With each closed-set system we associate a closure operation.

Definition 1.20. Let $\langle A, \mathcal{C} \rangle$ be a closed-set system. Define $\mathrm{Cl}_{\mathcal{C}} : : \boldsymbol{\mathcal{P}}(A) \to \boldsymbol{\mathcal{P}}(A)$ as follows. For every $X \subseteq A$,

$$\mathrm{Cl}_{\mathcal{C}}(X) = \bigcap \{\, C \in \mathcal{C} : X \subseteq C \,\}.$$

$\mathrm{Cl}_{\mathcal{C}}(X)$ is called the *closure of* X.

Theorem 1.21. *Let $\langle A, \mathcal{C} \rangle$ be a closed-set system. Then for all $X, Y \subseteq A$,*

$$X \subseteq \mathrm{Cl}_{\mathcal{C}}(X), \qquad (extensivity) \tag{11}$$

$$\mathrm{Cl}_{\mathcal{C}}\big(\mathrm{Cl}_{\mathcal{C}}(X)\big) = \mathrm{Cl}_{\mathcal{C}}(X), \qquad (idempotency) \tag{12}$$

$$X \subseteq Y \quad \textit{implies} \quad \mathrm{Cl}_{\mathcal{C}}(X) \subseteq \mathrm{Cl}_{\mathcal{C}}(Y), \qquad (monotonicity) \tag{13}$$

and if $\langle A, \mathcal{C} \rangle$ is algebraic,

$$\mathrm{Cl}_{\mathcal{C}}(X) = \bigcup \big\{ \mathrm{Cl}_{\mathcal{C}}(X') : X' \subseteq_\omega X \,\big\}. \quad (finitarity). \tag{14}$$

Proof. Note that since $\mathcal{C}$ is closed under intersection, $\mathrm{Cl}_{\mathcal{C}}(X) \in \mathcal{C}$ and thus $\mathrm{Cl}_{\mathcal{C}}(X)$ is the smallest member of $\mathcal{C}$ that includes X, and that $X \in \mathcal{C}$ iff $\mathrm{Cl}_{\mathcal{C}}(X) = X$. The conditions (11) and (12) are immediate consequences of this fact, and, if $X \subseteq Y$, then every member of $\mathcal{C}$ that includes Y, in particular $\mathrm{Cl}_{\mathcal{C}}(Y)$, necessarily includes X and hence also $\mathrm{Cl}_{\mathcal{C}}(X)$. Thus (13) holds.

Assume $\langle A, \mathcal{C} \rangle$ is algebraic. By (11) $\{\, \mathrm{Cl}_{\mathcal{C}}(X') : X' \subseteq_\omega X \,\}$ is directed, because, for all $X', X'' \subseteq_\omega X$, $\mathrm{Cl}_{\mathcal{S}}(X') \cup \mathrm{Cl}_{\mathcal{S}}(X'') \subseteq \mathrm{Cl}_{\mathcal{S}}(X' \cup X'')$, and $X' \cup X'' \subseteq_\omega X$.

$$X = \bigcup \{\, X' : X' \subseteq X \,\} \underset{(11)}{\subseteq} \bigcup \{\, \mathrm{Cl}_{\mathcal{C}}(X') : X' \subseteq_\omega X \,\} \in \mathcal{C}.$$

So $\mathrm{Cl}_{\mathcal{C}}(X) \subseteq \bigcup \big\{ \mathrm{Cl}_{\mathcal{C}}(X') : X' \subseteq_\omega X \,\big\}$. The opposite inclusion follows by monotonicity. Thus (14). □

