Proc. 3rd ICALP, S. Michaelson and R. Milner, eds., pages 508-530, Edinburgh University Press, Edinburgh 1976

H.-D. Ehrich

OUTLINE OF AN ALGEBRAIC THEORY OF STRUCTURED OBJECTS

ABSTRACT - An algebraic approach to data structures is presented which is somewhat related to the ideas of the Vienna method. After having introduced and investigated the basic operations of object space and their algebraic properties, continuous mappings are considered, thus getting the basis for solving fixpoint equations. Several interesting substructures of an object space are looked at: submonoids are associated whith data types; finite objects (corresponding to finite trees) as well as rational objects (corresponding to finite graphs) form subspaces. The rational objects are uniquely characterized as solutions of rational systems of equations which are particularly significant for the semantics of data type declarations.

1. INTRODUCTION

The algebraic theory outlined here is an attempt to get a systematic and rigorous framework for dealing with data structures, especially its semantics. The present concept is inspired by the ideas of the Vienna method (Lucas, Lauer and Stigleitner 197o, Wegner 1972) insomuch as the fundamental notion is that of a tree, represented by the set of words S^* over a finite alphabet S of selectors. The main differences, however, are that our trees may be infinite, and elementary data are attached to all nodes, not only to the leaves. Additionally, a monoid structure is assumed on the elementary objects which carries over to arbitrary objects. Besides monoid addition, two binary operations called selection and construction effect an algebraic structure on the set of objects which has several nice properties. For instance, both selection and construction are distributive over addition. One further operation, called amputation, is defined in terms of the former. The ternary μ-operation of the VDL is introduced involving amputation, construction, and addition. Several algebraic properties and relations of these operations are investigated in section 2.

In section 3, we consider continuous mappings on objects. Assuming that the monoid E of elementary objects is a complete partial ordering and that addition is continuous in E, we show the continuity of addition, selection, and construction with regard to all structured objects. Objects and sets of objects (e.g.responses to retrieval queries or data types) may often be characterized as solutions of fixpoint equations which are thus solvable by the fixpoint theorem (cf. Loeckx 1974). It is shown that a countable sum of continuous mappings is again continuous.

Vienna objects may be viewed as classes of our objects differing only at their inner nodes. It is shown in section 4 that this relation is a congruence. This means that Vienna objects, carrying information only on the leaves, may be suitably modelled by a quotient structure of the object space, retaining the algebraic structure.

The object space has several interesting substructures: submonoids are subsets closed with respect to addition, and subspaces are submonoids closed additionally with respect to both selection and construction. In section 5, the subspace of finite objects (Corresponding to finite trees) is investigated briefly. Then, submonoids are considered; they are characterized completely by the submonoids of E in a form that suggests to interpret them as data types. By this association, the relation between a data type and its instances is made precise.

Finally, in section 6, the pratically most interesting subspace of rational objects is considered. These correspond in some sense to finite graphs and thus describe the class of objects conveniently representable by finite means. We show that rational objects can uniquely be described as solutions of rational systems of equations (RSEs). Together with the submonoid considerations of section 5, RSEs represent a tool for declaring rational object types, thus providing some insight in the semantics of recursive type declarations (e.g. the so-called infinite modes in ALGOL 68 (van Wijngaarden et al. 1975; Heilbrunner 1973)

2. THE OBJECT SPACE

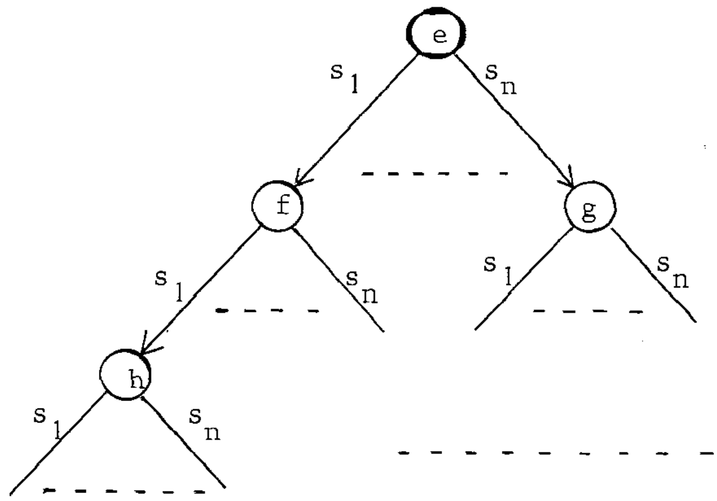
Let $(E,+,0)$ be a monoid, and let $S=\{s_1,\ldots,s_n\}$ be a finite set. The elements of S will be called selectors. The set of words over S is as usually denoted by S^*, and the empty word is written ε. The carrier set of our object space is defined as follows.

Definition 2.1: The set of objects is $\mathbb{D} := E^{S^*}$.

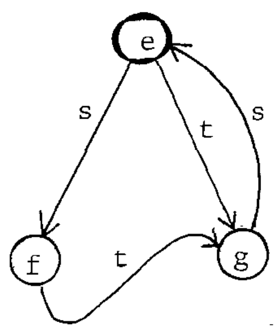
Subsequently, we will adopt the interpretation that E and S represent elementary units of information and access paths, respectively. Objects, then, are models of information structures, selecting pieces of information when following access paths.

Other interpretations are possible: if we consider S and E as sets of input resp. output symbols, we get a general notion of automata with starting state: every input string is answered by an output symbol. Furthermore, flow diagrams may be viewed as special objects, if E is a set of elementary statements and predicates, and S^* describes the control structure. Finally, formal languages over the alphabet S can be identified with objects over $E=\{o,1\}$.

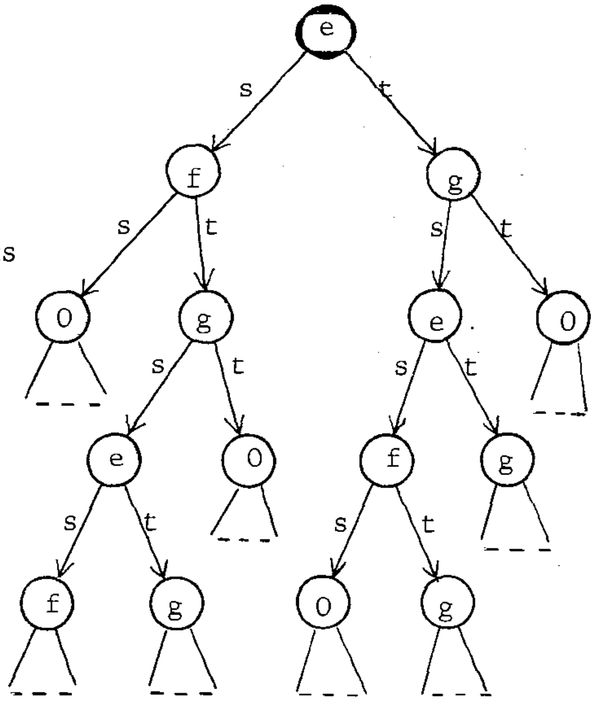
Objects will be represented graphically by (infinite) rooted trees with node marks from E and edge marks from S:



We omit subtrees all of whose nodes are marked with 0. If subtrees repeat periodically, we represent this graphically by loops:



represents



The concept of an infinite tree has been introduced into the computing literature by Rabin (1969), and infinite trees have been considered recently by several authors, e.g. Engelfriet (1972), Goguen and Thatcher (1974), and Nivat (1973). The approach taken here is, however, slightly different.

