# Appendix

## 2.4 Algebras

We have already encountered the concept of an algebra in example 2.3.1 and in the section containing theorem 2.3.1. An algebra is simply a pair $< A, F >$ consisting of a nonempty set $A$ together with a collection $F$ of functions also called operations, each function in $F$ being of the form $f : A^n \rightarrow A$, for some natural number $n > 0$. When working with algebras, one is often dealing with algebras having a *common structure*, in the sense that for any two algebras $< A, F >$ and $< B, G >$, there is a function $d : F \rightarrow G$ from the set of functions $F$ to the set of functions $G$. A more convenient way to indicate common structure is to define in advance the set $\Sigma$ of *function symbols* as a ranked alphabet used to name operators used in these algebras. Then, given an algebra $< A, F >$, each function name $f$ receives an *interpretation* $I(f)$ which is a function in $F$. In other words, the set $F$ of functions is defined by an interpretation $I : \Sigma \rightarrow F$ assigning a function of rank $n$ to every function symbol of rank $n$ in $\Sigma$. Given any two algebras $< A, I : \Sigma \rightarrow F >$ and $< B, J : \Sigma \rightarrow G >$, the mapping $d$ from $F$ to $G$ indicating common structure is the function such that $d(I(f)) = J(f)$, for every function name $f$ in $\Sigma$. This leads to the following formal definition.

### 2.4.1 Definition of an Algebra

Given a ranked alphabet $\Sigma$, a $\Sigma$-*algebra* $\mathbf{A}$ is a pair $< A, I >$ where $A$ is a nonempty set called the *carrier*, and $I$ is an *interpretation function* assigning functions to the function symbols as follows:

(i) Each symbol $f$ in $\Sigma$ of rank $n > 0$ is interpreted as a function $I(f) : A^n \to A$;

(ii) Each constant $c$ in $\Sigma$ is interpreted as an element $I(c)$ in $A$.

The following abbreviations will also be used: $I(f)$ will be denoted as $f_{\mathbf{A}}$ and $I(c)$ as $c_{\mathbf{A}}$.

Roughly speaking, the ranked alphabet describes the syntax, and the interpretation function $I$ describes the semantics.

**EXAMPLE 2.4.1**

Let $A$ be any (nonempty) set and let $2^A$ denote the *power-set* of $A$, that is, the set of all subsets of $A$. Let $\Sigma = \{0, 1, -, +, *\}$, where 0,1 are constants, that is of rank 0, $-$ has rank 1, and $+$ and $*$ have rank 2. If we define the interpretation function $I$ such that $I(0) = \emptyset$ (the empty set), $I(1) = A$, $I(-) =$ *set complementation*, $I(+) =$ *set union*, and $I(*) =$ *set intersection*, we have the algebra $\mathcal{B}_A =< 2^A, I >$. Such an algebra is a *boolean algebra of sets in the universe A*.

**EXAMPLE 2.4.2**

Let $\Sigma$ be a ranked alphabet. Recall that the set $CT_\Sigma$ denotes the set of all finite or infinite $\Sigma$-trees. Every function symbol $f$ of rank $n > 0$ defines the function $\overline{f} : CT_\Sigma^n \to CT_\Sigma$ as follows: for all $t_1, t_2, ..., t_n \in CT_\Sigma$, $\overline{f}(t_1, t_2, ..., t_n)$ is the tree denoted by $ft_1t_2...t_n$ and whose graph is the set of pairs

$$\{(e, f)\} \cup \bigcup_{i=1}^{i=n} \{(iu, t_i(u)) \mid u \in dom(t_i)\}.$$

The tree $ft_1...t_n$ is the tree with $f$ at the root and $t_i$ as the subtree at address $i$. Let $I$ be the interpretation function such that for every $f \in \Sigma_n$, $n > 0$, $I(f) = \overline{f}$. The pair $(CT_\Sigma, I)$ is a $\Sigma$-algebra. Similarly, $(T_\Sigma, J)$ is a $\Sigma$-algebra for the interpretation $J$ such that for every $f \in \Sigma_n$, $n > 0$, $J(f)$ is the restriction of $\overline{f}$ to finite trees. For simplicity of notation, the algebra $(CT_\Sigma, I)$ is denoted by $CT_\Sigma$ and $(T_\Sigma, J)$ by $T_\Sigma$.

The notion of a function preserving algebraic structure is defined below

### 2.4.2 Homomorphisms

Given two $\Sigma$-algebras $\mathbf{A}$ and $\mathbf{B}$, a function $h : A \to B$ is a *homomorphism* iff:

(i) For every function symbol $f$ of rank $n > 0$, for every $(a_1, ..., a_n) \in A^n$,

$h(f_{\mathbf{A}}(a_1, ..., a_n)) = f_{\mathbf{B}}(h(a_1), ..., h(a_n))$;

(ii) For every constant $c$, $h(c_{\mathbf{A}}) = c_{\mathbf{B}}$.

