Chapter 1

Universal algebra

The most basic assumption of work on algebraic specification is that programs are modelled as algebras. This point of view abstracts from the concrete details of code and algorithms, and regards the input/output behaviour of functions and the representation of data as primary. Representing programs in terms of sets (of data values) and ordinary mathematical functions over these sets greatly simplifies the task of reasoning about program correctness. See Section 0.1 for some illustrative examples and more introductory discussion on this point.

The branch of mathematics which deals with algebras in this general sense (as opposed to the study of specific classes of algebras, such as groups and rings) is called *universal algebra* or sometimes *general algebra*. However, work on universal algebra by mathematicians has concentrated almost exclusively on the special case of single-sorted algebras with first-order total functions. The generalization to *many-sorted* or *heterogeneous* algebras is required to model programs which manipulate several kinds or *sorts* of data; further generalizations are necessary to handle programs which fail to terminate on some inputs, which generate exceptions during execution, etc. This chapter summarizes the basic concepts and results of many-sorted universal algebra which will be required for the rest of this book. Some extensions useful for modelling more complex programs will be discussed later, in Section 2.7. In this chapter, all proofs are left as exercises for the reader.

1.1 Many-sorted sets

When using an algebra to model a program which manipulates several sorts of data, it is natural to partition the underlying set of values in the algebra so that there is one set of values for each sort of data. It is often convenient to manipulate such a family of sets as a unit, in such a way that operations on this unit respect the "typing" of data values.

The following sequence of definitions and notational conventions allow us to manipulate sorted families of sets (of functions, of relations, ...) in just the same way as ordinary

sets (functions, relations, ...). Ordinary sets (functions, relations, ...) correspond to the degenerate case in which there is just one sort, so these definitions also serve to recall the notation and terminology of set theory to be used throughout this book. Let S be a set; the notation $\langle X_s \rangle_{s \in S}$ is a standard shorthand for the family of objects X_s indexed by $s \in S$, i.e. the function with domain S which maps each $s \in S$ to X_s.

Throughout this section, let S be a set (of sorts).

Definition 1.1.1 : Many-sorted set

An *S-sorted set* is an S-indexed family of sets $X = \langle X_s \rangle_{s \in S}$, which is *empty* if X_s is empty for all $s \in S$. The empty S-sorted set will be written (ambiguously) as $\varnothing$. The S-sorted set X is *finite* if X_s is finite for all $s \in S$ and there is a finite set $\widehat{S} \subseteq S$ such that $X_s = \varnothing$ for all $s \in S \setminus \widehat{S}$.

Let $X = \langle X_s \rangle_{s \in S}$ and $Y = \langle Y_s \rangle_{s \in S}$ be S-sorted sets. Union, intersection, Cartesian product, disjoint union, inclusion (subset) and equality of X and Y are defined componentwise as follows:

$X \cup Y = \langle X_s \cup Y_s \rangle_{s \in S}$

$X \cap Y = \langle X_s \cap Y_s \rangle_{s \in S}$

$X \times Y = \langle X_s \times Y_s \rangle_{s \in S}$

$X \uplus Y = \langle X_s \uplus Y_s \rangle_{s \in S}$ (where $X_s \uplus Y_s = (\{1\} \times X_s) \cup (\{2\} \times Y_s)$)

$X \subseteq Y$ iff (if and only if) $X_s \subseteq Y_s$ for all $s \in S$

$X = Y$ iff $X \subseteq Y$ and $Y \subseteq X$ (equivalently, iff X and Y are equal as functions).

□

Exercise 1.1.2

Give a formal explanation of the above statement that "Ordinary sets ... correspond to the degenerate case [of many-sorted sets] in which there is just one sort". How many $\varnothing$-sorted sets are there? □

NOTATION: It will be very convenient to pretend that $X \subseteq X \uplus Y$ and $Y \subseteq X \uplus Y$. Although this is never actually the case, it allows us to treat disjoint union in the same way as ordinary union, the difference being that when $X \cap Y \neq \varnothing$, $X \uplus Y$ contains two "copies" of the common elements and keeps track of which copy is from X and which from Y. To see that this does not cause problems, observe that there are injective S-sorted functions (see the next definition) $i1 \colon X \to X \uplus Y$ and $i2 \colon Y \to X \uplus Y$ defined by $i1_s(x) = \langle 1, x \rangle$ for all $s \in S$ and $x \in X_s$ and similarly for $i2$. A pedant would be able to correct what follows by simply inserting the functions $i1$ and/or $i2$ where appropriate in expressions involving $\uplus$.

Exercise 1.1.3

Extend the above definitions of union, intersection, product and disjoint union to operations on I-indexed families of S-sorted sets, for an arbitrary index set I. For example, the definition for product is $(\prod\langle X_i\rangle_{i\in I})_s = \{f\colon I \to \bigcup_{i\in I}(X_i)_s \mid f(i) \in (X_i)_s \text{ for all } i \in I\}$ for each $s \in S$. □

Definition 1.1.4 : Many-sorted function

Let $X = \langle X_s\rangle_{s\in S}$ and $Y = \langle Y_s\rangle_{s\in S}$ be S-sorted sets. An *S-sorted function* $f\colon X \to Y$ is an S-indexed family of functions $f = \langle f_s\colon X_s \to Y_s\rangle_{s\in S}$; X is called the *domain* (or *source*) of f, and Y is called its *codomain* (or *target*). An S-sorted function $f\colon X \to Y$ is an *identity* (an *inclusion*, *surjective*, *injective*, *bijective*, ...) if for every $s \in S$, the function $f_s\colon X_s \to Y_s$ is an identity (an inclusion, surjective, injective, bijective, ...). The identity S-sorted function on X will be written as $id_X\colon X \to X$.

If $f\colon X \to Y$ and $g\colon Y \to Z$ are S-sorted functions then their *composition* $f;g\colon X \to Z$ is the S-sorted function defined by $f;g = \langle f_s;g_s\rangle_{s\in S}$. That is, if $s \in S$ and $x \in X_s$ then $(f;g)_s(x) = g_s(f_s(x))$.[1]

Let $f\colon X \to Y$ be an S-sorted function and $X' \subseteq X$, $Y' \subseteq Y$ be S-sorted sets. The *image of X' under f* is the S-sorted set $f(X') = \langle f_s(X'_s)\rangle_{s\in S} \subseteq Y$, where $f_s(X'_s) = \{f_s(x) \mid x \in X'_s\} \subseteq Y_s$ for all $s \in S$. The *coimage of Y' under f* is the S-sorted set $f^{-1}(Y') = \langle f_s^{-1}(Y'_s)\rangle_{s\in S} \subseteq X$, where $f_s^{-1}(Y'_s) = \{x \in X_s \mid f_s(x) \in Y'_s\} \subseteq X_s$ for all $s \in S$. □

Definition 1.1.5 : Many-sorted binary relation

Let $X = \langle X_s\rangle_{s\in S}$ be an S-sorted set. An *S-sorted binary relation on X*, written $R \subseteq X \times X$, is an S-indexed family of binary relations $R = \langle R_s \subseteq X_s \times X_s\rangle_{s\in S}$. For $s \in S$ and $x, y \in X_s$, xR_sy (sometimes written xRy) means $\langle x, y\rangle \in R_s$.

Let $f\colon X \to Y$ be an S-sorted function. The *kernel of f* is the S-sorted binary relation $K(f) = \langle K(f_s)\rangle_{s\in S} \subseteq X \times X$ where $K(f_s) = \{\langle x, y\rangle \mid x, y \in X_s \text{ and } f_s(x) = f_s(y)\} \subseteq X_s \times X_s$ is the kernel of f_s for all $s \in S$. □

Definition 1.1.6 : Many-sorted equivalence

Let $X = \langle X_s\rangle_{s\in S}$ be an S-sorted set. An S-sorted binary relation $R \subseteq X \times X$ is an *S-sorted equivalence (relation) on X* if it is:

- reflexive: xR_sx;
- symmetric: xR_sy implies yR_sx; and
- transitive: xR_sy and yR_sz implies xR_sz

[1] This "diagrammatic" order of composition will be used consistently throughout this book.

for all $s \in S$ and $x, y, z \in X_s$. The symbol $\equiv$ is often used for (S-sorted) equivalence relations.

Let $\equiv$ be an S-sorted equivalence on X. If $s \in S$ and $x \in X_s$ then the *equivalence class of x modulo $\equiv$* is the set $[x]_{\equiv_s} = \{y \in X_s \mid x \equiv_s y\}$. The *quotient of X modulo $\equiv$* is the S-sorted set $X/\equiv = \langle X_s/\equiv_s \rangle_{s \in S}$ where $X_s/\equiv_s = \{[x]_{\equiv_s} \mid x \in X_s\}$ for all $s \in S$. □

Example 1.1.7

Let $S = \{s_1, s_2\}$, and let X and Y be two S-sorted sets defined as follows:

$$X = \langle X_s \rangle_{s \in S} \text{ where } X_{s_1} = \{\square, \triangle\} \text{ and } X_{s_2} = \{\clubsuit, \heartsuit, \spadesuit\},$$
$$Y = \langle Y_s \rangle_{s \in S} \text{ where } Y_{s_1} = \{1, 2, 3\} \text{ and } Y_{s_2} = \{1, 2, 3\}.$$

Let $f\colon X \to Y$ be the S-sorted function such that

$$f_{s_1} = \{\square \mapsto 1, \triangle \mapsto 3\} \quad \text{and} \quad f_{s_2} = \{\clubsuit \mapsto 1, \heartsuit \mapsto 2, \spadesuit \mapsto 2\}.$$

(i.e., $f_{s_1}(\square) = 1$ and $f_{s_1}(\triangle) = 3$; analogously for f_{s_2}). Then the kernel of f is the S-sorted equivalence relation $K(f) = \langle K(f_s) \rangle_{s \in S}$ where

$$K(f_{s_1}) = \{\langle \square, \square \rangle, \langle \triangle, \triangle \rangle\} \quad \text{and} \quad K(f_{s_2}) = \{\langle \clubsuit, \clubsuit \rangle, \langle \heartsuit, \heartsuit \rangle, \langle \heartsuit, \spadesuit \rangle, \langle \spadesuit, \heartsuit \rangle, \langle \spadesuit, \spadesuit \rangle\}.$$

The quotient of X modulo $K(f)$ is the S-sorted set $X/K(f) = \langle X_s/K(f_s) \rangle_{s \in S}$ where

$$X_{s_1}/K(f_{s_1}) = \{\{\square\}, \{\triangle\}\} \quad \text{and} \quad X_{s_2}/K(f_{s_2}) = \{\{\clubsuit\}, \{\heartsuit, \spadesuit\}\}.$$

□

Exercise 1.1.8

Show that if $f\colon X \to Y$ is an S-sorted function, then $K(f)$ is an S-sorted equivalence on X. □

Exercise 1.1.9

Show that if $\equiv$ is an S-sorted equivalence on X then for all $s \in S$ and $x, y \in X_s$, $[x]_{\equiv_s} = [y]_{\equiv_s}$ iff $x \equiv_s y$. □

NOTATION: Subscripts selecting components of S-sorted sets (functions, relations, ...) are often omitted when there is no danger of confusion. Then Exercise 1.1.9 would read: "... for all $s \in S$ and $x, y \in X_s$, $[x]_{\equiv} = [y]_{\equiv}$ iff $x \equiv y$."

