

we deduce that $\gamma_{c,a}((x,f)+(y,g)) \sim ((y,g)+(x,f))\gamma_{d,b}$, therefore the quotient of $\mathbb{F}l_{X,B}$ by $\sim$ fulfils B10. By Corallary 1.2 this shows $\mathbb{F}l_{X,B}/\sim$ is a biflow. ∎

7.6 Proposition. If every morphism in Y is an aα-base morphism, then the simulation relation $\to_Y$ is the least congruence relation on $\mathbb{F}l_{X,B}$ which contains $((x+y)o(\gamma_{x,y}), i(\gamma_{x,y})(y+x))$ for $x,y \in X$.

Proof. As every aα-base morphism is an isomorphism we deduce from Lemmas 7.2, 7.3 and 7.4 that the simulation relation is a congruence relation which contains $((x+y)o(\gamma_{x,y}), i(\gamma_{x,y})(y+x))$ for $x,y \in X$. It follows from Lemma 7.5.(i) that $\to_Y$ is the least one. ∎

7.7 Lemma. Let B' be a biflow and $F: \mathbb{F}l_{X,B} \to B'$ a flow morphism. If $(x,f) \to_j (y,g)$ for an aα-base morphism j, then $F((x,f)) = F((y,g))$.

Proof. Let $\sim$ be the congruence relation on $\mathbb{F}l_{X,B}$ defined by $(x,f) \sim (y,g)$ iff $F((x,f)) = F((y,g))$. Since $F((x+y)o(\gamma_{x,y})) = (F(x)+F(y))\gamma_{F(o(x)),F(o(y))}$

$= \gamma_{F(i(x)),F(i(y))}(F(y)+F(x)) = F(i(\gamma_{x,y})(y+x))$, the result follows by applying Lemma 7.5.(i). ■

8. The Biflow of Flowchart Schemes

Assume X is a monoid, B is a biflow and $i,o: X \to Ob(B)$ are two monoid morphisms. Let $\sim$ be the least congruence relation on $\mathbb{F}\mathrm{l}_{X,B}$ which contains

$$((x+y)\gamma_{o(x),o(y)},\ \gamma_{i(x),i(y)}(y+x)), \quad \text{for } x,y \in X.$$

By Propositions 6.1 and 7.6 (and the observation that in $\mathbb{B}\mathrm{i}_X$ every morphism is an α-base morphism) we may identify <u>abstract flowchart schemes</u> with elements in $\mathbb{F}\mathrm{l}_{X,B}/\sim$. This quotient $\mathbb{F}\mathrm{l}_{X,B}/\sim$ is denoted by $\mathbb{FS}_{X,B}$. By Proposition 7.5.(ii) it is a biflow, which we call <u>the biflow of flowchart schemes with statements from X and connections from B.</u> The factorization morphism is denoted by $P^{\flat}: \mathbb{F}\mathrm{l}_{X,B} \to \mathbb{FS}_{X,B}$.

<u>8.1 Proposition.</u> For every flow morphism $F: \mathbb{F}\mathrm{l}_{X,B} \to B'$, where B' is a biflow, there is a unique flow morphism $F^{\#}: \mathbb{FS}_{X,B} \to B'$ such that $P^{\flat}F^{\#} = F$.

<u>Proof.</u> By B10, $F((x+y)\gamma_{o(x),o(y)}) = F(\gamma_{i(x),i(y)}(y+x))$ holds in B' for every $x,y \in X$. Hence the conclusion follows. ■

<u>8.2 Theorem.</u> If (I,H) is an interpretation of X and B in a biflow B', then there is a unique biflow morphism $(I,H)^b: \mathbb{FS}_{X,B} \to B'$ such that $E_X P^{\flat}(I,H)^b = I$ and $E_B P^{\flat}(I,H)^b = H$.

<u>Proof.</u> Apply Theorem 5.2 and Proposition 8.1. ■

<u>Final comment.</u> This theorem is the main result of this paper. A first version of this result was presented in [St86a], but there are significant improvements here (due to the first author): (i) bijections are axiomatized; (ii) the monoid of the objects of B is replaced here by an arbitrary monoid; (iii) the free monoid of statements is replaced here by an arbitrary monoid; and

(iv) the biflow morphisms are allowed here to change the objects in a monoid morphism manner. This theorem has many consequences. We sketch here two of them.

1. Suppose X is the free monoid generated by Σ, M is a monoid and $i,o:\Sigma^* \to M$ are two monoid morphisms. Moreover, assume we work in the category of M-biflows. Then an interpretation of Σ in an M-biflow B is a function $I:\Sigma \to Mor(B)$ such that $I(x)\in B(i(x),o(x))$ for every $x\in\Sigma$. An M-biflow B is freely generated by Σ if there is an interpretation I_Σ of Σ in B such that for every M-biflow B' and every interpretation I of Σ in B' there is a unique biflow morphism $I^\#:B \to B'$ such that $I_\Sigma I^\# = I$. We deduce from Theorem 8.2 that:

- if B is the initial M-biflow, then the M-biflow $\mathbb{FS}_{\Sigma^*,B}$ is freely generated by Σ ;

- for every M-biflow B the M-biflow $\mathbb{FS}_{\Sigma^*,B}$ is the coproduct of B with the M-biflow freely generated by Σ .