Now suppose that a mapping $\mathrm{Cl} : \boldsymbol{\mathcal{P}}(A) \to \boldsymbol{\mathcal{P}}(A)$ satisfies (11)–(13). Let $\mathcal{C} = \{\, X \subseteq A : \mathrm{Cl}(X) = X \,\}$ (called the *closed sets* of Cl). Then $\mathcal{C}$ is a closed-set system (exercise). Moreover, $\mathcal{C}$ is algebraic if Cl satisfies (14). To see this let $K \subseteq \mathcal{C}$ be upward directed. We must show $\mathrm{Cl}(\bigcup K) \subseteq \bigcup K$. By (14) $\mathrm{Cl}(\bigcup K) = \bigcup \big\{ \mathrm{Cl}(X) : X \subseteq_\omega \bigcup K \,\big\}$. Since K is directed, for every $X \subseteq_\omega \bigcup K$, there is a $C_X \in K$ such that $X \subseteq C_X$, and hence $\mathrm{Cl}(X) \subseteq C_X$, since C_X is closed. Thus

$$\bigcup \{\, \mathrm{Cl}(X) : X \subseteq_\omega \bigcup K \,\} \subseteq \bigcup \{\, C_X : X \subseteq_\omega K \,\} \subseteq \bigcup K.$$

Thus (algebraic) closed-set systems and (finitary) closure operators are equivalent in a natural sense, and we can go back-and-forth between them without hesitation. The next theorem shows that every (algebraic) closed-set system gives rise to an (algebraic) lattice.

Theorem 1.22. *Let $\langle A, \mathcal{C} \rangle$ be a closed-set system.*

(i) *$\langle \mathcal{C}, \subseteq \rangle$ is a complete lattice. For every $K \subseteq \mathcal{C}$, $\bigwedge K = \bigcap K$ and*

$$\bigvee K = \bigcap \{\, C \in \mathcal{C} : \bigcup K \subseteq C \,\} = \mathrm{Cl}_{\mathcal{C}}\big(\bigcup K\big).$$

(ii) *If $\langle A, \mathcal{C}\rangle$ is algebraic, then $\langle \mathcal{C}, \subseteq\rangle$ is an algebraic lattice. Moreover, the compact elements of $\langle \mathcal{C}, \subseteq\rangle$ are the closed sets of the form $\mathrm{Cl}_{\mathcal{C}}(X)$ with $X \subseteq_\omega A$.*

Proof. (i). Exercise. (The proof is just like the proofs that $\langle \mathrm{Sub}(\boldsymbol{G}), \subseteq\rangle$ and $\langle \mathrm{Eq}(A), \subseteq\rangle$ are complete lattices.)

(ii). Assume $\langle A, \mathcal{C}\rangle$ is algebraic. We first verify the claim that the compact elements are exactly those of the form $\mathrm{Cl}_{\mathcal{C}}(X)$ with $X \subseteq_\omega A$. Let $C = \mathrm{Cl}_{\mathcal{C}}(X)$ with $X \subseteq_\omega A$. Suppose

$$C \subseteq \bigvee K = \mathrm{Cl}_{\mathcal{C}}\big(\bigcup K\big) = \bigcup \big\{ \mathrm{Cl}_{\mathcal{C}}(Y) : Y \subseteq_\omega \bigcup K \,\big\}.$$

Since X is finite and $\big\{ \mathrm{Cl}_{\mathcal{C}}(Y) : Y \subseteq_\omega \bigcup K \,\big\}$ is directed, $X \subseteq \mathrm{Cl}_{\mathcal{C}}(Y)$ for some $Y \subseteq_\omega \bigcup K$. Thus there exist $D_1, \dots, D_n \in K$ such that $Y \subseteq D_1 \cup \cdots \cup D_n \subseteq D_1 \vee \cdots \vee D_n$. Hence

$$C = \mathrm{Cl}_{\mathcal{C}}(X) \subseteq \mathrm{Cl}_{\mathcal{C}}(Y) \subseteq D_1 \vee \cdots \vee D_n.$$

So C is compact in the lattice $\langle \mathcal{C}, \subseteq\rangle$.

