Algebraic Characterization of Reducible Flowcharts

HARTMUT SCHMECK

Institut für Informatik und Praktische Mathematik,
Christian-Albrechts-Universität Kiel,
D-2300 Kiel, West Germany

Received February 20, 1982

Reducible flowcharts as introduced by Hecht and Ullman have been algebraically characterized by Elgot and Shepherdson. They showed them to be freely generated from elementary flowcharts by means of composition, sum, and scalar iteration. In this paper the algebraic characterization is extended to the class of infinite almost accessible reducible flowcharts.

1. INTRODUCTION

Reducible flowcharts are of interest with respect to certain code improvement techniques [1]. Hecht and Ullman who introduced this class of flowcharts, provide a graph theoretic characterization [17, 18]. The flowcharts used by Thatcher, Wagner, and Wright in the target language of a compiler are reducible, too [7].

In [11, 12] Elgot and Shepherdson provide an algebraic characterization of the class of finite reducible and accessible flowcharts by showing it to be freely generated from certain simple flowcharts by means of composition, sum, and scalar iteration.

In this paper Elgot and Shepherdson's results are extended to an enlarged class of reducible flowcharts having neither to be fully accessible nor finite. To achieve this extension a new operation on flowcharts is defined, the strong composition, producing almost accessible flowcharts. Furthermore the flow theory over SUR as used in [12] is replaced by a new kind of flow theory over MAP. Finally to cope with infinite flowcharts a partial order on flowcharts is defined.

The fact that with respect to this partial order every infinite almost accessible flowchart is the least upper bound of an ω-chain of finite flowcharts then helps to extend our freeness results for classes of finite flowcharts to classes of infinite flowcharts.

The paper is organized as follows: In Section 2 we give some basic notation. Section 3 introduces the class of Γ-flowcharts, where Γ is a one-sorted signature. Further the subclasses of accessible, almost accessible, acyclic, and reducible Γ-flowcharts are defined. In Section 4 we define the operations of composition, strong composition, pairing, sum, and scalar iteration of flowcharts, allowing to view classes of flowcharts as strict monoidal categories [19]. The algebraic properties of these categories in Section 5 lead to the notion of a flow theory over MAP. In Section 6
freeness theorems are derived extending the results of Elgot and Shepherdson. The introduction of a partial order on Γ-flowcharts in Section 7 then leads to the derivation of freeness results for infinite flowcharts in Section 8. Finally Section 9 provides nondeterministic programs on a stack machine as an example of r-flowcharts.

2. Basic Notation

We begin with some basic notation. Let \( \text{Ord} \) denote the set of all ordinal numbers and let \( \omega \) be the least transfinite ordinal number. If \( \alpha \) is an ordinal number, define \( [\alpha] := \{ i + 1 \mid i < \alpha \} \). Thus \( [\omega] = \mathbb{N} \), the set of all natural numbers, \( [0] = \emptyset \) and for all \( n < \omega \) we have \( [n] = \{ 1, \ldots, n \} \). For ordinal numbers \( \alpha \) and \( \beta \) we define \( [\alpha] + [\beta] := [\alpha + \beta] \) and \( \alpha + [\beta] := [\alpha + i \mid i \in [\beta]] \).

If \( A \) is an arbitrary set, \( A^* \) denotes the free monoid of all words over \( A \), \( O_A \) is the unique mapping from \( \emptyset \) into \( A \) and \( 1_A \) is the identity on \( A \).

If \( f: A \rightarrow B \) is a mapping, the image of \( a \in A \) under \( f \) is denoted by \( a^f \). The composition of \( f \) with a mapping \( g: B \rightarrow C \) is written \( f \cdot g \), the pairing of \( f \) and \( g: C \rightarrow B \) is \( \langle f, g \rangle : A + C \rightarrow B \), and the sum of \( f \) and \( g: C \rightarrow D \) is \( f + g: A + C \rightarrow B + D \). All these mappings are defined as usual (see, e.g., [5]). Furthermore let \( f^* : A^* \rightarrow B^* \) denote the unique extension of \( f \) to a homomorphism of monoids.

For each \( n, p < \omega \) define \( \text{MAP}(n, p) = [[n] \rightarrow [p]] \) to be the set of all mappings from \( [n] \) to \( [p] \), \( \text{SUR}(n, p) = [[n] \twoheadrightarrow [p]] \) to be the set of all surjective mappings from \( [n] \) to \( [p] \), and \( \text{INJ}(n, p) = [[n] \hookrightarrow [p]] \) to be the set of all injective mappings from \( [n] \) to \( [p] \). Let MAP, SUR, and INJ denote the corresponding strict monoidal categories [19]. For each \( n, p < \omega \) define \( \iota^{n,p}_{11} : [n] \rightarrow [n + p] \) to be the embedding of \( [n] \) into \( [n + p] \), and \( \iota^{n,p}_{23} : [p] \rightarrow [n + p] \) to be the embedding of \( [p] \) into \( [n + p] \). That is, for all \( i \in [n], j \in [p] \) we have \( \iota^{n,p}_{11}(i) = i \) and \( \iota^{n,p}_{23}(j) = n + j \). For each \( n, m \) define \( \pi(n, m): [n + m] \rightarrow [m + n] \) by \( \pi(n, m) := \langle \iota^{m,n}_{12}, \iota^{n,m}_{21} \rangle \), i.e., \( \pi(n, m) \) is the permutation of \( [n] \) and \( n + [m] \).

A directed graph is a quadruple \( G = (V, E, \sigma, \tau) \) with set of vertices \( V \), set of edges \( E \), source mapping \( \sigma: E \rightarrow V \) and target mapping \( \tau: E \rightarrow V \). \( G \) is (partially) labeled iff there is a set of labels \( M \) and a (partial) labeling \( l: V \rightarrow M \). \( G \) is finite iff \( V \) and \( E \) are finite. \( G \) is locally finite iff for all \( v \in V \) the set \( v E^{-1} \) is finite. \( G \) is ordered iff for all \( v \in V \) there is a \( k < \omega \) such that \( v E^{-1} = [v] \times [k] \). A path \( \pi \) in \( G \) from \( v \in V \) to \( v' \in V \) of length \( k \) is a word \( e_1, \ldots, e_k \) over \( E \) such that \( e_i \sigma = v_i, e_k \tau = v'_k \), and for all \( i \in [k] \) we have \( e_i \tau = e_{i+1} \sigma \). Let \( 1_v = e \) be the empty path from \( v \) to \( v \). The set of all paths in \( G \) from \( v \) to \( v' \) is denoted by \( \Pi(G)(v, v') \). Thus \( \Pi(G) \) is the path category of \( G \) with set of objects \( V \), set of morphisms \( \Pi(G)(v, v') \) and concatenation of words over \( E \) as composition.
3. Γ-Flowcharts

Following Elgot and Shepherdson Γ-flowcharts are defined to be multi-begin multi-exit directed ordered graphs partially labeled with elements of a one-sorted signature \( \Gamma = \{ \Gamma_i \}_{i < \omega} \). The elements of \( \Gamma_i \) may be viewed as "machine operations," the elements of \( \Gamma_i \) with \( i > 1 \) may be viewed as "tests" or "choice operations" having two or more exits (for an example, see Section 9).

**Definition 3.1.** Let \( \Gamma \) be a one-sorted signature, \( n, p < \omega \).

(a) A \( \Gamma \)-flowchart \( f \) with \( n \) begins and \( p \) exits (or \( \Gamma \)-flow from \( n \) to \( p \)) is a quadruple \( f = (s, b, \tau, l) \) where

- \( s \) is an ordinal number, the weight of \( f \),
- \( [s] \) is the set of all interior vertices of \( f \),
- \( b : [n] \to [s + p] \) is the begin function of \( f \),
- \( \tau : [s] \to [s + p]^* \) is the graph of \( f \),

and

\[ l : [s] \to \Gamma \]

is the labeling of \( f \)

such that for all \( i \in [s] \) we have \( il \in \Gamma_{|ir|} \).

The vertices \( 1b, \ldots, nb \) are the begins of \( f \), the (unlabeled) vertices \( s + 1, \ldots, s + p \) are the exits of \( f \).

(b) \( \text{Flow}_\Gamma(n, p) \) denotes the set of all \( \Gamma \)-flows from \( n \) to \( p \). \( \text{FFlow}_\Gamma(n, p) \) denotes the set of all finite \( \Gamma \)-flows from \( n \) to \( p \).

Thus the set of vertices of a \( \Gamma \)-flow \( f \) from \( n \) to \( p \) of weight \( s \) is the set \( [s + p] \). Every vertex \( i \in [s] \) is the source of \( k = |ir| \) edges, and for any \( j \in [k] \) the target of edge \( (i, j) \) is the vertex \( (ir)_j \), the \( j \)th symbol of the word \( it \). The exits are the only unlabeled vertices of \( f \). The \( i \)th begin will be denoted by \( Bi \)

\[ \downarrow \]

and the \( i \)th exit will be denoted by \( [Ei] \).

![Figure 1](image_url)
EXAMPLE 3.1. The \( \Gamma \)-flow \( f = (2, b, \tau, l) \in \text{Flo}_\Gamma (1, 1) \) with \( 1b = 1, \ 1\tau = 32, \ 2\tau = 1, \ 1l = \pi, \) and \( 2l = \gamma \) is depicted in Fig. 1. The \( \Gamma \)-flows as defined above essentially differ from the models chosen by Gallier [13, 15] and Burstall and Thatcher [8]. Their flowcharts are bipointed directed graphs the edges of which are labeled with operations and predicates. Furthermore Gallier's results depend on his specific choice of operations and predicates.

The next definition shows how to embed MAP and \( \Gamma \) into \( \text{Flo}_\Gamma \).

**DEFINITION 3.2.** (a) For each \( f \in \text{MAP}(n, p) \) let \( \hat{f} := (0, 0, p), (0, r) \in \text{Flo}_\Gamma (n, p) \) be the representation of \( f \) as a trivial \( \Gamma \)-flow.

(b) For each \( k < \omega \) and \( \gamma \in \Gamma_k \) let \( \hat{\gamma} := (1, b, \tau, l) \in \text{Flo}_\Gamma (1, k) \) with \( 1b = 1, \ 1\tau = 2 \cdots k + 1, \) and \( 1l = \gamma \) be the representation of \( \gamma \) as an atomic \( \Gamma \)-flow from 1 to \( k \).

The trivial and atomic \( \Gamma \)-flows are also called elementary. If the context is obvious, we usually write \( f \) and \( \gamma \) instead of \( \hat{f} \) and \( \hat{\gamma} \).

**EXAMPLE 3.2.** (a) Let \( f : [3] \to [4] \) be the following mapping: \( 1f = 2f = 2, \ 3f = 4 \). Then \( \hat{f} \) is the \( \Gamma \)-flow from 3 to 4 shown in Fig. 2.

(b) For all \( \gamma \in \Gamma_k \) \( \hat{\gamma} \) is the \( \Gamma \)-flow shown in Fig. 3.

The path category of the graph of a \( \Gamma \)-flow \( f \) is denoted by \( \Pi(f) \).

**DEFINITION 3.3.** Let \( f = (s, b, \tau, l) \in \text{Flo}_\Gamma (n, p) \).

(a) A vertex \( j \in [s + p] \) is accessible (in \( f \)) iff there is an \( i \in [n] \) such that \( \Pi(f)(ib, j) \neq \emptyset \).

(b) Let \( A_f := \{ j \in [s] \mid j \text{ accessible in } f \} \) be the set of all interior vertices accessible in \( f \).

(c) Let \( E_f := \{ j \in [p] \mid s + j \text{ accessible in } f \} \) be the set of all exits accessible in \( f \).
(d) \( f \) is almost accessible iff \( A_f = [s] \).

(e) \( f \) is accessible iff \( A_f = [s] \) and \( E_f = [p] \).

The set of all accessible \( f \)-flows from \( n \) to \( p \) will be denoted by \( \text{AFlo}_f(n,p) \). The set of all almost accessible \( f \)-flows from \( n \) to \( p \) will be denoted by \( \text{AAFFlo}_f(n,p) \). The corresponding sets of finite \( f \)-flows from \( n \) to \( p \) will be denoted by \( \text{AFFlo}_f(n,p) \) and \( \text{AAFFlo}_f(n,p) \).

Elgot and Shepherdson only consider accessible \( f \)-flows. But solving flowchart equations or looking for fixpoint semantics of recursive flowchart schemes one naturally arrives at inaccessible flowcharts [21].

The almost accessible \( f \)-flows generalize the “CACI-schemes” defined in [10]. Gallier [13,15] is concerned only with a subclass of \( \text{AFFlo}_f \) since he requires every vertex to be on a path from the begin to the exit.

\( f \)-flows \( f = (s, b, \tau, l) \) and \( g = (s', b', \tau', l') \) from \( n \) to \( p \) usually are considered to be “essentially equal” or “isomorphic” if they differ only with respect to a renaming of their interior vertices. If the accessible parts of \( f \) and \( g \) differ only with respect to a renaming, we call \( f \) and \( g \) “weakly isomorphic”.

