On some algebraic formulation of algorithms

By Nobuo ZAMA

(Received July 10, 1968)

Introduction

This paper is the subsequent part of the author's paper [1] and [2]. Recursive functions can be defined by flow-charts, as is well known. The flow-charts can be considered not only as formulas which are written to define functions and predicates but also as diagrams. The author intends to develop a theory to discuss flow-charts, and he adopts the concept of category to describe their properties.

In this paper, the author writes only the most fundamental properties of the category which consists from flow-charts, but more useful developments might be expected. As an application of general discussion, the generalized notion of universal algorithms will be given.

1. Recursive structures

1.1. The definition of recursive structures.

Let $\boldsymbol{M}$ be an arbitrary set. By $\boldsymbol{F}$ we denote an arbitrary set of univalent functions whose ranges and domains are $\boldsymbol{M}$ or subsets of $\boldsymbol{M}$. We also assume $\boldsymbol{F}$ contains such the identity function $e(x)$ that $e(x)=x$ for all $x \in \boldsymbol{M}$. By $\boldsymbol{P}$ we denote an arbitrary set of predicates whose arguments are elements of $\boldsymbol{M}$. We can give the flow-chart* whose blocks are filled with functions of $\boldsymbol{F}$ and predicates of $\boldsymbol{P}$. We shall denote the set of these flow-charts by $\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle$. Using such flow-charts, we can define predicates and functions. By $\langle\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\rangle$ we denote the set of these functions and predicates, and call it the recursive structure over $\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle$, sometimes simply recursive structure over $\boldsymbol{M}$. $\boldsymbol{M}$ is called the base of $\langle\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\rangle$.

In this paper the number of variables of functions in $\boldsymbol{M}$ are disregarded, since if more than two variables are necessary, it is enough to consider the set of pairs of elements of $\boldsymbol{M}$.

The following lemmas 1, 2, 3, are fundamental in this paper, but the proofs are omitted here since they are in the author's paper [2]. Refer the paper about the details of the flow-charts.

Lemma 1. *Let $\langle\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\rangle$ and $\langle\langle \boldsymbol{G}, \boldsymbol{Q} | \boldsymbol{M} \rangle\rangle$ be two recursive structures over $\boldsymbol{M}$. Then*

* We shall understand flow-charts and logical flow-charts in the paper [2] by a word "flow-chart" if there is not any notice.

$$\langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle \supseteqq \langle\!\langle \boldsymbol{G}, \boldsymbol{Q} | \boldsymbol{M} \rangle\!\rangle \Longleftrightarrow \langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle \supseteqq \boldsymbol{G} \text{ and } \langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle \supseteqq \boldsymbol{Q} .$$

1.2. Mappings of recursive structures.

We consider two recursive structures $\mathscr{F} = \langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle$, $\mathscr{G} = \langle\!\langle \boldsymbol{G}, \boldsymbol{Q} | \boldsymbol{N} \rangle\!\rangle$, where $\boldsymbol{N}$ in allowed to be equal to $\boldsymbol{M}$. Let φ be a mapping from $\boldsymbol{M}$ in $\boldsymbol{N}$. If f is a function of $\mathscr{F}$ and p is a predicate of $\mathscr{F}$, we can define φf, φp by the following rule:

$$(\varphi f)(\varphi(x)) = \varphi(f(x)) \quad (\text{for all } x \in \boldsymbol{M})$$

$$(\varphi p)(\varphi(x)) \Longleftrightarrow p(x) \quad (\text{for all } x \in \boldsymbol{M})$$

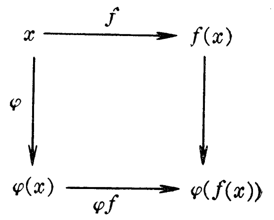


Fig. 1.