2. It is known in Equational Logic that an equation $\underline{\underline{e}}$ is deductible from a set E of equations if and only if $\underline{\underline{e}}$ is satisfied by every free algebra in the variety of algebras satisfying E (see [GM87]). Suppose we work in the category of S^*-biflows. We know that the biflow of S-sorted bijections $\mathbb{B}i_S$ is the initial S^*-biflow, hence $\mathbb{F}l_{\Sigma^*,\mathbb{B}i_S}$ is the S^*-biflow freely generated by Σ. An equation using biflow operations is satisfied in $\mathbb{F}l_{X,\mathbb{B}i_S}$ for all X if and only if it is deductible from the axioms of S^*-biflows. Therefore the axioms for S^*-biflows (and equational logic) give an axiomatization for flowchart schemes connected by S-sorted bijections.

References

[BC88] M. BAUDERON and B. COURCELLE, Graph expressions and graph rewritings, Math. System Theory 1988

[BE85] S.L. BLOOM and Z. ESIK, Axiomatizing schemes and their behaviour, J. Comput. System Sci. 31 (1985), 375-393.

[BEW80] S.L. BLOOM, C.C. ELGOT and J.B. WRIGHT, Vector iteration in pointed iterative theories, SIAM J. Comput. 9 (1980), 525-540.

[BK84] J.A. BERGSTRA and J.W. KLOP, Process algebra for synchronous communication, Inform. Control 60 (1984), 109-137.

[Ca85] V.E. CĂZĂNESCU, On context-free trees, Theoret. Comput. Sci. 41 (1985), 33-50.

[CG84] V.E. CĂZĂNESCU and Ş. GRAMA, On the definition of M-flowcharts, Preprint Series in Mathemetics No.56/1984, INCREST, Bucharest; also in: An. Univ. "Al.I. Cuza" Iasi, Mat. XXXIII, 4 (1987), 311-320.

[Co71] J.H. CONWAY, Regular algebra and finite machines, Chapman and Hall, London, 1971.

[CS87] V.E. CĂZĂNESCU and GH. ŞTEFĂNESCU, Towards a new algebraic foundation of flowchart scheme theory, Preprint Series in Mathematics No.43/1987, 45p, INCREST, Bucharest, December 1987.

[CS87a] V.E. CĂZĂNESCU and GH. ŞTEFĂNESCU, A formal representation of flowchart schemes, Preprint Series in Mathematics No.22/1987, 21p, INCREST, Bucharest, June 1987; also in: An. Univ. Bucureşti, Mat.-Inf. XXXVII, 2 (1988), 33-51.

[CS88] V.E. CĂZĂNESCU and GH. ŞTEFĂNESCU, Feedback, iteration and repetition, Preprint Series in Mathematics No.42/1988, 13p, INCREST, Bucureşti, August 1988.

[CS89] V.E. CĂZĂNESCU and GH. ŞTEFĂNESCU, Finite relations as initial abstract data types (to appear).

[CS89a] V.E. CĂZĂNESCU and GH. ŞTEFĂNESCU, Bi-Flow-Calculus (to appear).

[CU82] V.E. CĂZĂNESCU and C. UNGUREANU, Again on advice an structuring compilers and proving them correct, Preprint Series in Mathematics No.75/1982, 30p, INCREST, Bucharest, November 1982.

[Ei74] S. EILENBERG, Automata, languages and machines, Volume A, Academic Press, 1974.

[El75] C.C. ELGOT, Monadic computation and iterative algebraic theories, in: Logic Colloquium 1973 (H.E. Rose and J.C. Shepherdson, Eds.), North-Holland, Amsterdam, 1975, pp.175-230.

[El76] C.C. ELGOT, Matricial theories, J. Algebra 42 (1976), 391-422.

[Es80] Z. ÉSIK, Identities in iterative and rational theories, Comput. Linguistic and Comput. Language XIV (1980), 183-207; see also the paper in: J. Comput. System Sci. 27 (1983), 291-303.

[GM87] J.A. GOGUEN and J. MESEGUER, Completeness of many-sorted equational logic, Houston J. Math. 1987

[Kl56] S.C. KLEENE, Representation of events in nerve nets and finite automata, in: Automata studies (C.E. Shannon and J. McCarthy, Eds.), Ann. Math. Studies 34, Princeton Univers. Press, Princeton 1956, pp.3-41.

[Ma76] E.G. MANES, Algebraic theories, Springer-Verlag, 1976.

[Mi83] R. MILNER, Calculi for synchrony and asynchrony, Theoret. Comput. Sci. 25 (1983), 267-310.

[Mi88] R. MILNER, Operational and algebraic semantics of concurrent processes, Report ECS-LFCS-88-46, Computer Science Dept, University of Edinburgh, 1988.

[ML71] S. MAC LANE, Categories for the working mathematician, Springer-Verlag, 1971.

[Pa87] J. PARROW, Synchronisation flow algebra, Report ECS-LFCS-87-35, Computer Science Dept, University of Edinburgh, August 1987.

[St86] GH. ŞTEFĂNESCU, An algebraic theory of flowchart schemes, in: Proceedings CAAP'86 (P. Franchi-Zannettacci, Ed.), Lecture Notes in Computer Science 214, Springer-Verlag, 1986, pp. 60-73.

[St86a] GH. ŞTEFĂNESCU, Feedback theories (a calculus for isomorphism classes of flowchart schemes), Preprint Series in Mathematics No.24/1986, 11p, INCREST, Bucharest, April 1986.

[St87] GH. ŞTEFĂNESCU, On flowchart theories. Part I. The deterministic case, J. Comput. System Sci. 35 (1987), 163-191.

[TWW79] J.W. THATCHER, E.G. WAGNER and J.B. WRIGHT, Notes on algebraic fundamentals for theoretical computer science, in: Foundation of Computer Science III, Part 2. Language, logic, semantics (J.W. de Bakker and J. van Leeuwen, Eds.), Mathematical Centre Tracts 109, Math. Centrum, Amsterdam, 1979, pp. 83-164.