**DEFINITION 3.4.** (a) \( f \) and \( g \) are isomorphic \((f \approx g)\) iff there is a bijection \( \phi: [s] \rightarrow [s'] \) such that

\[
\begin{align*}
(\text{i}) & \quad b \cdot (\phi + 1_p) = b', \\
(\text{ii}) & \quad l = \phi \cdot l', \\
(\text{iii}) & \quad \phi \cdot \tau' = \tau \cdot (\phi + 1_p)^*.
\end{align*}
\]

(b) \( f \) and \( g \) are weakly isomorphic \((f \sim g)\) iff there is a bijection \( \phi: A_f \rightarrow A_g \) such that for any extension \( \bar{\phi}: [s] \rightarrow [s'] \) of \( \phi \) we have

\[
\begin{align*}
(\text{i}) & \quad b \cdot (\bar{\phi} + 1_p) = b', \\
(\text{ii}) & \quad \phi \cdot l' = l_{|A_f}, \\
(\text{iii}) & \quad \phi \cdot \tau' = \tau_{|A_f} \cdot (\phi + 1_p)^*.
\end{align*}
\]

Thus weakly isomorphic almost accessible \( f \)-flows are always isomorphic. Since all the operations on \( f \)-flows defined in the next section preserve isomorphism we are only interested in isomorphism classes of \( f \)-flows. Therefore we often write \( f = g \) instead of \( f \approx g \).

For each \( f = (s, b, \tau, l) \in \text{Flo}_f(n,p) \) there is a weakly isomorphic \( f \)-flow with a minimal number of vertices. This minimal \( f \)-flow will be denoted \( f^m = (s^m, b^m, \tau^m, l^m) \) and may be defined as follows:

Let \( s^m := \min\{\alpha \in \text{Ord} \mid |[\alpha]| = |A_f|\} \), let \( \phi: A_f \rightarrow [s^m] \) be a bijection, and let \( \bar{\phi}: [s] \rightarrow [s^m] \) be an extension of \( \phi \). Define

\[
\begin{align*}
b^m & := b \cdot (\bar{\phi} + 1_p), \\
l^m & := \phi^{-1} \cdot l, \\
\tau^m & := \phi^{-1} \cdot \tau \cdot (\bar{\phi} + 1_p)^*.
\end{align*}
\]
Thus in $f^m$ all interior vertices are accessible, i.e., up to isomorphism $f^m$ is the minimal almost accessible $I$-flow weakly isomorphic to $f$. The weight of $f^m$ is always less or equal to $\omega$. The construction of $f^m$ is similar to the reduction operation of [13] which would further eliminate all the vertices not on a path to an exit. Since the inaccessible vertices of a $I$-flow $f$ cannot contribute to any computation of $f$ starting at a begin vertex, $f^m$ contains all the semantically relevant information. In this paper we thus restrict our attention to the class of almost accessible $I$-flows.

Next the two classes of well-structured flowcharts are defined which will be of special interest in the following sections.

**Definition 3.5.** (a) A $I$-flow $f$ of weight $s$ is acyclic iff the graph of $f$ contains no cycles, i.e., for any $i \in [s]$ $\Pi(f)(i,i)$ contains no nontrivial paths.

(b) If $F$ is a subclass of $\Gamma_{lo}$, then $F^{ac}$ denotes the class of all acyclic $I$-flows in $F$.

An acyclic $I$-flow has a rather simple structure since every path through $f$ can pass any vertex of $f$ at most once, i.e., there are no backedges [11].

In order to define the second class of well-structured flowcharts we need the notion of strongly connected sets and dominators of such sets.

**Definition 3.6.** Let $f$ be a $I$-flow of weight $s$.

(a) A subset $U \subseteq [s]$ is strongly connected iff for any $i, j \in U$ there is a path from $i$ to $j$ and vice versa.

(b) A vertex $j \in U$ is a dominator of $U$ iff every path from a begin vertex of $f$ into $U$ enters $U$ via $j$.

A dominator can be seen as a unique entry to a strongly connected set [11].

We are now able to define the class of reducible flowcharts.

**Definition 3.7.** (a) A $I$-flow $f$ is reducible iff every strongly connected set of vertices of $f$ has a dominator.

(b) If $F$ is a subclass of the class of all $I$-flows, then $F^{red}$ denotes the class of all reducible $I$-flows in $F$.

As an immediate consequence, we get

**Proposition 3.1.** For any class $F$ of $I$-flows $F^{ac}$ is a subclass of $F^{red}$. The elementary $I$-flows are acyclic.

4. Operations on Flowcharts

The algebraic characterization of flowcharts heavily depends on the definition of operations on $I$-flows. Elgot and Shepherdson have introduced several such
operations the most essential of which are composition, sum (which they call separated pairing), and scalar iteration. As an operation on almost accessible $I$-flows we further need the strong composition \cite{20}. Definitions of these operations and of the pairing of flowcharts are given below.

In order to facilitate the schematic description a $I$-flow $f$ from $n$ to $p$ is denoted by Fig. 4.

4.1. Composition

For all $n, p, q < \omega$ the composition of $I$-flows is a mapping

$$\cdot : \text{Flo}_I(n, p) \times \text{Flo}_I(p, q) \rightarrow \text{Flo}_I(n, q).$$

Let $f = (s, b, \tau, l) \in \text{Flo}_I(n, p)$, $g = (s', b', \tau', l') \in \text{Flo}_I(p, q)$. We get $f \cdot g \in \text{Flo}_I(n, q)$, the composition of $f$ and $g$, by identifying the $p$ exits of $f$ with the $p$ begins of $g$ (see Fig. 5). Formally $f \cdot g = (s'', b'', \tau'', l'')$ is defined by

$$s'' := s + s',$$
$$b'' := b \cdot (1_s + b'),$$
$$\tau'' := \langle \tau \cdot (1_s + b')^*, \tau' \cdot (1_s + b')^* \rangle,$$
$$l'' := \langle l, l' \rangle.$$

From this, one easily derives

**Proposition 4.1.** The composition of $I$-flowcharts is associative.
The operation of composition provides a characterization of almost accessible \( I \)-flows in terms of accessible and trivial flows.

**Proposition 4.2.** For any \( f \in AAFlo_\omega(n, p) \) there exists an \( r < \omega \), a \( I \)-flow \( g \in AFlo_\omega(n, r) \) and an injective mapping \( \alpha : [r] \to [p] \) such that \( f = g \cdot \alpha \). For all \( r' < \omega \), \( g' \in AFlo_\omega(n, r') \) and \( \alpha' : [r'] \to [p] \) satisfying \( f = g' \cdot \alpha' \) we have \( r = r' \) and there exists a permutation \( \pi : [r] \to [r'] \) such that \( g = g' \cdot \pi \) and \( \alpha' = \pi \cdot \alpha \).

**Proof.** See [20].

Obviously \( AFlo_\omega \) is closed under composition. But this is not true for \( AAFlo_\omega \): Vertices which are accessible in \( g \) from begin \( i \) need be accessible no longer in \( f \cdot g \), if exit \( i \) of \( f \) is not accessible. Thus a new operation is needed.

**4.2. Strong Composition**

For all \( n, p, q < \omega \) the strong composition of \( I \)-flows is a mapping

\[ * : Flo_\omega(n, p) \times Flo_\omega(p, q) \to AAFlo_\omega(n, q). \]

Let \( f \) and \( g \) be as in 4.1. Then \( f \cdot g \in AAFlo_\omega(n, q) \), the strong composition of \( f \) and \( g \), is defined by \( f \cdot g := (f \cdot g)^m \), i.e., \( f \cdot g \) is the minimal \( I \)-flow weakly isomorphic to \( f \cdot g \). Obviously \( AAFlo_\omega \) is closed under strong composition. Furthermore

**Proposition 4.3.** The strong composition of \( I \)-flowcharts is associative.

**Proposition 4.4.** If \( f \in AFlo_\omega(n, p) \) and \( g \in AAFlo_\omega(p, q) \), then \( f \cdot g = f \cdot g \).

**4.3. Sum**

For all \( n, n', p, p' < \omega \) the sum of \( I \)-flows ("separated sum" [7], "separated pairing" [11]) is a mapping

\[ + : Flo_\omega(n, p) \times Flo_\omega(n', p') \to Flo_\omega(n + n', p + p'). \]

Let \( f = (s, b, i, l) \in Flo_\omega(n, p) \), \( g = (s', b', i', l') \in Flo_\omega(n', p') \). Then we get \( f + g \in Flo_\omega(n + n', p + p') \), the sum of \( f \) and \( g \), by laying \( f \) and \( g \) disjointly side by side (see Fig. 6). Formally \( f + g = (s'', b'', i'', l'') \) is defined by

\[ s'' := s + s', \]
\[ b'' := \langle b \cdot \phi, b' \cdot \psi \rangle, \]
\[ i'' := \langle i \cdot \phi^*, i' \cdot \psi^* \rangle, \]
\[ l'' := \langle l, l' \rangle, \]

where \( \phi = \langle t_1, t_2, t_3, t_4 \rangle \) and \( \psi = \langle t_1', t_2', t_3', t_4' \rangle \).

Obviously \( AFlo_\omega \) and \( AAFlo_\omega \) are closed under sum and we have
PROPOSITION 4.5. The sum of $\Gamma$-flowcharts is associative.

PROPOSITION 4.6. For all $f_1 \in \text{Flo}_r(n_1, p_1)$, $f_2 \in \text{Flo}_r(n_2, p_2)$, $g_1 \in \text{Flo}_r(p_1, q_1)$, and $g_2 \in \text{Flo}_r(p_2, q_2)$ we have

(a) $(f_1 + f_2)^m = f_1^m + f_2^m$,
(b) $(f_1 + f_2) \cdot (g_1 + g_2) = (f_1 \cdot g_1) + (f_2 \cdot g_2)$,
(c) $(f_1 + f_2) \ast (g_1 + g_2) = (f_1 \ast g_1) + (f_2 \ast g_2)$.

Proof. (a) Trivial.

(b) The equation may be visualized as in Fig. 7. The formal proof is easy but rather tedious.

(c) 

\[
(f_1 + f_2) \ast (g_1 + g_2) = ((f_1 + f_2) \cdot (g_1 + g_2))^m \\
= ((f_1 \cdot g_1) + (f_2 \cdot g_2))^m \\
= (f_1 \cdot g_1)^m + (f_2 \cdot g_2)^m \\
= (f_1 \ast g_1) + (f_2 \ast g_2).
\]
4.4. Pairing

For all \( n, m, p < \omega \) the pairing of \( \Gamma \)-flows ("source-pairing" [10], "coalesced sum" [7]) is a mapping

\[
\langle , \rangle : \text{Flo}_\Gamma(n, p) \times \text{Flo}_\Gamma(m, p) \rightarrow \text{Flo}_\Gamma(n + m, p).
\]

Let \( f \in \text{Flo}_\Gamma(n, p) \), \( g \in \text{Flo}_\Gamma(m, p) \). We get \( \langle f, g \rangle \in \text{Flo}_\Gamma(n + m, p) \), the pairing of \( f \) and \( g \), by laying \( f \) and \( g \) disjointly side by side and identifying the exits (see Fig. 8). Formally we define \( \langle f, g \rangle := (f + g) \cdot \langle 1_\Gamma, 1_\Gamma \rangle \). Obviously \( \text{AFlo}_\Gamma \) and \( \text{AAFlo}_\Gamma \) are closed under pairing. One easily verifies

**Proposition 4.7.** The pairing of \( \Gamma \)-flowcharts is associative.

**Proposition 4.8.** (a) For all \( f \in \text{Flo}_\Gamma(n, p) \), \( g \in \text{Flo}_\Gamma(n', p') \)

\[
(f + g) = \langle f \cdot t_{f(1)}^{p, p'}, g \cdot t_{g(2)}^{p, p'} \rangle.
\]

(b) For all \( i \in [2] \) \( f_i \in \text{AAFlo}_\Gamma(n_i, p) \)

\[
i_i^{n_i \rightarrow n_2} \circ \langle f_1, f_2 \rangle = f_i.
\]

**Proof.** (a)

\[
\langle f \cdot t_{f(1)}^{p, p'}, g \cdot t_{g(2)}^{p, p'} \rangle = (f \cdot t_{f(1)}^{p, p'} + g \cdot t_{g(2)}^{p, p'}) \cdot \langle 1_{p+p'}, 1_{p+p'} \rangle
\]

\[
= (f + g) \cdot (t_{f(1)}^{p, p'} + t_{g(2)}^{p, p'}) \cdot \langle 1_{p+p'}, 1_{p+p'} \rangle
\]

\[
= (f + g) \cdot \langle t_{f(1)}^{p, p'}, t_{g(2)}^{p, p'} \rangle
\]

\[
= (f + g).
\]

(b) The equation follows from the fact that only the vertices of \( f_i \) can be accessible in \( i_i^{n_i \rightarrow n_2} : \langle f_1, f_2 \rangle \).