Conversely, assume C is compact in $\langle \mathcal{C}, \subseteq\rangle$. Then $C = \bigcup \{\, \mathrm{Cl}_{\mathcal{C}}(X) : X \subseteq_\omega C \,\} = \bigvee \{\, \mathrm{Cl}_{\mathcal{C}}(X) : X \subseteq_\omega C \,\}$. So there exist $X_1, \dots, X_n \subseteq_\omega C$ such that $C = \mathrm{Cl}_{\mathcal{C}}(X_1) \vee \cdots \vee \mathrm{Cl}_{\mathcal{C}}(X_n) = \mathrm{Cl}_{\mathcal{C}}(X_1 \cup \cdots \cup X_n)$. Since $X_1 \cup \cdots \cup X_n$ is finite, we have, for every $C \in \mathcal{C}$,

For every $C \in \mathcal{C}$, $C = \bigcup \{\, \mathrm{Cl}_{\mathcal{C}}(X) : X' \subseteq X \,\} = \bigvee \{\, \mathrm{Cl}_{\mathcal{C}}(X) : X' \subseteq X \,\}$. So every $C \in \mathcal{C}$ is the join of compact elements. Hence $\langle \mathcal{C}, \subseteq\rangle$ is algebraic. □

For any group $\boldsymbol{G}$, $\mathrm{Cl}_{\mathrm{Sub}(\boldsymbol{G})}(X)$ the subgroup generated by X, which is usually denoted by $\langle X\rangle$. The finitely generated subgroups are the compact elements of $\mathbf{Sub}(\boldsymbol{G}) = \langle \mathrm{Sub}(\boldsymbol{G}), \subseteq\rangle$. The compact elements of $\mathbf{Eq}(A)$ are the equivalence relations "generated" by a finite set of ordered pairs.

The notion of a lattice was invented to abstract a number of difference phenomena in algebra, and other mathematical domains, that have to do with order. We have seen three levels of abstract so far: at the lowest level we have the lattices of subgroups and equivalence relations. At the next level the lattices of algebraic closed-set systems, and at the highest level the algebraic lattices in which all notions of set and subset have been abstracted away. The next theorem shows that in a real sense there is no loss in abstracting from algebraic closed-set systems to algebraic lattices.

Theorem 1.23. *Every algebraic lattice $\boldsymbol{A} = \langle A, \leq\rangle$ is isomorphic to the lattice of $\langle \mathcal{C}, \subseteq\rangle$ of closed sets for some algebraic closed-set system $\langle B, \mathcal{C}\rangle$.*

Proof. Let $B = \mathrm{Comp}(\boldsymbol{A})$, the set of compact elements of $\boldsymbol{A}$. For each $a \in A$, let $C_a = \{\, c \in \mathrm{Comp}(A) : c \leq a \,\}$. Let $\mathcal{C} = \{\, C_a : a \in A \,\}$. Because $\boldsymbol{A}$ is compactly generated, $a = \bigvee C_a$; hence the mapping $a \mapsto C_a$ is a bijection from A to $\mathcal{C}$. Moreover, the mapping is strictly order-preserving since, clearly, $a \leq b$ iff $C_a \subseteq C_b$. So by Theorem 1.8 $\langle \mathcal{C}, \subseteq\rangle$ is a complete lattice and the mapping $a \mapsto C_a$ is it is an isomorphism between the lattices $\langle A, \leq\rangle$ and $\langle \mathcal{C}, \subseteq\rangle$.

It only remains to show that $\langle \mathrm{Comp}(\boldsymbol{A}), \mathcal{C}\rangle$ is an algebraic closed-set system. Let $K \subseteq \mathcal{C}$; $K = \{\, C_x : x \in X \,\}$ for some $X \subseteq \mathrm{Comp}(\boldsymbol{A})$. Then

$$\bigcap \{\, C_x : x \in X \,\} = C_{\bigwedge X}.$$

To see this consider any $c \in \mathrm{Comp}(\boldsymbol{A})$. Then $c \in \bigcap \{\, C_x : x \in X \,\}$ iff, for all $x \in X$, $c \in C_x$ iff, for all $x \in X$, $c \leq x$ iff $c \in C_{\bigwedge X}$.