An object $e\varepsilon\mathbb{D}$ is <u>elementary</u> iff, for each selector word $x\varepsilon S^+$, we have $e(x)=0$, and it is the <u>null object</u> $0\varepsilon\mathbb{D}$ iff $0(x)=0$ for each word $x\varepsilon S^*$. The set of elementary objects will be denoted by $\mathbb{E}$, and we will use the notation $A:=\mathbb{D}-\mathbb{E}$ for the set of non-elementary or composite objects. There is an obvious one-to-one correspondence between E and $\mathbb{E}$, and we will make no notational difference between node marks and elementary objects, i.e. we identify E with $\mathbb{E}$.

The <u>characteristic set</u> of an object $a\varepsilon\mathbb{D}$ is

$$\chi(a):=\{x\varepsilon S^* \mid a(x)\neq 0\}.$$

Obviously, an object e is elementary iff $\chi(e)\subset\{\varepsilon\}$, and it is the null object iff $\chi(e)=\emptyset$.

We now introduce the basic operations of our object space: addition, selection, and construction. Unless otherwise specified, in the sequel index i ranges over an index set $I\subset\mathbb{N}$.

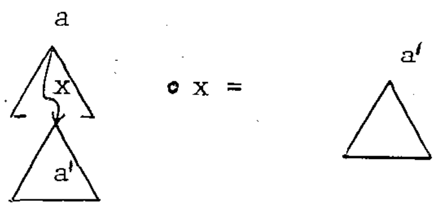
<u>Definition 2.2:</u> $\forall\ a_i, a\varepsilon\mathbb{D} \quad \forall\ x,y,z,\varepsilon S^*$

1. <u>addition</u> : $[\Sigma a_i](y):=\Sigma\ a_i(y)$
2. <u>selection</u> : $[a\circ x](y):= a(xy)$
3. <u>construction:</u> $[x\times a](y):=\begin{cases} a(z) & \text{if } y=xz \\ o & \text{otherwise}\end{cases}$

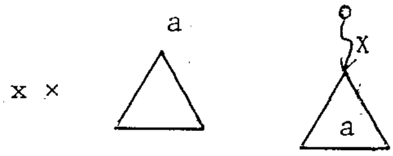
Addition is nothing else but the usual addition of functions. Intuitively this means merging the trees of a and b and adding the corresponding node marks. Of course, $(\mathbb{D}, +, 0)$ is a monoid. Infinite sums make sense if they make sense in $(E,+,0)$.

Selection and construction may be illustrated graphically as follows:

Selection:



Construction:



The construction operation implies that all marks of the new nodes on path x are zero.

Definition 2.3: An object space is any subset $R \subset \mathbb{D}$ satisfying $R+R \subset R$, $R \circ S \subset R$, and $S \times R \subset R$.

The following theorem states a number of basic rules for algebraic manipulations in an object space. The proofs are rather simple, so they are left to the reader.

Theorem 2.4: $\forall\, a_i, a, b, \in \mathbb{D} \quad \forall x, y, \in S^*$

1. $a \circ \varepsilon = a$ 2. $(a \circ x) \circ y = a \circ xy$
3. $\varepsilon \times a = a$ 4. $x \times (y \times a) = xy \times a$
5. $(\Sigma a_i) \circ x = \Sigma (a_i \circ x)$
6. $x \times (\Sigma a_i) = \Sigma (x \times a_i)$
7. $x \times 0 = 0$
8. $x \times a = x \times b \Rightarrow a = b$
9. $x \neq \varepsilon \wedge a \neq 0 \Rightarrow x \times a \varepsilon A$
10. $(x \times a) \circ y = \begin{cases} a \circ z & \text{if } y = xz \\ z \times a & \text{if } x = yz \\ 0 & \text{otherwise} \end{cases}$
11. $\chi(a) \cap \chi(b) = \emptyset \Rightarrow a+b = b+a$
12. $\chi(\Sigma a_i) \subset \bigcup \chi(a_i)$
13. $\chi(a \circ x) = \partial_x(\chi(a))$
14. $\chi(x \times a) = x\chi(a)$
15. $a = a(\varepsilon) + \sum_{s \varepsilon S} s \times (a \circ s)$
16. $a = \sum_{x \varepsilon \chi(a)} x \times a(x)$

In 12, equality holds if $(E,+,0)$ has "no zero sums", i.e. $\Sigma e_i = 0$ implies $e_i = 0$ for all $i \varepsilon I$, or if the characteristic sets $\chi(a_i)$ are pairwise disjoint. in 13, by ∂_x we mean the derivative of a word set with respect to a word, i.e. $\partial_x(X) = \{y \mid \exists\, z \varepsilon X : z = xy\}$. In 15 and 16, the order of summation is immaterial because of rule 11.

Now we introduce another operation, called amputation, which intuitively describes the clipping of subtrees. This operation can be defined in terms of addition and construction as follows.

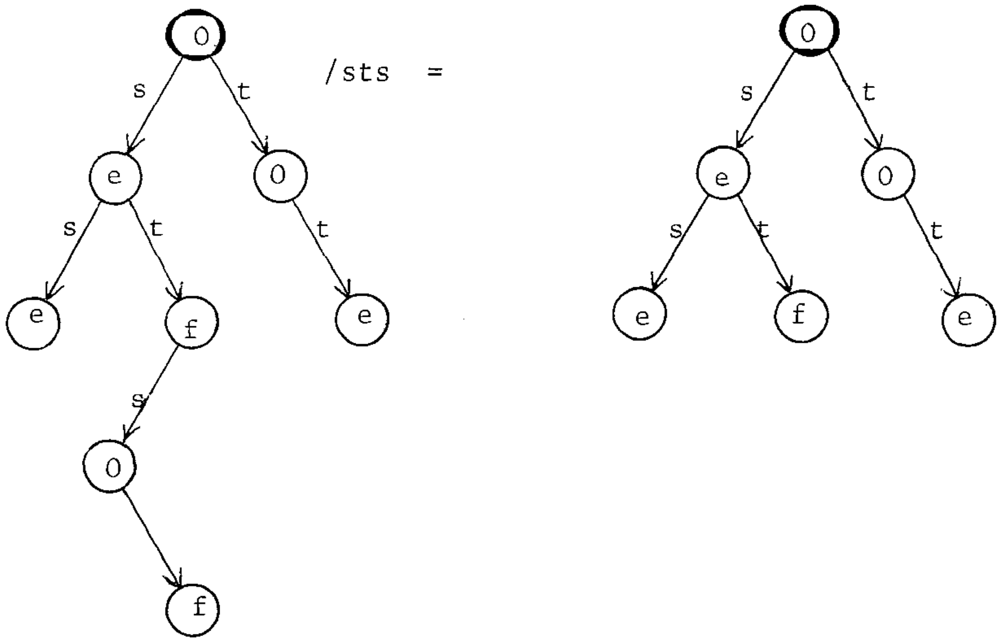
Definition 2.5: Let $a \in \mathbb{D}$ and $x \in S^*$. The operation

$$a/x := \sum_{z \in Z} z \times a(z) \quad , \text{ where } Z = \chi(a) - xS^*$$

is called the amputation of a by x.