1.2 Signatures and algebras

The functions and data types defined by a program have names. These names are used to compute with and reason about the program, and to build larger programs which rely on the functionality the program provides. The connection between a program and an algebra used to model it is provided by these names, which are attached to the corresponding components of the algebra. The set of names associated with an algebra is called its signature. The signature of an algebra defines the *syntax* of the algebra by characterising the ways in which its components may legally be combined; the algebra itself supplies the *semantics* by assigning interpretations to the names in the signature.

Definition 1.2.1 : Many-sorted signature

A *(many-sorted) signature* is a pair $\Sigma = \langle S, \Omega \rangle$, where:

- S is a set (of sort names); and
- Ω is an $S^* \times S$-sorted set (of operation names)

where S^* is the set of finite (including empty) sequences of elements of S. We will sometimes write $sorts(\Sigma)$ for S and $opns(\Sigma)$ for Ω. Σ is a *subsignature* of a signature $\Sigma' = \langle S', \Omega' \rangle$ if $S \subseteq S'$ and $\Omega_{w,s} \subseteq \Omega'_{w,s}$ for all $w \in S^*, s \in S$. □

Many-sorted signatures will be referred to as *algebraic* signatures when it is necessary to distinguish them from other kinds of signatures to be introduced later.

NOTATION: Saying that $f\colon s_1 \times \cdots \times s_n \to s$ is in $\Sigma = \langle S, \Omega \rangle$ means that $s_1 \ldots s_n \in S^*$, $s \in S$ and $f \in \Omega_{s_1 \ldots s_n, s}$. Then f is said to have *arity* $s_1 \ldots s_n$ and *result sort* s. The abbreviation $f\colon s$ will be used for $f\colon \varepsilon \to s$ (ε is the empty sequence).

This definition of signature does not accommodate programs containing higher-order functions, or functions returning multiple results. A possible extension to handle higher-order functions is briefly discussed in Section 2.7.6. As for functions with multiple results, a function $f\colon s_1 \times \cdots \times s_n \to t_1 \times \cdots \times t_m$ may be viewed as a family of m functions

$$f_1\colon s_1 \times \cdots \times s_n \to t_1 \qquad \ldots \qquad f_m\colon s_1 \times \cdots \times s_n \to t_m.$$

Generalizing the definition of signature to handle such functions in a more direct way is easy but makes subsequent developments somewhat messier in a non-interesting way.

The definition above *does* permit overloading, since it is possible to have both $f\colon s_1 \times \cdots \times s_n \to s$ and $f\colon t_1 \times \cdots \times t_m \to t$ in a signature Σ, where $s_1 \ldots s_n s \neq t_1 \ldots t_m t$. A more restrictive definition of signature, adequate for most purposes, would have a set Ω of operation names (and a set S of sort names) with functions $arity\colon \Omega \to S^*$ and $sort\colon \Omega \to S$. These two definitions are equivalent if each operation name in Ω is taken to be tagged with its arity and result sort.

In the rest of this section, let $\Sigma = \langle S, \Omega \rangle$ be a signature.

Definition 1.2.2 : Many-sorted algebra

A Σ-*algebra* A consists of:

- an S-sorted set $|A|$ of *carrier sets* (or *carriers*); and
- for each $f\colon s_1 \times \cdots \times s_n \to s$ in Σ, a function (or *operation*) $(f\colon s_1 \times \cdots \times s_n \to s)_A\colon |A|_{s_1} \times \cdots \times |A|_{s_n} \to |A|_s$.

□

If A is a Σ-algebra and s is a sort name in Σ then $|A|_s$, the carrier set of sort s in A, is the universe of data values of sort s; accordingly, we often refer to the elements of carrier sets as *values*. If $f\colon s_1 \times \cdots \times s_n \to s$ is in Σ then the operation $(f\colon s_1 \times \cdots \times s_n \to s)_A$ is a function on the corresponding carrier sets of A. If $n = 0$ (i.e. $f\colon s$), then $|A|_{s_1} \times \cdots \times |A|_{s_n}$ is a singleton set containing the empty tuple $\langle\rangle$, and then $(f\colon s)_A$ may be viewed as a constant denoting the value $(f\colon s)_A(\langle\rangle) \in |A|_s$. Notice that $(f\colon s_1 \times \cdots \times s_n \to s)_A$ is a *total* function[2] so algebras as defined here are only appropriate for modelling programs containing total functions. See Sections 2.7.3–2.7.5 for several ways of extending the definitions to cope with partial functions. Note also that there is no restriction on the cardinality of $|A|_s$; in particular, $|A|_s$ may be empty and need not be countable.

NOTATION: Let A be a Σ-algebra and let $f\colon s_1 \times \cdots \times s_n \to s$ be in Σ. We always write f_A in place of $(f\colon s_1 \times \cdots \times s_n \to s)_A$ when there is no danger of confusion. When $n = 0$ (i.e. $f\colon s$), we write $(f\colon s)_A$ or f_A in place of $(f\colon s)_A(\langle\rangle)$.

Exercise 1.2.3

If $\Omega_{\varepsilon,s} \neq \varnothing$ for some $s \in S$, then there are no $\langle S, \Omega\rangle$-algebras having an empty carrier of sort s. Characterise signatures for which all algebras have non-empty carriers of all sorts.

□

Example 1.2.4

Let $S1 = \{shape, suit\}$ and let $\Omega 1_{\varepsilon,shape} = \{box\}$, $\Omega 1_{\varepsilon,suit} = \{hearts\}$, $\Omega 1_{shape,shape} = \{boxify\}$, $\Omega 1_{shape\ suit,suit} = \{f\}$, and $\Omega 1_{w,s} = \varnothing$ for all other $w \in S1^*, s \in S1$. Then $\Sigma 1 = \langle S1, \Omega 1\rangle$ is a signature with sort names *shape* and *suit* and operation names $box\colon shape$, $hearts\colon suit$, $boxify\colon shape \to shape$ and $f\colon shape \times suit \to suit$. We can present $\Sigma 1$ in tabular form as follows (this notation will be used later with the obvious meaning):

$$\begin{array}{rcll} \Sigma 1 & = & \textbf{sorts} & shape, suit \\ & & \textbf{opns} & box\colon shape \\ & & & hearts\colon suit \\ & & & boxify\colon shape \to shape \\ & & & f\colon shape \times suit \to suit \end{array}$$

[2] All functions in this book are total except where they are explicitly designated as partial.

Let

$$|A1|_{shape} = \{\square, \triangle\} \quad \text{and} \quad |A1|_{suit} = \{\clubsuit, \heartsuit, \spadesuit\},$$

$$box_{A1} = \square \in |A1|_{shape}, \quad hearts_{A1} = \heartsuit \in |A1|_{suit},$$

$$boxify_{A1}\colon |A1|_{shape} \to |A1|_{shape} = \{\square \mapsto \square, \triangle \mapsto \square\},$$

and let

$$f_{A1}\colon |A1|_{shape} \times |A1|_{suit} \to |A1|_{suit}$$

be defined by the following table:

f_{A1}	$\clubsuit$	$\heartsuit$	$\spadesuit$
$\square$	$\clubsuit$	$\spadesuit$	$\heartsuit$
$\triangle$	$\heartsuit$	$\spadesuit$	$\spadesuit$

This defines a $\Sigma 1$-algebra $A1$. (NOTE: Reference will be made to $\Sigma 1$ and $A1$ in examples throughout the rest of this chapter.) □

Definition 1.2.5 : Subalgebra

Let A and B be Σ-algebras. B is a *subalgebra* of A if:

- $|B| \subseteq |A|$; and
- for $f\colon s_1 \times \cdots \times s_n \to s$ in Σ and $b_1 \in |B|_{s_1}, \ldots, b_n \in |B|_{s_n}$, $f_B(b_1, \ldots, b_n) = f_A(b_1, \ldots, b_n)$.

B is a *proper* subalgebra of A if it is a subalgebra of A and $|B| \neq |A|$. A subalgebra of A is determined by an S-sorted subset $|B|$ of $|A|$ which is closed under the operations of Σ, i.e. such that for each $f\colon s_1 \times \cdots \times s_n \to s$ in Σ and $b_1 \in |B|_{s_1}, \ldots, b_n \in |B|_{s_n}$, $f_A(b_1, \ldots, b_n) \in |B|_s$. □

If B is a (proper) subalgebra of A then B is "smaller" than A in the sense that it contains fewer *data values* than A. Both A and B are Σ-algebras though, so A and B contain interpretations for exactly the same sort and operation names.

Exercise 1.2.6

Let A be a Σ-algebra. Show that the intersection of any family of (carriers of) subalgebras of A is a (carrier of a) subalgebra of A. Use this to show that for any $X \subseteq |A|$, there is a least subalgebra of A that contains X. This is called the *subalgebra of A generated by X*. Give an explicit construction of this algebra. (HINT: Consider the family of S-sorted sets $X_i \subseteq |A|$, $i \geq 0$, where $X_0 = X$ and X_{i+1} is obtained from X_i by adding the results of applying the operations of A to arguments in X_i.) □

Definition 1.2.7 : Reachable algebra

Let A be a Σ-algebra. A is *reachable* if A has no proper subalgebra (equivalently, if A is generated by $\varnothing$). □

By Exercise 1.2.6, every algebra has a unique reachable subalgebra.