Assume now that K is upward directed. Since $x \leq y$ iff $C_x \subseteq C_y$, we see that X is also directed by the ordering $\leq$ of $\boldsymbol{A}$. We show that

$$\bigcup\{\, C_x : x \in X \,\} = C_{\bigvee X}.$$

Let $c \in \mathrm{Comp}(\boldsymbol{A})$.

$$\begin{aligned} c \in C_{\bigvee X} \quad &\text{iff} \quad c \leq \bigvee X \\ &\text{iff} \quad \text{for some } X' \subseteq_\omega X,\ c \leq \bigvee X', \quad \text{since } c \text{ is compact} \\ &\text{iff} \quad \text{for some } x \in X,\ c \leq x, \quad \text{since } X \text{ is directed} \\ &\text{iff} \quad \text{for some } x \in X,\ c \in C_x \\ &\text{iff} \quad c \in \bigcup\{\, C_x : x \in X \,\}. \end{aligned}$$

So $\bigvee\{\, C_x : x \in X \,\} = \bigcup\{\, C_x : x \in X \,\}$, and hence $\langle \mathcal{C}, \subseteq \rangle$ is algebraic. □

2. General Algebraic Structures

An algebraic structure is a simply a set with a possibly infinite set of operations on it of finite rank. For example a group is a set together with the binary operation of group multiplication, the inverse operation, which is of rank one, and what we call a "distinguished constant", the group identity. The latter can be viewed as as an operation of "rank zero". In order to compare two algebras of the same kind, it is useful to have some way of indexing the operations so that an operation on one algebra can matched with the corresponding operation of the other algebra. For instance, when we compare two rings we don't want to match addition on the first ring with multiplication on in the second ring. When one is dealing with only a few kinds of algebraic structures, like group, rings and vector spaces, this is not a problem. But in the general theory where a wide range of algebraic types are considered we have to be more precise. The custom now is to specify a the type of an algebraic structure by the formal language associated with it. This motivates the following definition.

Definition 2.1. A *signature* or *language type* is a set Σ together with a mapping $\rho\colon \Sigma \to \omega$. The elements of Σ are called *operations symbols*. For each $\sigma \in \Sigma$, $\rho(\sigma)$ is called the *arity* or *rank* of σ.

For simplicity we write Σ for $\langle \Sigma, \rho \rangle$, treating the rank function as implicit.

Definition 2.2. Let Σ be a signature. A *Σ-algebra* is a ordered couple $\boldsymbol{A} = \big\langle A, \langle \sigma^{\boldsymbol{A}} : \sigma \in \Sigma \rangle \big\rangle$, where A is a nonempty set and $\sigma^{\boldsymbol{A}}\colon A^{\rho(\sigma)} \to A$ for all $\sigma \in \Sigma$.

0-ary operations: if $\rho(\sigma) = 0$, $\sigma^{\boldsymbol{A}}\colon A^0 \to A$. But by definition $A^0 = \{\emptyset\}$. It is usual to identify the function $\sigma^{\boldsymbol{A}}$ with the unique element in its range, namely $\sigma^{\boldsymbol{A}}(\emptyset)$; the latter is called a *distinguished constant* of $\boldsymbol{A}$. In general the functions $\sigma^{\boldsymbol{A}}$ are called the *fundamental operations* of $\boldsymbol{A}$.

We give a number of examples of signatures and algebras. We consider two kinds of groups depending on the signature.

$\Sigma_1 = \{\cdot\}$; $\rho(\cdot) = 2$. $\Sigma_2 = \{\cdot, {}^{-1}, \mathrm{e}\}$; $\rho(\cdot) = 2$, $\rho({}^{-1}) = 1$, $\rho(\mathrm{e}) = 0$.

