BULLETIN DE L'ACADÉMIE
POLONAISE DES SCIENCES
Série des sciences math., astr.
et phys. — Vol. XXIII, No. 3, 1975

MATHEMATICS
(COMPUTER SCIENCE)

An Axiomatic Approach to the Theory of Data Structures

by

L. BANACHOWSKI

Presented by A. MOSTOWSKI on October 8, 1974

Summary. In this paper the axiomatizations of relational systems over data structures are considered by means of algorithmic logic [3]. The categorical axiomatization for the system of trees is given. The constructivity [4] of the system of trees is proved. The connections between data structures and regular languages are examined. In order to systematize the knowledge of the theory of data processing it was attempted to give a uniform model covering various types of data structures, for example [2]. In the present paper we accept the model first introduced in [1].

1. Formalized model of data structures. Let $\mathscr{J}$ be a finite nonempty set of objects. Let N be an infinite set disjoint with $\mathscr{J}$. Its elements will be called names. The set of forms Φ is the least set of expressions satisfying the following conditions:

1) $N \cup \mathscr{J} \cup \{[\ ]\} \subset \Phi$,
2) if $\varphi_1, \ldots, \varphi_m$ are forms and $\varphi_m \neq [\ ]$, then $[\varphi_1\ \varphi_2 \ldots \varphi_m]$ is a form.

By M we denote the set of all partial functions $m: N \to \Phi$ such that:

1) the domain of m is finite,
2) if $n \in \mathrm{Dom}(m)$ then $m(n) \notin \mathscr{J} \cup N \cup \{[\ ]\}$,
3) every name occurring in a form $m(n)$ for some $n \in \mathrm{Dom}(m)$ belongs to $\mathrm{Dom}(m)$.

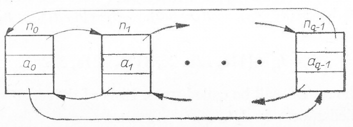
Elements of M are called memory states. Let m be a memory state. By L^m we shall denote the set of all pairs $\langle \varphi; m \rangle$ such that:

1) φ is a form,
2) every name occurring in φ belongs to $\mathrm{Dom}(m)$.

Elements of L^m are called lists with a memory state m. Let L be the set of all lists, i.e. $L = \bigcup_{m \in M} L^m$.

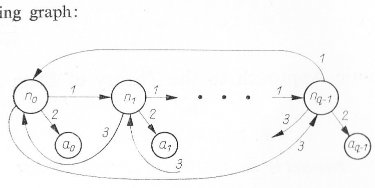
Lists possess the following two interpretations:

1) *Data structures.* Forms represent fields of memory which can be named by names from N. For example, the following ring



can be written as $l = \langle n_0; \{n_i \to [n_{(i-1) \bmod q}\, a_i\, n_{(i+1) \bmod q}]\}\ 0 \leqslant i < q\rangle$, where $n_0, n_1, \ldots, n_{q-1}$ are names of fields, $a_0, a_1, \ldots, a_{q-1}$ are objects.

2) *Graphs with ordered edges.* Names and objects can be considered as vertices of a graph. Objects are terminal vertices. For example the list 1 quoted above represents the following graph:



The set $W(\varphi)$ of all occurrences of subforms in a form φ is the least set satisfying the following conditions:

1) each element of $W(\varphi)$ is a pair (i, ψ), where $i \in \omega^*$, $\psi \in \Phi$,
2) (e, x) is in $W(\varphi)$,
3) if (i, ψ) is in $W(\varphi)$ and $\psi = [\psi_1 \psi_2 \ldots \psi_m]$, then for each $j = 1, \ldots, m$, $(i \cdot j, \psi_j) \in W(\varphi)$.

By a grammar of a list $l = \langle \varphi_0; \{n_i \to \varphi_i\}_{1 \leqslant i \leqslant m}\rangle$ we mean the quadruplet $G_l = (V, T, P, n_0)$ such that:

(1) $V = \omega \cup N \cup \mathcal{J}$,
(2) $T = \omega \cup \mathcal{J}$,
(3) n_0 is a name different from $n_1, \ldots, n_m$,
(4) $$P = \{n_i \to j\psi \mid \psi \in N \cup J,\ (j, \psi) \in W(\varphi_i),\ 0 \leqslant i \leqslant m\}.$$

By an information of a list 1 we mean the language I_l generated by the grammer G_l.