Example 1.2.8

Let $\Sigma 1 = \langle S1, \Omega 1\rangle$ and $A1$ be as defined in Example 1.2.4. Define a $\Sigma 1$-algebra $B1$ by

$$|B1|_{shape} = \{\square\}, \quad |B1|_{suit} = \{\heartsuit, \spadesuit\},$$

$$box_{B1} = \square \in |B1|_{shape}, \quad hearts_{B1} = \heartsuit \in |B1|_{suit},$$

$$boxify_{B1}\colon |B1|_{shape} \to |B1|_{shape} = \{\square \mapsto \square\},$$

$$f_{B1}\colon |B1|_{shape} \times |B1|_{suit} \to |B1|_{suit} = \{\langle \square, \heartsuit\rangle \mapsto \spadesuit, \langle \square, \spadesuit\rangle \mapsto \heartsuit\}.$$

$B1$ is the subalgebra of $A1$ generated by $\varnothing$. That is, $B1$ is the reachable subalgebra of $A1$. □

Definition 1.2.9 : Product algebra

Let A and B be Σ-algebras. The *product algebra* $A \times B$ is the Σ-algebra defined as follows:

- $|A \times B| = |A| \times |B|$; and
- for each $f\colon s_1 \times \cdots \times s_n \to s$ in Σ and $\langle a_1, b_1\rangle \in |A \times B|_{s_1}, \ldots, \langle a_n, b_n\rangle \in |A \times B|_{s_n}$, $f_{A\times B}(\langle a_1, b_1\rangle, \ldots, \langle a_n, b_n\rangle) = \langle f_A(a_1, \ldots, a_n), f_B(b_1, \ldots, b_n)\rangle \in |A \times B|_s$.

This generalizes to the product $\prod\langle A_i\rangle_{i\in I}$ of a family of Σ-algebras, indexed by an arbitrary set I (possibly empty), as follows:

- $|\prod\langle A_i\rangle_{i\in I}| = \prod\langle |A_i|\rangle_{i\in I}$; and
- for each $f\colon s_1 \times \cdots \times s_n \to s$ in Σ and $f_1 \in |\prod\langle A_i\rangle_{i\in I}|_{s_1}, \ldots, f_n \in |\prod\langle A_i\rangle_{i\in I}|_{s_n}$, $f_{\prod\langle A_i\rangle_{i\in I}}(f_1, \ldots, f_n)(i) = f_{A_i}(f_1(i), \ldots, f_n(i))$ for all $i \in I$.

□

Exercise 1.2.10

Definition 1.2.9 shows how two Σ-algebras can be combined to form a new Σ-algebra by taking the Cartesian product of their carriers. According to Exercise 1.2.6, the same thing can be done (with subalgebras of a fixed algebra) using intersection. Try to formulate definitions of *union* and *disjoint union* of algebras, where $|A \cup B| = |A| \cup |B|$ and $|A \uplus B| = |A| \uplus |B|$ respectively. What happens? □

1.3 Homomorphisms and congruences

A homomorphism between algebras is the analogue of a function between sets, and a congruence relation on an algebra is the analogue of an equivalence relation on a set. An algebra has more structure than a set, so homomorphisms and congruences are required to respect the additional structure (i.e. the behaviour of the operations). Homomorphisms and congruences are important basic tools for relating algebras and constructing new algebras from old ones.

Throughout this section, let $\Sigma = \langle S, \Omega \rangle$ be a signature.

Definition 1.3.1 : Homomorphism

Let A and B be Σ-algebras. A Σ-*homomorphism* $h\colon A \to B$ is an S-sorted function $h\colon |A| \to |B|$ which respects the operations of Σ, i.e. such that for all $f\colon s_1 \times \cdots \times s_n \to s$ in Σ and $a_1 \in |A|_{s_1}, \ldots, a_n \in |A|_{s_n}$, $h_s(f_A(a_1, \ldots, a_n)) = f_B(h_{s_1}(a_1), \ldots, h_{s_n}(a_n))$. A Σ-homomorphism $h\colon A \to B$ is an *identity* (an *inclusion*, *surjective*, ...) if it is an identity (an inclusion, surjective, ...) when viewed as an S-sorted function. □

NOTATION: If $h\colon A \to B$ is a Σ-homomorphism, then $|h|\colon |A| \to |B|$ denotes h viewed as an S-sorted function. The only difference between h and $|h|$ is that in the case of $|h|$ we have "forgotten" that the additional condition required of a homomorphism is satisfied.

Informally, the homomorphism condition says that the behaviour of the operations in A is reflected in that of the operations in B. This condition can be expressed in the form of a diagram as follows:

$$\begin{array}{ccc} |A|_{s_1} \times \cdots \times |A|_{s_n} & \xrightarrow{h_{s_1} \times \cdots \times h_{s_n}} & |B|_{s_1} \times \cdots \times |B|_{s_n} \\ \Big\downarrow{\scriptstyle f_A} & & \Big\downarrow{\scriptstyle f_B} \\ |A|_s & \xrightarrow[h_s]{} & |B|_s \end{array}$$

where $(h_{s_1} \times \cdots \times h_{s_n})(a_{s_1}, \ldots, a_{s_n}) = (h_{s_1}(a_{s_1}), \ldots, h_{s_n}(a_{s_n}))$ for all $a_1 \in |A|_{s_1}, \ldots, a_n \in |A|_{s_n}$. The homomorphism condition amounts to the requirement that this diagram *commutes*, i.e. that composing the functions on the top and right-hand arrows gives the same result as composing the functions on the left-hand and bottom arrows. Such commuting diagrams will be used heavily in later chapters, particularly in Chapter 3.

Example 1.3.2

Let $\Sigma 1 = \langle S1, \Omega 1 \rangle$ and $A1$ be as defined in Example 1.2.4. Define a $\Sigma 1$-algebra $B1$ by

$$|B1|_{shape} = |B1|_{suit} = \{1, 2, 3\},$$

$$box_{B1} = 1 \in |B1|_{shape}, \quad hearts_{B1} = 2 \in |B1|_{suit},$$

$$boxify_{B1}\colon |B1|_{shape} \to |B1|_{shape} = \{1 \mapsto 1, 2 \mapsto 3, 3 \mapsto 1\},$$

where

$$f_{B1}\colon |B1|_{shape} \times |B1|_{suit} \to |B1|_{suit}$$

is defined by the following table:

f_{B1}	1	2	3
1	1	2	3
2	2	1	2
3	2	2	1

Let $h1\colon |A1| \to |B1|$ be the $S1$-sorted function such that

$$h1_{shape} = \{\square \mapsto 1, \triangle \mapsto 3\} \quad \text{and} \quad h1_{suit} = \{\clubsuit \mapsto 1, \heartsuit \mapsto 2, \spadesuit \mapsto 2\}.$$

It is easy to verify that $h1\colon A1 \to B1$ is a $\Sigma 1$-homomorphism by checking the following:

$$\begin{aligned}
h1_{shape}(box_{A1}) &= box_{B1} \\
h1_{suit}(hearts_{A1}) &= hearts_{B1} \\
h1_{shape}(boxify_{A1}(\square)) &= boxify_{B1}(h1_{shape}(\square)) \\
h1_{shape}(boxify_{A1}(\triangle)) &= boxify_{B1}(h1_{shape}(\triangle)) \\
h1_{suit}(f_{A1}(\square, \clubsuit)) &= f_{B1}(h1_{shape}(\square), h1_{suit}(\clubsuit)) \\
h1_{suit}(f_{A1}(\square, \heartsuit)) &= f_{B1}(h1_{shape}(\square), h1_{suit}(\heartsuit)) \\
h1_{suit}(f_{A1}(\square, \spadesuit)) &= f_{B1}(h1_{shape}(\square), h1_{suit}(\spadesuit)) \\
h1_{suit}(f_{A1}(\triangle, \clubsuit)) &= f_{B1}(h1_{shape}(\triangle), h1_{suit}(\clubsuit)) \\
h1_{suit}(f_{A1}(\triangle, \heartsuit)) &= f_{B1}(h1_{shape}(\triangle), h1_{suit}(\heartsuit)) \\
h1_{suit}(f_{A1}(\triangle, \spadesuit)) &= f_{B1}(h1_{shape}(\triangle), h1_{suit}(\spadesuit)).
\end{aligned}$$

□

Exercise 1.3.3

Let A be a Σ-algebra. Show that $id_{|A|}\colon A \to A$ (the identity S-sorted function) is a Σ-homomorphism. Let $h\colon A \to B$ and $h'\colon B \to C$ be Σ-homomorphisms. Show that $|h|;|h'|\colon |A| \to |C|$ is a Σ-homomorphism $h;h'\colon A \to C$. □

Exercise 1.3.4

Let $h\colon A \to B$ be a Σ-homomorphism, and let A' be a subalgebra of A. Let the *image of* A' *under* h be the Σ-algebra $h(A')$ defined as follows:

- $|h(A')| = |h|(|A'|)$; and

- for each $f\colon s_1\times\cdots\times s_n \to s$ in Σ and $a_1 \in |A'|_{s_1}, \ldots, a_n \in |A'|_{s_n}$, $f_{h(A')}(h_{s_1}(a_1), \ldots, h_{s_n}(a_n)) = h_s(f_{A'}(a_1, \ldots, a_n))$.

Show that $h(A')$ is a well-defined Σ-algebra (in particular, that the function $f_{h(A')}\colon |h(A')|_{s_1} \times \cdots \times |h(A')|_{s_n} \to |h(A')|_s$ is well-defined for each $f\colon s_1 \times \cdots \times s_n \to s$ in Σ) and that it is a subalgebra of B. Formulate a definition of the *coimage* of a subalgebra B' of B under h, and show that it is a subalgebra of A. □

Exercise 1.3.5

Let $h\colon A \to B$ be a Σ-homomorphism, and suppose $X \subseteq |A|$. Show that the subalgebra of B generated by $|h|(X) \subseteq |B|$ is the image of the subalgebra of A generated by X. Show that it follows that if $h\colon A \to B$ is surjective and A is reachable then B is reachable. □

Exercise 1.3.6

Let B be a reachable Σ-algebra. Show that for any Σ-algebra A, there is at most one Σ-homomorphism $h\colon B \to A$, and that any Σ-homomorphism $h\colon A \to B$ is surjective. □

Definition 1.3.7 : Isomorphism

Let A and B be Σ-algebras. A Σ-homomorphism $h\colon A \to B$ is a Σ-*isomorphism* if it has an inverse, i.e. there is a Σ-homomorphism $h^{-1}\colon B \to A$ such that $h;h^{-1} = id_{|A|}$ and $h^{-1};h = id_{|B|}$. (**Exercise:** Show that if h^{-1} exists then it is unique.) Then A and B are called *isomorphic* and we write $h\colon A \cong B$ or just $A \cong B$. □

Exercise 1.3.8

Let $h\colon A \cong B$ and $h'\colon B \cong C$ be Σ-isomorphisms. Show that their composition is a Σ-isomorphism $h;h'\colon A \cong C$. Show that $\cong$ (as a binary relation on Σ-algebras) is reflexive and symmetric, and is therefore an equivalence relation. □

Two isomorphic algebras are typically regarded as indistinguishable for all practical purposes. It is easy to see why: the only way in which they can differ is in the particular choice of data values in the carriers. The size of the carriers and the way that the operations behave on the values in the carriers is exactly the same. For this reason we are often satisfied with a definition of an algebra "up to isomorphism", i.e. a description of an isomorphism class of algebras in a context where one would expect a definition of a single algebra. An example of this is in Fact 1.4.10 below. The notion of isomorphism can be generalized to other kinds of structures, where it embodies exactly the same concept of indistinguishability. See Chapter 3 for this generalization and for many more examples of definitions of objects "up to isomorphism".