$\boldsymbol{G} = \langle G, \{\cdot^{\boldsymbol{G}}\}\rangle$ is a group of type I if it satisfies the following two conditions.

$$\forall x, y, z\big((x \cdot y) \cdot z \approx x \cdot (y \cdot z)\big) \tag{15}$$

$$\exists x\big(\forall y\big(x \cdot y \approx y \textbf{ and } y \cdot x \approx y\big) \textbf{ and } \forall y \exists z(y \cdot z \approx x \textbf{ and } z \cdot y \approx x)\big). \tag{16}$$

$\boldsymbol{G} = \langle G, \{\cdot^{\boldsymbol{G}}, {}^{-1^{\boldsymbol{G}}}, e^{\boldsymbol{G}}\}\rangle$ is a group of type II if (15) holds together with the following:

$$\forall x(e \cdot x \approx x \textbf{ and } x \cdot e \approx x) \tag{17}$$

$$\forall x(x \cdot x^{-1} \approx e \textbf{ and } x^{-1} \cdot x \approx e). \tag{18}$$

In the general theory of algebras we are careful to distinguish between the symbol for the equality symbol, $\approx$, the identity relation $\{\, \langle a, a\rangle : a \in A \,\}$ on a given set A, which is usually denoted by the symbol "$=$". One should think of the identity relation as the interpretation of the symbol $\approx$ in the set A in much the same way $\cdot^{\boldsymbol{G}}$ is the interpretation of the operation symbol $\cdot$ in the group $\boldsymbol{G}$. In the spirit of the notation of signatures and algebras we can write $\approx^{\boldsymbol{A}}$ is $=$.

The two types of groups are equivalent in the sense that, if $\boldsymbol{G} = \langle G, \{\cdot^{\boldsymbol{G}}\}\rangle$ is a group of type I, then there is a unique $f\colon G \to G$ and $g \in G$ such that $\langle G, \cdot^{\boldsymbol{G}}, f, g\rangle$ is a group of type II. Conversely, if $\boldsymbol{G} = \langle G, \cdot^{\boldsymbol{G}}, {}^{-1^{\boldsymbol{G}}}, e^{\boldsymbol{G}}\rangle$ is a group of type II, then $\langle G, \cdot^{\boldsymbol{G}}\rangle$ is a group of type I.

However, from the viewpoint of the general theory of algebras, the two types of groups have very different properties. Note that the definition conditions (15), (17), and (18) are what we call *identities*: equations between terms with all the variables universally quantified. We note that although (17) is not strictly an identity, it is logically equivalent to the pair of identities $\forall x(e \cdot x \approx x)$ and $\forall x(x \cdot e \approx x)$.) (16) is not logically equivalent to any set of identities as we shall soon see. We mention also that is conventional to omit explicit reference to the universal quantifiers when writing an identity. Thus (15), the associative law, is nomally written simply "$(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$"

We shall use the following simplifying notation. If Σ is finite we write $\boldsymbol{A} = \langle A, \sigma_1^{\boldsymbol{A}}, \sigma_2^{\boldsymbol{A}}, \dots, \sigma_n^{\boldsymbol{A}}\rangle$, where $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ and $\rho(\sigma_1) \geq \rho(\sigma_2) \geq \cdots \geq \rho(\sigma_n)$. We omit the superscripts "$\boldsymbol{A}$" on the "$\sigma^{\boldsymbol{A}}$" when there is not chance of confusion.

More examples. $\boldsymbol{A} = \langle A, +, \cdot, -, 0\rangle$, where $+$ and $\cdot$ are binary, $-$ unary, and 0 nullary, is a *ring* if $\langle A, +, -, 0\rangle$ is *Abelian* group (of type II), i.e., it satisfies the identity $\forall x \forall y(x + y \approx y + x)$, and the $\cdot$ is associative and distributes over $+$, i.e.

$$\forall x, y(x \cdot (y + z) \approx (x \cdot y) + (x \cdot z) \textbf{ and } (y + z) \cdot x \approx (y \cdot x) + (z \cdot x)).$$

An *integral domain* is a ring satisfying

$$\forall x, y\big((x \cdot y \approx 0) \Longrightarrow (x \approx 0) \textbf{ or } (y \approx 0)\big).$$

Notice this is not an identity.