4.5. Scalar Iteration

To define the scalar iteration of a \( \Gamma \)-flow a distinguished nullary operator \( \bot \) ("undefined") is needed. Therefore from now on we assume \( \Gamma \) to contain this special operator.

![Figure 8](image-url)
For all $p < \omega$ the scalar iteration of a $l$-flow is a mapping

$$^+ : \text{Flo}_r(1, p+1) \rightarrow \text{Flo}_r(1, p)$$

Let $f = (s, b, t, l) \in \text{Flo}_r(1, p+1)$. We get $f^+ \in \text{Flo}_r(1, p)$, the scalar iteration of $f$, by identifying exit $p+1$ with the begin, unless exit $p+1$ is the begin of $f$ (see fig. 9). Formally we have to consider two cases:

(a) If $lb = s + p + 1$, then $f^+ = (s', b', t', l')$ is defined by

$$s' := s + 1,$$

$$lb' := s + 1,$$

for all $j \in [s + 1]$,

$$jt' := j \cdot \phi^*, \quad \text{if } j \in [s],$$

$$:= e, \quad \text{otherwise},$$

for all $j \in [s + 1]$,

$$jl' := jl, \quad \text{if } j \in [s],$$

$$:= \bot, \quad \text{otherwise},$$

where $\phi := [s + p + 1] \rightarrow [s + 1 + p]$ is defined by

for all $j \in [s + p + 1]$,

$$j\phi := j,$$

$$:= j + 1, \quad \text{if } j \in s + [p],$$

$$:= s + 1, \quad \text{otherwise}.$$

(b) If $lb \neq s + p + 1$, then define $f^+ := (s, b, t', l)$, where $t' := t \cdot \phi^*$ and $\phi : [s + p + 1] \rightarrow [s + p]$ is defined by: for all $j \in [s + p + 1]$,

$$j\phi := j,$$

$$:= 1, \quad \text{otherwise}.$$
This definition corresponds to the one chosen in [11, 12]. The vector iteration defined in [7] and [10] is not needed for the purpose of this paper.

One easily derives

**Proposition 4.9.** (a) \((1, i)^{++} = \bot\).

(b) For all \(f \in \text{Flo}_r(1, p)\), \((f \cdot i_{(1)})^{++} = f\).

(c) For all \(f \in \text{Flo}_r(1, p + 2)\), \(f^{++} = (f \cdot (1_p + \langle 1, 1, i \rangle))^{++}\).

(d) For all \(f \in \text{Flo}_r(1, p + 1)\), \(g \in \text{Flo}_r(p, q)\), \(f^{++} \cdot g = (f \cdot (g + 1))^{++}\).

*Proof.* See [20].

Summarizing we get the following statements:

\[
(\text{Flo}_r, \cdot, +, 1_0), (\text{AAFlo}_r, *, +, 1_0) \text{ and } (\text{AFlo}_r, \cdot, +, 1_0) \text{ are strict monoidal categories with set of objects } \mathbb{N}_0. \tag{4.1}
\]

\[
(\text{Flo}_r, \cdot, +, 1_0) \text{ and } (\text{AAFlo}_r, *, +, 1_0) \text{ are extensions of } (\text{AFlo}_r, \cdot, +, 1_0) \text{ and of } (\text{MAP}, \cdot, +, 1_0). \tag{4.2}
\]

\[
(\text{AFlo}_r, \cdot, +, 1_0) \text{ is an extension of } (\text{SUR}, \cdot, +, 1_0) \text{ but not of } (\text{MAP}, \cdot, +, 1_0). \tag{4.3}
\]

Since composition, strong composition, and sum respect acyclicity and reducibility, we get the same statements as (4.1)–(4.3) for the subclasses \(\text{Flo}_{r\text{c}}\), \(\text{AFlo}_{r\text{c}}\), and \(\text{AAFlo}_{r\text{c}}\) as well as for \(\text{Flo}_{r\text{d}}\), \(\text{AFlo}_{r\text{d}}\), and \(\text{AAFlo}_{r\text{d}}\).

In view of Proposition 4.2 we can also state the following easy extensions of theorems stated in [11] for accessible \(\Gamma\)-flows:

**Theorem 4.1.** \(\text{AAFlo}_{r\text{c}}\) is the least category containing the elementary \(\Gamma\)-flows and being closed under strong composition and sum.

**Theorem 4.2.** \(\text{AAFFlo}_{r\text{d}}\) is the least category containing the elementary \(\Gamma\)-flows and being closed under strong composition, sum, and scalar iteration.

5. Flow Theories

In this section we define the category of flow theories over MAP in which the class of almost accessible reducible flowcharts later is shown to be a free object. We begin with a slightly more general notion, the flow theories over SUR [12].

**Definition 5.1.** (a) A flow theory over SUR is a strict monoidal category \((T, \cdot, +, 1_0)\) extending SUR which satisfies the block permutation axiom

\[
\forall f_1 \in T(n_1, p_1), \forall f_2 \in T(n_2, p_2), \ (f_1 + f_2) \cdot \pi(p_1, p_2) = \pi(n_1, n_2) \cdot (f_2 + f_1).
\]
(b) Let $T$ and $T'$ be flow theories over SUR. A mapping $\phi : T \to T'$ is called an FIS-\textit{morphism} iff we have for all $f_1, f_2 \in T$,
\[(f_1 \cdot f_2) \phi = f_1 \phi \cdot f_2 \phi, \quad (f_1 + f_2) \phi = f_1 \phi + f_2 \phi,\]
and for all $f \in \text{SUR}$
\[f \phi = f.\]

(c) Let $(\text{FlS}, \cdot )$ denote the category having flow theories over SUR as objects and FIS-morphisms as morphisms, where $\cdot$ is the usual composition of mappings.

The block permutation axiom is easily shown to be valid for $I$-flows [20]. Thus the statements at the end of the last section imply

\begin{proposition}
Flo$_r$, AAFlo$_r$, and AFlo$_r$ are flow theories over SUR.
\end{proposition}

A reasonable requirement of morphisms of $I$-flows is to be the identity on trivial $I$-flows. Thus there cannot exist a morphism from AAFlo$_r^{fd}$ into AFlo$_r^{fd}$, i.e., AAFlo$_r^{fd}$ cannot be a free object of FlS. Therefore we define the more suitable flow theories over MAP [20]:

\begin{definition}
(a) A flow theory over MAP is a strict monoidal category $(T, \cdot, +, 1_0)$ extending MAP and satisfying the block permutation axiom (as in Definition 5.1) and the injection axiom
\[\forall i \in [2] \forall f_1 \in T(n_1, p_i) \quad t_i^{n_1,n_2} \cdot (f_1 + f_2) = f_1 \cdot t_i^{n_1,n_2}.\]

(b) Let $T$ and $T'$ be flow theories over MAP. A mapping $\phi : T \to T'$ is called an FIM-\textit{morphism} iff for all $f_1, f_2 \in T$ and for all $f \in \text{MAP}$ we have
\[(f_1 \cdot f_2) \phi = f_1 \phi \cdot f_2 \phi, \quad (f_1 + f_2) \phi = f_1 \phi + f_2 \phi, \text{ and } f \phi = f.\]

(c) Let $(\text{FIM}, \cdot )$ denote the category having flow theories over MAP as objects and FIM-morphisms as morphisms, where $\cdot$ is the usual composition of mappings.

The definition implies that every flow theory over MAP is a flow theory over SUR, i.e., FIM is a subcategory of FlS. Since the injection axiom is not valid in Flo$_r$, and since AFlo$_r$ does not extend MAP, FIM is a proper subcategory of FlS.

The following proposition illustrates the richness of FIM:

\begin{proposition}
Every algebraic theory extending MAP is a flow theory over MAP.
\end{proposition}

\textit{Proof}: In an algebraic theory $T$ as defined in [6] we have for all $f_1 \in T(n_1, p_1)$ and $f_2 \in T(n_2, p_2)$,
\[(f_1 + f_2) \cdot \pi(p_1, p_2)\]
\[= (f_1 \cdot \pi_{i_{(1)}^{P_1}}^{p_1}, f_2 \cdot \pi_{i_{(2)}^{P_2}}^{p_2}) \cdot \pi(p_1, p_2)\]
\[= (f_1 \cdot \pi_{i_{(1)}^{P_1}}^{p_1}, f_2 \cdot \pi_{i_{(2)}^{P_2}}^{p_2} \cdot \pi(p_1, p_2))\]
\[= (f_1 \cdot \pi_{i_{(2)}^{P_1}}^{p_1}, f_2 \cdot \pi_{i_{(1)}^{P_2}}^{p_2})\]
\[= \langle \pi_{i_{(2)}^{P_1}}^{n_1}, \pi_{i_{(1)}^{P_2}}^{n_2}, \langle f_2 \cdot \pi_{i_{(2)}^{P_1}}^{p_1}, f_1 \cdot \pi_{i_{(1)}^{P_2}}^{p_2} \rangle \rangle \]
\[= \langle \pi_{i_{(2)}^{P_1}}^{n_1}, \pi_{i_{(1)}^{P_2}}^{n_2}, \langle f_2 \cdot \pi_{i_{(2)}^{P_1}}^{p_1}, f_1 \cdot \pi_{i_{(1)}^{P_2}}^{p_2} \rangle \rangle \]
\[= \pi(n_1, n_2) \cdot (f_2 + f_1).\]

The injection axiom is trivially satisfied. Finally, every algebraic theory is a strict monoidal category.  

From Proposition 4.8 we know the injection axiom to be valid in (AAFlo, +, 1). However, in general, the equation

\[\langle \pi_{i_{(2)}^{P_1}}^{n_1}, \pi_{i_{(1)}^{P_2}}^{n_2}, \langle f_2 \cdot \pi_{i_{(2)}^{P_1}}^{p_1}, f_1 \cdot \pi_{i_{(1)}^{P_2}}^{p_2} \rangle \rangle \]

does not hold: For \( i \in [2] \) let \( f_i \in AAFlo \langle (n_i, p) \rangle \) and \( g \in AAFlo \langle (p, q) \rangle \). If \( g \) is nontrivial, we have

\[\pi_{i_{(1)}^{P_1}}^{n_1}, \pi_{i_{(2)}^{P_2}}^{n_2} \cdot (f_1, f_2) \cdot g = f_1 \cdot g, \quad \pi_{i_{(1)}^{P_1}}^{n_1}, \pi_{i_{(2)}^{P_2}}^{n_2} \cdot (f_1, f_2) \cdot g = f_2 \cdot g.\]

Hence

\[\langle \pi_{i_{(1)}^{P_1}}^{n_1}, \pi_{i_{(2)}^{P_2}}^{n_2}, \langle f_1, f_2 \rangle \cdot g \rangle = \langle f_1 \cdot g, f_2 \cdot g \rangle \neq \langle f_1, f_2 \rangle \cdot g.\]

This implies

**Proposition 5.3.** (AAFlo, +, 1) is a flow theory over MAP which is not an algebraic theory.

We shall need further the following subcategories of FlM:

**Definition 5.3.** (a) A flow theory \( T \) over MAP is pointed iff \( T(1, 0) \) contains a distinguished element \( \perp \).

(b) A PFIM-morphism is an FlM-morphism preserving \( \perp \).

(c) Let PFIM denote the subcategory of FlM having pointed flow theories over MAP as objects and PFIM-morphisms as morphisms.

(d) A mapping from \( I \) into a pointed flow theory is strict iff it preserves the distinguished element \( \perp \).

**Definition 5.4.** (a) A pointed flow theory \( T \) over MAP is a scalar iteration flow theory over MAP iff \( T \) is equipped with an operation \( ^\dagger \), the scalar iteration, satisfying

\[\forall p \in \mathbb{N}, ^\dagger : T(1, p + 1) \rightarrow T(1, p),\]
\[\forall f \in T(0, p), (f + 1)^\dagger = f + \perp,\]
(13) \( \forall f \in T(1, p), (f \cdot t_{(1)}^{p})^{\dagger} = f \),

(14) \( \forall f \in T(1, p + 2), f^{\dagger \dagger} = (f \cdot (1_p + \langle 1_1, 1_1 \rangle))^{\dagger} \),

(15) \( \forall f \in T(1, p + 1), \forall g \in T(p, q), f^{\dagger} \cdot g = (f \cdot (g + 1_1))^{\dagger} \).

(b) An IFIM-morphism is a PFIM-morphism respecting scalar iteration, i.e.,

\[ f^{\dagger} \phi = (f \phi)^{\dagger} \]

(c) Let IFIM denote the subcategory of PFIM having scalar iteration flow theories over MAP as objects and IFIM-morphisms as morphisms.

Since scalar iteration on flowcharts preserves reducibility, Proposition 4.9 implies

**Proposition 5.4.** AAFlo and AAFlo\(_{\text{red}}\) are scalar iteration flow theories over MAP.

Furthermore we have [20]

**Proposition 5.5.** Every \( \omega \)-continuous algebraic theory extending MAP is a scalar iteration flow theory over MAP.