The diagram in the left shows the meaning of the rule. By $\varphi \boldsymbol{F}$ we denote the set $\{\varphi f;\ f \in \boldsymbol{F}\}$, and by $\varphi \boldsymbol{P}$ the set $\{\varphi p;\ p \in \boldsymbol{P}\}$. Let $\varphi\langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle$ be $\langle\!\langle \varphi\boldsymbol{F}, \varphi\boldsymbol{P} | \boldsymbol{N} \rangle\!\rangle$. Then we have the following lemma according to Lemma 1.

Lemma 2. $\langle\!\langle \boldsymbol{G}, \boldsymbol{Q} | \boldsymbol{N} \rangle\!\rangle \supseteq \varphi \langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle \Longleftrightarrow$
$\langle\!\langle \boldsymbol{G}, \boldsymbol{Q} | \boldsymbol{N} \rangle\!\rangle \supseteq \varphi \boldsymbol{F}$ *and* $\langle\!\langle \boldsymbol{G}, \boldsymbol{Q} | \boldsymbol{N} \rangle\!\rangle \supseteq \varphi \boldsymbol{P}$.

Let a function or a predicate $f \in \langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle$ be defined by a flow-chart F. By φF we shall understand the flow-chart which is made by substituting $f \in \boldsymbol{F}$ in F to φf, $p \in \boldsymbol{P}$ in F to φp. Then we have the following lemma:

Lemma 3. *Let F be a flow-chart of $\langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle$. Then*

$$(\varphi F)(\varphi(x)) = \varphi(F(x)) \quad (\textit{if } F \textit{ is a function})$$

$$(\varphi F)(\varphi(x)) \Longleftrightarrow F(x) \quad (\textit{if } F \textit{ defines a predicate})$$

where $F(x)$ is the result of the computation of F for an element x.

In the following, we shall assume that a mapping $\Phi : \mathscr{F} \to \mathscr{G}$ is a mapping from $\mathscr{F}$ in $\mathscr{G}$ which assigns elements of $\boldsymbol{M}$ to elements of $\boldsymbol{N}$, functions in $\boldsymbol{F}$ to functions in $\boldsymbol{G}$, and predicates in $\boldsymbol{F}$ to predicates in $\boldsymbol{G}$. A mapping $\Phi : \mathscr{F} \to \mathscr{G}$ is a homomorphism if and only if

1) $\Phi f \in \boldsymbol{G}$ (for all $f \in \boldsymbol{F}$), $\Phi p \in \boldsymbol{Q}$ (for all $p \in \boldsymbol{P}$);

2) $(\Phi f)(\Phi(x)) = \Phi(f(x))$ (for all functions f in $\mathscr{F}$).

$(\Phi p)(\Phi(x)) \Longleftrightarrow p(x)$ (for all predicates p in $\mathscr{F}$).

A homomorphism is called an isomorphism if and only if it is a one-to-one correspondence.

Theorem 1. *When $\mathscr{F}$ is $\langle\!\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle\!\rangle$ and $\mathscr{G}$ is $\langle\!\langle \boldsymbol{G}, \boldsymbol{Q} | \boldsymbol{N} \rangle\!\rangle$, $\Phi : \mathscr{F} \to \mathscr{G}$ is a homomorphism if and only if*

1) $\Phi(f)(\Phi(x))=\Phi(f(x))$ ($f \in \boldsymbol{F}$, *not for all* $f \in \mathscr{F}$),

$(\Phi p)(\Phi(x)) \Longleftrightarrow p(x)$ ($p \in \boldsymbol{P}$, *not for all* $p \in \mathscr{F}$).

2) *If f is a predicate or a function in $\mathscr{F}$, Φf is defined. Then if f is a predicate or a function in $\mathscr{F}$ by a flow-chart f, Φf is defined by Φf.*

Proof. a) If 1) and 2) hold, Φ is clearly homomorphism according to Lemma 2.

b) Conversely, let us assume that Φ is a homomorphism. Then we have 1) since $f \in F$ and $p \in P$ are defined by the flow-charts of Fig. 2. Next, $\Phi F \in G$ according to Lemma 2. And $\Phi\mathscr{F} \subseteqq \mathscr{G}$ by Lemma 3. q.e.d.