A *field* is an algebra $\langle A, +, \cdot, -, 0, 1\rangle$ such that $\langle A, +, \cdot, -, 0\rangle$ and the following conditions are satisfied.

$$\forall x(x \cdot y \approx y \cdot x)$$
$$\forall x(1 \cdot x \approx x)$$
$$\forall x\big(\textbf{not}(x \approx 0) \Longrightarrow \exists y(x \cdot y \approx 1)\big).$$

We cannot define a field as an algebra of type $\langle A, +, \cdot, -, ^{-1}, 0, 1\rangle$ because 0^{-1} is not defined and by definition every operation of an algebra must be defined for all elements of the algebra.

Lattices are Σ-algebras, with $\Sigma = \{\vee, \wedge\}$, defined by identities.

We now consider an algebra of quite different character, the algebra of *nondeterministic while programs*. Let $\Sigma = \{\mathbf{or}, ;, \mathbf{do}\}$, where **or** and ; are binary and **do** is unary. These operation symbols denote three different ways of controling the flow of a program. If P and Q are programs, then $P\,\mathbf{or}\,Q$ is the program that nondeterministically passes control to P or Q. $P;Q$ passes control first to P and when P terminates to Q. $\mathbf{do}\,P$ loops a nondeterministic number of times, possibly zero, through P. A set W of programs is said to be closed with respect to these control structures if $P, Q \in W$ imply $(P\ \mathbf{or}\ Q), (P\ ;\ Q), (\mathbf{do}\,P) \in W$. For any set S of "atomic programs", let $\mathrm{WP}(S)$ be the smallest closed set containing S. $\mathbf{WP}(S) = \langle \mathrm{WP}(S), \mathbf{or}, ;, \mathbf{do}\rangle$ is an algebra of *nondeterministic while programs*.

$\mathbf{WP}(S)$ is different from the other algebras we have considered in that we have not specificed any conditions that it must satisfy (other than its signature). It is typical of algebras that arise from programming languages in this regard; we will study this kind of algebra in more detail later.

A *vector space* over a field $\langle F, +, \cdot, -, 0, 1\rangle$ is an Abelian group $\langle A, +, -, 0\rangle$ with a scalar multiplication $F \times A \to A$ satisfying the following conditions, for all $r, r' \in F$ and $a, a' \in A$.

$$\begin{aligned} 1a &= a, \\ r(a + a') &= ra + ra', \\ (r + r')a &= ra + r'a, \\ (r \cdot r')b &= r(r'a). \end{aligned}$$

This is not an algebra in our sense but can be made into one by expanding the signature of Abelian groups by adjoining a new unary operation for each element of the field. Let $\Sigma = \{+, -, 0\} \cup \{\sigma_r : r \in F\}$, where $\rho(\sigma_r) = 1$ for every $r \in F$. (Note that Σ is infinite if F is infinite.) A vector space is a Σ-algebra $\boldsymbol{A} = \langle A, +^{\boldsymbol{A}}, -^{\boldsymbol{A}}, 0^{\boldsymbol{A}}, \sigma_r^{\boldsymbol{A}}\rangle_{r\in F}$ such that $\langle A, +^{\boldsymbol{A}}, -^{\boldsymbol{A}}, 0^{\boldsymbol{A}}\rangle$ is an Abelian group and, for every $r \in F$ and $a \in A$, $\sigma_r^{\boldsymbol{A}}(a) = ra$, the scalar product of a by r.

A vector space in this sense is defined by identities, but in general an infinite number. The properties of both the scalar multiplication the field must be expressed in terms of identities. For example, the last of the four defining condtions on scalar multiplication takes the form of a possibly infinite set of identities, namely, $\{\sigma_{r\cdot r'}(x) \approx \sigma_r(\sigma_{r'}(x) : r, r' \in F\}$, while the the commutativity of the ring multiplication is reflected in the set of identities $\{\sigma_r(\sigma_{r'}(x)) = \sigma_{r'}(\sigma_r(x)) : r, r' \in F\}$.

A more satisfactory treatment of vector spaces requires a generalization of the notion of a signature.