**Proof.** Let \( T \) be an \( \omega \)-continuous algebraic theory [6] extending MAP. For all \( n, p < \omega \) \( T(n, p) \) contains a least element \( \bot \), i.e., \( T \) is pointed.

The scalar iteration may be defined on \( T \) as follows [2]: For \( p < \omega \) and \( f \in T(1, p + 1) \) let the sequence \( (f^{(i)})_{i \in \omega} \) be defined by \( f^{(0)} := \bot_{1_p} \) and for all \( i < \omega \)

\[ f^{(i+1)} := f \cdot \langle 1_p, f^{(i)} \rangle. \]

Obviously we have for all \( i < \omega \)

\[ f^{(i)} \leq f^{(i+1)}. \]

Define \( f^{\dagger} := \bigcup_{i < \omega} f^{(i)} \). It is well known [2] that \( f^{\dagger} \) is the least solution of the equation \( n = f \cdot \langle 1_p, \eta \rangle \) as well as of the inequation \( f \cdot \langle 1_p, \eta \rangle \leq \eta \).

Conditions (11)–(13) of Definition 5.4 are easily verified. In order to show condition (14) we need the following lemma, which may be proven by induction.

**Lemma 5.1.**

(a) \( \forall f \in T(1, p + 1), \forall i < \omega, f^{(i+1)} = f \cdot \langle \langle t_{(1)}^{p}, f \rangle \rangle^i \cdot \langle 1_p, \bot_{1_p} \rangle \).

(b) \( \forall f \in T(1, p + 2), \forall \eta \in T(1, p) \forall i < \omega, \eta = f \cdot \langle 1_p, \eta, \eta \rangle \Rightarrow \eta = f \cdot \langle 1_p, \eta, \eta \rangle \).

(c) \( \forall f \in T(1, p + 2), \forall i < \omega, (f \cdot (1_p + \langle 1_1, 1_1 \rangle))^{(i+1)} = f \cdot \langle \langle t_{(1)}^{p}, f, f \rangle \rangle^i \cdot \langle 1_p, \bot_{1_p}, \bot_{1_p} \rangle \).

Now for any \( f \in T(1, p + 2) \) we have

\[ f^{\dagger \dagger} = f^{\dagger} \cdot \langle 1_p, f^{\dagger \dagger} \rangle \]

\[ = f \cdot \langle 1_p, f^{\dagger} \rangle \cdot \langle 1_p, f^{\dagger \dagger} \rangle \]

\[ = f \cdot \langle 1_p, f^{\dagger \dagger}, f^{\dagger} \rangle \cdot \langle 1_p, f^{\dagger \dagger} \rangle \]

\[ = f \cdot \langle 1_p, f^{\dagger \dagger}, f^{\dagger} \rangle \]

\[ = f \cdot (1_p + \langle 1_1, 1_1 \rangle) \cdot \langle 1_p, f^{\dagger \dagger} \rangle, \]

implying \( (f \cdot (1_p + \langle 1_1, 1_1 \rangle))^{\dagger} \leq f^{\dagger \dagger} \).
Conversely by means of Lemma 5.1 we can show that for any \( \eta \in T(1, p) \) with 
\[ \eta = f \cdot \langle 1_p, \eta, \eta \rangle \]
we have \( f^{++} \leq \eta \). Furthermore we get
\[
(f \cdot (1_p + \langle 1_1, 1_1 \rangle))^+ = \bigcup_{i < \omega} (f \cdot (1_p + \langle 1_1, 1_1 \rangle))^{(i)}
\]
\[
= \bigcup_{i < \omega} \left( f \cdot (1_p + \langle 1_1, 1_1 \rangle) \right)^{(i+2)}
\]
\[
= \bigcup_{i < \omega} \left( f \cdot \left( \langle t^{p,2}_{11}, f, f \rangle \right)^{(i+1)} \cdot \langle 1_p, 1_p, 1_p \rangle \right)
\]
\[
= f \cdot \bigcup_{i < \omega} \langle 1_p, f \cdot \left( \langle t^{p,2}_{11}, f, f \rangle \right)^{(i)} \cdot \langle 1_p, 1_p, 1_p \rangle \rangle
\]
\[
= f \cdot \bigcup_{i < \omega} \left( f \cdot \left( \langle 1_p + \langle 1_1, 1_1 \rangle \rangle \right)^{(i+1)} \cdot (f \cdot (1_p + \langle 1_1, 1_1 \rangle))^{(i+1)} \right)
\]
Hence \( f^{++} \leq (f \cdot (1_p + \langle 1_1, 1_1 \rangle))^+ \).

Condition (15) is verified by noticing that for all \( i < \omega \)
\[
(f \cdot (g + 1_1))^{(i)} = f^{(i)} \cdot g.
\]

6. FREEDOM RESULTS FOR FINITE FLOWCHARTS

In [12] Elgot and Shepherdson show AFFloF to be freely generated from \( \Gamma \) by composition and sum, and AFFloF(r) to be freely generated from \( \Gamma \) by composition, sum, and scalar iteration. In this section their results are extended to almost accessible \( \Gamma \)-flowcharts.

We begin by defining the notion of a factorization of a \( \Gamma \)-flow (following [12]).

**Definition 6.1.** Let \( f \in \text{AFFlo}^F(n, p) \).

(a) An alternating factorization of \( f \) is defined as a triple \((\alpha, A, g)\), where \( \alpha \in \text{SUR}(n, r + q) \), \( A : [r] \to \Gamma \) and \( g \in \text{AFFlo}^F(t + q, p) \) such that \( f = \alpha * (\Sigma A + 1_q) * g \) (\( \Sigma A := 1A + 2A + \cdots + rA \)).

(b) A normal factorization of \( f \) is an alternating factorization of \( f \) in which \( r + q \) is minimum and among all such minimum \( r + q \) \( r \) is maximum.

In view of Proposition 4.2 we could have phrased this definition using the notion of alternating factorization as defined in [12].

**Definition 6.1'.** Let \( f \in \text{AFFlo}^F(n, p) \), \( f' \in \text{AFFlo}^F(n, r) \), \( \beta \in \text{INJ}(r, p) \) such that \( f = f' \cdot \beta \). An alternating factorization of \( f \) then is a triple \((\alpha, A, g * \beta)\) such that \((\alpha, A, g)\) is an alternating factorization of the accessible \( \Gamma \)-flow \( f' \).
Thus statements on alternating factorizations of accessible acyclic \(I\)-flows directly imply statements on alternating factorizations of almost accessible acyclic \(I\)-flows. As shown for \(\text{AFFlo}_I\) in [12], we thus have

**Proposition 6.1.** Up to interspersed permutations every acyclic \(I\)-flow in \(\text{AAFFlo}_I\) has a unique normal factorization.

**Proof.** The extension to infinite \(I\)-flows is possible since the proof of the corresponding result in [12] does not make use of the weight of a \(I\)-flow. ■

As mentioned above, Elgot and Shepherdson show \(\text{AFFlo}_I^{\text{ec}}\) to be a free object of \(\text{FIS}\). We thus have

**Theorem 6.1.** If \(T\) is a flow theory over \(\text{SUR}\), then every mapping \(\phi_0 : I \to T\) has a unique extension to an \(\text{FIS}\)-morphism \(\phi : \text{AFFlo}_I^{\text{ec}} \to T\).

This theorem is quite useful in deriving an "extension lemma" for \(\text{AAFFlo}_I^{\text{ec}}\):

**Lemma 6.1.** If \((T, \cdot)\) is a flow theory over \(\text{MAP}\), every mapping \(\phi_0 : I \to T\) has a unique extension to a mapping \(\phi : \text{AFFlo}_I^{\text{ec}} \to T\) such that for all \(f \in \text{AFFlo}_I^{\text{ec}}, f' \in \text{AFFlo}_I^{\text{ec}},\) and \(\alpha \in \text{INJ}\) such that \(f = f' * \alpha\) we have \(f\phi = f'\phi \cdot \alpha\).

**Proof.** Since every flow theory over \(\text{MAP}\) is a flow theory over \(\text{SUR}\) Theorem 6.1 implies that \(\phi_0\) has a unique extension to an \(\text{FIS}\)-morphism \(\phi : \text{AFFlo}_I^{\text{ec}} \to T\).

If we have \(f_1, f_2 \in \text{AFFlo}_I^{\text{ec}},\) \(\alpha_1, \alpha_2 \in \text{INJ}\) such that \(f = f_1 * \alpha_1 = f_2 * \alpha_2\) there exists a permutation \(\pi\) such that \(f_1 = f_2 * \pi\) and \(\alpha_2 = \pi \cdot \alpha_1\). Thus

\[
f_1\phi \cdot \alpha_1 = (f_2 * \pi)\phi \cdot \alpha_1 = f_2\phi \cdot \pi \cdot \alpha_1 = f_2\phi \cdot \alpha_2.
\]

Therefore the anticipated extension of \(\phi_0\) is uniquely defined by \(f\phi := f_1\phi \cdot \alpha_1\). ■

In order to show the mapping \(\phi : \text{AAFFlo}_I^{\text{ec}} \to T\) to be an \(\text{FIM}\)-morphism we need

**Lemma 6.2.** If \((T, \cdot)\) is a flow theory over \(\text{MAP}\), \(\phi_0 : I \to T\) a mapping and \(\phi : \text{AAFFlo}_I^{\text{ec}} \to T\) the unique extension of \(\phi_0\), then we have

\[
\begin{align*}
(1) & \quad \forall f \in \text{AFFlo}_I^{\text{ec}}(n, p), \forall g \in \text{AFFlo}_I^{\text{ec}}(p, q), (f * g) \phi = f\phi \cdot g\phi, \\
(2) & \quad \forall \alpha \in \text{INJ}(n, p), \forall f \in \text{AFFlo}_I^{\text{ec}}(p, q), (\alpha * f) \phi = \alpha \cdot f\phi, \\
(3) & \quad \forall f \in \text{AFFlo}_I^{\text{ec}}(n, p), \forall \alpha \in \text{INJ}(p, q), (f * \alpha) \phi = f\phi \cdot \alpha.
\end{align*}
\]

**Proof.** Statement (1) is a simple consequence of Proposition 4.2 and of the fact that \(\phi\) restricted to \(\text{AFFlo}_I^{\text{ec}}\) is an \(\text{FIS}\)-morphism. Statement (3) is almost trivial since for any \(f' \in \text{AFFlo}_I^{\text{ec}},\) \(\alpha, \beta \in \text{INJ}\) we have \((f' * \beta * \alpha) \phi = (f' * (\beta * \alpha)) \phi = f'\phi \cdot \beta \cdot \alpha - (f' * \beta) \phi \cdot \alpha\). The proof of statement (2) proceeds by induction on the length of the normal factorization of \(f\). Let \(fl_n\) be defined as follows:
(i) $f \in \text{MAP}$ implies $f_{l_N} := 0$.

(ii) If $f \in \text{MAP}$ and if $(\beta, A, g)$ is a normal factorization of $f$, then $f_{l_N} := 1 + g_{l_N}$.

If $f_{l_N} = 0$, we trivially have $(\alpha * f) \phi = \alpha * f \phi = \alpha * f$. Assume $f_{l_N} > 0$ and suppose statement (2) to be true for every $f'$ such that $f'_{l_N} < f_{l_N}$. Let $(\beta, A, g)$ be a normal factorization of $f$. Then

$$\alpha * f = \alpha * (\beta * (\Sigma A + 1_q) * g)$$

$$= (\alpha * \beta) * (\Sigma A + 1_q) * g.$$  

It is a simple exercise to prove that the injection and the block permutation axioms imply the existence of $\sigma \in \text{SUR}$, $B : [r'] \to T$, $q' < \omega$, and $\gamma \in \text{INJ}$ such that

$$(\alpha * \beta) * (\Sigma A + 1_q) * g = \sigma * (\Sigma B + 1_{q'}) * \gamma * g.$$  

Therefore

$$(\alpha * f) \phi = ((\sigma * (\Sigma B + 1_{q'})) * (\gamma * g)) \phi$$

$$= (\sigma * (\Sigma R + 1_{q'})) \phi * (\gamma * g) \phi$$

$$= \sigma \cdot (\Sigma B \phi + 1_{q'}) \cdot \gamma \cdot g \phi$$

and

$$\alpha * f \phi = \alpha \cdot (\beta * (\Sigma A + 1_q) * g) \phi$$

$$= \alpha \cdot (\beta * (\Sigma A \phi + 1_{q}) * g \phi)$$

$$= \sigma \cdot (\Sigma B \phi + 1_{q'}) \cdot \gamma \cdot g \phi$$

which completes the proof. 