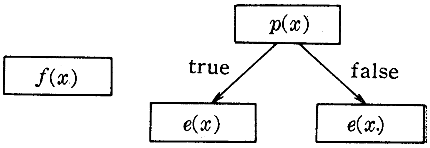


Fig. 2.

Theorem 1 means that a homomorphism of recursive categories can be induced by the mapping between their bases. When $\mathscr{F}=\langle\langle \boldsymbol{F}, \boldsymbol{P}|\boldsymbol{M}\rangle\rangle$ and $\mathscr{G}=\langle\langle \boldsymbol{G}, \boldsymbol{Q}|\boldsymbol{N}\rangle\rangle$. By $\varphi : \boldsymbol{M}\rightarrow\boldsymbol{N}$ we shall understand a mapping from $\boldsymbol{M}$ in $\boldsymbol{N}$. For $f \in \boldsymbol{F}$ and $p \in \boldsymbol{P}$, we define Φf, Φp as follows:

$$(\Phi f)(\varphi(x))=\varphi(f(x))$$

$$(\Phi p)(\varphi(x)) \Longleftrightarrow p(x) .$$

If $f \in \mathscr{F}$ is defined by a flow-chart F, we define Φf by the flow-chart ΦF. Then Φ is a homomorphism by Theorem 1, and we shall write it by $[\varphi]$. Then we have easily the following lemma.

Lemma 4. *Let $\Phi : \mathscr{F} \rightarrow \mathscr{G}$ be a homomorphism. If φ is a mapping from $\boldsymbol{M}$ in $\boldsymbol{N}$ defined by Φ, $\Phi=[\varphi]$.*

2. Recursive categories

2.1. The definition of recursive categories.

A recursive category $\mathfrak{R}$ is the category whose class of objects is the class of the recursive structures over a set $\boldsymbol{M}$. If $\mathscr{R}_1$ and $\mathscr{R}_2$ are objects in $\mathfrak{R}$, the morphism from $\mathscr{R}_1$ to $\mathscr{R}_2$ is the set of the homomorphisms from $\mathscr{R}_1$ in $\mathscr{R}_2$. We shall write $[\mathscr{R}_1, \mathscr{R}_2]$ to denote the morphism from $\mathscr{R}_1$ to $\mathscr{R}_2$. $1_{\mathfrak{R}1}$ is the identity mapping from $\mathscr{R}_1$ on itself.

When $\mathscr{R}_1=\langle\langle \boldsymbol{F}, \boldsymbol{P}|\boldsymbol{M}\rangle\rangle$ and $\mathscr{R}_2=\langle\langle \boldsymbol{G}, \boldsymbol{Q}, |\boldsymbol{M}\rangle\rangle$, every homomorphism Φ from $\mathscr{R}_1$ in $\mathscr{R}_2$ is induced by a mapping $\varphi : \boldsymbol{M}\rightarrow\boldsymbol{M}$. φ can be considered as a function in $\boldsymbol{M}$. Then we have the next theorem.

Theorem 2. *Under the above assumption, if φ is a function in*

$\mathscr{R}_1$, $\varphi\mathscr{R}_1 \subseteq \mathscr{R}_1$.

Proof. For all $f \in \boldsymbol{F}$ and $p \in \boldsymbol{P}$, φf and φp can be defined by flow-charts in $\mathscr{R}_1$. Therefore, when F is a flow-chart of $\mathscr{R}_1$, φF is in $\mathscr{R}_1$. q.e.d.

That is, we must consider φ which is not contained in $\mathscr{R}_1$, when we wish to observe a recursive category. This facts leads us to the concept "degrees of unsolvability", but the author will give an announcement about it in another paper.

In spite of it, morphisms and functors of a recursive category can be written as flow-charts by taking new functions and predicates other than the members of $\boldsymbol{F}$ and $\boldsymbol{P}$.