If we define $A^0 = \{e\}$ and the function $h^n : A^n \to B^n$ by $h(a_1, ..., a_n) = (h(a_1), ..., h(a_n))$, then the fact that the function $h$ is a homomorphism is expressed by the commutativity of the following diagram:

$$\begin{array}{ccc} A^n & \xrightarrow{f_{\mathbf{A}}} & A \\ h^n \downarrow & & \downarrow h \\ B^n & \xrightarrow[f_{\mathbf{B}}]{} & B \end{array}$$

We say that a homomorphism $h : \mathbf{A} \to \mathbf{B}$ is an *isomorphism* iff there is a homomorphism $g : \mathbf{B} \to \mathbf{A}$ such that $h \circ g = I_A$ and $g \circ h = I_B$. Note that if $h$ is an isomorphism, then it is a bijection.

Inductive sets correspond to subalgebras. This concept is defined as follows.

### 2.4.3 Subalgebras

An algebra $\mathbf{B} =< B, J >$ is a *subalgebra* of an algebra $\mathbf{A} =< A, I >$ (with the same ranked alphabet $\Sigma$) iff:

(1) $B$ is a subset of $A$;

(2) For every constant $c$ in $\Sigma$, $c_{\mathbf{B}} = c_{\mathbf{A}}$, and for every function symbol $f$ of rank $n > 0$, $f_{\mathbf{B}} : B^n \to B$ is the restriction of $f_{\mathbf{A}} : A^n \to A$.

The fact that $\mathbf{B}$ is an algebra implies that $B$ is *closed under the operations*; that is, for every function symbol $f$ of rank $n > 0$ in $\Sigma$, for all $b_1, ..., b_n \in B$, $f_{\mathbf{B}}(b_1, ..., b_n) \in B$. Now, inductive closures correspond to least subalgebras.

### 2.4.4 Least Subalgebra Generated by a Subset

Given an algebra $\mathbf{A}$ and a subset $X$ of $A$, the inductive closure of $X$ in $\mathbf{A}$ is the *least subalgebra of* $\mathbf{A}$ *containing* $X$. It is also called the least subalgebra of $\mathbf{A}$ generated by $X$. This algebra can be defined as in lemma 2.3.1. Let $([X]_i)_{i \geq 0}$ be the sequence of subsets of A defined by induction as follows:

$$[X]_0 = X \cup \{c_{\mathbf{A}} \mid c \text{ is a constant in } \Sigma\};$$
$$[X]_{i+1} = [X]_i \cup \{f_{\mathbf{A}}(a_1, ..., a_n) \mid a_1, ..., a_n \in [X]_i, f \in \Sigma_n, n \geq 1\}.$$

Let

$$[X] = \bigcup_{i \geq 0} [X]_i.$$

One can verify easily that $[X]$ is closed under the operations of $\mathbf{A}$. Hence, $[X]$ together with the restriction of the operators to $[X]$ is a subalgebra of $\mathbf{A}$.

Let $[\mathbf{X}]$ denote this subalgebra. The following lemma is easily proved (using a proof similar to that of lemma 2.3.1).

**Lemma 2.4.1** Given any algebra $\mathbf{A}$ and any subset $X$ of $A$, $[\mathbf{X}]$ is the least subalgebra of $\mathbf{A}$ containing $X$.

*Important note*: The carrier of an algebra is always *nonempty*. To avoid having the carrier $[X]$ empty, we will make the assumption that $[X]_0 \neq \emptyset$ (either $X$ is nonempty or there are constant symbols).

Finally, the notion of free generation is generalized as follows.

### 2.4.5 Subalgebras Freely Generated by a Set $X$

We say that the algebra $[\mathbf{X}]$ is *freely generated by $X$ in* $\mathbf{A}$ iff the following conditions hold:

(1) For every $f$ (not a constant) in $\Sigma$, the restriction of the function $f_{\mathbf{A}} : A^m \rightarrow A$ to $[X]^m$ is injective.

(2) For every $f_{\mathbf{A}} : A^m \rightarrow A$, $g_{\mathbf{A}} : A^n \rightarrow A$ with $f, g \in \Sigma$, $f_{\mathbf{A}}([X]^m)$ is disjoint from $g_{\mathbf{A}}([X]^n)$ whenever $f \neq g$ (and $c_{\mathbf{A}} \neq d_{\mathbf{A}}$ for constants $c \neq d$).

(3) For every $f_{\mathbf{A}} : A^n \rightarrow A$ with $f \in \Sigma$ and every $(x_1, ..., x_n) \in [X]^n$, $f_{\mathbf{A}}(x_1, ..., x_n) \notin X$ (and $c_{\mathbf{A}} \notin X$, for a constant $c$).

As in lemma 2.3.3, it can be shown that for every $(x_1, ..., x_n) \in [X]^n_i - [X]^n_{i-1}$, $f_{\mathbf{A}}(x_1, ..., x_n) \notin [X]_i$, ($i \geq 0$, with $[X]_{-1} = \emptyset$).

We have the following version of theorem 2.3.1.