Two lists $l, l' \in L$ are equiva'ent, $l \sim l'$, provided $I_l = I_{l'}$. Since the information I_l of a list l is a regu'ar language, then the equivalence of lists is decidable.

It is proved [2] that for any list l we can effectively find out a list l' such that:

(1) l' is equivalent to l,
(2) l' needs the least area of memory to be placed among all the lists equivalent to l.

For example, the following two lists are equivalent:

$$l_1 = \langle [[ab]\ c\ [[a]\ [ab]]];\ \emptyset \rangle$$

and

$$l_2 = \langle n_0;\ n_0 \to [n_1\ cn_2],\quad n_1 \to [ab],\quad n_2 \to [n_3\ n_1],\quad n_3 \to [a] \rangle.$$

Namely,

$$I_{l_1} = I_{l_2} = \{11a, 12b, 2c, 311a, 321a, 322b\}.$$

Elements of the set $L^{\emptyset}$ will be called trees. Lists of the set $A^m = \{\langle j; m\rangle \mid j \in \mathcal{J}, m \in$

2. Systems of lists. We shall impose an algebraical, LISP-like structure upon the set of lists.

Let v and g be the following auxiliary functions:

$$v(l) = \begin{cases} s(n) & \text{if } \varphi = n \in N, \\ \varphi & \text{otherwise;} \end{cases}$$

$$g_l = \begin{cases} k & \text{if } v(l) = [\varphi_1, \ldots, \varphi_k], \\ 0 & \text{otherwise.} \end{cases}$$

Let nil, car, cdr, cons be the following operations in the set L^s:

$$\text{nil} = \langle [\ ]; s\rangle,$$

$$\text{car}(l) = \begin{cases} \langle \varphi_1; s\rangle & \text{if } v(l) = [\varphi_1 \ldots \varphi_k],\ k \geqslant 1, \\ \text{nil} & \text{otherwise;} \end{cases}$$

$$\text{cdr}(l) = \begin{cases} \langle [\varphi_2 \ldots \varphi_k]; s\rangle & \text{if } v(l) = [\varphi_1 \ldots \varphi_k],\ k \geqslant 1, \\ \text{nil} & \text{otherwise;} \end{cases}$$

$$\text{cons}(l, l') = \begin{cases} \langle [\varphi\, \psi_1 \ldots \psi_k]; s\rangle & \text{if } v(l') = [\psi_1 \ldots \psi_k],\ k \geqslant 0, \\ \langle [\varphi]; s\rangle & \text{otherwise.} \end{cases}$$

The algebra $\mathcal{L}^s = \langle L^s, A^s \cup \{\text{nil, car, cdr, cons}\}\rangle$ is called the algebra of lists with the memory state s.

Now we shall extend the operation cons on the whole set L. Let $l = \langle \varphi; s\rangle$, $k = \langle \psi; m\rangle$ be two arbitrary lists. Let χ: Dom $(m) \xrightarrow{1-1} N$-Dom (s).

If λ is a form then by $\overline{\chi\lambda}$ we denote the form obtained from λ by the simultaneous replacement of all occurrences of names $n \in$ Dom (χ) by $\chi(n)$, respectively.

Let $l' = \langle \varphi'; s'\rangle$, $k' = \langle \psi'; s'\rangle$ be the following lists:

(1) $\varphi' = \varphi$,
(2) $\psi' = \overline{\chi\psi}$,
(3) Dom (s') = Dom $(s) \cup \chi$ (Dom (m)),
(4) if $n \in$ Dom (s) then $s'(n) = s(n)$,
(5) if $n \in \chi$ (Dom (m)) then $s'(n) = \overline{\chi m(\chi^{-1}(n))}$.

We put cons (l, k) = cons (l', k') where the value cons (l', k') is calculated in $\mathcal{L}^{s'}$.