A *mult-sorted signature* consists of a nonempty set S of *sorts* together with a set Σ of *operation symbols* and, for each $\sigma \in \Sigma$, a nonempty sequence $\rho(\sigma) = \langle s_1, \dots, s_n, t\rangle$ of sorts, called the *type* of σ. We usually write the type in the form $s_1, \dots, s_n \to t$. The sequence of sorts $s_1, \dots, s_n$ and the single sort t are called respectively the *arity* and *target sort* of σ.

A Σ-*algebra* is an ordered pair

$$\boldsymbol{A} = \big\langle \langle A_s : s \in S\rangle,\ \langle \sigma^{\boldsymbol{A}} : \sigma \in \Sigma\rangle \big\rangle,$$

where $\langle A_s : s \in S\rangle$, which is usually denoted by A, is a nonempty finite sequence of nonempty sets. For each $\sigma \in \Sigma$, if $s_1, \dots, s_n \to t$ is the type of σ, then

$$\sigma^{\boldsymbol{A}} : A_{s_1} \times \cdots \times A_{s_n} \to A_t.$$

Vector spaces over a field can be most naturally viewed as multi-sorted algebras where $S = \{V, F\}$ and $\Sigma = \{+_V, -_V, 0_V, +_F, \cdot_F, -_F, 0_F, +_V, -_V, 0_V, 1_V, *\}$, where the types of the various operation symbols are given in the following table. $*$ denotes scalar multiplica-

operation	type	operation	type
$+_V$	$V, V \to V$	$+_F$	$F, F \to F$
$-_V$	$V \to V$	$-_F$	$F \to F$
0_V	$\to V$	0_F	$\to F$
		1_F	$\to F$
		$\cdot_F$	$F, F \to F$
$*$	$F, V \to V$		

tion. The defining identities are left to the reader.

We give an example of a multi-sorted algebra that arises in the algebraic theory of data types, the algebra of *lists of data.* $S = \{D, L\}$, $\Sigma = \{\mathtt{head}, \mathtt{tail}, \mathtt{append}, \mathtt{derror}, \mathtt{lerror}\}$. The type of each operation symbol is as follows: $\mathtt{head} : L \to D$; $\mathtt{tail} : L \to L$; $\mathtt{append} : D, L \to L$, $\mathtt{derror} : \to D$; $\mathtt{lerror} : \to L$.

The *algebra of lists* over a nonempty set A is

$$\mathbf{List}(A) = \langle \mathrm{List}(A), \mathtt{head}^{\mathbf{List}(A)}, \mathtt{tail}^{\mathbf{List}(A)}, \mathtt{append}^{\mathbf{List}(A)}, \mathtt{derror}^{\mathbf{List}(A)}, \mathtt{lerror}^{\mathbf{List}(A)} \rangle,$$

where $A^* = \{\langle a_1, \dots, a_n\rangle : n \in \omega, a_1, \dots, a_n \in A\}$, the set of all finite sequences of elements of A.

$$\begin{aligned}
\mathtt{head}^{\mathbf{List}(A)}(\langle a_1, \dots, a_n\rangle) &= a_1 \quad \text{if } \langle a_1, \dots, a_n\rangle \text{ is not empty},\\
\mathtt{head}^{\mathbf{List}(A)}(\langle\rangle) &= \mathtt{derror},\\
\mathtt{head}^{\mathbf{List}(A)}(\langle a_1, \dots, a_n\rangle) &= \langle a_2, \dots, a_n\rangle \quad \text{if } \langle a_1, \dots, a_n\rangle \text{ is not empty},\\
\mathtt{head}^{\mathbf{List}(A)}(\langle\rangle) &= \mathtt{lerror},\\
\mathtt{append}^{\mathbf{List}(A)}(b, \langle a_1, \dots, a_n\rangle) &= \langle b, a_1, \dots, a_n\rangle,\\
\mathtt{derror}^{\mathbf{List}(A)} &= e_D,\\
\mathtt{lerror}^{\mathbf{List}(A)} &= e_L.
\end{aligned}$$