**Theorem 6.2.** The mapping $\phi : \text{AFFlo}^{\text{uc}}_T \to T$ given in Lemma 6.1 is an $\text{FlM}$-morphism.

**Proof.** For $f \in \text{MAP}$ we trivially have $f \phi = f$. For any $f \in \text{AFFlo}^{\text{uc}}_T(n, p)$, $g \in \text{AFFlo}^{\text{uc}}_T(p, q)$ there exist $f', g' \in \text{AFFlo}^{\text{uc}}_T$ and $\alpha, \beta \in \text{INJ}$ such that $f = f' * \alpha$ and $g = g' * \beta$. Hence

$$(f * g) \phi = (f' * ((\alpha * g') * \beta)) \phi$$

$$= f' \phi \cdot (\alpha * g') \phi \cdot \beta$$

$$= f' \phi \cdot \alpha \cdot g' \phi \cdot \beta$$

$$= f \phi \cdot g \phi.$$  

The proof that for any $f \in \text{AFFlo}^{\text{uc}}_T(n_1, p_1)$ and $g \in \text{AFFlo}^{\text{uc}}_T(n_2, p_2)$ we have $(f + g) \phi = f \phi + g \phi$ is left to the reader. 

Summarizing we have derived the freeness of $\text{AFFlo}^{\text{uc}}_T$ over $T$. 


The crucial idea in showing a corresponding freeness result for $\text{AAFFlo}^\text{red}_I$ lies in transforming a reducible $I$-flow $f$ into an "isomorphic" acyclic flowchart by decomposing $f$ into components.

**Definition 6.2.** Let $f = (s, b, r, l) \in \text{AAFFlo}^\text{red}_I(n, p)$.

(a) A component of $f$ is a maximally strongly connected subset of $[s + p]$.

(b) A vertex $j$ of a component $C$ of $f$ is called begin of $C$ iff $j$ is the dominator of $C$.

(c) A vertex $j \in [s + p]$ is called exit of the component $C$ iff $j \not\in C$ and there are $i \in C$ and $u, v \in [s + p]^*$ such that $it = uvb$.

(d) A component $C$ of $f$ is called trivial iff $C$ contains an exit of $f$.

Note that any component containing an exit is a one-element set. Every component $C$ of $f$ defines a $I$-flow $\hat{C} \in \text{AFFlo}^\text{red}_I$, the component flow of $C$.

**Definition 6.3.** Let $C \subseteq [s + p]$ be a nontrivial component of $f = (s, b, r, l) \in \text{AAFFlo}^\text{red}_I(n, p)$. Define $E_C := \{j \in [s + p] | j \text{ is an exit of } C\}$, $r := |C|$, $q := |E_C|$. Let $j \in C$ be the begin of $C$.

(a) Let $\phi : C \rightarrow [r]$, $\psi : E_C \rightarrow [q]$ be bijections, and let $\bar{\phi} : [s] \rightarrow [r]$ and $\bar{\psi} : [p] \rightarrow [q]$ be extensions of $\phi$ and $\psi$. Then the component flow $C(\phi, \psi) = (r, b_C, r_C, l_C) \in \text{AFFlo}^\text{red}_I(1, q)$ is defined by

$$
\begin{align*}
1b_c & := j\phi, \\
r_c & := \phi^{-1} \cdot r \cdot (\bar{\phi} + \bar{\psi})^*, \\
l_c & := \phi^{-1} \cdot l.
\end{align*}
$$

(b) If $\phi' : C \rightarrow [r]$ and $\psi' : E_C \rightarrow [q]$ are also bijections, then $C(\phi, \psi)$ and $C(\phi', \psi')$ are called component equivalent.

(c) Define the component flow $\hat{C}$ as a fixed representative of the class of all component equivalent component flows of $C$.

If $C$ is trivial, define $\hat{C} := 1$. This is the only trivial component flow.

The definition is restricted to finite $I$-flows since otherwise we should have components and thus component flows with a possibly infinite number of exits. From the definition we see that different components can define identical component flows. Isomorphic component flows are always component equivalent. The converse is not true.

For any $f \in \text{AAFFlo}^\text{red}_I$ define $C_f := \{\hat{C} | C \text{ is a nontrivial component of } f\}$.

Obviously we have for all $f, g \in \text{AAFFlo}^\text{red}_I$

$$
C_{f+g} = C_f \cup C_g \supseteq C_{f \circ g}.
$$
Furthermore define $\mathcal{R}_r$ as the set of all component flows defined by components of reducible $\Gamma$-flows, i.e., $\mathcal{R}_r := \bigcup \{ \mathcal{R}_f \mid f \in \text{AFFl}_{\text{red}}^r \}$. $\mathcal{R}_r$ can be viewed as an $\mathbb{N}_0$-sorted set: $(\mathcal{R}_r)_n$ is the set of all component flows with $n$ exits.

Theorem 6.3 provides the key idea for the proof of the freeness of $\text{AFFl}_{\text{red}}^r$.

**Theorem 6.3 ("Component-decomposition theorem").** In the category FIM there exist for any signature $\Gamma$ a signature $\Omega$ and FIM-morphisms

\[
\text{RED} : \text{AFFl}_{\text{red}}^r \rightarrow \text{AFFl}_{\text{red}}^\Omega \quad \text{and} \quad \text{SUB} : \text{AFFl}_{\text{red}}^\Omega \rightarrow \text{AFFl}_{\text{red}}^r
\]

such that

\[
\text{RED} \cdot \text{SUB} = 1_{\text{AFFl}_{\text{red}}^r} \quad \text{and} \quad \text{SUB} \cdot \text{RED} = 1_{\text{AFFl}_{\text{red}}^\Omega}.
\]

In [12] this theorem is stated for $\text{AFFl}_r$. The proof proceeds as follows:

First a one-sorted signature $\Omega$ is defined in bijective correspondence with $\mathcal{R}_r$, i.e., for every $\mathcal{C} \in (\mathcal{R}_r)_n$ we have a unique operator symbol $\omega_{\mathcal{C}} \in \Omega_n$. RED is defined to transform any reducible $\Gamma$-flow $f$ into an acyclic $\Omega$-flow $f_{\text{RED}}$ by replacing every component of $f$ by the appropriate operator symbol. SUB then maps every operator symbol into the corresponding component flow. From Theorem 6.2 we know that SUB has a unique extension to an FIM-morphism. RED is called reduction and SUB is called substitution. The details of the proof may be found in [12, 20].

As a consequence of Theorem 6.3 and Proposition 6.1 we get

**Corollary 6.1.** Up to interspersed permutations every reducible $\Gamma$-flow $f$ in $\text{AFFl}_r$ has a unique "normal factorization" $(a, A, g)$ into components, i.e., there exist $a \in \text{SUR}$, $A : [r] \rightarrow \mathcal{R}_r$, and $g \in \text{AFFl}_{\text{red}}^r$ such that $f = a \ast (\Sigma A + 1_q) \ast g$.

A few more notions are needed before we can arrive at the next freeness theorem.

**Definition 6.4.** Let $f$ be a $\Gamma$-flow from $n$ to $p$.

(a) For $i \in [n]$ define $f_{G_i}$, the degree of the $i$th begin of $f$, as the number of edges ending in the $i$th begin vertex.

(b) In case $n = 1$ the degree of $f$ is defined as the degree of the begin of $f$, i.e., $f_G := f_{G_1}$.

The following lemmas provide a deeper characterization of the component flows [12, 20].

**Lemma 6.3.** In $\mathcal{R}_r$ nontrivial component flows of degree 0 are of the form $\gamma \ast \sigma$, where $\gamma \in \Gamma$ and $\sigma \in \text{SUR}$.

**Lemma 6.4.** For every $f \in \text{AFFl}_{\text{red}}^r(1, n)$ there is a factorization $f = g \ast h$, where $g$ is the component flow defined by the begin of $f$. For any $g'$, $h'$ satisfying $f = g' \ast h$ and $g' \in \text{MAP}$, $f_G > 0$ implies $g'_G > 0$. 
**Lemma 6.5.** A $\Gamma$-flow $f \in \text{AFFFlo}_r^{\text{red}}(1, p)$ has exactly one nontrivial component iff $f$ is nontrivial and $\ast$-irreducible. Then $f$ is a component flow iff $f$ is accessible and $\ast$-irreducible.

**Lemma 6.6.** A $\Gamma$-flow $f \in \text{AFFFlo}_r^{\text{red}}(1, p)$ is of positive degree iff there is a $g \in \text{AFFFlo}_r^{\text{red}}(1, p + 1)$ such that $f = g^\dagger$, $g \in \text{MAP}$ and for any $g' \in \text{AFFFlo}_r^{\text{red}}(1, p)$ $g \neq g' \ast t_{(1)}^r$.

Lemma 6.6 motivates the following definition where a new operation on $\Gamma$-flows, the anti-iteration, is introduced.

**Definition 6.5.** Let $f = (s, b, r, l) \in \text{Flo}_r(1, p)$ and let $\phi : [s + p] \rightarrow [s + p + 1]$ be an injective mapping satisfying 

\[ j\phi = s + p + 1, \quad \text{if} \quad j = lb, \]

\[ = j, \quad \text{otherwise.} \]

Then the anti-iteration $f^\dagger$ of $f$ is defined as $f^\dagger = (s', b', r', l')$ from 1 to $p + 1$, where $s' := s$, $b' := b \cdot t_{(1)}^r$, $r' := r \cdot \phi^*$, and $l' := l$.

Thus in $f^\dagger$ all edges of $f$ ending in the begin of $f$ are redirected into the new exit $p + 1$, i.e., $f^\dagger$ is the $\Gamma$-flow $g$ of Lemma 6.6. Properties of the anti-iteration are summarized in

**Lemma 6.7.** For all $f = (s, b, r, l) \in \text{AFFFlo}_r(1, p)$ we have

1. $f^{\dagger \dagger} = f$, $f^{\dagger} G = 0$, $f^{\dagger} \in \text{AFFFlo}_r(1, p + 1)$.
2. $f$ accessible and $f G > 0 \Rightarrow f^{\dagger}$ accessible.
3. $f$ reducible $\Rightarrow f^{\dagger}$ reducible.
4. $f G = 0 \Rightarrow f^{\dagger} = f \ast t_{(1)}^r$.
5. $p > 0$, $lb \neq s + p$ and $f G = 0 \Rightarrow f^{\dagger} = f$
6. $\forall g, h \in \text{AFFFlo}_r$, $p > 0 \land f^{\dagger} = g \ast h \land g \in \text{MAP} \land f G = 0 \land 1b \neq s + p \Rightarrow f = g^{\dagger} \ast (h + 1)$.

Now the iteration depth of a reducible $\Gamma$-flow may be defined.

**Definition 6.6.** For every $f \in \text{AFFFlo}_r^{\text{red}}$ the iteration depth $f^\dagger$ of $f$ is defined by

\[ f^\dagger := 1 + \max \{ g^\dagger | g \in \mathbb{C}_r \land g G > 0 \}. \]

In [12, 20] the function "it" is shown to be well defined. The following lemma considerably paves the way towards the proof of the next freeness theorem.

**Lemma 6.8 ([12, 20]).** For every nontrivial $f \in \text{AFFFlo}_r^{\text{red}}(1, p + 1)$ there exist
g ∈ AFFlo_n^*(1, n), h ∈ AFFlo_n^a(1, n, p) such that (∈ RED)^t = g * h and
g ∈ ∈ RED^+(n, p) ⊆ ∈.n. For this g we have g^+ ∈ AFFlo_n^a, and

(1) f = g^+SUB * (hSUB + 1).
(2) f^+ = (g^+SUB)^+ * hSUB.
(3) gG > 0 ⇒ (g^+SUB)^it ∈ ∈_t and (g^+SUB)^+it = 1 + (g^+SUB)^it.
(4) ∃!ω ∈ ∈, ∃!j ∈ AFFlo_n^*, g^+ = ω * j.

For these ω and j we have in case gG > 0,

(5) (g^+SUB) G > 0 ⇒ g^+SUB ≠ (g^+SUB)^+ = (ωSUB)^+ * (jSUB + 1) * (1, n + (1, 1, 1)).
(6) ωSUBit ≤ jSUBit ⇒ (g^+SUB)^it = 1 + g^+SUBit.
(7) ωSUB > jSUB ⇒ (g^+SUB)^it = g^+SUBit and (ωSUB)G, (g^+SUB)G > 0.

Finally we get

(8) fit ≤ f^it ≤ fit + 1.

For sake of brevity we only give the proof of (8). The other proofs may be found in [20].