The sum may be infinite. This makes sense even if infinite sums in E do not, since in each component z only one term is different from zero.

Example: Let $e,f \neq 0$



We see immediately that amputation satisfies the following simple rules for all $a,b \in \mathbb{D}$ and all $x,y \in S^*$:

1. $0/x = 0$
2. $a/\varepsilon = 0$
3. $a \circ x = 0 \Rightarrow a/x = a$
4. $a = a/x + x \times (a \circ x)$
5. $(\Sigma a_i)/x = \Sigma a_i/x$
6. $\chi(a/x) = \chi(a) - x\, S^*$
7. $[a/x](y) = \begin{cases} 0 \text{ if } y = xz,\ z \in S^* \\ a(y) \text{ otherwise} \end{cases}$

Furthermore, the following relations hold true:

Theorem 2.6: $\forall a \in \mathbb{D} \quad \forall x,y, \in S^*$

1. $a/xy = a/x + x \times (a \circ x/y)$

$$2.\ (a/x)\circ y = \begin{cases} 0 & \text{if } y=xz,\ z\varepsilon S^* \\ a\circ y/z & \text{if } x=yz,\ z\varepsilon S^* \\ a\circ y & \text{otherwise} \end{cases}$$

$$3.\ (x\times a)/y = \begin{cases} x\times(a/z) & \text{if } y=xz,\ z\varepsilon S^* \\ 0 & \text{if } x=yz,\ z\varepsilon S^* \\ x\times a & \text{otherwise} \end{cases}$$

Proof: 1. Let $b:=a/x+x\times(a\circ x/y)$. Then,

$$b = \sum_{z\varepsilon Z} z\times a(z) \;+\; x \times \sum_{z\varepsilon Y} z\times a(xz)\ ,$$

Where $Z = \chi(a) - xS^*$, and $Y = \chi(a\circ x) - y\ S^*$
$= \partial_x(\chi(a)) - y\ S^*$.

Now we have

$$b = \sum_{z\varepsilon Z} z\times a(z) + \sum_{z\varepsilon Y} xz\times a(xz)$$

$$= \sum_{z\varepsilon Z'} z\times a(z)$$

where $Z' = Z \cup xY = \chi(a) - xy\ S^*$. This implies $b=a/xy$

2. Let $b:=(a/x)\circ y$ and $W:=\chi(a/x)$. Then,

$$b = \Big(\sum_{w\varepsilon W} w\times a(w)\Big)\circ y$$

$$= \sum_{w\varepsilon W} (w\times a(w))\circ y$$

In all non-zero terms of this sum, y is a prefix of w. Let $w=yv$, $v\varepsilon S^*$. Then,

$$b = \sum_{v\varepsilon V} v\times a(yv) = \sum_{v\varepsilon V} v\times\big[a\circ y\big](v),$$

where $V=\partial_y(W)=\partial_y(\chi(a)) -\partial_y(x\ S^*)$. If $y=xz$, we get $V=\emptyset$, and thus $b=0$. If $x=yz$, we have $V= \chi(a\circ y) - \partial_y(yz\ S^*)$ $=\chi(a\circ y/z)$, yielding $b=a\circ y/z$. In all other cases we have $V=\partial_y(\chi(a))$ and thus $b= a\circ y$.

3. Let $b:=(x\times a)/y$ and $W = \chi(x\times a) - yS^*$. Then,

$$b = \sum_{w\varepsilon W} w\times\big[x\times a\big]\ (w)$$

Since x is a prefix of all selector words, we may write

$$b = \sum_{v\varepsilon V} xv\times\big[x\times a\big]\ (xv) = x\times\sum_{v\varepsilon V} v\times a(v)$$

where $V = \partial_x(W) = \chi(a) - \partial_x(yS^*)$. If y=xz, we have $V = \chi(a) - zS^*$, and thus $b=x\times(a/z)$. If x=yz, we get $V=\emptyset$, implying b=0. In all other cases, we have $V = \chi(a)$ and get $b = x\times a$.

Corollary 2.7: $a/xy/x = a/x/xy = a/x$

Proof: Since $(a/x)\circ x = 0$, we have $a/x/x = a/x$. Now, $a/xy/x = a/x/x + (x\times(a\circ x/y))/x = a/x$, and $a/x/xy = a/x/x + x\times((a/x)\circ x/y) = a/x$.

Corollary 2.8: $a/x/y = a/y/x$

Proof: If x is a prefix of y or vice versa, corollary 2.7 applies. Otherwise we conclude: $a = a/x + x\times(a\circ x) \Rightarrow a/y = a/x/y + (x\times(a\circ x))/y$. The last term is zero, so we go on: $a/y/x = a/x/y/x$. Since $(a/x/y)\circ x = (a/x)\circ x = 0$, we see that $a/x/y/x = a/x/y$.

Another useful operator is the μ-operator which is fundamental in the VDL (Lucas, Lauer and Stigleitner 197o, Wegner 1972) and is also used by Rosen (1973). Defining it as a basic operation, however, does not seem to be very advantageous, because it has three arguments. Therefore one cannot expect a priori that it has any desirable algebraic properties. Intuitively, the μ-operator replaces a subtree by another one. Within our framework it can be defined as follows.

Definition 2.9: $\forall a,b\in D\ \forall x\in S^*$: $\mu(a,x,b) := a/x + x\times b$

As special cases we have $a/x = \mu(a,x,0) = \mu(a,x,b)/x$, $x\times a=\mu(0,x,a)$, and $b = \mu(a,\varepsilon,b)$. We see immediately that the characteristic set is

$$\chi(\mu(a,x,b)) = (\chi(a) - xS^*)\cup x\chi(b).$$

Furthermore, the following rules for the μ-operation can be proven by straightforward application of theorems 2.4/6 and corollaries 2.7/8.

Theorem 2.9: $\forall a,b,c\in D\ \forall x,y\in S^*$

1. $\mu(a,x,b)\circ y = \begin{cases} b\circ z & \text{if } y=xz \\ \mu(a\circ y,z,b) & \text{if } x=yz \\ a\circ y & \text{otherwise} \end{cases}$
2. $x\times\mu(a,y,b) = \mu(x\times a,xy,b)$
3. $\mu(a,x,b)/y = \begin{cases} \mu(a,x,b/z) & \text{if } y=xz \\ a/y & \text{if } x=yz \\ \mu(a/y,x,b) & \text{otherwise} \end{cases}$
4. $\mu(\mu(a,x,b),y,c) = \begin{cases} \mu(a,x,\mu(b,z,c)) & \text{if } y=xz \\ \mu(a,y,c) & \text{if } x=yz \\ \mu(\mu(a,y,c),x,b) & \text{otherwise} \end{cases}$

5. $\mu(a,xy,b) = \mu(a,x,\mu(a\circ x,y,b))$
6. $\mu(a+b,x,c)= \mu(a,x,0) + \mu(b,x,c)$
7. $\mu(a,x,b+c)= \mu(a,x,b) + \mu(0,x,c)$

3. CONTINUITY

Consider the equation $u\circ x= b$, where u is an unknow object, and $x\in S^*$ as well as $b\in\mathbb{D}$ are given. This equation corresponds to a common query type in information retrieval, where all objects with "property (x,b)" are being sought, i.e. all objects that have "value" b under "attribute" x.