Let $A = A^{\emptyset}$.

The algebra $\mathcal{L} = \langle L, A \cup \{\text{nil, car, cdr, cons}\}\rangle$ is called the algebra of lists. If $X \subset (\mathcal{J} \cup \omega)^*$, $j \in \omega$ then by $X|j$ we denote the set $\{x \in (\mathcal{J} \cup \omega)^* \mid jx \in X\}$.

We shall consider the following relational system of regular languages over the alphabet $\mathcal{J} \cup \omega$, $\mathcal{R} = \langle R, \{\{j\}\}_{j \in \mathcal{J}} \cup \{\emptyset, |1, \|, \otimes\}\rangle$, where:

(1) $R = \{I_l \mid l \in L\}$,
(2) $\emptyset$ is the empty language,
(3) $$X\| = \bigcup_{j=2}^{\infty} (j-1) \cdot (X/j),$$
(4) $X \otimes Y = 1 \cdot X \cup \bigcup^{\infty} (j+1) \cdot (Y/j)$

The following lemma is fundamental:

LEMMA 1. *Let r^m $(m \in M)$ and r be the following mappings:*

$$r^m(l) = I_l \quad \text{for } l \in L^m,$$
$$r(l) = I_l \quad \text{for } l \in L.$$

(1) *The mappings r^m are homomorphisms from $\mathscr{L}^m$ into $\mathscr{R}$.*
(2) *The mapping r is an epimorphism from $\mathscr{L}$ onto R.*
(3) *The relation $\sim$ is a congruence of the systems $\mathscr{L}^m$ and $\mathscr{L}$.*

Let $|l|$ denote the equivalence class of the relation containing l. By $|\mathscr{L}^m| = \langle |L^m|, A \cup \{\varepsilon, h, t, c\}\rangle$ we denote the quotient algebra of $\mathscr{L}^m$ by $\sim$.

By $|\mathscr{L}| = \langle |L|, A \cup \{\varepsilon, h, t, c\}\rangle$ we denote the quotient algebra of $\mathscr{L}$ by $\sim$. In the sequel we shall concentrate on an examination of the algebras $|\mathscr{L}^m|$ and $|\mathscr{L}|$.

The following lemma justifies such an approach.

LEMMA 2. *Let $\overline{\mathscr{L}}^m$, $\overline{\mathscr{L}}$, $|\overline{\mathscr{L}}^m|$ and $|\overline{\mathscr{L}}|$ be the following relational systems:*

$$\overline{\mathscr{L}}^m = \langle L^m, A^m \cup \{\text{nil, car, cdr, cons}\} \cup \{\sim\}\rangle,$$
$$\overline{\mathscr{L}} = \langle L, A \cup \{\text{nil, car, cdr, cons}\} \cup \{\sim\}\rangle,$$
$$|\overline{\mathscr{L}}^m| = \langle |L^m|, A \cup \{\varepsilon, \mathrm{h, t, c}\} \cup \{=\}\rangle,$$
$$|\overline{\mathscr{L}}| = \langle |L|, A \cup \{\varepsilon, h, t, c\} \cup \{=\}\rangle,$$

where m is a memory state in M.

For every algorithmic formula $\alpha(x_1, \ldots, x_k)$

$$\alpha_{\overline{\mathscr{L}}^m}(l_1, \ldots, l_k) = \alpha_{|\overline{\mathscr{L}}^m|}(|l_1|, \ldots, |l_k|) \quad \text{for any lists } l_1, \ldots, l_k \in L^m,$$
$$\alpha_{\overline{\mathscr{L}}}(l_1, \ldots, l_k) = \alpha_{|\overline{\mathscr{L}}|}(|l_1|, \ldots, |l_k|) \quad \text{for any lists } l_1, \ldots, l_k \in L.$$

The above lemma follows from a theorem on isomorphism for the algorithmic logic [5].