In case f^+G = fG we have f = f^+ * t_{(1)} and ∈_t = ∈_t. Thus

\[ f^it = 1 + \max \{ g^it | g' \in ∈_t \land g'G > 0 \} \]
\[ = 1 + \max \{ g^it | g' \in ∈_t \land g'G > 0 \} \]
\[ = fit. \]

Otherwise we have f^+G > fG and thus (∈ RED)^G > (∈ RED)G = 0. Since (∈ RED)^t = g * h and g ∈ ∈ RED^+(n, p), we also have gG > 0. Furthermore (1) implies

\[ fit = \max (g^+SUBit, hSUBit) \] and (2) implies f^it = \max ((g^+SUB)^it, hSUBit). From (6) and (7) we get g^+SUBit ≤ (g^+SUB)^it ≤ 1 + g^+SUBit. Summarized this implies

\[ fit = \max (g^+SUBit, hSUBit) \]
\[ ≤ \max ((g^+SUB)^it, hSUBit) = f^it \]
\[ ≤ \max (1 + g^+SUBit, hSUBit) \]
\[ ≤ 1 + \max (g^+SUBit, hSUBit) \]
\[ = 1 + fit. \]

From Definition 6.6 we know that \( \bigcup_{r \in N} [r]it^{-1} = AFFlo_n^\ast. \) Let \( \Omega^{(r)} \subseteq \Omega \) be defined by \( \Omega^{(r)} := \{ Ωc \in Ω | \tilde{c} \in ∈_t \land \tilde{c}it \in [r] \}. \) Further define RED_r := RED|_{[r]it^{-1}} and SUB_r := SUB|_{AFFlo_n^a(r)}. This implies

\[ \text{RED}_r \cdot \text{SUB}_r = 1_{[r]it^{-1}} \quad \text{and} \quad \text{SUB}_r \cdot \text{RED}_r = 1_{AFFlo_n^a(r)}. \]
REDUCIBLE FLOWCHARTS

Since in FLM $\text{AAFFlo}_F^{\text{red}}(r)$ is free over $\Omega^{(r)}$ for all $r \in \mathbb{N}$, the isomorphisms RED, and SUB, show $[r]^{\text{it}^{-1}}$ to be free over $\mathcal{G}^{(r)} := \Omega^{(r)} \text{SUB}_r$. Hence if $T$ is a flow theory over MAP, then every mapping $\phi^{(r)}_0: \mathcal{G}^{(r)} \to T$ has a unique extension to an FLM-morphism $\phi^{(r)}: [r]^{\text{it}^{-1}} \to T$. These observations lead to the main theorem of this section.

**THEOREM 6.4.** If $T$ is a scalar iteration flow theory over MAP, then every strict mapping $\phi_0: \Gamma \to T$ has a unique extension to an FLM-morphism $\phi: \text{AAFFlo}^{\text{red}}_F \to T$.

**Proof.** The proof uses induction on the iteration depth of the elements of $\Lambda \text{AAFFlo}^{\text{red}}$. Since $[1]^{\text{it}^{-1}} = \Lambda \text{AAFFlo}^{\text{red}}$, $\phi_0$ has a unique extension to a PFLM morphism $\phi^{(1)}: \text{AAFFlo}^{\text{red}}_F \to T$. Suppose $\phi_0$ has a unique extension to a PFLM-morphism $\phi^{(r)}: [r]^{\text{it}^{-1}} \to T$.

For $g \in \mathcal{G}^{(r+1)} \setminus \mathcal{G}^{(r)}$ we have $gG > 0$ and $\text{git} > \text{g}^* \text{it} \leq r$. Thus $\phi^{(r+1)}: \mathcal{G}^{(r+1)} \to T$ may be defined by

$$g\phi^{(r+1)} := (g\phi^{(r)})^\dagger, \quad \text{if} \quad \text{git} = r + 1,$$

$$:= g\phi^{(r)}, \quad \text{otherwise}.$$

As mentioned above $\phi^{(r+1)}_0$ has a unique extension to a PFLM-morphism $\phi^{(r+1)}: [r+1]^{\text{it}^{-1}} \to T$, i.e., $\phi^{(r+1)}$ is a unique extension of $\phi^{(r)}$. Define $\phi := \bigcup_{r \in \mathbb{N}} \phi^{(r)}: \text{AAFFlo}^{\text{red}}_F \to T$. Then obviously

$$g \in \mathcal{G}_r \quad \text{and} \quad \text{git} > 1 \Rightarrow g\phi = (g\phi)^\dagger, \quad (***)$$

$\phi$ is a PFLM-morphism. (***)

It remains to be shown that $\phi$ respects the operation of scalar iteration, i.e., for every $f \in \text{AAFFlo}^{\text{red}}_F(1, p + 1)$ we have to show $f^\dagger \phi = (f\phi)^\dagger$. In case $f^\dagger G = fG$ we have $f = f^\dagger \ast t_{p+1}^1$, implying $f\phi = f\phi \cdot t_{p+1}^1$. Since $T$ is a scalar iteration flow theory we thus get $(f\phi)^\dagger = f\phi$.

Otherwise $f^\dagger G > fG$ implies $(f^{\text{RED}})^\dagger G > (f^{\text{RED}})G = 0$. Thus for $(f^{\text{RED}})^\dagger = g \ast h$ we have $gG > 0$, whence all statements of Lemma 3.10 may be applied

$$(f\phi)^\dagger = ((g^{\text{SUB}} \ast (h^{\text{SUB}} + 1)))\phi)^\dagger = (g^{\text{SUB}}\phi \cdot (h^{\text{SUB}}\phi + 1))$$

$$= (g^{\text{SUB}}\phi)^\dagger \cdot h^{\text{SUB}}\phi$$

and

$$f^\dagger \phi = ((g^{\text{SUB}})^\dagger \ast h^{\text{SUB}})\phi = (g^{\text{SUB}})^\dagger \phi \cdot h^{\text{SUB}}\phi.$$
\[(g^+\text{SUB})^t \in \mathcal{F}_r \text{ and } (g^+\text{SUB})^t \mu > 1. \] Thus \((*)\) implies \((g^+\text{SUB})^t \phi = ((g^+\text{SUB})^t \phi)^t\) and it remains to prove
\[(g^+\text{SUB})^t = ((g^+\text{SUB})^t \phi)^t.\]

In case \((g^+\text{SUB})G = 0\) we have \((g^+\text{SUB})^t = g^+\text{SUB}\), i.e., the equation holds. Otherwise there exist \(\omega \in \Omega\) and \(j \in \text{AFF}_{\omega}^{\text{red}}\) such that
\[g^+\text{SUB}\phi = \omega^+\text{SUB}\phi \cdot j^+\text{SUB}\phi\]
\[= ((\omega^+\text{SUB})^t \phi)^t \cdot j^+\text{SUB}\phi\]
\[= ((\omega^+\text{SUB})^t \phi \cdot (j^+\text{SUB}\phi + 1,))\]
implying
\[(g^+\text{SUB})^t = ((\omega^+\text{SUB})^t \phi \cdot (j^+\text{SUB}\phi + 1,))\]
\[= ((\omega^+\text{SUB})^t \phi \cdot (j^+\text{SUB}\phi + 1,))\cdot (1_n + \langle 1, 1, \rangle))\]
\[= ((\omega^+\text{SUB})^t \phi \cdot (j^+\text{SUB}\phi + 1,))\cdot (1_n + \langle 1, 1, \rangle))\phi\]
\[= ((g^+\text{SUB})^t \phi)^t.\]

Since according to Theorem 4.2 \(\text{AFF}_{\omega}^{\text{red}}\) is the least category containing the elementary \(\Gamma\)-flows and being closed under strong composition, sum, and scalar iteration, \(\phi\) is obviously uniquely defined.

The restriction to strict mappings is not essential: If \(\Gamma\) is not assumed to contain the distinguished operator \(\perp\), then any mapping \(\phi : \Gamma \rightarrow T\) obviously has a unique extension to a strict mapping \(\phi' : \Gamma_{\perp} \rightarrow T\).

We thus have extended the freeness results of [12] to the case of almost accessible \(\Gamma\)-flows. The next two sections provide a further extension to infinite \(\Gamma\)-flows.

### 7. Partial Order of Flowcharts

Assume the trivial partial order on \(\Gamma\), i.e., for all \(\gamma, \gamma' \in \Gamma\) we have \(\gamma \leq \gamma'\) iff \(\gamma = \perp\) or \(\gamma = \gamma'\) (remember that \(\Gamma\) is assumed to contain \(\perp\)). Further define for all \(f \in \text{Flo}_r\)
\[A_f = \{j \in A_f | jI = \perp\}.\]

The following definition of a partial order on \(\Gamma\)-flowcharts is close to corresponding notions for \(\Sigma\)-trees [3] and graphs [13].

**Definition 7.1.** Let \(f = (s, b, \tau, l), g = (s', b', \tau', l') \in \text{Flo}_r(n, p).

(a) In case \(A_f \neq \emptyset\) define \(f \leq g\) iff there is a surjective mapping \(\phi : A_f \rightarrow A_g\), such that for any extension \(\phi' : |s'| \rightarrow |s|\) of \(\phi\) we have

(i) \(\forall j \in A_f, |j\phi^{-1}| = 1\),
(ii) \( b = b' \cdot (\tilde{\phi} + 1_p) \),

(iii) \( \phi \cdot l \leq l|_{A^*} \),

(iv) \( \forall j \in A^*, j\phi l \neq \bot \Rightarrow j\tau'(\tilde{\phi} + 1_p)^* = j\phi \).

If necessary the dependence on \( \phi \) is indicated by writing \( f \leq \phi \cdot g \).

(b) In case \( A_\tau = \emptyset \) define \( f \leq g \) iff \( A_\tau = \emptyset \) and for all \( i \in [n] \) we have \( ib - s = ib' - s' \).

Thus whenever we have \( f \leq \phi \cdot g, f \) and \( g \) coincide in the accessible part of \( f \) which is not labeled with \( \bot \), and the part of \( g \) differing from \( f \) is mapped by \( \phi \) onto those accessible vertices of \( f \) labeled with \( \bot \).

**EXAMPLE 7.1.** Let \( f \) and \( g \) be given as in Fig. 10. Then we have \( f \leq \phi \cdot g \), where \( \phi \) is mapping those vertices of \( g \) which are labeled with \( y_1 \) and \( y_2 \) onto the corresponding vertices of \( f \), and the vertices labeled with \( y_3 \) and \( y_4 \) onto the vertex of \( f \) which is labeled with \( \bot \).

Another important notion characterizes the “distance” between a vertex and the begin of a \( \Gamma \)-flow.

**DEFINITION 7.2.** Let \( f = (s, b, \tau, l) \in \text{Flo}_\tau(n, p) \).

(a) The depth of a vertex \( j \in A_f \) is defined by

\[
jD := \min \left\{ |\pi| \mid \pi \in \bigcup_{i \in [n]} \Pi(f)(ib, j) \right\},
\]

i.e., \( jD \) is the minimal length of a path from a begin to \( j \).

(b) The depth of \( f \) is defined by \( jD := \sup\{ j \in A_f \} \), i.e., the depth of \( f \) may be infinite.

(c) For \( k < \omega \) define \( A^f_{\Gamma^k} \) to be the set of all accessible interior vertices of \( f \) of depth maximally \( k \).

Thus the begins of a \( \Gamma \)-flow have depth 0. Since every accessible vertex is accessible via a path of finite length, \( A_f \) is always countable.

---

**FIGURE 10**

- **f:**
  - B1
  - Y3
  - Y2
  - Y5
  - E1
- **g:**
  - B2
  - Y1
  - Y3
  - Y2
  - E1
  - E2
PROPOSITION 7.1. (a) The relation "<" on Flo is reflexive and transitive.
(b) For all \( f, g \in Flo \) \( f \leq g \) and \( g \leq f \) imply \( f \equiv g \).

Proof: (a) Reflexivity of "<" is trivial. Transitivity is verified by an easy calculation.
(b) In case \( A_f = \emptyset \) the implication is trivial. Otherwise assume \( f \leq g \) and \( g \leq f \).
Then we have \( \phi \cdot \psi : A_g \to A_g \) and \( \psi \cdot \phi : A_f \to A_f \). By an easy induction on the depth of the accessible vertices we get \( \phi \cdot \psi = 1_{A_g} \) and \( \psi \cdot \phi = 1_{A_f} \) implying \( \phi = \psi^{-1} \). Thus \( \phi \) and \( \psi \) are bijective and \( f \equiv g \). \( \square \)

In case of almost accessible \( f \) and \( g \), \( f \leq g \) and \( g \leq f \) even imply \( f \equiv g \). This leads to

COROLLARY 7.1. The relation "<" is a partial order on the isomorphism classes of AAFlor.

For all \( n, p < \omega \) and \( n > 0 \) we define a special \( I \)-flow \( n \) \( p \in \text{AAFlor}(n, p) \) by \( n \) \( p \) := \( (1, b, \tau, l) \), where \( |n| = \{1\} \), \( 1 \tau = c \), and \( 1l = \bot \). Thus \( n \) \( p \) is the \( I \)-flow depicted in Fig. 11.

Obviously for all \( f \in \text{Flo}(n, p) \) \( A_f \neq \emptyset \) implies \( n \) \( p \leq f \). In AAFlor \( n, p \) all elements of MAP \( n, p \) are isolated, i.e., incomparable, with respect to the partial order on \( I \)-flows. Let us call a partially ordered set weakly strict (weakly \( \omega \)-complete) iff it is a strict (\( \omega \)-complete) set augmented by isolated elements. The least element of the maximal strict subset of a weakly strict set \( M \) is called least element of \( M \), usually denoted by \( \bot_M \). We thus have

PROPOSITION 7.2. AAFlor \( n, p \) is a weakly strict set.

The crucial idea for extending the results of Section 6 to infinite \( I \)-flows is to represent every \( I \)-flow as the least upper bound of an \( \omega \)-chain of finite \( I \)-flows.