2.2. Flow-chart as diagrams.

Let F be a flow-chart of $\langle \boldsymbol{F}, \boldsymbol{P} | \boldsymbol{M} \rangle$. A flow-chart is a figure which is made by connecting blocks with arrows, and every blocks contains either a function or a predicate. So we have the following definition.

A scheme is a triple (I, M, d), where I is the set of vertices, M is a set of arrows, and d is a function from M to I. The function d shows the origins and the ends of arrows. It i and j are vertices, and if arrow m is $k \to j$, $d(m)=(i, j)$. If $F(i)$ is a function from I to $\boldsymbol{F}$ and $\boldsymbol{P}$, $\mathscr{F}$ can be considered as the pair $(\sum, F)$. That is, $F(i)$ shows the function or the predicate corresponding to the vertice i. We shall write flow-charts as such pairs, for example $(\sum, F)$. $F(i)$ is called a content of i.

Let $\mathscr{R}_1 = \langle\!\langle \boldsymbol{F}_1, \boldsymbol{P}_1 | \boldsymbol{M} \rangle\!\rangle$ and $\mathscr{R}_2 = \langle\!\langle \boldsymbol{F}_2, \boldsymbol{P}_2 | \boldsymbol{M} \rangle\!\rangle$ be two recursive structures in a recursive category $\mathfrak{R}$. According to Theorem 1, $\Phi : \mathscr{R}_1 \to \mathscr{R}_2$ is the function from flow-charts in $\mathscr{R}_1$ to flow-charts in $\mathscr{R}_2$ as follows.

1) If $f \in \boldsymbol{F}_1$, Φf is a flow-chart $F_f = (\sum_f, G_f)$
2) If $p \in \boldsymbol{P}_1$, Φp is a flow-chart $F_p = (\sum_p, G_p)$
3) If $F = (\sum, F) \in \mathscr{R}_1$, the flow-chart of $\varphi(F)$ is $(\sum^*, G)$

where $\sum^*$ is the insertion of $\sum_f$'s and $\sum_p$'s to $\sum$. The insertion is the substitution of a block i by a flow-chart $(\sum, G)$, for example as Fig. 3.

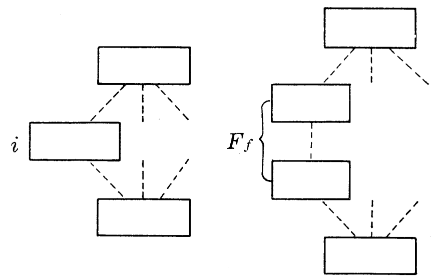


Fig. 3.

We define an ordering of vertices in a flow-chart arbitrarily, and write $\bar{0}, \bar{1}, \bar{2}, \cdots$ to denote them.

Theorem 2. *Let $F = (\sum, F)$ be a flow-chart in $\mathscr{F}$. If Φ is a homomorphism, Φ can considered as a function from $\mathscr{F}$ to $\Phi\mathscr{F}$. Then Φ can be written as a flow-chart whose contents are functions or* *predicates as follows:*

the function $\mathrm{Ver}_F\ (\bar{i})=$ *the next vertices of the vertice i,*

the function $\mathrm{Ins}_F\ (\bar{i}, G)=$ *the insertion of a flow-chart G to the vertice i in F.*

the predicate $\mathrm{Con}_F\ (\bar{i}) \Longleftrightarrow F$ *contains the vertice i.*

(Note: These functions and predicates have two or three variables. But as each of them are of different types, no confusion is expected in the substitution).

Proof. If we wish to give ΦF, we must test all vertices i of F, and substitute each content f to Φf. The procedure can be written in the form of a flow-chart. q.e.d.

The same theorem as Theorem 2 holds about homomorphisms between recursive structures over different bases.

3. Functors between recursive categories

3.1. Remarks about functions.

We shall consider functors between two recursive categories in the following. The concept of recursive categories can be regarded as an abstraction of so-called algorithms. Therefore, functors will be formulations of meta-mathematical concepts about algorithms, and some examples will be given in the following.