Example 1.3.9

Let $\Sigma 1 = \langle S1, \Omega 1\rangle$ and $A1$ be as defined in Example 1.2.4. Define a $\Sigma 1$-algebra $B1$ by

$$|B1|_{shape} = \{\square, \triangle\}, \quad |B1|_{suit} = \{1, 2, 3\},$$

$$box_{B1} = \triangle \in |B1|_{shape}, \quad hearts_{B1} = 2 \in |B1|_{suit},$$

$$boxify_{B1}\colon |B1|_{shape} \to |B1|_{shape} = \{\square \mapsto \triangle, \triangle \mapsto \triangle\},$$

where

$$f_{B1}\colon |B1|_{shape} \times |B1|_{suit} \to |B1|_{suit}$$

is defined by the following table:

f_{B1}	1	2	3
$\square$	2	3	3
$\triangle$	1	3	2

Let $i1\colon |A1| \to |B1|$ be the $S1$-sorted function such that

$$i1_{shape} = \{\square \mapsto \triangle, \triangle \mapsto \square\} \quad \text{and} \quad i1_{suit} = \{\clubsuit \mapsto 1, \heartsuit \mapsto 2, \spadesuit \mapsto 3\}.$$

This defines a $\Sigma 1$-homomorphism $i1\colon A1 \to B1$ which is a $\Sigma 1$-isomorphism, so $A1 \cong B1$.

□

Exercise 1.3.10

Show that a homomorphism is an isomorphism iff it is bijective. □

Exercise 1.3.11

Show that there is an injective homomorphism $h\colon A \to B$ iff A is isomorphic to a subalgebra of B. □

Example 1.3.12

Let $\Sigma = \langle S, \Omega \rangle$ be the signature

sorts s
opns $a\colon s$
$f\colon s \to s$

and define Σ-algebras A and B by

$$|A|_s = Nat \text{ (the natural numbers)},$$

$$a_A = 0 \in |A|_s,$$

$$f_A\colon |A|_s \to |A|_s = \{n \mapsto n + 1 \mid n \in Nat\},$$

$$|B|_s = \{n \in Nat \mid \text{the Turing machine with Gödel number } n \text{ halts on all inputs}\},$$

$$a_B = \text{the smallest } n \in |B|_s,$$

$$f_B\colon |B|_s \to |B|_s = \{n \in |B|_s \mapsto \text{the smallest } m \in |B|_s \text{ such that } m > n\}.$$

Let $i\colon |A| \to |B|$ be the S-sorted function such that

$$i_s(n) = \text{the } (n+1)^{\text{st}} \text{ smallest element of } |B|_s$$

for all $n \in |A|_s$. The function i_s is well-defined since $|B|_s$ is infinite. This defines a Σ-homomorphism $i\colon A \to B$ which is an isomorphism.

Although $A \cong B$, the Σ-algebras A and B are not "the same" from the point of view of computability: everything in A is computable, in contrast to B ($|B|_s$ is not recursively enumerable and f_B is not computable). Isomorphisms capture *structural* similarity, ignoring what the values in the carriers are and what the operations actually compute. This example shows that, for some purposes, properties stronger than structural similarity are important. □

Definition 1.3.13 : Congruence

Let A be a Σ-algebra. A Σ-*congruence on* A is an (S-sorted) equivalence $\equiv$ on $|A|$ which respects the operations of Σ: for all $f\colon s_1 \times \cdots \times s_n \to s$ in Σ and $a_1, a_1' \in |A|_{s_1}, \ldots, a_n, a_n' \in |A|_{s_n}$, if $a_1 \equiv_{s_1} a_1'$ and $\ldots$ and $a_n \equiv_{s_n} a_n'$ then $f_A(a_1, \ldots, a_n) \equiv_s f_A(a_1', \ldots, a_n')$. □

Exercise 1.3.14

Show that the intersection of any family of Σ-congruences on A is a Σ-congruence on A. Use this to show that for any S-sorted binary relation R on $|A|$ there is a least (with respect to $\subseteq$) Σ-congruence on A which includes R.

Show that the kernel of any Σ-homomorphism $h\colon A \to B$ is a Σ-congruence on A.

Show that a surjective Σ-homomorphism is an isomorphism iff its kernel is the identity. □

Definition 1.3.15 : Quotient algebra

Let A be a Σ-algebra, and let $\equiv$ be a Σ-congruence on A. The *quotient algebra of* A *modulo* $\equiv$ is the Σ-algebra $A/{\equiv}$ defined by:

- $|A/{\equiv}| = |A|/{\equiv}$; and
- for each $f\colon s_1 \times \cdots \times s_n \to s$ and $a_1 \in |A|_{s_1}, \ldots, a_n \in |A|_{s_n}$, $f_{A/\equiv}([a_1]_{\equiv_{s_1}}, \ldots, [a_n]_{\equiv_{s_n}}) = [f_A(a_1, \ldots, a_n)]_{\equiv_s}$.

□

Exercise 1.3.16

Show that $A/{\equiv}$ in Definition 1.3.15 is a well-defined Σ-algebra. □

Example 1.3.17

Let $\Sigma 1 = \langle S1, \Omega 1\rangle$ and $A1$ be as defined in Example 1.2.4, and let $\equiv = \langle \equiv_s \rangle_{s \in S1}$ be the $S1$-sorted binary relation on $|A1|$ defined by

$$\equiv_{shape} = \{\langle \square, \square\rangle, \langle \triangle, \triangle\rangle\} \quad \text{and} \quad \equiv_{suit} = \{\langle \clubsuit, \clubsuit\rangle, \langle \heartsuit, \heartsuit\rangle, \langle \heartsuit, \spadesuit\rangle, \langle \spadesuit, \heartsuit\rangle, \langle \spadesuit, \spadesuit\rangle\}.$$

This defines a congruence on $A1$. $A1/\equiv$ is the $\Sigma 1$-algebra defined by

$$|A1/\equiv|_{shape} = \{\{\square\}, \{\triangle\}\}, \quad |A1/\equiv|_{suit} = \{\{\clubsuit\}, \{\heartsuit, \spadesuit\}\},$$

$$box_{A1/\equiv} = \{\square\} \in |A1/\equiv|_{shape}, \quad hearts_{A1/\equiv} = \{\heartsuit, \spadesuit\} \in |A1/\equiv|_{suit},$$

$$boxify_{A1/\equiv}\colon |A1/\equiv|_{shape} \to |A1/\equiv|_{shape} = \{\{\square\} \mapsto \{\square\}, \{\triangle\} \mapsto \{\square\}\},$$

where

$$f_{A1/\equiv}\colon |A1/\equiv|_{shape} \times |A1/\equiv|_{suit} \to |A1/\equiv|_{suit}$$

is defined by the following table:

$f_{A1/\equiv}$	$\{\clubsuit\}$	$\{\heartsuit, \spadesuit\}$
$\{\square\}$	$\{\clubsuit\}$	$\{\heartsuit, \spadesuit\}$
$\{\triangle\}$	$\{\heartsuit, \spadesuit\}$	$\{\heartsuit, \spadesuit\}$

□

Exercise 1.3.18

Let $\equiv$ be a Σ-congruence on A, and let $h_s(a) = [a]_{\equiv_s}$ for $s \in S$, $a \in |A|_s$. Show that $\langle h_s\colon |A|_s \to (|A|/\equiv)_s\rangle_{s\in S}$ is a Σ-homomorphism $h\colon A \to A/\equiv$ with $K(h) = \equiv$. □

Exercise 1.3.19

Let $h\colon A \to B$ be a Σ-homomorphism. Show that $A/K(h)$ is isomorphic to $h(A)$. (HINT: The isomorphism is given by $[a]_{K(h_s)} \mapsto h_s(a)$ for $s \in S$, $a \in |A|_s$.) □

Exercise 1.3.20

Let $\equiv$ be a Σ-congruence on A. Show that for any Σ-homomorphism $h\colon A \to B$ such that $\equiv \subseteq K(h)$, there exists a unique Σ-homomorphism $g\colon A/\equiv \to B$ such that $h_s(a) = g_s([a]_{\equiv_s})$ for all $s \in S$, $a \in |A|_s$. □

Exercise 1.3.21

Show that there is a surjective homomorphism $h\colon A \to B$ iff there is a congruence $\equiv$ on A such that B is isomorphic to $A/\equiv$. □

Exercise 1.3.22

Let A be a Σ-algebra, let $\equiv$ be a congruence on A and let B be a subalgebra of $A/\equiv$. Show that there is a subalgebra C of A and congruence $\equiv'$ on C such that $B = C/\equiv'$. □

Exercise 1.3.23

Let $h\colon A \to B$ be a Σ-homomorphism. Show that there is a unique Σ-congruence $\equiv$ on A and a unique injective Σ-homomorphism $g\colon A/\equiv \to B$ such that $h_s(a) = g_s([a]_{\equiv_s})$ for all $s \in S$, $a \in |A|_s$. □

1.4 Term algebras

For any signature Σ there is a special Σ-algebra whose values are just well-formed terms (i.e. expressions) built from the operation names in Σ. A Σ-algebra of terms with variables is similarly determined by a signature $\Sigma = \langle S, \Omega \rangle$ and an S-sorted set of variables. These algebras are rather boring insofar as modelling programs is concerned — the term algebra models a program which does no real computation. But the homomorphisms from these algebras to *other* algebras turn out to be very useful technical tools, as shown by the definitions below.

Throughout this section, let $\Sigma = \langle S, \Omega \rangle$ be a signature and let X be an S-sorted set (of variables), where $x \in X_s$ for $s \in S$ means that the variable x is of sort s (written $x\colon s$). Note that "overloading" of variable names is permitted here, since there is no requirement that X_s and $X_{s'}$ be disjoint for $s \neq s' \in S$.