DEFINITION 7.3. Let \( f = (s, b, \tau, l) \in \text{AAFlor}(n, p) \). For each \( i < \omega \) the \( i \)-truncate of \( f^{(i)} = (s_i, b_i, \tau_i, l_i) \in \text{AAFlor}(n, p) \) is defined as follows:

In case \( s = 0 \) define \( f^{(i)} := f \). Otherwise define \( s_i := |A_f^{(i)}| \). Let \( \phi_i : A_f^{(i)} \to |s_i| \) be a bijection, and let \( \psi_i : [s] \to [s_i] \) be an extension of \( \phi_i \). Then define \( b_i := b \cdot (\phi_i + 1_p) \) and for each \( j \in [s_i] \),

\[
jl_i := j \phi_i^{-1}l, \quad \text{if } j \phi_i^{-1}D < i,
\]

\[
jl_i := \bot, \quad \text{otherwise,}
\]

\[
\begin{array}{c}
B1 \ldots Bn \\
\end{array}
\]

\[
\begin{array}{c}
E1 \ldots En
\end{array}
\]

FIGURE 11
and

\[ j\phi_i := j\phi_i^{-1} \tau(\tilde{\phi}_i + 1_p), \quad \text{if} \quad j\phi_i^{-1} D < i, \]
\[ := \varepsilon, \quad \text{otherwise}. \]

Thus, as is well known in the case of \( \Sigma \)-trees [3], \( f^{(t)} \) can be viewed as a finite approximation of \( f \).

In [20] the following important theorems are derived:

**THEOREM 7.1.** For all \( n, p < \omega \) \( \text{AAFlo}_T(n, p) \) is weakly \( \omega \)-complete.

Let us call a mapping \( \phi \) mapping a weakly strict set \( A \) into a weakly strict set \( B \) strict iff (1) for each \( a \in A \), \( a\phi \) isolated in \( B \) implies a isolated in \( A \) and (2) \( \downarrow_a \phi = \downarrow_B \). A strict mapping \( \phi : A \to B \) is \( \omega \)-continuous iff \( \phi \) is monotonic and for each \( \omega \)-chain \( (a_i)_{i<\omega} \) in \( A \) we have \( (\bigcup_{i<\omega} a_i) \phi = \bigcup_{i<\omega} a_i \phi \).

**THEOREM 7.2.** Strong composition, sum, and scalar iteration are \( \omega \)-continuous operations on \( \text{AAFlo}_T \).

**THEOREM 7.3.** Each \( f \in \text{AAFlo}_T \) is the least upper bound of an \( \omega \)-chain of finite almost accessible \( \Gamma \)-flows : \( f = \bigcup_{i<\omega} f^{(i)} \).

The rather lengthy proofs may be found in [20]. As an obvious consequence of these theorems and Theorems 4.1 and 4.2 we get

**THEOREM 7.4.** \( \text{AAFlo}^{\omega_c}_T \) is the least category with weakly \( \omega \)-complete sets of morphisms containing the elementary \( \Gamma \)-flows and being closed under strong composition and sum.

**THEOREM 7.5.** \( \text{AAFlo}^{\omega_{ed}}_T \) is the least category with weakly \( \omega \)-complete sets of morphisms containing the elementary \( \Gamma \)-flows and being closed under strong composition, sum, and scalar iteration.

8. **Freeness Results for Infinite Flowcharts**

In this section the freeness results of Section 5 are extended to infinite \( \Gamma \)-flows. Our approach is closely related to the technique employed by [3] in the case of \( \Sigma \)-trees.

First we have to introduce partially ordered and \( \omega \)-continuous flow theories.

**DEFINITION 8.1.** (a) A pointed flow theory \( T \) over MAP is partially ordered iff

(i) For each \( n, p < \omega \) \( T(n, p) \) is partially ordered and weakly strict with least element \( \perp_{n, p} \).
For each \( n, p < \omega \) the set of isolated elements of \( T(n, p) \) is a subset of \( \text{MAP}(n, p) \).

The operations \( \cdot \) and \( + \) are strict and monotonic.

A \textit{PPFlm-morphism} is a monotonic \textit{PFIM-morphism}.

Let \textit{PPFlm} denote the subcategory of \textit{PFIM} having partially ordered pointed flow theories over \textit{MAP} as objects and \textit{PPFlm}-morphisms as morphisms.

A mapping \( \phi \) from \( T \) into a partially ordered flow theory \( T \) over \textit{MAP} is called \textit{strict} iff \( \bot \phi = \bot_{1,0} \in T(1, 0) \) and \( T \phi \) contains no isolated elements.

\textbf{Definition 8.2.}  
(a) A partially ordered pointed flow theory \( T \) over \textit{MAP} is \( \omega \)-continuous iff

(i) For each \( n, p < \omega \) \( T(n, p) \) is weakly \( \omega \)-complete

(ii) The operations \( \cdot \) and \( + \) are \( \omega \)-continuous.

(b) A \textit{CPFIM morphism} is an \( \omega \)-continuous \textit{PPFlm-morphism}.

(c) Let \textit{CPFIM} denote the subcategory of \textit{PPFlm} having \( \omega \)-continuous pointed flow theories as objects and \textit{CPFIM}-morphisms as morphisms.

The results of the last section imply

\textbf{Proposition 8.1.}  
(a) \textit{AAFFlo}_{\text{ac}}^{\text{uc}} \text{ and } \textit{AAFFlo}_{\text{red}}^{\text{uc}} \text{ are partially ordered pointed flow theories over MAP.}

(b) \( \textit{AAFl}_{\text{ac}}^{\text{uc}} \text{ and } \textit{AAFl}_{\text{red}}^{\text{uc}} \text{ are } \omega \text{-continuous pointed flow theories over MAP.}

Furthermore one directly verifies

\textbf{Proposition 8.2.}  
(a) Every partially ordered algebraic theory extending \textit{MAP} is a partially ordered pointed flow theory over \textit{MAP}.

(b) Every \( \omega \)-continuous algebraic theory extending \textit{MAP} is an \( \omega \)-continuous pointed flow theory over \textit{MAP}.

A few lemmas are needed in the proofs of this section.

\textbf{Lemma 8.1.}  
For each \( f, g \in \textit{AAFl}_{\text{uc}}^{\text{uc}}(n, p), h \in \textit{AAFl}_{\text{uc}}^{\text{uc}}(p, q) \) and \( \chi \in \text{MAP}(p, q) \)
\( f \cdot \chi \leq g \cdot h \) and \( f \leq g \) imply \( f \cdot \chi = f \cdot h. \)

\textbf{Lemma 8.2.}  
(a) If \( (f_i)_{i<\omega} \) is an \( \omega \)-chain in \( \textit{AAFl}_{\text{uc}}^{\text{uc}}(n, p) \) and \( f = \cup_{i<\omega} f_i \), then for each \( n < \omega \) there exists an \( i < \omega \) such that \( f^{(n)} \leq f_i \).

(b) If \( f, g \in \textit{AAFl}_{\text{uc}}^{\text{uc}}(n, p) \) and \( f \leq g \), then for each \( i < \omega \) we have \( f^{(i)} \leq g^{(i)} \)

\textbf{Lemma 8.3.}  
For each \( f_1, f_2 \in \textit{AAFFlo}_{\text{red}}^{\text{uc}} \) with \( f_1 \leq f_2 \) there exist \( g \in \textit{AAFFlo}_{\text{red}}^{\text{uc}} \) and \( h_1, h_2 \in \textit{AAFFlo}_{\text{red}}^{\text{uc}} \) such that \( f_1 = g \cdot h_1, f_2 = g \cdot h_2 \), and \( h_1 \leq h_2 \).

As a corollary we get
Lemma 8.4. For each $f_1, f_2 \in \text{AAFFlo}^\infty_T$ with $f_1 \leq f_2$ there exist $g \in \text{AFFlo}^\infty_T$, $h_1, h_2 \in \text{AAFFlo}^\infty_T$ such that $f_1 = g \ast h_1, f_2 = g \ast h_2, h_1 \leq h_2$, and all interior vertices This lemma facilitates the proof of

Theorem 8.1. If $T$ is a partially ordered flow theory over MAP, then every strict mapping $\phi_0 : \Gamma \to T$ has a unique extension to a PPFIM-morphism $\phi : \text{AAFFlo}^\infty_T \to T$.

Proof. According to Theorem 6.2 $\phi_0$ has a unique extension to a PFIIM-morphism $\phi : \text{AAFFlo}^\infty_T \to T$. It remains to show $\phi$ to be monotonic.

Assume $f_1, f_2 \in \text{AAFFlo}^\infty_T$ such that $f_1 \leq f_2$. If $f_1$ or $f_2$ is in MAP, we have $f_1 \phi = f_2 \phi$. Thus assume $f_1$ and $f_2$ to be nontrivial. We distinguish two cases:

(a) All the interior vertices of $f_1$ are labeled with $\bot$. Since $\phi$ is a PFIIM-morphism, we have $\bot \phi = \bot$. Since furthermore the operations $\cdot$ and $+$ of $T$ are monotonic, we get $f_1 \phi \leq f_2 \phi$.

(b) $f_1$ has an interior vertex not labeled with $\bot$. According to Lemma 8.4 there exist $g, h, h_1$ in $\text{AAFFlo}^\infty_T$ such that $f_1 = g \ast h_1, f_2 = g \ast h_2, h_1 \leq h_2$, and all interior vertices of $h_1$ are labeled with $\bot$. Thus (a) implies $h_1 \phi \leq h_2 \phi$ whence

$$f_1 \phi = (g \ast h_1) \phi = g \phi \cdot h_1 \phi \leq g \phi \cdot h_2 \phi = (g \ast h_2) \phi = f_2 \phi.$$  

This theorem now helps to establish the first freeness result for infinite $\Gamma$-flows.

Theorem 8.2. If $T$ is an $\omega$-continuous pointed flow theory over MAP, then every strict mapping $\phi_0 : \Gamma \to T$ has a unique extension to a CPFIM-morphism $\phi : \text{AAFFlo}^\infty_T \to T$.

Proof. According to Theorem 8.1 $\phi_0$ has a unique extension to a CPFIM-morphism $\phi : \text{AAFFlo}^\infty_T \to T$. Each $f \in \text{AAFFlo}^\infty_T$ is the least upper bound of the $\omega$-chain $(f^{(i)})_{i < \omega}$. Furthermore for all $i < \omega$ we have $f^{(i)} \in \text{AAFFlo}^\infty_T$. Define $f_\phi := \bigcup_{i < \omega} f^{(i)} \phi$. This least upper bound exists since $\phi$ is monotonic and $T$ is $\omega$-continuous. Obviously $f$ is an extension of $\phi$. It remains to show that $\phi$ is a CPFIM-morphism and that it is unique.

Assume $f, g \in \text{AAFFlo}^\infty_T(n, p)$ such that $f \leq g$. Lemma 8.2(b) implies for all $i < \omega f^{(i)} \phi \leq g^{(i)} \phi$. Since $\phi$ is monotonic, we have for all $i < \omega f^{(i)} \phi \leq g^{(i)} \phi$ whence $f_\phi = \bigcup_{i < \omega} f^{(i)} \phi \leq \bigcup_{i < \omega} g^{(i)} \phi = g_\phi$, i.e., $\phi$ is monotonic.

Let $(f_i)_{i < \omega}$ be an $\omega$-chain in $\text{AAFFlo}^\infty_T(n, p)$ and $f := \bigcup_{i < \omega} f_i$. Since for all $i, f_i \leq f$, we have $f_\phi \leq f_\phi$. Thus $f_\phi$ is an upper bound of the $f_i \phi$. Let $g$ be another upper bound. According to Lemma 8.2(a) there exists for each $n < \omega$ a $j < \omega$ such that $f^{(n)} \phi \leq f_j \phi \leq g$ implying $f_\phi = \bigcup_{i < \omega} f^{(i)} \phi \leq g$. Thus $f_\phi = \bigcup_{i < \omega} f_i \phi$ and $\phi$ is $\omega$-continuous.

To show $\phi$ to respect the operations on $\Gamma$-flows assume $f \in \text{AAFFlo}^\infty_T(n, p), g \in \text{AAFFlo}^\infty_T(p, q)$. Then we have
\[(f * g) \phi = \left( \bigcup_{i<\omega} f^{(i)} \ast \bigcup_{i<\omega} g^{(i)} \right) \phi \]
\[= \left( \bigcup_{i<\omega} (f^{(i)} \ast g^{(i)}) \right) \phi \]
\[= \bigcup_{i<\omega} (f^{(i)} \ast g^{(i)}) \tilde{\phi} \]
\[= \bigcup_{i<\omega} \left( f^{(i)} \tilde{\phi} \cdot g^{(i)} \tilde{\phi} \right) \]
\[= \bigcup_{i<\omega} f^{(i)} \tilde{\phi} \cdot \bigcup_{i<\omega} g^{(i)} \tilde{\phi} \]
\[= f\tilde{\phi} \cdot g\tilde{\phi}. \]

In the case of sum the proof is analogous. It remains to show that $\phi$ is unique: Assume $\tilde{\phi}$ to be another CPFLM-morphism extending $\phi_0$. Then we have for each $f \in AAFlop_r^{\mathbb{R}}$

\[f\tilde{\phi} = \left( \bigcup_{i<\omega} f^{(i)} \right) \tilde{\phi} = \bigcup_{i<\omega} f^{(i)} \tilde{\phi} = \bigcup_{i<\omega} f^{(i)} \tilde{\phi} = f\tilde{\phi}. \]

Thus $AAFlop_r^{\mathbb{R}}$ is a free object in CPFLM with respect to strict mappings.