**Theorem 2.4.1** (Unique homomorphic extension theorem) Let $\mathbf{A}$ and $\mathbf{B}$ be two $\Sigma$-algebras, $X$ a subset of $A$, let $[\mathbf{X}]$ be the least subalgebra of $\mathbf{A}$ containing $X$, and assume that $[\mathbf{X}]$ is freely generated by $X$. For every function $h : X \rightarrow B$, there is a unique homomorphism $\widehat{h} : [\mathbf{X}] \rightarrow \mathbf{B}$ such that:

(1) For all $x \in X$, $\widehat{h}(x) = h(x)$;

For every function symbol $f$ of rank $n > 0$, for all $(x_1, ..., x_n) \in [X]^n$,

(2) $\widehat{h}(f_{\mathbf{A}}(x_1, ..., x_n)) = f_{\mathbf{B}}(\widehat{h}(x_1), ..., \widehat{h}(x_n))$, and $\widehat{h}(c_{\mathbf{A}}) = c_{\mathbf{B}}$, for each constant $c$.

This is also expressed by the following diagram.

$$
\begin{array}{ccc}
X & \longrightarrow & [\mathbf{X}] \\
 & {\scriptstyle h} \searrow & \downarrow {\scriptstyle \widehat{h}} \\
 & & \mathbf{B}
\end{array}
$$

The following lemma shows the importance of the algebra of finite trees.

**Lemma 2.4.2** For every ranked alphabet $\Sigma$, for every set $X$, if $X \cap \Sigma = \emptyset$ and $X \cup \Sigma_0 \neq \emptyset$ (where $\Sigma_0$ denotes the set of constants in $\Sigma$), then the $\Sigma$-algebra $T_\Sigma(X)$ of finite trees over the ranked alphabet $\Sigma \cup X$ obtained by adjoining the set $X$ to $\Sigma_0$ is freely generated by $X$.

*Proof*: First, it is easy to show from the definitions that for every tree $t$ such that $depth(t) > 0$,

$$t = \overline{f}(t/1, ..., t/n) \quad \text{and} \quad depth(t/i) < depth(t),\ 1 \leq i \leq n,$$

where $t(e) = f$ (the label of the root of $t$), $n = r(f)$ (the rank of $f$) and $t/i$ is the "i-th subtree" of $t$, $1 \leq i \leq n$. Using this property, we prove by induction on the depth of trees that every tree in $T_\Sigma(X)$ belongs to the inductive closure of $X$. Since a tree of depth 0 is a one-node tree labeled with either a constant or an element of $X$, the base case of the induction holds. Assume by induction that every tree of depth at most $k$ belongs to $X_k$, the $k$-th stage of the inductive closure of $X$. Since every tree of depth $k+1$ can be written as $t = \overline{f}(t/1, ..., t/n)$ where every subtree $t/i$ has depth at most $k$, the induction hypothesis applies. Hence, each $t/i$ belongs to $X_k$. But then $\overline{f}(t/1, ..., t/n) = t$ belongs to $X_{k+1}$. This concludes the induction showing that $T_\Sigma(X)$ is a subset of $X_+$. For every $f \in \Sigma_n$, $n > 0$, by the definition of $\overline{f}$, if $t_1$,...,$t_n$ are finite trees, $\overline{f}(t_1, ..., t_n)$ is a finite tree (because its domain is a finite union of finite domains). Hence, every $X_k$ is a set of finite trees, and thus a subset of $T_\Sigma(X)$. But then $X_+$ is a subset of $T_\Sigma(X)$ and $T_\Sigma(X) = X_+$. Note also that for any two trees $t$ and $t'$ (even infinite), $t = t'$ if and only if, either

(1) $t = \overline{f}(t_1, ..., t_m) = t'$, for some unique $f \in \Sigma_m$, $(m > 0)$, and some unique trees $t_1$,...,$t_m \in CT_\Sigma(X)$, or

(2) $t = a = t'$, for some unique $a \in X \cup \Sigma_0$.

Then, it is clear that each function $\overline{f}$ is injective, that $range(\overline{f})$ and $range(\overline{g})$ are disjoint whenever $f \neq g$, and that for every $f$ of rank $n > 0$, $\overline{f}(t_1, ..., t_n)$ is never a one-node tree labeled with a constant. Hence, conditions (1),(2),(3) for free generation are satisfied. □

In Chapter 5, when proving the completeness theorem for first-order logic, it is necessary to define the quotient of an algebra by a type of equivalence relation. Actually, in order to define an algebraic structure on the set of equivalence classes, we need a stronger concept known as a *congruence relation*.

### 2.4.6 Congruences

Given a $\Sigma$-algebra $\mathbf{A}$, a *congruence* $\cong$ on $A$ is an equivalence relation on the carrier $A$ satisfying the following conditions: For every function symbol $f$ of rank $n > 0$ in $\Sigma$, for all $x_1, ..., x_n$, $y_1, ..., y_n \in A$, if $x_i \cong y_i$ for all $i, 1 \leq i \leq n$, then

$$f_{\mathbf{A}}(x_1, ..., x_n) \cong f_{\mathbf{A}}(y_1, ..., y_n).$$

The equivalence class of $x$ modulo $\cong$ is denoted by $[x]_{\cong}$, or more simply by $[x]$ or $\widehat{x}$.