The functor "$\mathfrak{Hom}$" is considered to define equivalency of algorithms in terms of a category. "$\mathfrak{Univ}$" is used to give a definition of universal algorithms.

3.2. $\mathfrak{Hom}$.

Consider two recursive categories $\mathfrak{R}_1$ and $\mathfrak{R}_2$. Let φ be a univalent mapping from $\mathscr{B}$ on $\mathscr{C}$, where $\mathscr{B}$ and $\mathscr{C}$ are the bases of $\mathfrak{R}_1$ and $\mathfrak{R}_2$ respectively.

If $\mathfrak{Hom}_\varphi$ is defined by the following rule, where $\mathscr{R}_1$ and $\mathscr{R}_2$ are objects in $\mathscr{R}_1$,

$$\mathfrak{Hom}_\varphi\, \mathscr{R}_1=\varphi(\mathscr{R}_1)$$

$$\mathfrak{Hom}_\varphi\, (\Phi)(\varphi(\mathscr{R}_1))=\Phi(\mathscr{R}_2)\ .$$

Theorem 3. $\mathfrak{Hom}_\varphi$ *is a functor.*

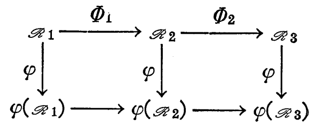
Proof. 1) $1_{\mathscr{R}}=[\varphi]$, $1_{\varphi[\mathscr{R}_1]}=[\varphi\mathscr{R}_1)]$. Therefore $\mathfrak{Hom}_\varphi(1_{\mathscr{R}_1})=1_{\varphi(\mathscr{R}_1)}$.

2) Let $\Phi_1 : \mathscr{R}_1 \rightarrow \mathscr{R}_2$ and $\Phi_2 : \mathscr{R}_2 \rightarrow \mathscr{R}_3$. Then,

$\mathfrak{Hom}_\varphi(\Phi_1)(\varphi(\mathscr{R}_1))=\varphi_1(\mathscr{R}_2)$

$\mathfrak{Hom}_\varphi(\Phi_2)(\varphi(\mathscr{R}_2))=\varphi_2(\mathscr{R}_3)$.

$\mathfrak{Hom}_\varphi(\Phi_1)\,\mathfrak{Hom}_\varphi(\Phi_2)(\varphi(\mathscr{R}_1))=(\varphi(\mathscr{R}_3))$.



On the other hand, as $\Phi_1\Phi_2\ \mathscr{R}_1\to\mathscr{R}_3$, we have

$$\mathfrak{Hom}_\varphi(\Phi_1\Phi_2)(\varphi(\mathscr{R}_1))=\varphi(\mathscr{R}_3)\ .$$

$$\mathfrak{Hom}_\varphi(\Phi_1\Phi_2)=\mathfrak{Hom}_\varphi(\Phi_1)\,\mathfrak{Hom}_\varphi(\Phi_2)\ .$$ q.e.d.

4. Universality

4.1. Universal structures.

By $\mathscr{F}$ we shall understand a recursive structure $\langle\!\langle \boldsymbol{F}, \boldsymbol{P}|\boldsymbol{M}\rangle\!\rangle$. Let (x, F) be a pair of $x\in\boldsymbol{M}$ and a flow-chart $F\in\mathscr{F}$. The function Comp is defined by the rule that

$$\mathrm{Comp}_{\mathscr{F}}((x, F))=(F(x), e(x))$$

where F is a flow-chart. If F is a logical flow-chart, Comp $((x, F))$ is $(true, e(x))$ or $(false, e(x))$ whether $F(x)$ is true or not. Let U_F be the set of all pairs (x, F), $F_u=\{\mathrm{Comp}_F\}$, and P be empty. Then we have a recursive structure $U_F=\langle\!\langle F_u, P_u|U_F\rangle\!\rangle$. U_F is called a universal closure of F.