We easily verify that, for each $c\in\mathbb{D}$,

$$u = c/x + x\times b = \mu(c,x,b)$$

is a solution of our equation. One the other hand, each solution can be expressed this way, since if a is a solution, we have $a = a/x + x\times b$.

A more sophisticated query type is of the form

$$u\circ x=u\circ y \ ,$$

i.e. "find all objects with equal values under selectors x and y". By the above reasoning, we get

$$u = c/x + x\times(u\circ y) \ , \quad c\in\mathbb{D}.$$

This is, however, not an explicit solution, but a fixpoint equation. In order to solve such equations by application of the fixpoint theorem, we must introduce a complete partial ordering (CPO) on $\mathbb{D}$ and the notion of continuity, such that the algebraic operations are continuous functions.

Let us assume that a CPO $\leq$ on E is given, such that $0\leq e$ for all $e\in E$. Completeness means that each chain $e_1\leq e_2\leq\ldots$ over E has a least upper bound $\bigcup e_i$ in E. This CPO is extended to $\mathbb{D}$ in a natural way.

<u>Definition 3.1:</u> $\forall a,b\in\mathbb{D}\quad a\leq b \ :\Leftrightarrow \forall x\in S^*\ a(x)\leq b(x)$

Obviously, $\leq$ is a CPO on $\mathbb{D}$, and $0\leq a$ holds true for each object $a\in\mathbb{D}$. If $x\in S^*$, the x-component of the limit of a chain $\bigcup a_i$ is given by

$$(\bigcup a_i)(x) = \bigcup a_i(x)$$

The next lemma follows immediately from the definitions.

<u>Lemma 3.2:</u> $\forall a,b\in\mathbb{D}$

1. $a\leq b \Leftrightarrow \forall x\in S^*\ a\circ x\leq b\circ x$
2. $\forall x\in S^*\ (a\leq b \Leftrightarrow x\times a\leq x\times b)$
3. $a\leq b \Rightarrow \chi(a)\subset\chi(b)$

Definition 3.3: A mapping $\phi : \mathbb{D} \to \mathbb{D}$ is called continuous iff, for each chain $a_1 \leq a_2 \leq \ldots$ over $\mathbb{D}$, we have

$$\phi(\bigcup a_i) = \bigcup \phi(a_i)$$

We remark that every continuous mapping ϕ is monotonous: $a \leq b$ implies $\phi(a) \leq \phi(b)$.

Theorem 3.4: Selection and construction by fixed selector words are continuous mappings on $\mathbb{D}$. Addition of a fixed object is continuous on $\mathbb{D}$ iff it is continuous on E.

Proof: Let $a_1 \leq a_2 \leq \ldots$ be a chain over $\mathbb{D}$, and let $x \in S^*$.

1. Selection: $\forall y \in S^*$:

$$\begin{aligned}[\bigcup(a_i \circ x)](y) &= \bigcup[a_i \circ x](y) = \bigcup a_i(xy) \\ &= [\bigcup a_i](xy) \\ &= [(\bigcup a_i) \circ x](y)\end{aligned}$$

2. Construction: $\forall y \in S^*$:
Case 1: if x is not a prefix of y, we have

$$\begin{aligned}[\bigcup(x \times a_i)](y) &= \bigcup[x \times a_i](y) = 0 \\ &= [x \times \bigcup a_i](y)\end{aligned}$$

Case 2: if y=xz, we conclude

$$\begin{aligned}[\bigcup(x \times a_i)](y) &= \bigcup[x \times a_i](y) = \bigcup a_i(z) \\ &= [\bigcup a_i](z) \\ &= [x \times \bigcup a_i](y)\end{aligned}$$

3. Addition: The only-if part is trivial. Let + be continuous on E, and let $b \in \mathbb{D}$.

$$\begin{aligned}[(\bigcup a_i)+b](x) &= [\bigcup a_i](x) + b(x) \\ &= \bigcup a_i(x) + b(x) \\ &= \bigcup(a_i(x)+b(x)) \\ &= \bigcup[a_i+b](x) \\ &= [\bigcup(a_i+b)](x)\end{aligned}$$

Thus, $(\bigcup a_i)+b = \bigcup (a_i+b)$. In the same way we show that $b+\bigcup a_i = \bigcup(b+a_i)$ which completes the proof.

This theorem shows that we only have to assume continuity of addition in E in order to make shure that the algebraic operations on $\mathbb{D}$ are continuous. This will be taken for granted in the sequel. Continuity of addition implies several interrelations between partial ordering and addition, the most important of which are shown in the next lemma.

Lemma 3.5: $\forall a,b,c,d,\in \mathbb{D}$:

1. $a \le c \wedge b \le d \quad \Rightarrow \quad a + b \le c + d$
2. $a \le a + b \quad ; \quad b \le a + b$
3. $a + b = 0 \quad \Rightarrow \quad a = b = 0$,

i.e. $\mathbb{D}$ has no zero sums.

Proof: 1. $a \le c \Rightarrow a+b \le c+b$, and $b \le d \Rightarrow c+b \le c+d$
2. $0 \le b \Rightarrow a=a+0 \le a+b$, similar: $b=0+b \le a+b$
3. $a \le a+b = 0 \Rightarrow a=0$. Similarly with b.

A remarkable fact is that countable sums of continuous functions are again continuous. In order to prove this, we need the following lemma.

Lemma 3.6: Let $\{a_{ik} \mid i,k\in \mathbb{N}\}$ be a set of objects such that

(i) $a_{ik} \le a_{i(k+1)}$ and (ii) $a_{ik} \le a_{(i+1)k}$

for all $i,k\in\mathbb{N}$. Then,

$$\bigcup_i \bigcup_k a_{ik} = \bigcup_k \bigcup_i a_{ik}$$

Proof: Let $a_i := \bigcup_k a_{ik}$ and $a^k := \bigcup_i a_{ik}$. Obviously, we have $a_i \le a_j$ and $a^i \le a^j$ for $i \le j$. Let $a_* := \bigcup a_i$ and $a^* := \bigcup a^k$. For all $i,k,\in\mathbb{N}$, we have $a_{ik} \le a^k \le a^* \Rightarrow a_i \le a^* \Rightarrow a_* \le a^*$. Similarly, the converse realtion $a^* \le a_*$ is proven.

Let $(\phi_1,\phi_2,\ldots)$ be a sequence of functions from $\mathbb{D}$ into $\mathbb{D}$ such that $\phi_i(a) \le \phi_{i+1}(a)$ for all $i\in\mathbb{N}$ and all $a\in\mathbb{D}$. The we define $\phi := \bigcup \phi_i$ to be the function mapping each object $a\in\mathbb{D}$ onto $\phi(a) := \bigcup \phi_i(a)$.