Given any function symbol $f$ of rank $n > 0$ in $\Sigma$ and any $n$ subsets $B_1, ..., B_n$ of $A$, we define

$$f_{\mathbf{A}}(B_1, ..., B_n) = \{f_{\mathbf{A}}(b_1, ..., b_n) \mid b_i \in B_i, 1 \leq i \leq n\}.$$

If $\cong$ is a congruence on $A$, for any $a_1, ..., a_n \in A$, $f_{\mathbf{A}}([a_1], ..., [a_n])$ is a subset of some unique equivalence class which is in fact $[f_{\mathbf{A}}(a_1, ..., a_n)]$. Hence, we can define a structure of $\Sigma$-algebra on the set $A/\cong$ of equivalence classes.

## 2.4.7 Quotient Algebras

Given a $\Sigma$-algebra $\mathbf{A}$ and a congruence $\cong$ on $A$, the *quotient algebra* $\mathbf{A}/\cong$ has the set $A/\cong$ of equivalence classes modulo $\cong$ as its carrier, and its operations are defined as follows: For every function symbol $f$ of rank $n > 0$ in $\Sigma$, for all $[a_1], ..., [a_n] \in A/\cong$,

$$f_{\mathbf{A}/\cong}([a_1], ..., [a_n]) = [f_{\mathbf{A}}(a_1, ..., a_n)],$$

and for every constant $c$,

$$c_{\mathbf{A}/\cong} = [c].$$

One can easily verify that the function $h_{\cong} : A \to A/\cong$ such that $h_{\cong}(x) = [x]$ is a homomorphism from $\mathbf{A}$ to $\mathbf{A}/\cong$.

**EXAMPLE 2.4.3**

Let $\Sigma = \{0, 1, -, +, *\}$ be the ranked alphabet of example 2.4.1. Let $A$ be any nonempty set and let $T_{\Sigma}(2^A)$ be the tree algebra freely generated by $2^A$. By lemma 2.4.2 and theorem 2.4.1, the identity function $Id : 2^A \to 2^A$ extends to a unique homomorphism $h : T_{\Sigma}(2^A) \to \mathcal{B}_A$, where the range of the function $h$ is the boolean algebra $\mathcal{B}_A$ (and not just its carrier $2^A$ as in the function $Id$). Let $\cong$ be the relation defined on $T_{\Sigma}(2^A)$ such that:

$$t_1 \cong t_2 \quad \text{if and only if} \quad h(t_1) = h(t_2).$$

For example, if $A = \{a, b, c, d\}$, for the trees $t_1$ and $t_2$ given by the terms $t_1 = +(\{a\}, *(\{b, c, d\}, \{a, b, c\}))$ and $t_2 = +(\{a, b\}, \{c\})$, we have

$$h(t_1) = h(t_2) = \{a, b, c\}.$$

It can be verified that $\cong$ is a congruence, and that the quotient algebra $T_{\Sigma}(2^A)/\cong$ is isomorphic to $\mathcal{B}_A$.

## 2.5 Many-Sorted Algebras

For many computer science applications, and for the definition of data types in particular, it is convenient to generalize algebras by allowing domains and operations of different types (also called sorts). A convenient way to do so is to introduce the concept of a *many-sorted algebra*. In Chapter 10, a generalization of first-order logic known as many-sorted first-order logic is also presented. Since the semantics of this logic is based on many-sorted algebras, we present in this section some basic material on many-sorted algebra.

Let $S$ be a set of *sorts* (or *types*). Typically, $S$ consists of types in a programming language (such as *integer*, *real*, *boolean*, *character*, etc.).

### 2.5.1 $S$-Ranked Alphabets

An *S-ranked alphabet* is pair $(\Sigma, r)$ consisting of a set $\Sigma$ together with a function $r : \Sigma \to S^* \times S$ assigning a *rank* $(u, s)$ to each symbol $f$ in $\Sigma$.

The string $u$ in $S^*$ is the *arity* of $f$ and $s$ is the *sort* (or *type*) of $f$. If $u = s_1...s_n$, $(n \geq 1)$, a symbol $f$ of rank $(u, s)$ is to be interpreted as an operation taking arguments, the $i$-th argument being of type $s_i$ and yielding a result of type $s$. A symbol of rank $(e, s)$ (when $u$ is the empty string ) is called a *constant of sort s*. For simplicity, a ranked alphabet $(\Sigma, r)$ is often denoted by $\Sigma$.

### 2.5.2 Definition of a Many-Sorted Algebra

Given an $S$-ranked alphabet $\Sigma$, a *many-sorted* $\Sigma$-algebra $\mathbf{A}$ is a pair $< A, I >$, where $A = (A_s)_{s \in S}$ is an $S$-indexed family of nonempty sets, each $A_s$ being called a *carrier of sort s*, and $I$ is an *interpretation function* assigning functions to the function symbols as follows:

(i) Each symbol $f$ of rank $(u, s)$ where $u = s_1...s_n$ is interpreted as a function $I(f) : A_{s_1} \times ... \times A_{s_n} \to A_s$;

(ii) Each constant $c$ of sort $s$ is interpreted as an element $I(c)$ in $A_s$.