Lemma 5. *For every flow-chart* F_u of U, *we have*

$$F_u((x, F))=\mathrm{Comp}\,(x, F)\ .$$

if the left side can be defined.

Proof. According to the definition of the function "Comp", the values of the $F_u((x, F))$ must be $(F(x), e(x))$. For $(F(x), e(x))$ we have only the value $(F(x), e(x))$ after any computation. q.e.d.

Theorem 4. *If* φ *is the mapping*: $F\to U$ *that* $\varphi(x)=(x, e(x))$,

$$F_\varphi\supseteq U_F\ .$$

Proof. By Lemma 2 and Lemma 4.

4. The functor "$\mathfrak{Univ}$".

Let $\mathfrak{R}$ be a recursive category, and $\mathscr{R}$ be its object. By $U_v(\mathscr{R})$ we shall understand the universal closure for $\mathscr{R}$. If $\Phi:\mathscr{R}_1\to\mathscr{R}_2$ is a homomorphism, the rule that

$$\bar{\Phi}((x, F))=(\Phi(x), \Phi F)$$

$$\bar{\Phi}(\mathrm{Comp}_{\mathscr{F}}((x, F))) = \mathrm{Comp}_{\Phi(\mathscr{F})}(\Phi(x), \Phi(F))$$

gives us a homomorphism $\bar{\Phi} : U_v(\mathscr{R}_1) \rightarrow U_v(\mathscr{R}_2)$. We have a category $U_v(R)$ whose objects are $U_v(\mathscr{R})$'s, and whose morphisms are the set of $\bar{\Phi}$'s. The category is much different from a recursive category, since the bases of its objects are different each other.

If the mapping $\mathfrak{Univ}$ is defined by the rule that

1) $\mathfrak{Univ}(\mathscr{R}) = U_v(\mathscr{R})$.

2) $\mathfrak{Univ}(\Phi)(\mathscr{R}_1) = U_v(\mathscr{R}_2)$, where $\Phi : \mathscr{R}_1 \rightarrow \mathscr{R}_2$ is a morphism of $\mathfrak{R}$,

we have easily,

Theorem 5. $\mathfrak{Univ}$ *is a functor from* $\mathfrak{R}$ *to* $U_v(\mathfrak{R})$.

4.2. Comp as a function of flow-state.

Let F be a flow-charts, and i be one of its vertice. We substitute an element $m \in \mathfrak{M}$ to the content $f(x)$ of i. By $F(m; i)$ we shall understand the figure, and it is called a flow-state.

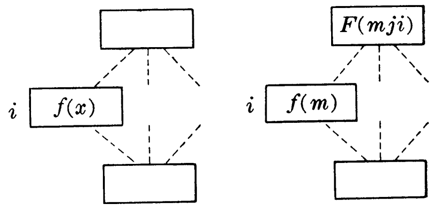


Fig. 4.

Theorem 6. *The function* Comp *can be written as a flow-chart whose contents are following functions and predicates:*

1) *functions.*

a) $\varphi_1((m, F)) = F(\bar{m}, 1)$,
where 1 *is the beginning vertice of* F.

b) $\varphi_2(F(m, \bar{i}_1)) = F(m', \bar{i}_2)$,
where the next vertice of i *along the arrow is* i,
m' *is the value gained in the vertice* i.

c) $\varphi_3(F(m, \bar{i}_1)) = F(m, \bar{i}_2)$,
where the next vertice of $\bar{i}_1$ *along the truth arrow is* i.

d) $\varphi_4(F(m, \bar{i}_1)) = F(m, \bar{i}_2)$,
where the next vertice of $\bar{i}_1$ *along the false arrow is* i.