Before we can derive an analogous result for reducible $\Gamma$-flows partially ordered and $\omega$-continuous scalar iteration flow theories have to be introduced.

**Definition 8.3.** (a) A scalar iteration flow theory $T$ over MAP is **partially ordered** iff $T$ is a partially ordered pointed flow theory over MAP and the iteration is monotonic.

(b) A PIFLM-morphism is a monotonic IFLM-morphism.

(c) Let PIFLM denote the subcategory of IFLM having partially ordered scalar iteration flow theories over MAP as objects and PIFLM-morphisms as morphisms.

**Definition 8.4.** (a) A partially ordered scalar iteration flow theory $T$ over MAP is **$\omega$-continuous** iff $T$ is an $\omega$-continuous pointed flow theory and the iteration is $\omega$ continuous.

(b) A CIFLM-morphism is an $\omega$-continuous PIFLM-morphism.

(c) Let CIFLM denote the subcategory of PIFLM having $\omega$-continuous scalar iteration flow theories over MAP as objects and CIFLM-morphisms as morphisms.

The results of Section 7 imply

**Proposition 8.3.** (a) $AAFlop_r^{\omega \mathbb{R}}$ is a partially ordered scalar iteration flow theory over MAP.
(b) $\text{AAFflo}^{\text{red}}_I$ is an $\omega$-continuous scalar iteration flow theory over $\text{MAP}$.

By means of Proposition 5.5 we also get

**Proposition 8.4.** Every $\omega$-continuous algebraic theory extending $\text{MAP}$ is an $\omega$-continuous scalar iteration flow theory over $\text{MAP}$.

Now we can show $\text{AAFflo}^{\text{red}}_I$ to be a free object of $\text{PIFlM}$ with respect to strict mappings.

**Theorem 8.3.** If $T$ is a partially ordered scalar iteration flow theory over $\text{MAP}$, then every strict mapping $\phi_o: \Gamma \to T$ has a unique extension to a $\text{PIFlM}$-morphism $\phi: \text{AAFflo}^{\text{red}}_I \to T$.

**Proof.** The proof is by induction on the iteration depth. According to Theorem 6.4 $\phi_0$ has a unique extension to an $\text{IFlM}$-morphism $\phi: \text{AAFflo}^{\text{red}}_I \to T$. It remains to show $\phi$ to be monotonic:

Assume $f_1, f_2 \in \text{AAFflo}^{\text{red}}_I(n, p)$ such that $f_1 \leq f_2$. Obviously this implies $f_1$ it $f_2$ it.

If $f_2$ it $1$, $f_1$ and $f_2$ are acyclic. Thus Theorem 8.1 implies $f_1 \phi \leq f_2 \phi$. Otherwise assume monotonicity to be proven for all iteration depths less than $f_2$ it. We distinguish two cases:

(a) $f_2 \in \mathcal{G}_I$. Since $f_2$ it $1$ we have $f_2 G > 0$, whence $f_1^+ \phi \leq f_2^+ \phi$. Furthermore $f_1 \leq f_2$ implies $f_1^+ \leq f_2^+$. Thus the induction hypothesis implies $f_1^+ \phi \leq f_2^+ \phi$ and we have

$$f_1 \phi = f_1^+ \phi = (f_1^+ \phi)^+ \leq (f_2^+ \phi)^+ = f_2^+ \phi = f_2 \phi.$$ 

(b) $f_2 \notin \mathcal{G}_I$. According to Lemma 8.3 there exist $g \in \text{AAFflo}^{\text{red}}_I(n, r + q)$ and $h_1, h_2 \in \text{AAFflo}^{\text{red}}_I(r + q, p)$ such that $f_1 = g * h_1, f_2 = g * h_2$, and $h_1 \leq h_2$.

It is easily seen that there further exist $A: [r] \to \mathcal{G}_I, B: [r] \to \text{AAFflo}^{\text{red}}_I, h \in \text{AAFflo}^{\text{red}}_I$ and $\chi \in \text{SUR}$ such that

$$h_1 \leq h_1^* = (\Sigma B + 1_\rho) * \chi \leq h_2 = (\Sigma A + 1_\rho) * h$$

and for all $i \in [r], iB \leq iA$ holds. Lemma 8.1 then implies $h_1^* = (\Sigma B + 1_\rho) * h$. Since $h_1$ and $h_1^*$ differ only in vertices labeled with $\perp$ another application of Lemma 8.3 shows $h_1 \phi \leq h_1^* \phi$. Furthermore for all $i \in [r]$ we have $iA \phi \leq iA \phi$ which implies $\Sigma A \phi \leq \Sigma B \phi$. Summarizing we have

$$f_1 \phi = (g * h_1) \phi = g \phi \cdot h_1 \phi$$

$$\leq g \phi \cdot h_1^* \phi = g \phi \cdot ((\Sigma B + 1_\rho) * h) \phi$$

$$= g \phi \cdot (\Sigma B \phi + 1_\rho) \cdot h \phi$$

$$\leq g \phi \cdot (\Sigma A \phi + 1_\rho) \cdot h \phi$$
As in the case of acyclic \( \Gamma \)-flows this theorem allows to derive the freeness of \( \text{AAFlo}_\Gamma^{\text{red}} \) in CIFIM.

**Theorem 8.4.** If \( T \) is an \( \omega \)-continuous scalar iteration flow theory over MAP, then every strict mapping \( \phi_0: \Gamma \to T \) has a unique extension to a CIFIM-morphism \( \phi: \text{AAFlo}_\Gamma^{\text{red}} \to T \).

The proof is completely analogous to the proof of Theorem 8.2.

9. **An Example: Nondeterministic Flowcharts**

In this section we present an operator system \( \Gamma \) which provides an extension to the class of flowcharts serving as the target language of a compiler in [7].

We start from the same abstract data type as Thatcher, Wagner, and Wright: Let \( \Sigma \) and \( E \) be the \{int, Bool\}-sorted signature and set of axioms as defined in [7]. Let \( S \) be a \((\Sigma, E)\)-algebra such that \( S_{\text{int}} = \mathbb{Z} \) and \( S_{\text{bool}} = \{1, 2\} \), where \( t_S = 2 \). The axioms ensure a reasonable integer arithmetic within \( S \).

The operations of \( \Gamma \) are to be interpreted as operations on a stack machine consisting of a random access memory and a stack. Each memory cell contains an integer and is represented by an element of \( \text{Id} \), a nonempty set of identifiers. The contents of each cell may be loaded on top of the stack. The topmost stack elements are available for arithmetic and Boolean operations as defined in \( S \). Furthermore it is possible to load the topmost stack element into the memory, and there is a nondeterministic choice operation.

This informal description of the stack machine is made precise by the following definition of \( \Gamma \) and of an interpretation \( I \).

Let the one-sorted operator system \( \Gamma \) be defined as follows:

\[
\Gamma_0 := \{ \bot \},
\]

\[
\Gamma_1 := \{ \text{load}_x, \text{store}_x | x \in \text{Id} \} \cup \{ \text{switch}, \text{cont} \} \cup \{ \Sigma_{\text{int}, w} | w \in \{\text{int}\}^* \},
\]

\[
\Gamma_2 := \{ \text{choose}_2 \} \cup \{ \Sigma_{\text{bool}, w} | w \in \{\text{int}\}^* \},
\]

and for all \( n \geq 3 \) \( \Gamma_n := \{ \text{choose}_n \} \).

To interpret \( \Gamma \) we begin by defining \( \text{Stk} := [(\omega) \to \mathbb{Z}] \), the set of all possible stacks and \( \text{Env} := [\text{Id} \to \mathbb{Z}] \), the set of all memory contents. The representation of stacks as infinite words simplifies the formal treatment. The first symbol always represents the top of stack.
For \( e \in \text{Env} \), \( x \in \text{Id} \), and \( v \in \mathbb{Z} \) the mapping \( e[v/x] \in \text{Env} \) is as usual defined by

\[
y e[v/x] := v, \quad \text{if } y = x,
\]

\[
:= y e, \quad \text{otherwise},
\]

for all \( y \in \text{Id} \). Define \( A := \text{Stk} \times \text{Env} \). For each \( n, p < \omega \) let \( \text{RSUM}_A(n, p) \) be the set of all relations between \( A \times [n] \) and \( A \times [p] \), i.e.,

\[
\text{RSUM}_A(n, p) = \mathcal{P}(\text{Stck} \times \text{Env} \times [n] \times \text{Stk} \times \text{Env} \times [p]).
\]

(In [7] the well-known algebraic theory \( \Sigma_A \) is used as a semantic domain.) It is easily verified that \( \text{RSUM}_A \) is an \( \omega \)-continuous algebraic theory extending \( \text{MAP} \). Thus according to Proposition 8.4 it is also an \( \omega \)-continuous scalar iteration flow theory over \( \text{MAP} \).

The interpretation \( I \) of \( \Gamma \) in \( \text{RSUM}_A \) now may be defined as follows: Assume \( \sigma \in \text{Stk} \), \( e \in \text{Env} \), \( x \in \text{Id} \), \( v, v_1, v_2 \in \mathbb{Z} \), Then

\[
(11) \quad \Upsilon := \Upsilon_{1,0},
\]

\[
(12) \quad (\sigma, e)(\text{load}_x I)((xe) \sigma, e),
\]

\[
(13) \quad (v \sigma, e)(\text{store}_x I)(\sigma, e[v/x]),
\]

\[
(14) \quad (v_1, v_2 \sigma, e)(\text{switch} I)(v_2 v_1 \sigma, e),
\]

\[
(15) \quad (\sigma, e)(\text{cont} I)(\sigma, e),
\]

\[
(16) \quad \text{for all } c \in \Sigma_{\text{int},c}, (\sigma, e)(c\text{c}f)(c \sigma, e),
\]

\[
(17) \quad \text{for all } \text{aop}1 \in \Sigma_{\text{int},\text{int}} (u \sigma, e)(\text{aop}1\text{f})(\text{u}\text{aop}1 \sigma, e)
\]

\[
(18) \quad \text{for all } \text{aop}2 \in \Sigma_{\text{int},\text{intint}} (v_2 v_1 \sigma, e)(\text{aop}2\text{f})(\text{(v}_1, v_2) \text{aop}2 \sigma, e),
\]

\[
(19) \quad \text{for all } b \in \Sigma_{\text{int}}, (\sigma, e)(\text{bl}f)(\sigma, e, b \sigma),
\]

\[
(10) \quad (v \sigma, e)(\text{even} I)(\sigma, e, v \text{even} s)
\]

\[
(11) \quad \text{for all } \text{rel} \in \Sigma_{\text{int},\text{intint}} (v_2 v_1 \sigma, e)(\text{rel} f)(\sigma, e, (v_1, v_2) \text{rel} s),
\]

\[
(12) \quad \text{for all } n \geq 2, \text{ for all } j \in [n], (\sigma, e)(\text{choose} _n I)(\sigma, e, j).
\]

For sake of brevity in this definition of \( I A \times [1] \) is identified with \( A \), and the infix notation for relations is used.

According to Theorem 8.4 \( I \) has a unique extension to a CIFLM-morphism \( \bar{I} : \text{AAFlo}^{\text{fed}}_\Gamma \rightarrow \text{RSUM}_A \). Thus \( \bar{I} \) assigns to each reducible \( \Gamma \)-flow \( f \) in \( \text{AAFlo}^{\text{fed}}_\Gamma(n, p) \) a relation \( f \bar{I} \) between \( \text{Stk} \times \text{Env} \times [n] \) and \( \text{Stk} \times \text{Env} \times [p] \).

10. Conclusion

The main result of this paper is the freeness of \( \text{AAFlo}^{\text{fed}}_\Gamma \) in the category of all \( \omega \)-continuous scalar iteration flow theories over \( \text{MAP} \). It provides an extension to analogous results for finite flowcharts obtained by Elgot and Shepherdson in [12].
Section 9 demonstrates how this result reduces the investigation of specific statements in programming languages to the definition of an operator system $I$, and to choosing an appropriate $\omega$-continuous scalar iteration flow theory over MAP as a semantic domain. It should be interesting to include statements like "fork" and "join" known from parallel programming languages.

As is demonstrated in [20, 21] the freeness of $\text{AAFlo}_F$ further allows to define fixpoint semantics of recursive flowchart schemes in the same mathematically elegant way as is well known in the case of recursive tree schemes [2, 13]. This might lead to a significant extension of the compiler correctness results of [7] by allowing recursive structures in the source language.

REFERENCES