Theorem 3.7: Let $\phi_1, \phi_2, \ldots$ be a sequence of continuous functions with the above property. Then, $\phi = \bigcup \phi_i$ is continuous.

Proof: Let $a_1 \le a_2 \le \ldots$ be a chain over $\mathbb{D}$. Since all ϕ_i

are continuous, we have for all $i,k \in \mathbb{N}$

$$\text{(i)}\quad \phi_i(a_k) \leq \phi_i(a_{k+1}) \quad \text{and (ii)}\quad \phi_i(a_k) \leq \phi_{i+1}(a_k).$$

By lemma 3.6, we conclude

$$\begin{aligned}\phi(\bigcup_k a_k) &= \bigcup_i \phi_i(\bigcup_k a_k) = \bigcup_i \bigcup_k \phi_i(a_k)\\ &= \bigcup_k \bigcup_i \phi_i(a_k)\\ &= \bigcup_k \phi(a_k)\end{aligned}$$

Corollary 3.8: A countable sum of continuous mappings is continuous.

Now we are in a position to solve fixpoint equations. We start with our introductory example $u \circ x = u \circ y$, and then give one further example.

Example 3.9: Equation $u \circ x = u \circ y$ leads to

$$u = c/x + x \times (u \circ y), \quad c \in \mathbb{D}.$$

By the fixpoint theorem, we get

$$\begin{aligned}u_o &= 0\\ u_1 &= c/x\\ u_2 &= c/x + x \times ((c/x) \circ y)\end{aligned}$$

We consider only the case that neither x is a prefix of y nor vice versa. Then we can simplify

$$\begin{aligned}u_2 &= c/x + x \times (c \circ y)\\ u_3 &= c/x + x \times ((c/x + x \times (c \circ y)) \circ y)\\ &= c/x + x \times ((c/x) \circ y + (x \times (c \circ y)) \circ y)\\ &= c/x + x \times (c \circ y)\end{aligned}$$

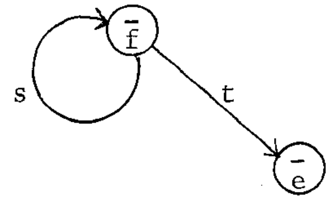
Since $u_3 = u_2$, we have $u_2 = u_3 = u_4 \ldots$. Thus, the minimal fixpoint is

$$u = \bigcup u_i = c/x + x \times (c \circ y) = \mu(c, x, c \circ y)$$

Example 3.1o: $u = u + s \times u + t \times e + f$

$$\begin{aligned}u_o &= 0\\ u_1 &= t \times e + f\\ u_2 &= t \times e + f + st \times e + s \times f + t \times e + f\\ &= st \times e + s \times f + t \times (e+e) + f+f\\ u_3 &= sst \times e + ss \times f + st \times (e+e+e) + s \times (f+f+f)\\ &\quad + t \times (e+e+e) + f+f+f\end{aligned}$$

The solution is $u = \sum_{i=o}^{\infty} s^i t \times \bar{e} + \sum_{i=o}^{\infty} s^i \times \bar{f}$, where $\bar{e} = \sum_{i=o}^{\infty} e$ and $\bar{f} = \sum_{i=o}^{\infty} f$. This solution can be represented graphically as follows.



There is another equation having the same minimal fixpoint, namely

$$u = s \times u + t \times \bar{e} + \bar{f}$$

Equations and systems of equations of this latter type will be considered in greater depth in section 6.

4. VIENNA OBJECTS

Vienna objects differ from our objects mainly with respect to the association of information to the nodes of the tree: only leaves are allowed to carry elementary data, inner nodes are not. In fact, Vienna objects are restricted to finite trees. We want to give up this restriction here and consider "generalized" Vienna objects. They are introduced as a quotient space of an object space with respect to a congruence relation to be defined below.

To begin with, we define a set of selector words associated with an object which can be understood to represent the leaves.

Definition 4.1: Let $a \in \mathbf{D}$. The crown of a is

$$\eta(a) := \{x \in \chi(a) \mid a \circ x \in \mathbf{E} - \{0\}\}$$

Obviously, no $x \in \eta(a)$ is prefix of any other word from $\eta(a)$, save x itself. Let max be an operator operating on subsets of S^* by removing all prefixes of other words, formally

$$\underline{\max}\ T := \{x \in T \mid \forall\, y \in S^* : xy \in T \Rightarrow y = \varepsilon\}$$

for $T \subset S^*$. Then we have

$$\eta(a) = \underline{\max}\ \chi(a)$$

Lemma 4.2: $\forall a_i, b \in \mathbf{D}\quad \forall x \in S^*$:

1. $\eta(\Sigma a_i) = \underline{\max} \bigcup \eta(a_i)$, provided that $\mathbf{D}$ has no zero sums
2. $\eta(a \circ x) = \partial_x(\eta(a))$
3. $\eta(x \frown a) = x\eta(a)$

Proof: 2 and 3 are rather trivial. We only prove 1:

$x\varepsilon\ \eta(\Sigma a_i) \Leftrightarrow (\Sigma a_i)\circ x\varepsilon\mathbf{E}-\{0\} \Leftrightarrow \forall i: a_i\circ x\varepsilon\mathbf{E} \wedge \exists j: a_j\circ x\neq 0 \Leftrightarrow$

$\exists j: x\varepsilon\eta(a_j) \wedge \forall y\varepsilon S^* \forall i: (y\neq\varepsilon \Rightarrow xy\notin \eta(a_i)) \Leftrightarrow x\varepsilon\ \underline{\max} \bigcup \eta(a_i).$

We now introduce the congruence relation $\sim$ on $\mathbf{D}$ which will be used subsequently to define the generalized Vienna objects.

Definition 4.3: $\forall a,b\varepsilon\mathbf{D} : a\sim b : \Leftrightarrow \forall x\varepsilon\ \eta(a)\cup\eta(b): a\circ x = b\circ x$

Intuitively, two objects are congruent iff they coincide on their crowns. This will be made clearer in the sequel.

Lemma 4.4: $\forall a,b\varepsilon\mathbf{D} : a\sim b \Rightarrow \eta(a) = \eta(b)$

Proof: $x\varepsilon\eta(a) \Rightarrow a\circ x\varepsilon\mathbf{E}-\{0\}$. $a\sim b \Rightarrow a\circ x=b\circ x\varepsilon\mathbf{E}-\{0\} \Rightarrow \eta(a)\subset\eta(b)$. In the same way the converse inclusion is proven.

It is obvious that $\sim$ is an equivalence relation. Each equivalence class has a particular representative: let $a\varepsilon\mathbf{D}$ be an object and $[a]$ be its equivalence class; now restrict the partial ordering $\leq$ on $\mathbf{D}$ to $[a]$.