2) *predicates.*

a) Pred $(F(m; \bar{i}_1)) \Longleftrightarrow$ *the contents of the vertice* $\bar{i}$ *is a predicate.*

b) True $(F(m; \bar{i}_1)) \Longleftrightarrow p_i(m)$ *is true, where* p_i *is the predicate in the vertice* $\bar{i}$.

c) False$(F(m; \bar{i}_1)) \Longleftrightarrow p_i(m)$ *is false, where* p_i *is the predicate in the vertice* $\bar{i}$.

d) End $(F(m; \bar{i})) \Longleftrightarrow$ *the vertice* $\bar{i}$ *is the end-vertice.*

e) Et $(F(m; \bar{i})) \Longleftrightarrow$ *the vertice* $\bar{i}$ *is the true-end vertice.*

f) Ef $(F(m; \bar{i})) \Longleftrightarrow$ *the vertice* $\bar{i}$ *is the false-end vertice.*

Proof. The computation of f to an element m M is consisted in the following procedures;

a) substitution of m to the beginning vertice,

b) decision whether a vertice is the end vertice or not,

c) decision whether the content of a vertice is a function or a predicate,

d) substitution of an element to the variable of the content of a vertice,

e) decision whether the content is true or not, if the content is a predicate,

f) decision of the vertice to go in the following step.

These procedures can easily be written as a flow-chart with the functions and predicates in the above. The idea of the flow-chart is given in Fig. 5. q.e.d.

4.3. Universal structures.

Generally speaking, the function "Comp" can not be considered as a function in a recursive structure $\mathscr{F}$. But in special cases, "Comp" can be given as a flow-chart of $\mathscr{F}$.

If a recursive structure $\mathscr{F}$ is isomorphic to $U_v(\mathscr{F})$, $\mathscr{F}$ is called universal. That is, $\mathscr{F}$ is universal if and only if there is an isomor-

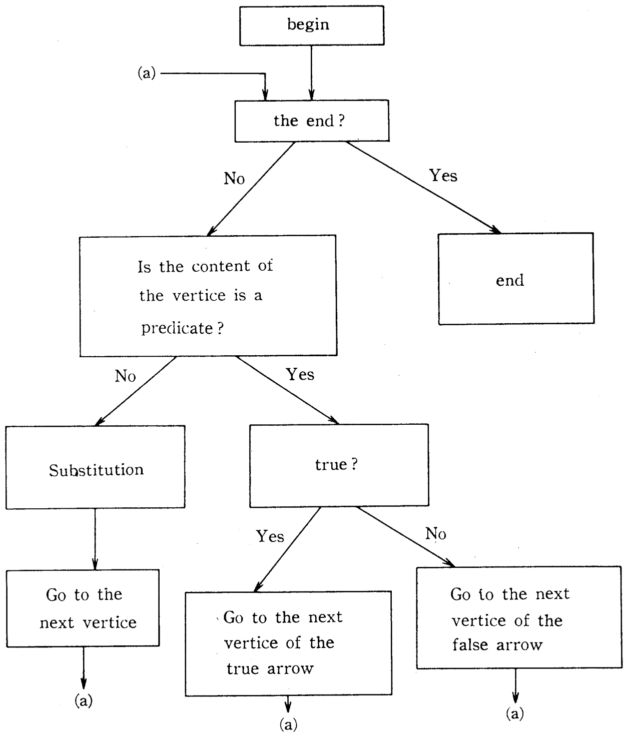


Fig. 5.

phism: $\mathscr{F} \rightarrow U_v(\mathscr{F})$.

Lemma 5. *If $\mathscr{F}$ is a recursive structure, $U_v(U_v(\mathscr{F}))$ is isomorphic to $U_v(\mathscr{F})$.*

Proof. The base of $U_v(U_v(\mathscr{F}))$ consists of elements of $((x, F), F_U)$ where F_U is a flow-chart in $U_v(\mathscr{F})$. By Lemma, we can only define $\mathrm{Comp}_{\mathscr{F}}$ by F_U. If $\varphi((x, F), F_U)=(x, F)$, φ is an isomorphism $U_v(U_v(\mathscr{F}))\rightarrow U_v(\mathscr{F})$. q.e.d.