By Lemma 1 we get

LEMMA 3.
3.1. *$|\mathscr{L}|$ is isomorphic with $\mathscr{R}$.*
3.2. *Every algebra $|\mathscr{L}^m|$ is a subalgebra of $|\mathscr{L}|$.*
3.3. *$\mathscr{L}^{\varnothing}$ is isomorphic with $|\mathscr{L}^{\varnothing}|$.*

In $|\mathscr{L}|$ we can construct only trees by means of constants of $|\mathscr{L}|$, i.e. by means of atoms and the empty list ε. Hence the system $|\mathscr{L}|$ is not constructive. However, $|\mathscr{L}|$ possesses properties similar to those of constructive systems.

For example, the natural numbers are definable in $|\mathscr{L}|$. Namely, let $j \in \mathscr{J}$, $\hat{\imath} = |\langle[\underbrace{j \ldots j}_{i\text{-times}}]; \varnothing\rangle|$.

As zero we take $\hat{0}$ and as a successor the following function

$$S(\hat{\imath}) = c(|\langle j; \varnothing\rangle|, \hat{\imath}).$$

In the system $|\mathscr{L}|$ a stack can be formed. Namely,

empty stack $= \varepsilon$,

put x onto stack $y - y := \mathrm{c}(x, y)$,

take the top of stack $y = h(y)$,

delete the top of stack $y - y := t(y)$.

Hence recursive procedures with parameters called by value can be replaced by equivalent nonrecursive ones.

In the sequel we shall assume that $A = \{a_1, \ldots, a_r\}$ for some $r > 0$ and we shall use the abbreviation $a(x)$ for $x = a_1 \vee \ldots \vee x = a_r$.

3. Categorical axiomatization of the relational system of trees.

LEMMA 4. *The system of lists $|\mathscr{L}|$ is a model of the axioms for identity and of the following formulas:*

A1) $$\bigwedge_{\substack{i,j=1\\ i \neq j}}^{r} (a_i \neq a_j),$$

A2) $$\neg a(\varepsilon) \wedge \neg a(c(x, y)) \wedge \neg a(t(x)),$$

A3) $$c(x, y) = \varepsilon \Leftrightarrow x = \varepsilon \wedge (y = \varepsilon \vee a(y)),$$

A4) $$\neg a(y) \wedge \neg a(u) \wedge c(x, y) = c(z, u) \Rightarrow x = z \wedge y = u,$$

A5) $$a(y) \Rightarrow c(x, y) = c(x, \varepsilon),$$

A6) $$a(x) \Rightarrow (h(x) = \varepsilon \wedge t(x) = \varepsilon),$$

A7) $$h(c(x, y)) = x,$$

A8) $$\neg a(y) \Rightarrow t(c(x, y)) = y,$$

A9) $$\neg a(x) \Rightarrow x = c(h(x), t(x)).$$

The system of trees $\mathscr{L}^{\varnothing}$ is a model for the following formula.

T) $$K_T(x)\, \mathbf{1},$$

where

$$K_T(x)\colon * \big[x \neq \varepsilon \veebar \big[a(h(x)) \vee h(x) = \varepsilon [x/t(x)]\, \big[x/c\big(h(h(x)), c\big(t(h(x)), t(x)\big)\big)\big]\big]\big].$$

Let us observe that $K_T(x)\,\mathbf{1}$ is equivalent to the fact of halting of the following recursive procedure

Tree $(x) \Leftarrow$ if a $(x) \vee x = \varepsilon$ then true else Tree $(h(x)) \wedge$ Tree $(t(x))$.

The proof of Lemma 4 can be carried out by applying either the definition of the system $|\mathscr{L}|$ or Lemma 1. For example, we shall prove by the second method that $|\mathscr{L}|$ is a model for A4).