Theorem 4.5: Let $a\varepsilon\mathbf{D}$. The partal ordering $([a],\leq)$ has a minimal element a_{min} given by

$$a_{min} = \sum_{x\varepsilon\eta(a)} x\times a(x)$$

Proof: By definition 4.3 it is clear that $a_{min}\sim a$. Let $b\varepsilon\mathbf{D}$ be an arbitrary object with $b\sim a$, and let $x\varepsilon S^*$. Then, if $x\varepsilon\eta(a)=\eta(a_{min})=\eta(b)$, we have $a_{min}(x) = b(x)$. If $x\notin\eta(a)$, we have $a_{min}(x)=0\leq b(x)$. Thus, by definition 3.1 we have $a_{min}\leq b$.

Corollary 4.6: $\forall a\varepsilon\mathbf{D} : \mu(a) = \bigcap_{b\sim a} \chi(b)$

The mapping $a \mapsto a_{min}$ on $\mathbf{D}$ is compatible with selection and construction in the following sense.

Theorem 4.7: $\forall a\varepsilon\mathbf{D}\ \ \forall x\varepsilon S^*$:

1. $(a\circ x)_{min} = a_{min}\circ x$
2. $(x\times a)_{min} = x\times a_{min}$

Proof: $(a \circ x)_{min} = \sum_{y \in \partial_x(\eta(a))} y \times [a \circ x](y)$

$= [\sum_{y \in \partial_x(\eta(a))} xy \times a(xy)] \circ x$

$= [\sum_{z \in \eta(a)} z \times a(z)] \circ x$

$= a_{min} \circ x$

2. $(x \times a)_{min} = \sum_{y \in x\eta(a)} y \times [x \times a](y)$

$= \sum_{z \in \eta(a)} xz \times [x \times a](xz)$

$= x \times \sum_{z \in \eta(a)} z \times a(z) = x \times a_{min}$

Unfortunately, there is no similar compatibility with addition nor amputation. The next theorem shows that $\sim$ is a congruence.

Theorem 4.8: $\forall a_i, b_i, a, b \in \mathbf{D}$:

1. $(\forall i : a_i \sim b_i) \Rightarrow \sum a_i \sim \sum b_i$, provided that $\mathbf{D}$ has no zero sums
2. $a \sim b \Leftrightarrow (\forall x \in S^*: a \circ x \sim b \circ x)$
3. $\forall x \in S^*: (a \sim b \Leftrightarrow x \times a \sim x \times b)$

Proof: 1. By lemma 4.4, we have $\eta(a_i) = \eta(b_i)$ for all i. Thus, by lemma 4.2.1, we have $\eta(\Sigma a_i) = \eta(\Sigma b_i)$. Let X be this set, and let $x \in X$, say $x \in \eta(a_j)(= \eta(b_j))$. Then, $xy \notin \eta(a_i)$ for all i and all $y \neq \varepsilon$, and $a_i \circ x \in \mathbf{E}$ for all i. The same holds true for the b_i. Thus, $a_i \circ x = b_i \circ x$ for all i, and we conclude $(\Sigma a_i) \circ x = \Sigma(a_i \circ x) = \Sigma(b_i \circ x) = (\Sigma b_i) \circ x$. The proofs of 2 and 3 are rather obvious. They are omitted here.

This theorem shows that $\sim$ is a congruence relation if E has no zero sums. Assuming this, the set $\mathbf{D}/\sim$, whose elements will be called (generalized) Vienna objects, bears the same algebraic structure as $\mathbf{D}$, at least as far as the operations $+, \circ, \times$ are concerned. The amputation operation, however, makes some difficulties, because in general the implication $a \sim b \Rightarrow a/x \sim b/x$ is not true. The reason is that the definition of amputation depends on the characteristic set, and this concept is meaningless for Vienna objects (as defined here). If, however amputation of the latter is redefined using the crown $\eta(a)$,

$$a/x := \sum_{x \in X} x \times a(x)$$

$$X := \eta(a) - xS^*,$$

then all relationships and rules from section 2 carry over to Vienna objects. Especially, the μ-operator then allows the usual interpretation of the VDL. Also, nearly all discussions of the remaining sections carry over to Vienna objects with only slight modifications.

5. SUBSTRUCTURES AND TYPES

In this section, several important substructures of the object space $\mathbf{D}$ will be considered, subspaces as well as submonoids. By a submonoid is meant a subset of $\mathbf{D}$ closed with respect to addition. A subspace is a submonoid closed additionally with respect to selection and construction, i.e. a subset of $\mathbf{D}$ which is an object space.

Definition 5.1: The set of

1. finite objects is $\mathbf{D}_f := \{a \in \mathbf{D} \mid |\chi(a)| < \infty\}$
2. rational objects is $\mathbf{D}_r := \{a \in \mathbf{D} \mid |\{a \circ x \mid x \in S^*\}| < \infty\}$

Obviously, these sets form subspaces, and we have

$$\mathbf{D}_f \subsetneq \mathbf{D}_r \subsetneq \mathbf{D} .$$

Rational objects are of considerable practical interest, since they represent in some sense the objects that are representable by finite graphs. We will consider them in greater depth in the next section.

Finite objects are representable by finite trees and therefore are especially convenient to describe and manipulate. This is manifested by their widespread use in data description and manipulation devices. It is easy to see that $\mathbf{D}_f$ determines the smallest subspace of $\mathbf{D}$ which makes full use of E, i.e. the only subspaces contained in $\mathbf{D}_f$ are the finite objects over submonoids of E. The set of finite objects can be generated by the finite algebraic expressions built by elementary objects, selector words, and the operations $+, \circ, \times$.

We next show that $\mathbf{D}$ itself is the smallest subspace (over E) which is a complete partial ordering, provided that E is one. For each subset $C \subset \mathbf{D}$, let C^∞ be the closure of C with respect to limits of chains over C. Of course, $\mathbf{D}^\infty = \mathbf{D}$.

Theorem 5.2: $\mathbf{D}_f^\infty = \mathbf{D}_r^\infty = \mathbf{D}$.

Proof: We need only show that $\mathbf{D}_f^\infty = \mathbf{D}$. Let $a \in \mathbf{D}$. By theorem 2.4.16, we have

$$a = \sum_{x\in\chi(a)} x\times a(x)$$

We order $\chi(a)$ somehow, e.g. lexicographically, getting $\chi(a)=\{x_1,x_2,x_3,\ldots\}$. If $\chi(a)$ is finite, nothing has to be shown. Otherwise, let

$$a_i := \sum_{j=1}^{i} x_j\times a(x_j) .$$

Then, by lemma 3.5.2, $a_1\leq a_2\leq\ldots$ is a chain over $\mathbb{D}_f$, and $\bigcup a_i=a$.

The possible submonoids of $\mathbb{D}$ can be characterized by the submonoids of E. Let $X\subset S^*$, and for each $x\in X$ let E_x be some submonoid of E. Then,

$$M:=\{\ a=\sum_{x\in X} x\times e_x \mid e_x\in E_x\ \}$$

is a submonoid of $\mathbb{D}$. We represent M formally by

$$(*)\qquad M = \sum_{x\in X} x\times E_x .$$

If $a\in M$, we have clearly $\chi(a)\subset X$.