A flow-chart can be written as a pair $(\sum, F)$ as mentioned in the section 2.2, so it can be written as a sequence of symbols. That is, arrows can be written as $i_1 \rightarrow i_2$, so a flow-chart can be given as a sequence of such formulas as

$$i_1;\ F(i_1)\rightarrow i_2;\ F(i_2)\ .$$

For example, the flow-chart of the Fig. 4 can be written;

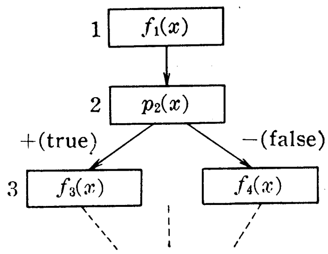


Fig. 6.

$$1;\ f_1(x)\rightarrow 2;\ p_2(x):\ 2;\ p_2(x)\rightarrow 3;\ f_3(x):\ 2;\ p_2(x)\rightarrow f_4(x):\ \cdots .$$

Flow-states can also be written in similar form as above. By $W(F)$ we shall understand the sequence corresponded to a flow-chart $\mathscr{F}$. If we call such sequences flow-words, the functions and predicates in Theorem 6 can be rewritten as a functions and predicates of flow-words. Moreover, these functions and predicates are of the following types;

1) functions each of which substitutes a specified consecutive part of a word to other specified word,
2) predicates each of which states that a word contains a specified word as its consecutive part,
3) the functions $f_i(x)$ each of which is a content of F,
4) the predicates $p_i(x)$ each of which is a content of F.

Let W be a set of flow-words of a recursive structure F. By U we shall understand the set of the functions to define "Comp", and by V the set of the predicates to define it.

Theorem 7. *A recursive structure $\mathscr{R}$ is universal, if and only if there is such an isomorphism φ that*

$$\varphi(R)\subset\langle\!\langle U, V|W\rangle\!\rangle$$

Proof. R is universal if and only if $\varphi(\mathrm{Comp}_{\mathscr{R}})\in\varphi(\mathscr{R})$. And it is equivalent to (1). q.e.d.

4.5. Strictly universal categories.

In the preceding part, we didn't restrict the type of functions and predicates to define a recursive category. If we restrict the type of functions or predicates in a structure $\langle\langle F, P|M\rangle\rangle$ to 1) and 2) in 4.4. where M is the set of word, we call the category as a recursive structure of words. The category, whose objects are recursive structures of words, is called the recursive category of words.

Let $\mathscr{W}$ be a recursive structure of words, and $\mathscr{R}$ be a recursive structure. If $\mathscr{R}$ is universal and $\varphi(\mathrm{Comp}_{\mathscr{R}}) \in \mathscr{W}$ where φ is the isomorphism in Theorem 7, we call $\mathscr{R}$ is strictly universal.

If there is an isomorphism $\mathscr{R}_1 \rightarrow \mathscr{R}_2$ where $\mathscr{R}_1$ and $\mathscr{R}_2$ are recursive structures, $\mathscr{R}_1$ is isomorphic to $\mathscr{R}_2$. If every object of a recursive category $\mathscr{R}_1$ is isomorphic to an object of another category $\mathscr{R}_2$, we call $\mathscr{R}_1$ is isomorphic to $\mathscr{R}_2$.

By Theorem 7, we can easily have the next theorem:

Theorem 8. *If $\mathscr{R}$ is a strictly universal recursive structure, $\mathscr{R}$ is isomorphic to a recursive structure of words.*

If every object of a recursive category is strictly universal, we call that the category is strictly universal.

By Theorem 7, we also have:

Theorem 9. *A recursive category $\mathfrak{R}$ is strictly universal if and only if it is isomorphic to the recursive category of words.*

References

[1] Mitchell, B., Theory of Categories. Academic Press, 1965.

[2] Zama, N., On models of algorithms and flow-charts, Commentarii mathematici Universitatis Sancti Pauli XIV (1966) 123–134.

[3] Zama, N., On a generalization of algorithms and one of its applications (I), ibid XV (1967) 110–116.