Let l, k, l', k' be any lists such that:

(1) $$I_k \cap J = \emptyset,$$

(2) $$I_{k'} \cap J = \emptyset,$$

(3) $$I_l \otimes I_k = I_{l'} \otimes I_{k'}$$

i.e.

$$1 \cdot I_l \cup \bigcup_{j=1}^{g_k} (j+1) \cdot (I_k/j) = 1 \cdot I_{l'} \cup \bigcup_{j=1}^{g_{k'}} (j+1) \cdot (I_{k'}/j).$$

Hence $I_l = I_{l'}$, i.e. $|l| = |l'|$ and for every j, $I_k/j = I_{k'}/j$. From the last equations by (1) and (2) it follows that

$$I_k = I_{k'}, \quad \text{i.e. } |k| = |k'|.$$

THEOREM 1. *Formulas* A1)—A9) *and* T) *form a categorical axiomatization of the system of trees.*

Sketch of the proof. Let $\mathfrak{B} = \langle B, A \cup \{\varepsilon, h, t, c\}\rangle$ be a model for A0)—A9) and T). We shall define an isomorphism from $\mathscr{L}^{\varnothing}$ onto $\mathfrak{B}$. We use the fact that $L^{\varnothing}$ is the least set satisfying the following conditions:

(1) $$A \cup \{\text{nil}\} \subset L^{\varnothing},$$

(2) if $l \in L^{\varnothing}$, $k \in L^{\varnothing} A$ and ($l \neq \text{nil}$ or $k = \text{nil}$) then cons $(l, k) \in L^{\varnothing}$.

Now we put:

(i) $$f(a^{\mathscr{L}^{\varnothing}}) = a^{\mathfrak{B}} \quad \text{for each } a \in A,$$

(ii) $$f(\text{nil}) = \varepsilon,$$

(iii) $f(\text{cons}(l, k)) = c(f(l), f(k))$ for trees l, k satisfying the condition in (2) above.

The only more difficult case is to prove that f is a mapping onto $\mathfrak{B}$.

Axiom T) secures that every element $x \in B$ can be decomposed into elementary components of the set $A \cup \{\varepsilon\} \subset f(L^{\varnothing})$. Axiom A9) makes it possible to compose x of elementary components by decomposing it beforehand.

THEOREM 2. *The system of trees is constructive.*

Sketch of the proof. We shall use the shorter notation of the system $|\mathscr{L}^{\varnothing}|$ which is isomorphic with $\mathscr{L}^{\varnothing}$. We can regard trees of the form $c(l, k)$ as pairs (l, k). We can order all trees analogously to the ordering of all pairs of natural numbers, i.e. as shown in the Table.

TABLE

ε

	$c(c(a_1, \varepsilon), \varepsilon)$	$c(c(a_2, \varepsilon), \varepsilon)$	
$c(\varepsilon, c(a_1, \varepsilon))$	$c(c(a_1, \varepsilon), c(a_1, \varepsilon))$	↗	
$c(\varepsilon, c(a_2, \varepsilon))$	↗		
↗			

$\xrightarrow{a_1, \ldots, a_r}$

$\xrightarrow{c(a_1, \varepsilon), \ldots, c(a_r, \varepsilon)}$

$\xrightarrow{c(a_1, c(a_1, \varepsilon)), \ldots, c(a_r, c(a_1, \varepsilon))}$

$\xrightarrow{c(a_1, c(a_2, \varepsilon)), \ldots, c(a_r, c(a_2, \varepsilon))}$

. . . .

The successor and the predecessor of a tree in this ordering can be defined as follows:

$$S(d)=\begin{cases} a_{i+1} & \text{if } d=a_i,\ i=1,\ldots,r-1 \\ \varepsilon & \text{if } d=a_r, \\ c(a_1,\varepsilon) & \text{if } d=\varepsilon, \\ c\big(a_1, c(a_1,\varepsilon)\big) & \text{if } d=c(a_r,\varepsilon), \\ c\big(S(a_i), y\big) & \text{if } d=c(a_i,y),\ i=1,\ldots,r, \\ c\big(a_1, S(x)\big) & \text{if } d=c(x,\varepsilon), \\ c\big(S(x), P(y)\big) & \text{if } d=c(x,y), \end{cases}$$

$$P(d)=\begin{cases} \text{undefined} & \text{if } d=a_1, \\ a_{i-1} & \text{if } d=a_1, \\ a_r & \text{if } d=\varepsilon, \\ c(a_r,\varepsilon) & \text{if } d=c\big(a_1, c(a_1,\varepsilon)\big), \\ c\big(P(x),\varepsilon\big) & \text{if } d=c(a_1,x), \\ c\big(P(x), y\big) & \text{if } d=c(x,y),\ x=a_2,\ldots,a_r,\varepsilon, \\ c\big(P(x), S(y)\big) & \text{if } d=c(x,y). \end{cases}$$

Since in $\mathscr{L}^{\varnothing}$ we can define the system of natural numbers and use a stack then the successor S is programmable in $\mathscr{L}^{\varnothing}$.