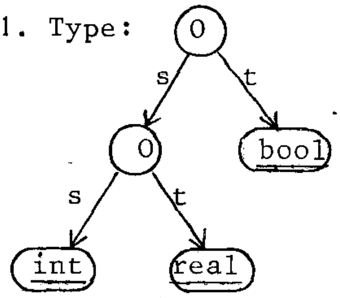
Theorem 5.3: Each submonoid $M\subset\mathbb{D}$ is contained in a submonoid of the form $(*)$ for a suitable set $X\subset S^*$ and suitable submonoids $E_x\subset E$.

Proof: Let M be any submonoid of $\mathbb{D}$, and let $X:=\bigcup_{a\in M}\chi(a)$. For each $x\in X$, let $E_x:=\{a(x) \mid a\in M\}$. Clearly, each E_x is a submonoid of E. If $a\in M$, we have by theorem 2.4.16 $a=\sum_{x\in\chi(a)} x\times e_x$, $e_x\in E_x$. Since $0\in E_x$ for all $x\in X$, we may extend the sum by some zero terms, getting $a = \sum_{x\in X} x\times e_x$, $e_x\in E_x$.

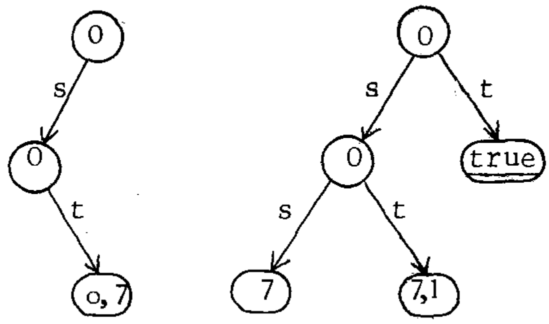
Let $\mathcal{E}$ be the set of submonoids of E. In view of theorem 2.4.16, any formal algebraic expression over $\mathcal{E}$, S and the operations $+,\circ,\times$ uniquely specifies a submonoid of $\mathbb{D}$. These circumstances can be exploited to characterize data types and their descriptions. We suggest to define a data types as a submonoid of $\mathbb{D}$, and its description to be a suitable formal expression involving $\mathcal{E}$, resp. its graphical representation.

Example 5.4:

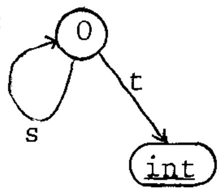
1. Type:



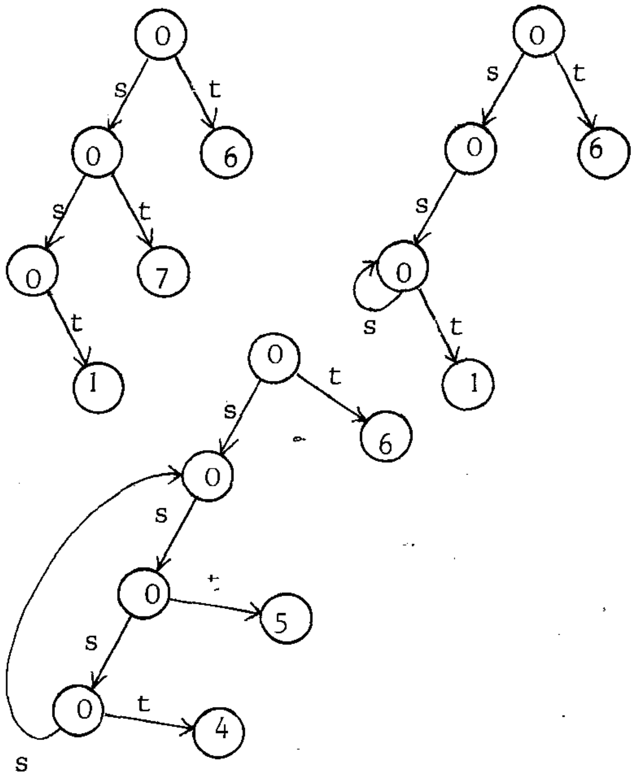
some instances: 0



2. Type:



some instances: 0



6. RATIONAL OBJECTS .

Rational objects are of particular interest because they correspond to finite graphs and are thus immediately representable by finite means. We will characterize them as solutions of certain systems of equations which therefore are called rational. These systems provide a suitable basis to understand the semantics of type declarations.

To begin with, we give a necessary condition for an object to be rational.

Lemma 6.1: If $a \in D$ is rational, its characteristic set $\chi(a)$ is regular.

Proof: Since $\{a \circ x \mid x \in S^*\}$ is finite, there is only a finite number of derivatives $\partial_x(\chi(a))$, where x ranges over S^*.

Let $U = \{u_1, u_2, \ldots, u_m\}$ be a finite ordered set of symbols not in E, the "unknows", and let $V := U \cup E$.

Definition 6.2: A system of equations of the form

$$u_i = \sum_{k=1}^{n} s_k \times v_{ik} + e_i \ , \ i=1,\ldots,m,$$

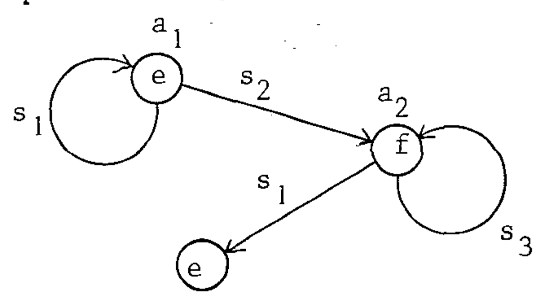
where $u_i \in U$, $v_{ik} \in V$, and $e_i \in E$ for all $i=1,\ldots,m$ and all $k=1,\ldots,n$ is called rational (or briefly an RSE).

An object a_1 is a solution of an RSE if there are objects $a_2,\ldots,a_m$ such that all equations become identities when substituting a_i for u_i , $i=1,\ldots,m$.

Example 6.3:

$$u_1 = s_1 \times u_1 + s_2 \times u_2 \qquad + e$$
$$u_2 = s_1 \times e \ + \qquad + s_3 \times u_2 + f$$

is an RSE. Its graphical solution is as follows:



That this is a solution is easily verified, since $a_1 \circ s_1 = a_1$, $a_1 \circ s_2 = a_2$, $a_1(\varepsilon) = e$, etc.

The following lemma provides the basis for the solution of an RSE. We will make use of the shortcut notation

$$X \times a \ := \sum_{x \in X} x \times a \ ,$$

which is meaningful if X is an independent set, i.e. no word in X is prefix of any other word in X (cf. Thm. 2.4.11).

Lemma 6.4: Let $R \subset S^*$ be a regular independent set $R \neq \{\varepsilon\}$. Let $b \in \mathbf{D}$ be such that $RS^* \cap \chi(b) = \emptyset$. Then the equation

$$u = R \times u + b$$

has the unique solution

$$a = R^* \times b .$$

This solution is rational if b is rational.

Proof: We verify the solution by substitution:

$$R \times (R^* \times b) + b = R^+ \times b + b = R^* \times b.$$

The first equality holds because R is independent, and the second one uses the assumption $RS^* \cap \chi(b) = \emptyset$, from which commutativity of addition follows.

To show uniqueness, let $x \in S^*$, and let u be any solution.