The following abbreviations will also be used: $I(f)$ will be denoted as $f_{\mathbf{A}}$ and $I(c)$ as $c_{\mathbf{A}}$; If $u = s_1...s_n$, we let $A^u = A_{s_1} \times ... \times A_{s_n}$ and $A^e = \{e\}$. (Since there is a bijection between $A$ and $A \times \{e\}$, $A$ and $A \times \{e\}$ will be identified.) Given an $S$-indexed family $h = (h_s)_{s \in S}$ of functions $h_s : A_s \to B_s$, the function $h^u : A^u \to B^u$ is defined so that, for all $(a_1, ..., a_n) \in A^u$, $h^u(a_1, ..., a_n) = (h_{s_1}(a_1), ..., h_{s_n}(a_n))$.

**EXAMPLE 2.5.1**

The algebra of *stacks of natural numbers* is defined as follows. Let $S = \{\mathbf{int}, \mathbf{stack}\}$, $\Sigma = \{\Lambda, ERROR, POP, PUSH, TOP\}$, where $\Lambda$ is a constant of sort **stack**, $ERROR$ is a constant of sort **int**, $PUSH$ is a function symbol of rank (**int**.**stack**,**stack**), $POP$ a function symbol of rank (**stack**,**stack**) and $TOP$ a function symbol of rank (**stack**,**int**). The carrier of sort **int** is $\mathbf{N} \cup \{error\}$, and the carrier of sort **stack**, the set of functions of the form $X : [n] \to \mathbf{N} \cup \{error\}$, where $n \in \mathbf{N}$. When $n = 0$, the unique function from the empty set ([0]) is denoted by $\Lambda$ and is called the empty stack. The constant $ERROR$ is interpreted as the element *error*. Given any stack $X : [n] \to \mathbf{N} \cup \{error\}$ and any element $a \in \mathbf{N} \cup \{error\}$, $PUSH(a, X)$ is the stack $X' : [n+1] \to \mathbf{N} \cup \{error\}$ such that $X'(k) = X(k)$ for all $k$, $1 \leq k \leq n$, and $X'(n+1) = a$; $TOP(X)$ is the top element $X(n)$ of $X$ if $n > 0$, *error* if $n = 0$; $POP(X)$ is the empty stack $\Lambda$ if either $X$ is the empty stack or $n = 1$, or the stack $X' : [n-1] \to \mathbf{N} \cup \{error\}$ such that $X'(k) = X(k)$ for all $k$, $1 \leq k \leq n-1$ if $n > 1$.

*Note*: This formalization of a stack is not perfectly faithful because $TOP(X) = error$ does not necessarily imply that $X = \Lambda$. However, it is good enough as an example.

## 2.5.3 Homomorphisms

Given two many-sorted $\Sigma$-algebras $\mathbf{A}$ and $\mathbf{B}$, an $S$-indexed family $h = (h_s)_{s \in S}$ of functions $h_s : A_s \to B_s$ is a *homomorphism* iff:

(i) For every function symbol $f$ or rank $(u, s)$ with $u = s_1 ... s_n$, for every $(a_1, ..., a_n) \in A^u$,

$$h_s(f_{\mathbf{A}}(a_1, ..., a_n)) = f_{\mathbf{B}}(h_{s_1}(a_1), ..., h_{s_n}(a_n));$$

(ii) For every sort $s$, for every constant $c$ of sort $s$, $h_s(c_{\mathbf{A}}) = c_{\mathbf{B}}$.

These conditions can be represented by the following commutative diagram.

$$\begin{array}{ccc} A^u & \xrightarrow{f_{\mathbf{A}}} & A_s \\ h^u \downarrow & & \downarrow h \\ B^u & \xrightarrow[f_{\mathbf{B}}]{} & B_s \end{array}$$

## 2.5.4 Subalgebras

An algebra $\mathbf{B} =< B, J >$ is a *subalgebra* of an algebra $\mathbf{A} =< A, I >$ (with the same ranked alphabet $\Sigma$) iff:

(1) Each $B_s$ is a subset of $A_s$, for every sort $s \in S$;

(2) For every sort $s \in S$, for every constant $c$ of sort $s$, $c_{\mathbf{B}} = c_{\mathbf{A}}$, and for every function symbol $f$ of rank $(u, s)$ with $u = s_1 ... s_n$, $f_{\mathbf{B}} : B^u \to B_s$ is the restriction of $f_{\mathbf{A}} : A^u \to A_s$.

The fact that $\mathbf{B}$ is an algebra implies that $B$ is *closed under the operations*; that is, for every function symbol $f$ of rank $(u, s)$ with $u = s_1 ... s_n$, for every $(b_1, ..., b_n) \in B^u$, $f_{\mathbf{A}}(b_1, ..., b_n)$ is in $B_s$, and for every constant $c$ of sort $s$, $c_{\mathbf{A}}$ is in $B_s$.

### 2.5.5 Least Subalgebras

Given a $\Sigma$-algebra $\mathbf{A}$, let $X = (X_s)_{s \in S}$ be an $S$-indexed family with every $X_s$ a subset of $A_s$. As in the one-sorted case, the least subalgebra of $\mathbf{A}$ containing $X$ can be characterized by a bottom-up definition. We define this algebra $[\mathbf{X}]$ as follows.