Definition 1.4.1 : Term algebra

The Σ-*algebra* $T_\Sigma(X)$ *of terms with variables* X is the Σ-algebra defined as follows:

- $|T_\Sigma(X)|$ is the least (with respect to $\subseteq$) S-sorted set of words (sequences) over the alphabet

$$S \cup \bigcup_{\substack{w \in S^* \\ s \in S}} \Omega_{w,s} \cup \bigcup_{s \in S} X_s \cup \{:, (, , ,)\}$$

  such that:

  - the word "$x\colon s$" $\in |T_\Sigma(X)|_s$ for all $s \in S$ and $x \in X_s$; and
  - for all $f\colon s_1 \times \cdots \times s_n \to s$ in Σ and all words $t_1 \in |T_\Sigma(X)|_{s_1}, \ldots, t_n \in |T_\Sigma(X)|_{s_n}$, the word "$f(t_1, \ldots, t_n)\colon s$" $\in |T_\Sigma(X)|_s$.

- for all $f\colon s_1 \times \cdots \times s_n \to s$ in Σ and all words $t_1 \in |T_\Sigma(X)|_{s_1}, \ldots, t_n \in |T_\Sigma(X)|_{s_n}$, $f_{T_\Sigma(X)}(t_1, \ldots, t_n) =$ (the word) "$f(t_1, \ldots, t_n)\colon s$" $\in |T_\Sigma(X)|_s$.

If $s \in S$ and $t \in |T_\Sigma(X)|_s$ then t is a Σ-*term of sort* s *with variables* X; the *free variables of* t is the set $FV(t) \subseteq X$ of variables that actually occur in t.

The Σ-*algebra of ground terms* is the Σ-algebra $T_\Sigma = T_\Sigma(\varnothing)$ of terms without variables. If $s \in S$ and $t \in |T_\Sigma|_s$ then t is a *ground* Σ-*term*. □

The values of $T_\Sigma(X)$ are "fully-typed" terms formed using the variables in X and the operation names in Σ, and the operations of $T_\Sigma(X)$ just build complicated terms from simpler terms. Note that a term $t \in |T_\Sigma(X)|$ need not contain all the variables in X, and that some variables may occur more than once in t. T_Σ is also called the Σ-*word algebra*, and its carriers $|T_\Sigma|$ are sometimes called the *Herbrand universe for* Σ.

Example 1.4.2

Let $\Sigma 1 = \langle S1, \Omega 1 \rangle$ be as defined in Example 1.2.4. Then $T_{\Sigma 1}$ is the $\Sigma 1$-algebra defined by

$$\begin{aligned} |T_{\Sigma 1}|_{shape} = \{ \ & \text{“}box(): shape\text{”}, \\ & \text{“}boxify(box(): shape): shape\text{”}, \\ & \text{“}boxify(boxify(box(): shape): shape): shape\text{”}, \\ & \ldots \}, \end{aligned}$$

$$\begin{aligned} |T_{\Sigma 1}|_{suit} = \{ \ & \text{“}hearts(): suit\text{”}, \\ & \text{“}f(box(): shape, hearts(): suit): suit\text{”}, \\ & \text{“}f(boxify(box(): shape): shape, hearts(): suit): suit\text{”}, \\ & \text{“}f(box(): shape, f(box(): shape, hearts(): suit): suit): suit\text{”}, \\ & \ldots \} \end{aligned}$$

where the operations of $T_{\Sigma 1}$ are the term formation operations

$$box_{T_{\Sigma 1}} = \text{“}box(): shape\text{”} \in |T_{\Sigma 1}|_{shape}, \quad hearts_{T_{\Sigma 1}} = \text{“}hearts(): suit\text{”} \in |T_{\Sigma 1}|_{suit},$$

$$\begin{aligned} boxify_{T_{\Sigma 1}}: & |T_{\Sigma 1}|_{shape} \to |T_{\Sigma 1}|_{shape} \\ & = \{ \ \text{“}box(): shape\text{”} \mapsto \text{“}boxify(box(): shape): shape\text{”}, \\ & \quad \text{“}boxify(box(): shape): shape\text{”} \mapsto \text{“}boxify(boxify(box(): shape): shape): shape\text{”}, \\ & \quad \ldots \}, \end{aligned}$$

and similarly for $f: shape \times suit \to suit$. □

NOTATION: Sort decorations (e.g. “$: shape$” in “$box(): shape$”) are often unambiguously determined, and they will usually be omitted when this is the case. When $\Omega_{\epsilon,s} \cap X_s = \varnothing$ for some $s \in S$, then variables of sort s cannot be confused with constants (0-ary operations) of sort s and so we will usually drop the parentheses “()” in the latter. We will omit quotation marks whenever it is clear from the context that we are dealing with terms. Finally, in examples we will use infix notation for binary operations when convenient.

Example 1.4.2 (revisited)

We repeat Example 1.4.2, making use of these notational conventions.

Let $\Sigma 1 = \langle S1, \Omega 1 \rangle$ be as defined in Example 1.2.4. Then $T_{\Sigma 1}$ is the $\Sigma 1$-algebra defined by

$$|T_{\Sigma 1}|_{shape} = \{box, boxify(box), boxify(boxify(box)), \ldots\},$$

$$|T_{\Sigma 1}|_{suit} = \{hearts, f(box, hearts), f(boxify(box), hearts), f(box, f(box, hearts)), \ldots\}$$

where the operations of $T_{\Sigma 1}$ are the term formation operations

$$box_{T_{\Sigma 1}} = box \in |T_{\Sigma 1}|_{shape}, \quad hearts_{T_{\Sigma 1}} = hearts \in |T_{\Sigma 1}|_{suit},$$

$boxify_{T_{\Sigma 1}}\colon |T_{\Sigma 1}|_{shape} \to |T_{\Sigma 1}|_{shape} = \{box \mapsto boxify(box), boxify(box) \mapsto boxify(boxify(box)), \ldots\}$,

and similarly for $f\colon shape \times suit \to suit$. □

Example 1.4.3

The notational conventions above will almost always be applicable. They cannot be adopted from the outset (i.e. in Definition 1.4.1) because of the relatively rare examples where confusion can arise. For example, let $\Sigma 2 = \langle S2, \Omega 2\rangle$ be the signature

$$\begin{array}{ll} \textbf{sorts} & s, s', t \\ \textbf{opns} & a\colon s \\ & a\colon s' \\ & f\colon s \to t \\ & f\colon s' \to t \end{array}$$

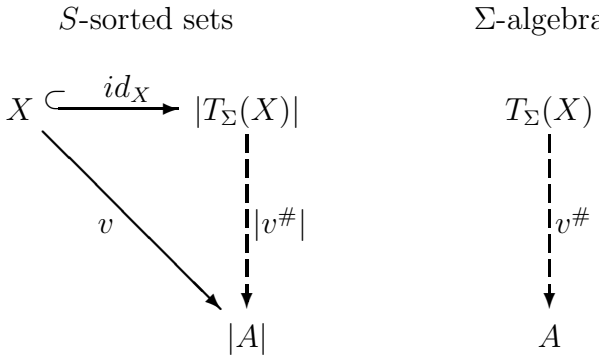
According to the definition, $|T_{\Sigma 2}|_t = \{\text{“}f(a()\colon s)\colon t\text{”}, \text{“}f(a()\colon s')\colon t\text{”}\}$. If all sort decorations were omitted then both of the terms in this set would become “$f(a())$” and so $|T_{\Sigma 2}|_t$ would have just this single element. The “outer” decoration can be omitted but the “inner” decoration is required, thus e.g. “$f(a()\colon s)$”.

Similarly, if X is an $S2$-sorted set of variables such that $a \in X_s$, then “$f(a()\colon s)$” and “$f(a\colon s)$” are different terms in $|T_{\Sigma 2}(X)|_t$, so the convention of writing “$a()\colon s$” as “$a\colon s$” cannot be used.

Since the definitions permit variables and operation names like $f(a()\colon s)$ and even “ or , or (), the custom of writing terms in the form of character strings without explicit separators can cause confusion. Luckily, such names never arise in practice and so this problem can safely be forgotten. □

Fact 1.4.4

For any Σ-algebra A and S-sorted function $v\colon X \to |A|$ there is exactly one Σ-homomorphism $v^{\#}\colon T_\Sigma(X) \to A$ that extends v, i.e. such that $v^{\#}_s(x) = v_s(x)$ for all $s \in S$, $x \in X_s$.



□

The existence and uniqueness of $v^\#$ follow easily from the requirement that $v^\#$ extends v (this fixes the value of $v^\#(x)$ for all the variables in $|T_\Sigma(X)|$) and that $v^\#$ is a Σ-homomorphism (this determines the value of $v^\#$ for a term $f(t_1,\ldots,t_n) \in |T_\Sigma(X)|$ as a function of the values of $v^\#$ for its immediate subterms $t_1,\ldots,t_n \in |T_\Sigma(X)|$). The homomorphism which results is the function which evaluates Σ-terms based on the assignment of values in A to variables in X given by v.

Definition 1.4.5 : Term evaluation

Let A be a Σ-algebra A and let $v\colon X \to |A|$ be an S-sorted function. By Fact 1.4.4 there is a unique Σ-homomorphism $v^\#\colon T_\Sigma(X) \to A$ that extends v. Let $s \in S$ and let $t \in |T_\Sigma(X)|_s$ be a Σ-term of sort s; the *value of t in A under the valuation v* is $v^\#(t) \in |A|_s$. When $t \in |T_\Sigma|_s$ the value of t does not depend on v; then the *value of t in A* is $\varnothing^\#(t)$ where $\varnothing\colon \varnothing \to |A|$ is the empty function. To make the algebra explicit, we write $t_A(v)$ for $v^\#(t)$, and t_A for $t_A(\varnothing)$ when t is ground. □

Exercise 1.4.6

Let $t \in |T_\Sigma(X)|$ be a Σ-term and let A be a Σ-algebra. Show that if $v\colon X \to |A|$ and $w\colon X \to |A|$ coincide on $FV(t)$, then $t_A(v) = t_A(w)$. □

Exercise 1.4.7

Define evaluation of terms in an inductive fashion. Convince yourself that the result is the same as that given by Definition 1.4.5. □

Exercise 1.4.8

Let $h\colon A \to B$ be a Σ-homomorphism, let $v\colon X \to |A|$ be an S-sorted function, and let $t \in |T_\Sigma(X)|$ be a Σ-term. Using Fact 1.4.4, prove that $h(v^\#(t)) = (v;h)^\#(t)$. Compare this with a proof of the same thing using your inductive definition of term evaluation from Exercise 1.4.7. □