1. If $x \in RS^*$, we have $x = ry$ for some $r \in R$ and some $y \in S^*$; thus we get
   $$u(x) = (R \times u)(x) + b(x) = u(y)$$
   This implies: if $x = zy$ for some $z \in R^+$ and some $y \notin RS^*$, we have $u(x) = u(y)$.
2. If $x \notin RS^*$, we get
   $$u(x) = (R \times u)(x) + b(x) = b(x)$$
   In each case, u(x) is determined uniquely by R and b.

   In order to show that the solution is rational if b is we observe that, for any $y \in S^*$, we have
   $$u \circ y = \partial_y(R) \times u + b \circ y$$
   Since R is regular and b is rational, we have only a finite number of different derivatives of R resp. objects $b \circ y$. Thus, the number of objects $u \circ y$ is finite.

This lemma can be applied for the stepwise solution of an RSE. We revert to example 6.3 . The second equation is solved by

$$a_2 = s_3^* \times (s_1 \times e + f) = s_3^* s_1 \times e + s_3^* \times f$$

Substituting this for u_2 in the first equation, we get

$$u_1 = s_1 \times u_1 + s_2 s_3^* s_1 \times e + s_2 s_3^* \times f + e .$$

The solution is

$$a_1 = s_1^*\times(s_2 s_3^* s_1\times e + s_2 s_3^*\times f + e)$$
$$= (s_1^* s_2 s_3^* s_1 \vee s_1^*)\times e + s_1^* s_2 s_3^*\times f$$

Theorem 6.5: Each RSE has a unique rational solution. Conversely, each rational object is the solution of some RSE.

Proof: We apply lemma 6.4 to eliminate succesively $u_m, u_{m-1}, \ldots, u_2$. After having eliminated u_{j+1}, let the equations have the following form:

$$u_i = \sum_{k\leq j} P_{ik}\times u_k + \sum_{h=1}^{p} Q_{ih}\times e_h \ , \ i=1,\ldots,j \ ,$$

where P_{ik}, Q_{ih} are regular sets, and $e_1,\ldots,e_p$ are the elementary objects occuring as coefficients in the RSE. Eliminating u_j yields the following equations:

$$u_i = \sum_{k<j} (P_{ik}\cup P_{ij}P_{jj}^*P_{jk})\times u_k + \sum_{h=1}^{p} (Q_{ih}\cup P_{ij}P_{jj}^*Q_{jh})\times e_h \ ,$$

$i=1,\ldots,j-1$. It is not difficult to see that the following properties of the coefficient sets remain invariant by this elimination step:

1. All P_{ik}, Q_{ih} are regular
2. $\forall i,k : \varepsilon\notin P_{ik}$
3. $\forall i : \bigcup_k P_{ik}$ is independent
4. $\forall i,k,h : P_{ik}S^*\cap Q_{ih}=\emptyset$

These properties guarantee that the assumptions of lemma 6.4 are satisfied at each elimination step.

The converse statement of the theorem is rather evident: given an object $a\in\mathbf{D}_r$, we can construct an RSE whose solution is a by repeated application of theorem 2.4.15.

From this theorem we see that RSEs are a suitable means for specifying rational objects. Together with the submonoid considerations of the last section, we arrive at a tool for declaring rational object types, providing some insight in the semantics of recursive data type declarations, e.g. the socalled "infinite modes" in ALGOL 68 (Heilbrunner 1973).

Example 6.6: The mode declaration

mode u = struct (bool r, ref u s, ref v t)

mode v = struct (ref u s, int t)

is represented by the RSE

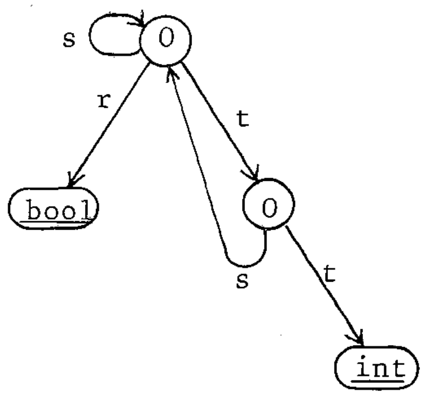
$$u = r\times\underline{bool} + s\times u + t\times v$$

$$v = s\times u + t\times\underline{int} ,$$

whose solution is

$$a = (s\vee ts)^{*}r\times\underline{bool} + (s\vee ts)^{*}tt\times\underline{int} ,$$

shown graphically by the following diagram.



The set of instances of this type is determined by the corresponding submonoid of $\mathbf{D}$.

7. CONCLUSIONS

We have outlined an algebraic theory of structured objects which seems to be useful for the semantic description of data structures. Referring to another possible interpretation mentioned in section 2, namely automata, we realize some interesting relationships to automata theory. For instance, rational objects correspond to finite automata of Moore type. In this interpretation, RSEs refer to a very common representation of finite automata, namely flow tables. However, the operations on objects introduced here do not have any correspondence with know concepts in automata theory.

Nevertheless, some problems with data structures do have their automata-theoretic counterparts. For example, consider the problem of type equivalence: do different RSEs specify the same object (data type)? This is nothing else but the equivalence problem of automata, and it may be solved by carrying over the relevant methods.

Another important problem is that of type recognition: to which data type does a given instance of a data structure belong? Provided that we can recognize the elementary types, we get from the instance a description of its type in terms of selector words associated with elementary types. Recognizing the type from a minimum of such information is somewhat similar to testing automata by minimal experiments.

These remarks may suffice for showing that automata theory has some significance for data structures and that these relations should be subject to further studies.

REFERENCES

Ehrich, H.-D. (1974). Ein axiomatischer Ansatz für eine Algebra strukturierter Objekte. Bericht Nr. 5/74. Dortmund: Abteilung Informatik der Universität.

Engelfriet, J. (1972). A note on infinite trees. Inf.Proc. Letters 1, 229-232.

Goguen, J.A. & J.W. Thatcher (1974) Initial algebra semantics. 15th Ann.Symp. SWAT, 63-77.

Heilbrunner, S. (1973) Gleichungssysteme für Zeichenreihen. Dissertation. Bericht 7311. München: Technische Universität, Abteilung Mathematik.

Loeckx, J. (1974) The fixpoint theorem and the principle of computational induction. Bericht A74/o8. Saarbrücken: Universität des Saarlandes, Fachbereich Angew. Mathematik und Informatik.

Lucas, P., P.Lauer & H. Stigleitner (197o) Method and notation for the formal definition of programming languages. TR 25.o87, Vienna: IBM Lab.

Nivat, M. (1973) On the interpretation of recursive program schemes. Symposia Mathematica-Atti del convegno d'Informatica teorica, Roma.

Rabin, M.O. (1969) Decidability of second-order theories and automata on infinite trees. Transactions of the American Mathematical Society 141, 1-35.

Rosen, B.K. (1973) Tree-manipulating systems and Church-Rosser theorems. Journal of the ACM 2o, 16o-187.

Wand, M. (1973) Mathematical foundations of Formal Language theory. MAC TR-1o8. Cambridge: Mass. Inst. of Technology.

Wegner, P. (1972) The Vienna Definition Language. Comp. Surveys 4, 7-63.

van Wijngaarden, A. et al. (1975) Revised report on the algorithmic language ALGOL 68. Acta Informatica 5, 1-236