The sequence of $S$-indexed families of sets $([X]_i)$, $(i \geq 0)$ is defined by induction: For every sort $s$,

$$[X_s]_0 = X_s \cup \{c_{\mathbf{A}} \mid c \;\text{ a constant of sort } s\},$$
$$[X_s]_{i+1} = [X_s]_i \cup \{f_{\mathbf{A}}(x_1, ..., x_n) \mid r(f) = (u, s), (x_1, ..., x_n) \in ([X]_i)^u\},$$

with $u = s_1 ... s_n$, $n \geq 1$.

The carrier of sort $s$ of the algebra $[\mathbf{X}]$ is

$$\bigcup_{i \geq 0} [X_s]_i.$$

*Important note*: The carriers of an algebra are always *nonempty*. To avoid having any carrier $[X_s]$ empty, we will make the assumption that for every sort $s$, $[X_s]_0 \neq \emptyset$. This will be satisfied if either $X_s$ is nonempty or there are constants of sort $s$. A more general condition can be given. Call a sort $s$ *nonvoid* if either there is some constant of sort $s$, or there is some function symbol $f$ of rank $(s_1 ... s_n, s)$ such that $s_1, ..., s_n$ are all non-void $(n \geq 1)$. Then, $[X]_s$ is nonempty if and only if either $X_s \neq \emptyset$ or $s$ is non-void.

We also have the following induction principle.

**Induction Principle for Least Subalgebras**

If $[\mathbf{X}]$ is the least subalgebra of $\mathbf{A}$ containing $X$, for every subfamily $Y$ of $[X]$, if $Y$ contains $X$ and is closed under the operations of $\mathbf{A}$ (and contains $\{c_{\mathbf{A}} \mid c \text{ is a constant}\}$) then $\mathbf{Y} = [\mathbf{X}]$.

### 2.5.6 Freely Generated Subalgebras

We say that $[\mathbf{X}]$ is *freely generated by $X$ in* $\mathbf{A}$ iff the following conditions hold:

(1) For every $f$ (not a constant) in $\Sigma$, the restriction of the function $f_{\mathbf{A}} : A^u \to A_s$ to $[X]^u$ is injective.

(2) For every $f_{\mathbf{A}} : A^u \to A_s$, $g_{\mathbf{A}} : A^v \to A_{s'}$ with $f, g \in \Sigma$, $f_{\mathbf{A}}([X]^u)$ is disjoint from $g_{\mathbf{A}}([X]^v)$ whenever $f \neq g$ (and $c_{\mathbf{A}} \neq d_{\mathbf{A}}$ for constants $c \neq d$).

(3) For every $f_{\mathbf{A}} : A^u \to A_s$ with $f \in \Sigma$ and every $(x_1, ..., x_n) \in [X]^u$, $f_{\mathbf{A}}(x_1, ..., x_n) \notin X_s$ (and $c_{\mathbf{A}} \notin X_s$, for a constant $c$ of sort $s$).

As in lemma 2.3.3 , it can be shown that for every $(x_1, ..., x_n)$ in $[X]^u_i - [X]^u_{i-1}$, $f_{\mathbf{A}}(x_1, ..., x_n) \notin [X_s]_i$, ($i \geq 0$, with $[X_s]_{-1} = \emptyset$).

We have the following generalization of theorem 2.4.1.

**Theorem 2.5.1** (Unique homomorphic extension theorem) Let $\mathbf{A}$ and $\mathbf{B}$ be two many-sorted $\Sigma$-algebras, $X$ an $S$-indexed family of subsets of $A$, let $[\mathbf{X}]$ be the least subalgebra of $\mathbf{A}$ containing $X$, and assume that $[\mathbf{X}]$ is freely generated by $X$. For every $S$-indexed family $h : X \to B$ of functions $h_s : X_s \to B_s$, there is a unique homomorphism $\widehat{h} : [\mathbf{X}] \to \mathbf{B}$ such that:

(1) For all $x \in X_s$, $\widehat{h}_s(x) = h_s(x)$ for every sort $s$;

For every function symbol $f$ of rank $(u, s)$, with $u = s_1 ... s_n$, for all $(x_1, ..., x_n) \in [X]^u$,

(2) $\widehat{h}_s(f_{\mathbf{A}}(x_1, ..., x_n)) = f_{\mathbf{B}}(\widehat{h}_{s_1}(x_1), ..., \widehat{h}_{s_n}(x_n))$, and $\widehat{h}_s(c_{\mathbf{A}}) = c_{\mathbf{B}}$ for a constant $c$ of sort $s$.

$$
\begin{array}{ccc}
X & \longrightarrow & [\mathbf{X}] \\
 & {\scriptstyle h} \searrow & \downarrow {\scriptstyle \widehat{h}} \\
 & & \mathbf{B}
\end{array}
$$

## 2.5.7 Congruences

Given a $\Sigma$-algebra $\mathbf{A}$, a *congruence* $\cong$ on $A$ is an $S$-indexed family $(\cong_s)_{s \in S}$ of relations, each $\cong_s$ being an equivalence relation on the carrier $A_s$, and satisfying the following conditions: For every function symbol $f$ of rank $(u, s)$, with $u = s_1 ... s_n$, for all $(x_1, ..., x_n)$ and $(y_1, ..., y_n)$ in $A^u$,

if $x_i \cong_{s_i} y_i$ for all $i$, $1 \leq i \leq n$, then

$$f_{\mathbf{A}}(x_1, ..., x_n) \cong_s f_{\mathbf{A}}(y_1, ..., y_n).$$

The equivalence class of $x$ modulo $\cong_s$ is denoted by $[x]_{\cong_s}$, or more simply by $[x]_s$ or $\widehat{x}_s$.