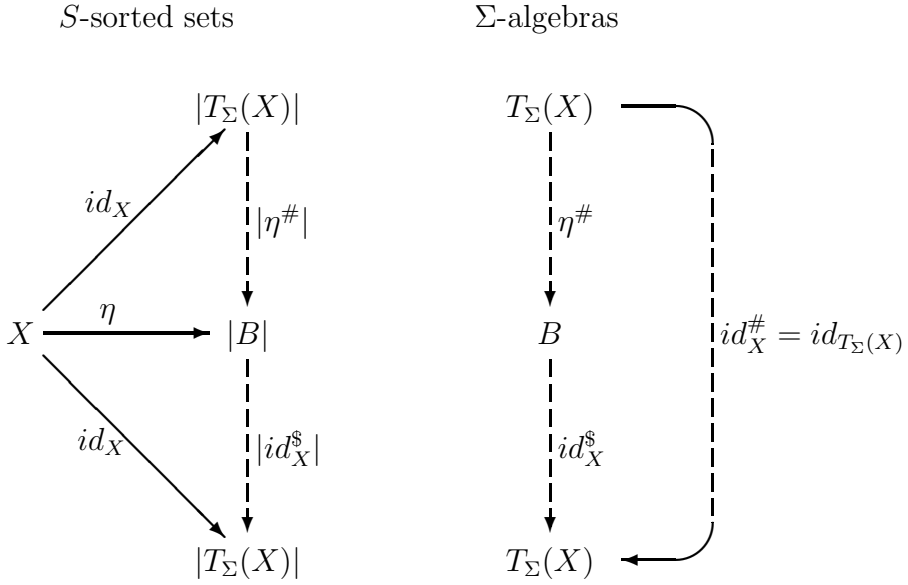
Exercise 1.4.9

Functions $\theta\colon X \to |T_\Sigma(Y)|$ are sometimes called *substitutions* (of terms in $T_\Sigma(Y)$ for variables in X). Using Fact 1.4.4, define the Σ-term $t[\theta]$ resulting from applying the substitution θ to a Σ-term $t \in |T_\Sigma(X)|$. Show that $t[id_X] = t$ for any $t \in |T_\Sigma(X)|$. Define the composition $\theta;\theta'$ of substitutions $\theta\colon X \to |T_\Sigma(Y)|$ and $\theta'\colon Y \to |T_\Sigma(Z)|$, and show that $(t[\theta])[\theta'] = t[\theta;\theta']$ for any Σ-term t and substitutions θ and θ'. □

NOTATION: Suppose $u \in |T_\Sigma(Y)|_s$ for some sort $s \in S$. Then $[x \mapsto u]$ (when used as a substitution $\{x{:}s\} \cup X \to |T_\Sigma(X \cup Y)|$) is shorthand for the function $\{x{:}s \mapsto u\} \cup \{z \mapsto z \mid z \in X, z \neq x{:}s\}$. For $t \in |T_\Sigma(\{x{:}s\} \cup X)|$, $t[x \mapsto u] \in |T_\Sigma(X \cup Y)|$ thus stands for the term obtained by substituting u for x in t. This notation generalizes straightforwardly to $[x_1 \mapsto u_1,\ldots,x_n \mapsto u_n]$ and $t[x_1 \mapsto u_1,\ldots,x_n \mapsto u_n]$ provided $x_1,\ldots,x_n$ are distinct variables.

Fact 1.4.10

The property of $T_\Sigma(X)$ in Fact 1.4.4 defines $T_\Sigma(X)$ up to isomorphism: **if** B is a Σ-algebra and $\eta\colon X \to |B|$ is an S-sorted function such that for any Σ-algebra A and S-sorted function $v\colon X \to |A|$ there is a unique Σ-homomorphism $v^\$\colon B \to A$ such that $\eta;|v^\$| = v$ **then** B is isomorphic to $T_\Sigma(X)$, where $\eta^\#\colon T_\Sigma(X) \to B$ is an isomorphism with inverse $id_X^\$\colon B \to T_\Sigma(X)$.



□

Fact 1.4.4 says that the definition of $T_\Sigma(X)$ fixes the definition of the term evaluation function "for free" (see Definition 1.4.5). Fact 1.4.10 says that this property is unique (up to isomorphism) to $T_\Sigma(X)$, so in fact the explicit definition of $T_\Sigma(X)$ is superfluous — it would be enough to define $T_\Sigma(X)$ as "the" (unique up to isomorphism) Σ-algebra for which Definition 1.4.5 makes sense. $T_\Sigma(X)$ is a particular example of a *free object* — see Section 3.5 for more on this topic.

Example 1.4.11

Let $\Sigma 1 = \langle S1, \Omega 1\rangle$ be as defined in Example 1.2.4. Then $T_{\Sigma 1}$ is the $\Sigma 1$-algebra described in Example 1.4.2. Let $B1$ be the $\Sigma 1$-algebra defined by

$$|B1|_{shape} = \{box, box\ boxify, box\ boxify\ boxify, \ldots\},$$

$$|B1|_{suit} = \{hearts, box\ hearts\ f, box\ boxify\ hearts\ f, box\ box\ hearts\ f\ f, \ldots\}$$

where the operations of $B1$ are the postfix term formation operations

$$box_{B1} = box \in |B1|_{shape}, \quad hearts_{B1} = hearts \in |B1|_{suit},$$

$$boxify_{B1}\colon |B1|_{shape} \to |B1|_{shape} = \{box \mapsto box\ boxify, box\ boxify \mapsto box\ boxify\ boxify, \ldots\},$$

and similarly for $f\colon shape \times suit \to suit$. Then $B1$ satisfies the property of $T_{\Sigma 1}$ in Fact 1.4.4 (the fact that $X = \varnothing$ here makes this easy to check — there is only one function $v\colon \varnothing \to |A1|$ for any $\Sigma 1$-algebra $A1$), so by Fact 1.4.10 (where $\eta\colon \varnothing \to |B1|$ is the empty function) $B1$ is isomorphic to $T_{\Sigma 1}$. The isomorphism $\varnothing^{\#}\colon T_{\Sigma 1} \to B1$ converts a $\Sigma 1$-term to its postfix form. □

Exercise 1.4.12

Prove Facts 1.4.4 and 1.4.10. □

Exercise 1.4.13

Let A be a Σ-algebra and let $\varnothing\colon \varnothing \to |A|$ be the empty function. Show that A is reachable iff the unique homomorphism $\varnothing^{\#} : T_\Sigma \to A$ is surjective, i.e., iff every element in $|A|$ is the value of a ground Σ-term. □

Exercise 1.4.14

Show that T_Σ is reachable. Put this fact together with previous results to show that a Σ-algebra is reachable iff it is isomorphic to a quotient of T_Σ, and that there is a one-to-one correspondence between isomorphism classes of reachable Σ-algebras and congruences on T_Σ. □

Exercise 1.4.15

Let G be a context-free grammar over an alphabet T of terminal symbols. Consider the signature $\Sigma_G = \langle S_G, \Omega_G \rangle$, where S_G is the set of non-terminal symbols of G and each production $X \to Y_1 \ldots Y_n$ in G corresponds to an operation in Ω_G with result sort X and arity given by the sequence of non-terminal symbols in $Y_1 \ldots Y_n$. The Σ_G-algebra A_G has carriers $|A_G|_X = T^*$ for all $X \in S_G$, and for any $p\colon X_1 \times \cdots \times X_n \to X$ in Σ_G and $a_1, \ldots, a_n \in T^*$, $p_{A_G}(a_1, \ldots, a_n)$ is the sequence obtained by substituting a_j for the j^{th} non-terminal symbol on the right-hand side of the production associated with p. Prove the following:

1. For any $X \in S_G$, the carrier of sort X in the reachable subalgebra of A_G is the set of sequences generated from the non-terminal X in G.

2. The algebra T_{Σ_G} is isomorphic to the algebra of parse trees of G.

3. The grammar G is unambiguous iff the reachable subalgebra of A_G is isomorphic to T_{Σ_G}.

□

1.5 Changing signatures

A signature morphism defines a mapping from the sort and operation names in one signature to those in another signature, in such a way that the arity and result sort of operations are respected. (This requirement is analogous to the requirement that homomorphisms respect the behaviour of the operations.) Signature morphisms will be used extensively in later chapters to mediate constructions involving multiple signatures. The crucial point that makes these constructions work is that a signature morphism from Σ to Σ' induces translations of syntax (terms — later, also logical formulae) and semantics (algebras and homomorphisms) between Σ and Σ'.

Two kinds of signature morphisms are introduced in this section. Only the first kind will be used in the rest of the book. The second kind, *derived signature morphisms*, are introduced mainly as an example of one way in which a basic definition could be modified. Such a modification would not affect later definitions and results, since these depend only on the induced translations of terms, algebras and homomorphisms.

1.5.1 Signature morphisms

Definition 1.5.1 : Signature morphism

Let $\Sigma = \langle S, \Omega \rangle$ and $\Sigma' = \langle S', \Omega' \rangle$ be signatures. A *signature morphism* $\sigma\colon \Sigma \to \Sigma'$ is a pair $\sigma = \langle \sigma_{sorts}, \sigma_{opns} \rangle$ where $\sigma_{sorts}\colon S \to S'$ and σ_{opns} is a family of functions respecting the arities and result sorts of operation names in Σ, that is $\sigma_{opns} = \langle \sigma_{w,s}\colon \Omega_{w,s} \to \Omega'_{\sigma^*_{sorts}(w),\sigma_{sorts}(s)} \rangle_{w \in S^*, s \in S}$ (where for $w = s_1 \ldots s_n \in S^*$, $\sigma^*_{sorts}(w) = \sigma_{sorts}(s_1) \ldots \sigma_{sorts}(s_n)$). A signature morphism $\sigma\colon \Sigma \to \Sigma'$ is a *signature inclusion* $\sigma\colon \Sigma \hookrightarrow \Sigma'$ if σ_{sorts} is an inclusion and $\sigma_{w,s}$ is an inclusion for all $w \in S^*, s \in S$. □

Signature morphisms as defined above will be referred to as *algebraic* signature morphisms when it is necessary to distinguish them from other kinds of signature morphisms to be introduced later. Note that σ_{sorts} and (the functions constituting) σ_{opns} are not required to be either surjective or injective.

NOTATION: When $\sigma\colon \Sigma \to \Sigma'$, both σ_{sorts} and σ_{opns} (and its components $\sigma_{w,s}$ for all $w \in S^*, s \in S$) will be denoted by σ.

Example 1.5.2

Let $\Sigma = \langle S, \Omega \rangle$ be the signature

sorts *polygon*, *figure*, *trump*

opns *square*: *polygon*

boxify: *polygon* → *polygon*

boxify: *polygon* → *figure*

h: *figure* × *trump* → *trump*

Let $\Sigma 1 = \langle S1, \Omega 1\rangle$ be the signature defined in Example 1.2.4.