Conversely, in the system $\langle L^{\varnothing}, S, a_1\rangle$ which is isomorphic with the system of natural numbers we can calculate the numbers of the trees $c(x,y)$, $h(x)$ and $t(x)$ from the numbers of x and y.

Since the number of a tree is this tree, then the functions c, h and t are programmable in $\langle L^{\varnothing}, S, a_1\rangle$.

In algorithmic logic we can express the finiteness of lists by means of the following axiom analogous to axiom T) for trees:

$$L)\qquad K_L(x)\,\mathbf{1},\quad \text{where}$$

$$K_L(x):\circ\big[[v/c(x,\varepsilon)]*\big[\neg a(x)\wedge x\neq\varepsilon\circ\big[[u/h(x),\ x/t(x)]\,\underline{\vee}$$
$$\underline{\vee}\big[u=\varepsilon\vee a(u)\,[\ ]\circ\big[\mathrm{Is}\ u,v;z)\,\underline{\vee}\big[z\,[\ ]\circ[[v/c(u,v)]\,\mathrm{Append}\,(u,x)]]]]]]],$$

$$\mathrm{Is}\,(u,v;z):\circ\big[[z/\mathbf{0},\ y/v]*\big[y\neq\varepsilon\wedge\neg z\ \underline{\vee}\big[u=h(y)\ [z/\mathbf{1}]\ [y\,|\,t(y)]]]]],$$

$$\mathrm{Append}\ (u,x):*\big[u\neq\varepsilon\ \big[x/c\big(h(u),x\big),\ u/t(u)\big]\big].$$

To prove that $|\mathscr{L}|$ is a model for axiom L) it is sufficient to observe that for any regular language $X\in R$, the set $\{X/a|a\in\omega^*\}$ is finite. This implies that the program K_L halts in $\mathscr{R}$ for any $X\in R$.

By Lemma 3, K_L halts in $|\mathscr{L}|$ for any $|l|\in|L|$.

Since the system of lists possesses subsystems not isomorphic with it, hence there is no categorical axiomatization for it in algorithmic logic. However, such axiomatization may exist in algorithmic logic extended by classical quantifiers. The system $|\mathscr{L}|$ is not constructive and the question arises what operation should be adjointed to $|\mathscr{L}|$ in order to obtain a constructive system.

I would like to thank Dr A. Sawicki for his valuable aid during the preparation of this manuscript.

INSTITUTE OF MATHEMATICAL MACHINES, UNIVERSITY, 00-901 WARSAW
(INSTYTUT MASZYN MATEMATYCZNYCH, UNIWERSYTET WARSZAWSKI)

REFERENCES

[1] L. Banachowski, *Formalization of the notions of data structures and programs on data structures*, Reports of the Warsaw University Computation Centre, No. 46, 1974 [in Polish].

[2] W. Turski, *Data structure*, Wyd. Naukowo-Techniczne, Warsaw, 1971 [in Polish].

[3] A. Salwicki, *Formalized algorithmic languages*, Bull. Acad. Polon. Sci., Sér. Sci. Math. Astronom. Phys., **18** (1970), 227—232.

[4] — , *Programmability and recursiveness*, Warsaw University, 1974.

[5] G. Mirkowska, *Algorithmic logic and its applications in the theory of programs*, doctoral dissertation, Warsaw University, 1972 [in Polish].

Л. Банаховски, **Аксиоматический подход к теории структур данных**

Содержание. На базе алгоритмической логики рассматриваются аксиоматизации алгебраических систем над структурами данных. Приводится категоричная аксиоматизация систем деревьев. Доказывается коструктивность системы деревьев. Исследуются соотношения между структурами данных и регулярными явыками.