Given any function symbol $f$ of rank $(u, s)$, with $u = s_1 ... s_n$, and any $n$ subsets $B_1, ..., B_n$ such that $B_i$ is a subset of $A_{s_i}$, we define

$$f_{\mathbf{A}}(B_1, ..., B_n) = \{f_{\mathbf{A}}(b_1, ..., b_n) \mid (b_1, ..., b_n) \in A^u\}.$$

If $\cong$ is a congruence on $A$, for any $(a_1, ..., a_n) \in A^u$, $f_{\mathbf{A}}([a_1]_{s_1}, ..., [a_n]_{s_n})$ is a subset of some unique equivalence class which is in fact $[f_{\mathbf{A}}(a_1, ..., a_n)]_s$. Hence, we can define a structure of $\Sigma$-algebra on the $S$-indexed family $A/\cong$ of sets of equivalence classes.

## 2.5.8 Quotient Algebras

Given a $\Sigma$-algebra $\mathbf{A}$ and a congruence $\cong$ on $A$, the *quotient algebra* $\mathbf{A}/\cong$ has the $S$-indexed family $A/\cong$ of sets of equivalence classes modulo $\cong_s$ as its carriers, and its operations are defined as follows: For every function symbol $f$ of rank $(u, s)$, with $u = s_1...s_n$, for all $([a_1]_{s_1}, ..., [a_n]_{s_n}) \in (A/\cong)^u$,

$$f_{\mathbf{A}/\cong}([a_1]_{s_1}, ..., [a_n]_{s_n}) = [f_{\mathbf{A}}(a_1, ..., a_n)]_s,$$

and for every constant $c$ of sort $s$,

$$c_{\mathbf{A}/\cong_s} = [c]_s.$$

The $S$-indexed family $h_{\cong}$ of functions $h_{\cong_s} : A_s \to A/\cong_s$ such that $h_{\cong_s}(x) = [x]_{\cong_s}$ is a homomorphism from $\mathbf{A}$ to $\mathbf{A}/\cong$.

Finally, many-sorted trees are defined as follows.

## 2.5.9 Many-Sorted Trees

Given a many-sorted alphabet $\Sigma$ (with set $S$ of sorts), a $\Sigma$-*tree of sort* $s$ is any function $t : D \to \Sigma$ where $D$ is a tree domain denoted by $dom(t)$ and $t$ satisfies the following conditions:

1) The root of $t$ is labeled with a symbol $t(e)$ in $\Sigma$ of sort $s$.

2) For every node $u \in dom(t)$, if $\{i \mid ui \in dom(t)\} = [n]$, then if $n > 0$, for each $ui$, $i \in [n]$, if $t(ui)$ is a symbol of sort $v_i$, then $t(u)$ has rank $(v, s')$, with $v = v_1...v_n$, else if $n = 0$, then $t(u)$ has rank $(e, s')$, for some $s' \in S$.

The set of all finite trees of sort $s$ is denoted by $T^s_\Sigma$, and the set of all finite trees by $T_\Sigma$. Given an $S$-indexed family $X = (X_s)_{s \in S}$, we can form the sets of trees $T^s_\Sigma(X_s)$ obtained by adjoining each set $X_s$ to the set of constants of sort $s$. $T_\Sigma(X)$ is a $\Sigma$-algebra, and lemma 2.4.2 generalizes as follows.

**Lemma 2.5.1** For every many-sorted $\Sigma$-algebra $\mathbf{A}$ and $S$-indexed family $X = (X_s)_{s \in S}$, if $X \cap \Sigma = \emptyset$, then the $\Sigma$-algebra $T_\Sigma(X)$ of finite trees over the ranked alphabet obtained by adjoining each set $X_s$ to the set $\Sigma_{e,s}$ of constants of sort $s$ is freely generated by $X$.

**EXAMPLE 2.5.2**

Referring to example 2.5.1, let $\Sigma$ be the ranked alphabet of the algebra $\mathbf{A}$ of stacks. Let $T_\Sigma(\mathbf{N})$ be the algebra freely generated by the pair of sets $(\emptyset, \mathbf{N})$. The identity function on $(\emptyset, \mathbf{N})$ extends to a unique homormophism $h$ from $T_\Sigma(\mathbf{N})$ to $\mathbf{A}$. Define the relations $\cong_{\mathbf{int}}$ and $\cong_{\mathbf{stack}}$ on $T_\Sigma(\mathbf{N})$ as follows: For all $t_1, t_2$ of sort **stack**,