Define $\sigma_{sorts}\colon S \to S1$ and $\sigma_{opns} = \langle \sigma_{w,s}\colon \Omega_{w,s} \to \Omega 1_{\sigma^*_{sorts}(w),\sigma_{sorts}(s)}\rangle_{w\in S^*, s\in S}$ by

$$\sigma_{sorts} = \{\mathit{polygon} \mapsto \mathit{shape}, \mathit{figure} \mapsto \mathit{shape}, \mathit{trump} \mapsto \mathit{suit}\},$$

$$\sigma_{\varepsilon,\mathit{polygon}} = \{\mathit{square} \mapsto \mathit{box}\}, \quad \sigma_{\mathit{polygon},\mathit{polygon}} = \{\mathit{boxify} \mapsto \mathit{boxify}\},$$

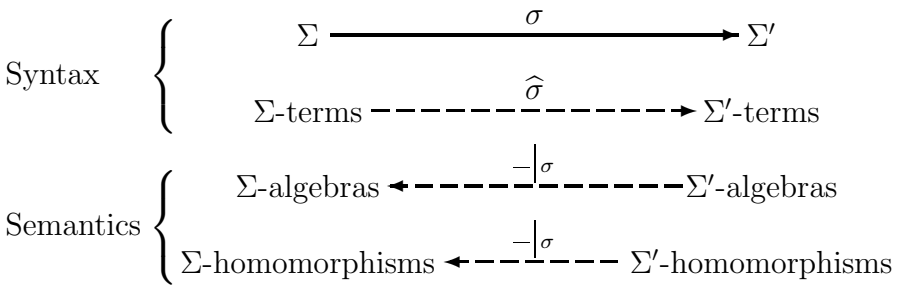
$$\sigma_{\mathit{polygon},\mathit{figure}} = \{\mathit{boxify} \mapsto \mathit{boxify}\}, \quad \sigma_{\mathit{figure}\,\mathit{trump},\mathit{trump}} = \{h \mapsto f\},$$

and $\sigma_{w,s} = \varnothing$ for all other $w \in S^*, s \in S$. Then $\sigma\colon \Sigma \to \Sigma 1$ is a signature morphism. □

Exercise 1.5.3

Let $\sigma\colon \Sigma \to \Sigma'$ and $\sigma'\colon \Sigma' \to \Sigma''$ be signature morphisms. Let $(\sigma;\sigma')_{sorts} = \sigma_{sorts};\sigma'_{sorts}$ and $(\sigma;\sigma')_{opns} = \sigma_{opns};\sigma'_{opns}$. Show that this defines a signature morphism $\sigma;\sigma'\colon \Sigma \to \Sigma''$. □

In the rest of this section, let $\sigma\colon \Sigma \to \Sigma'$ be a signature morphism, where $\Sigma = \langle S, \Omega\rangle$ and $\Sigma' = \langle S', \Omega'\rangle$. As will be explained below, any such signature morphism gives rise to a translation of Σ-terms to Σ'-terms, and of Σ'-algebras and homomorphisms to Σ-algebras and homomorphisms (note that the direction of translation of algebras and homomorphisms is "backwards" with respect to the direction of the signature morphism).



Definition 1.5.4 : Reduct algebra

Let A' be a Σ'-algebra. The *σ-reduct of A'* is the Σ-algebra $A'\big|_\sigma$ defined as follows:

- $|A'\big|_\sigma|_s = |A'|_{\sigma(s)}$ for all $s \in S$; and
- for all $f\colon s_1 \times \cdots \times s_n \to s$ in Σ,

$$f_{A'\big|_\sigma}\colon |A'\big|_\sigma|_{s_1} \times \cdots \times |A'\big|_\sigma|_{s_n} \to |A'\big|_\sigma|_s = \sigma(f)_{A'}\colon |A'|_{\sigma(s_1)} \times \cdots \times |A'|_{\sigma(s_n)} \to |A'|_{\sigma(s)}.$$

□

If Σ is a subsignature of Σ', $\sigma\colon \Sigma \hookrightarrow \Sigma'$ is the signature inclusion, and A' is a Σ'-algebra, then $A'\big|_\sigma$ is a Σ-algebra which is just A' with some carriers and/or operations removed.

NOTATION: We sometimes write $A'\big|_\Sigma$ for $A'\big|_\sigma$ when $\sigma\colon \Sigma \to \Sigma'$ is obvious, such as when σ is a signature inclusion.

Example 1.5.5

Let $\sigma\colon \Sigma \to \Sigma 1$ be the signature morphism defined in Example 1.5.2 and let $A1$ be the $\Sigma 1$-algebra defined in Example 1.2.4. Then $A1\big|_\sigma$ is the Σ-algebra such that

$$|A1\big|_\sigma|_{polygon} = |A1\big|_\sigma|_{figure} = \{\square, \triangle\} = |A1|_{shape},$$

$$|A1\big|_\sigma|_{trump} = \{\clubsuit, \heartsuit, \spadesuit\} = |A1|_{suit},$$

$$square_{A1\big|_\sigma} = \square = box_{A1},$$

$$boxify_{A1\big|_\sigma}\colon |A1\big|_\sigma|_{polygon} \to |A1\big|_\sigma|_{polygon} = \{\square \mapsto \square, \triangle \mapsto \square\} = boxify_{A1}\colon |A1|_{shape} \to |A1|_{shape},$$

$$boxify_{A1\big|_\sigma}\colon |A1\big|_\sigma|_{polygon} \to |A1\big|_\sigma|_{figure} = \{\square \mapsto \square, \triangle \mapsto \square\} = boxify_{A1}\colon |A1|_{shape} \to |A1|_{shape},$$

$$\begin{aligned} h_{A1\big|_\sigma}\colon |A1\big|_\sigma|_{figure} \times |A1\big|_\sigma|_{trump} \to |A1\big|_\sigma|_{trump} &= \{\langle \square, \clubsuit\rangle \mapsto \clubsuit, \langle \square, \heartsuit\rangle \mapsto \spadesuit, \ldots\} \\ &= f_{A1}\colon |A1|_{shape} \times |A1|_{suit} \to |A1|_{suit}. \end{aligned}$$

□

Exercise 1.5.6

A Σ-algebra A can be regarded as a function mapping the names in Σ to their interpretations; the σ-reduct of A is then the composition $\sigma;A$. Spell out the details. □

Exercise 1.5.7

Let $\sigma\colon \Sigma \to \Sigma'$ be a signature morphism that is surjective on sort names, and let A' be a Σ'-algebra. Show that if $A'\big|_\sigma$ is reachable then A' is reachable. Give counterexamples showing that the opposite implication does not hold, and that the implication itself does not hold if some sort names in Σ' are not in the image of Σ under σ. □

Recall that $\sigma\colon \Sigma \to \Sigma'$ is a signature morphism, where $\Sigma = \langle S, \Omega\rangle$ and $\Sigma' = \langle S', \Omega'\rangle$.

Definition 1.5.8 : Reduct homomorphism

Let $h'\colon A' \to B'$ be a Σ'-homomorphism. The *σ-reduct of h'* is the S-sorted function $h'\big|_\sigma\colon |A'\big|_\sigma| \to |B'\big|_\sigma|$ such that $(h'\big|_\sigma)_s = h'_{\sigma(s)}$ for all $s \in S$. (**Exercise:** Show that $h'\big|_\sigma\colon A'\big|_\sigma \to B'\big|_\sigma$ is a Σ-homomorphism.) □

Exercise 1.5.9

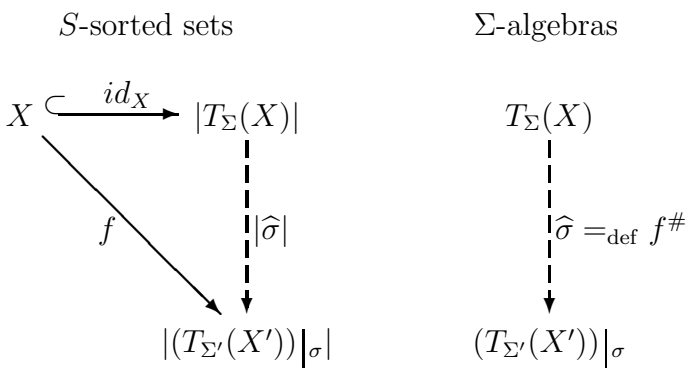
Define the σ-reduct $\equiv'\big|_\sigma$ of a Σ'-congruence $\equiv'$ on a Σ'-algebra A', and prove that it is a Σ-congruence on $A'\big|_\sigma$. Show that σ-reduct distributes over quotient, i.e. $(A'/\equiv')\big|_\sigma = (A'\big|_\sigma)/(\equiv'\big|_\sigma)$ for all Σ'-algebras A' and Σ'-congruences $\equiv'$ on A'. □

Definition 1.5.10 : Term translation

Let X be an S-sorted set of variables. Define $X' = \langle X'_{s'} \rangle_{s' \in S'}$ to be the S'-sorted set such that

$$X'_{s'} = \biguplus_{\sigma(s)=s'} X_s \qquad \text{for each } s' \in S'.$$

Then $(T_{\Sigma'}(X'))\big|_\sigma$ is a Σ-algebra. Let $f\colon X \to |(T_{\Sigma'}(X'))\big|_\sigma|$ be the obvious embedding (if not for the disjoint union in the definition of X', we would have $X \subseteq |(T_{\Sigma'}(X'))\big|_\sigma|$ and then f would be an inclusion). Then by Fact 1.4.4 there is a unique Σ-homomorphism $\widehat{\sigma}\colon T_\Sigma(X) \to (T_{\Sigma'}(X'))\big|_\sigma$ extending f:



The *translation of a Σ-term $t \in |T_\Sigma(X)|$ by σ* is the Σ'-term $\widehat{\sigma}(t) \in |T_{\Sigma'}(X')|$ (we usually write just $\sigma(t)$). □

This definition may look somewhat daunting, but its upshot is to define $\widehat{\sigma}(t)$ for $t \in |T_\Sigma(X)|$ as the Σ'-term obtained by replacing each operation name $f\colon s_1 \times \cdots \times s_n \to s$ in t by $\sigma(f)\colon \sigma(s_1) \times \cdots \times \sigma(s_n) \to \sigma(s)$. Some care must be taken in the treatment of variables since X_s and $X_{s'}$ are not required to be disjoint for $s \neq s' \in S$ — this is the reason why a disjoint union is used in the definition of X'.

Example 1.5.11

Let $\sigma\colon \Sigma \to \Sigma 1$ be the signature morphism defined in Example 1.5.2, where $\Sigma = \langle S, \Omega \rangle$ and $\Sigma 1 = \langle S1, \Omega 1 \rangle$. Let X be the S-sorted set of variables x: *polygon*, x: *figure*, y: *figure*, z: *trump*. The $S1$-sorted set of variables X' in Definition 1.5.10 is then x: *shape*, x': *shape*, y: *shape*, z: *suit*, and

$$\widehat{\sigma}(h(boxify(x\colon polygon), h(x\colon figure, z))) = f(boxify(x), f(x', z)),$$

$$\widehat{\sigma}(h(x\colon figure, h(boxify(boxify(square)), z))) = f(x', f(boxify(boxify(box)), z)),$$

and so on. □

Exercise 1.5.12

Let $t \in |T_\Sigma|$ be a ground Σ-term and let A' be a Σ'-algebra. Show that the value of t is invariant under change of signature, i.e. $\sigma(t)_{A'} = t_{A'|_\sigma}$.

Formulate and prove a more general version of this result in which t may contain variables. □

*1.5.2 Derived signature morphisms

A derived signature morphism from Σ to Σ' is like an algebraic signature morphism from Σ to Σ' except that operation names in Σ are mapped to *terms* over Σ'. This allows operation names in Σ to be mapped to combinations of operations in Σ', and also handles the case where the order of arguments of the corresponding operations in Σ and Σ' are different.

Definition 1.5.13 : Derived signature

Let $\Sigma = \langle S, \Omega \rangle$ be a signature. For any sequence $s_1 \dots s_n \in S^*$, let $I_{s_1 \dots s_n}$ be the S-sorted set $\boxed{1}\colon s_1, \dots, \boxed{n}\colon s_n$. The *derived signature of* Σ is the signature $\Sigma^{der} = \langle S, \Omega^{der} \rangle$ where for each $s_1 \dots s_n \in S^*$ and $s \in S$, $\Omega^{der}_{s_1 \dots s_n, s} = |T_\Sigma(I_{s_1 \dots s_n})|_s$. □

In the derived signature of Σ, a Σ-term t of sort s with variables $I_{s_1 \dots s_n}$ represents an operation $t\colon s_1 \times \dots \times s_n \to s$. The variable $\boxed{i}\colon s_i$ in $I_{s_1 \dots s_n}$ thus stands for the ith argument of t. Note that a "bare" variable $\boxed{i} \in |T_\Sigma(I_{s_1 \dots s_n})|_{s_i}$ is an operation $i\colon s_1 \times \dots \times s_n \to s_i$ in Σ^{der}, corresponding to a projection function.

Definition 1.5.14 : Derived signature morphism

Let Σ and Σ' be signatures. A *derived signature morphism* $\delta\colon \Sigma \to \Sigma'$ is an algebraic signature morphism $\delta\colon \Sigma \to (\Sigma')^{der}$. □

Definition 1.5.15 : Derived algebra

Let $\Sigma = \langle S, \Omega \rangle$ be a signature, and let A be a Σ-algebra. The *derived algebra of* A is the Σ^{der}-algebra A^{der} defined as follows:

- $|A^{der}| = |A|$; and
- for each $t\colon s_1 \times \dots \times s_n \to s$ in Σ^{der} and $a_1 \in |A^{der}|_{s_1}, \dots, a_n \in |A^{der}|_{s_n}$, $t_{A^{der}}(a_1, \dots, a_n) = t_A(v) \in |A^{der}|_s$ where v is the S-sorted function $\{(\boxed{1}\colon s_1) \mapsto a_1, \dots, (\boxed{n}\colon s_n) \mapsto a_n\}$.

□

In the rest of this section, let $\delta\colon \Sigma \to \Sigma'$ be a derived signature morphism. The following corresponds to Definition 1.5.4 for algebraic signature morphisms; later exercises correspond to Definitions 1.5.8 and 1.5.10 and related results.

Definition 1.5.16 : Reduct algebra w.r.t. a derived signature morphism

Let A' be a Σ'-algebra. The *δ-reduct of* A' is the Σ-algebra $A'\big|_\delta$ defined as follows:

- $|A'\big|_\delta|_s = |A'|_{\delta(s)}$ for all $s \in S$; and

- for all $f\colon s_1 \times \cdots \times s_n \to s$ in Σ, $f_{A'\restriction_\delta}\colon |A'\restriction_\delta|_{s_1} \times \cdots \times |A'\restriction_\delta|_{s_n} \to |A'\restriction_\delta|_s = \delta(f)_{(A')^{der}}$.

Equivalently, $A'\restriction_\delta$ is the Σ-algebra $(A')^{der}\restriction_\delta$, viewing δ as the algebraic signature morphism $\delta\colon \Sigma \to (\Sigma')^{der}$. □

Exercise 1.5.17 : Reduct homomorphism w.r.t. a derived signature morphism

What is the δ-reduct $h'\restriction_\delta$ of a Σ'-homomorphism $h'\colon A' \to B'$? Prove that $h'\restriction_\delta\colon A'\restriction_\delta \to B'\restriction_\delta$ is a Σ-homomorphism. □

Exercise 1.5.18 : Term translation w.r.t. a derived signature morphism

Let $t \in |T_\Sigma(X)|$ be a Σ-term, where X is an S-sorted set of variables. Define $\widehat{\delta}(t)$, the translation of t by δ (the result should be a Σ'-term). □

Example 1.5.19

Let $\Sigma = \langle S, \Omega\rangle$ be the signature defined in Example 1.5.2, and let $\Sigma 1 = \langle S1, \Omega 1\rangle$ be the signature defined in Example 1.2.4. Let $\delta\colon \Sigma \to \Sigma 1$ be the derived signature morphism defined by

$$\delta_{sorts} = \{polygon \mapsto shape, figure \mapsto shape, trump \mapsto suit\},$$
$$\delta_{\varepsilon,polygon} = \{square \mapsto boxify(box)\}, \quad \delta_{polygon,polygon} = \{boxify \mapsto \boxed{1}\colon shape\},$$
$$\delta_{polygon,figure} = \{boxify \mapsto boxify(boxify(\boxed{1}\colon shape))\},$$
$$\delta_{figure\ trump,trump} = \{h \mapsto f(boxify(\boxed{1}\colon shape), f(\boxed{1}\colon shape, \boxed{2}\colon suit))\},$$

and $\delta_{w,s} = \varnothing$ for all other $w \in S^*, s \in S$.

Let $A1$ be the $\Sigma 1$-algebra defined in Example 1.2.4. Then $A1\restriction_\delta$ is the Σ-algebra such that

$$|A1\restriction_\delta|_{polygon} = |A1\restriction_\delta|_{figure} = \{\square, \triangle\},$$
$$|A1\restriction_\delta|_{trump} = \{\clubsuit, \heartsuit, \spadesuit\}, \quad square_{A1\restriction_\delta} = \square,$$
$$boxify_{A1\restriction_\delta}\colon |A1\restriction_\delta|_{polygon} \to |A1\restriction_\delta|_{polygon} = \{\square \mapsto \square, \triangle \mapsto \triangle\}$$
$$boxify_{A1\restriction_\delta}\colon |A1\restriction_\delta|_{polygon} \to |A1\restriction_\delta|_{figure} = \{\square \mapsto \square, \triangle \mapsto \square\},$$

and $h_{A1\restriction_\delta}\colon |A1\restriction_\delta|_{figure} \times |A1\restriction_\delta|_{trump} \to |A1\restriction_\delta|_{trump}$ is defined by the following table:

$h_{A1\restriction_\delta}$	$\clubsuit$	$\heartsuit$	$\spadesuit$
$\square$	$\clubsuit$	$\heartsuit$	$\spadesuit$
$\triangle$	$\spadesuit$	$\heartsuit$	$\heartsuit$

Let X be the S-sorted set of variables $x\colon polygon, x\colon figure, y\colon figure, z\colon trump$. A correct solution to Exercise 1.5.18 would translate $h(boxify(x\colon polygon), h(x\colon figure, z))$ (a Σ-term with variables X) to

$$f(boxify(\underbrace{boxify(boxify(x))}_{=\widehat{\delta}(boxify(x:polygon))}),\ f(\underbrace{boxify(boxify(x))}_{=\widehat{\delta}(boxify(x:polygon))},\ \underbrace{f(boxify(x'), f(x', z))}_{=\widehat{\delta}(h(x:figure,z))})).$$

□

Exercise 1.5.20

Repeat Exercise 1.5.12 for the case of derived signature morphisms. □

Exercise 1.5.21

A more complex definition of derived signature morphism $\delta: \Sigma \to \Sigma'$ would allow a sort name s in Σ to be mapped to a *Cartesian product* $s'_1 \times \cdots \times s'_n$ of sorts $s'_1, \ldots, s'_n$ in Σ'. Give versions of the above definitions which permit this. □

Exercise 1.5.22

Another variation on the definition of derived signature morphism would permit operation names in Σ to be mapped to recursively defined functions in terms of the operation names in Σ'. Give versions of the above definitions which would allow this. (HINT: Look at a book like [Schmidt-denotational-semantics] before attempting this exercise.) □

1.6 Further reading

This chapter presents the basic notions of universal algebra that are required in the sequel. There is a vast literature on universal algebra as a branch of mathematics, and the concepts and results we need here are a tiny fraction of this.

For much more on universal algebra see e.g. [Grätzer] or [Cohn] (both of these handle only the single-sorted case). A presentation of some of this material for a computer scientist audience is [Wechler], see also [MT92] where applications to topics in computer science other than the ones covered in this book are indicated (???check).

The style of presentation here is relaxed but it might be too dense for some readers, who might prefer the gentler style of presentation, with proofs of many of the results which we omit here, in [adj1978], [Ehrig/Mahr-V1], [Meseguer/Goguen-initiality], or [Ehrich/Gogolla/Lipeck] (possibly [Ehrich/Loeckx/Wolf] instead).

The generalisation from single-sorted to many-sorted algebras originates with [Birkhoff/Lipson] (???check). The generalisation is straightforward from a purely mathematical standpoint, but there are a few subtle issues that will surface in later chapters. For instance, we admit empty carrier sets in Definition **??**, unlike most logic books and, for instance, [Bergstra/Tucker] and [CASL]. Admitting empty carrier sets requires more care in the presentation of rules for reasoning, see Exercise 2.4.9, but also makes some results smoother, see Exercise 2.5.18.

There are different definitions of many-sorted signature in the literature. The one here is quite general, allowing overloading of operation names etc., and originates with [ADJ73]. A definition that does not permit overloading is used in [Wir90].

Bergstra module algebra paper: signature morphisms (renamings) are forced to be injective.

Derived signature morphisms originate in ???. They have made only sporadic appearances in the literature since then. Related to (syntactic part of) theory interpretation in logic.