$$t_1 \cong_{\mathbf{int}} t_2 \text{ iff } h(t_1) = h(t_2),$$

and for all $t_1, t_2$ of sort **int**,

$$t_1 \cong_{\mathbf{stack}} t_2 \text{ iff } h(t_1) = h(t_2).$$

One can check that $\cong$ is a congruence, and that $T_\Sigma(\mathbf{N})/\cong$ is isomorphic to $\mathbf{A}$. One can also check that the following holds for all trees $X$ of sort **stack** and all trees $a$ of sort **int**:

$$\begin{aligned} POP(PUSH(a, X)) &\cong_{\mathbf{stack}} X, \\ POP(\Lambda) &\cong_{\mathbf{stack}} \Lambda, \\ TOP(PUSH(a, X)) &\cong_{\mathbf{int}} a, \\ TOP(\Lambda) &\cong_{\mathbf{int}} ERROR. \end{aligned}$$

The reader is referred to Cohn, 1981, or Gratzer, 1979, for a complete exposition of universal algebra. For more details on many-sorted algebras, the reader is referred to the article by Goguen,Thatcher,Wagner and Wright in Yeh, 1978, or the survey article by Huet and Oppen, in Book, 1980.

## PROBLEMS

**2.4.1.** Let $\mathbf{A}$ and $\mathbf{B}$ two $\Sigma$-algebras and $X$ a subset of $A$. Assume that $\mathbf{A}$ is the least subalgebra generated by $X$. Show that if $h_1$ and $\mathrm{h}_2$ are any two homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ such that $h_1$ and $h_2$ agree on $X$ (that is, $h_1(x) = h_2(x)$ for all $x \in X$), then $h_1 = h_2$.

**2.4.2.** Let $h : \mathbf{A} \to \mathbf{B}$ be a homomorphism of $\Sigma$-algebras.

(a) Given any subalgebra $\mathbf{X}$ of $\mathbf{A}$, prove that $h(X)$ is a subalgebra of $\mathbf{B}$ (denoted by $h(\mathbf{X})$).

(b) Given any subalgebra $\mathbf{Y}$ of $\mathbf{B}$, prove that $h^{-1}(Y)$ is a subalgebra of $\mathbf{A}$ (denoted by $h^{-1}(\mathbf{Y})$).

**2.4.3.** Let $h : \mathbf{A} \to \mathbf{B}$ be a homomorphism of $\Sigma$-algebras. Let $\cong$ be the relation defined on $A$ such that, for all $x, y \in A$,

$$x \cong y \quad \text{if and only if} \quad h(x) = h(y).$$

Prove that $\cong$ is a congruence on $A$, and that $h(\mathbf{A})$ is isomorphic to $\mathbf{A}/\cong$.

**2.4.4.** Prove that for every $\Sigma$-algebra $\mathbf{A}$, there is some tree algebra $T_\Sigma(X)$ freely generated by some set $X$ and some congruence $\cong$ on $T_\Sigma(X)$ such that $T_\Sigma(X)/\cong$ is isomorphic to $\mathbf{A}$.

**2.4.5.** Let $\mathbf{A}$ be a $\Sigma$-algebra, $X$ a subset of $A$, and assume that $[\mathbf{X}] = \mathbf{A}$, that is, $X$ generates $\mathbf{A}$.

Prove that if for every $\Sigma$-algebra $\mathbf{B}$ and function $h : X \to \mathbf{B}$ there is a unique homomorphism $\widehat{h} : \mathbf{A} \to \mathbf{B}$ extending $h$, then $\mathbf{A}$ is freely generated by $X$.

∗ **2.4.6.** Given a $\Sigma$-algebra $\mathbf{A}$ and any relation $R$ on $A$, prove that there is a least congruence $\cong$ containing $R$.

**2.5.1.** Do problem 2.4.1 for many-sorted algebras.

**2.5.2.** Do problem 2.4.2 for many-sorted algebras.

**2.5.3.** Do problem 2.4.3 for many-sorted algebras.

**2.5.4.** Do problem 2.4.4 for many-sorted algebras.

**2.5.5.** Do problem 2.4.5 for many-sorted algebras.

∗ **2.5.6.** Do problem 2.4.6 for many-sorted algebras.

∗ **2.5.7.** Referring to example 2.5.2, prove that the quotient algebra $T_\Sigma(\mathbf{N})/\cong$ is isomorphic to the stack algebra $\mathbf{A}$.

∗ **2.5.8.** Prove that the least congruence containing the relation $R$ defined below is the congruence $\cong$ of problem 2.5.7. The relation $R$ is defined such that, for all trees $X$ of sort **stack** and all trees $a$ of sort **int**:

$$\begin{aligned} POP(PUSH(a, X)) &\ R_{\mathbf{stack}}\ X, \\ POP(\Lambda) &\ R_{\mathbf{stack}}\ \Lambda, \\ TOP(PUSH(a, X)) &\ R_{\mathbf{int}}\ a, \\ TOP(\Lambda) &\ R_{\mathbf{int}}\ ERROR. \end{aligned}$$

This problem shows that the stack algebra is isomorphic to the quotient of the tree algebra $T_\Sigma(\mathbf{N})$ by the least congruence $\cong$ containing the above relation.