1

Universal algebra

The function of mathematical logic is to provide formal languages for describing the structures with which mathematicians work, and the methods of proof available to them. Obviously, the more complicated we make our language, the more powerful it will be as an instrument for expressing our ideas; but in these notes we are going to begin with what is perhaps the simplest *useful* language, that of universal algebra (sometimes called equational logic). Although simple, it displays many of the characteristic features of more complicated languages, which is why it makes a good introduction to them.

Universal algebra begins by abstracting the common features of a number of familiar mathematical structures, including groups, rings and vector spaces. In each of these cases, the structure is defined to be a set equipped with certain finitary *operations*, which satisfy certain *equations*. For example, a group is a set G equipped with

$$\begin{array}{ll} \text{a binary operation } m: G \times G \to G & \text{(multiplication)}, \\ \text{a unary operation } i: G \to G & \text{(inversion)}, \\ \text{and a nullary operation } e: G^0 \to G & \text{(identity)} \end{array}$$

[note: we adopt the convention that G^0 is a singleton set for any G, so that a 0-ary operation – also called a constant – simply picks out a single element of G], satisfying the equations

$$\begin{array}{ll} m(x, m(y, z)) = m(m(x, y), z) & \text{(associative law)}, \\ m(e, x) = x & \text{(left identity law)}, \\ \text{and } m(i(x), x) = e & \text{(left inverse law)}, \end{array}$$

which are supposed to hold for all possible values of x, y, z in G.

We leave it as an exercise for the reader to write down similar

descriptions of the notion of ring (with 1), and of the notion of K-vector space for a given field K. Note that for the latter it is necessary (or at least convenient) to take (scalar multiplication by) each element of K as a unary operation; thus in general the set of operations and/or equations required to define a given type of structure may be infinite.

Abstracting from the above examples, we introduce the notion of an operational type. An *operational type* is a pair (Ω, α) where Ω is a set of *operation-symbols* and α is a function assigning to each $\omega \in \Omega$ a natural number $\alpha(\omega)$, called its *arity*. [N.B.: throughout these notes, 0 is considered to be a natural number.] Frequently, we suppress any explicit mention of the function α, and simply write 'Ω is an operational type'. Thus in our example above we have $\Omega = \{m, i, e\}$ with $\alpha(m) = 2$, $\alpha(i) = 1$, $\alpha(e) = 0$.

Given an operational type (Ω, α), a *structure* of type (Ω, α) (or Ω-structure, or Ω-algebra) is a set A equipped with functions $\omega_A : A^{\alpha(\omega)} \to A$ for each $\omega \in \Omega$. We call ω_A the *interpretation* of the abstract symbol ω in the structure A; we also speak of the family of functions $(\omega_A \mid \omega \in \Omega)$ as an Ω-structure on the set A. A *homomorphism* $f : A \to B$ of Ω-structures is a function such that

$$f(\omega_A(a_1, \ldots, a_{\alpha(\omega)})) = \omega_B(f(a_1), \ldots, f(a_{\alpha(\omega)}))$$

for all $\omega \in \Omega$ and all $a_1, a_2, \ldots, a_{\alpha(\omega)}$ in A.

So much for the operations; how about the equations? Before answering this question, we turn to a seemingly different topic: the notion of a *term* or *derived operation*. Let Ω be an operational type and X a set (whose elements we shall call *variables*; we assume for convenience $\Omega \cap X = \varnothing$); then the set $F_\Omega(X)$ (or simply FX) of Ω-*terms* in X is defined inductively as follows:

(a) If $x \in X$, then $x \in F_\Omega(X)$.
(b) If $\omega \in \Omega$, $\alpha(\omega) = n$ and $t_1, t_2, \ldots, t_n \in F_\Omega(X)$, then $\omega t_1 t_2 \ldots t_n \in F_\Omega(X)$.
(c) That's all.

[Inductive definitions of this type are very common in the construction of formal languages. Formally, what this one means is that $F_\Omega(X)$ is the smallest subset of the set $(M$, say) of all finite strings of elements of $\Omega \cup X$ which satisfies the closure properties (a) and (b), i.e. the intersection of all such subsets.]

Remark 1.1. Note in passing that we have a simple algorithm for determining whether a given finite string in M belongs to $F_\Omega(X)$: start at the right-hand end of the string with counter set to 0, and move leftwards increasing the count by 1 each time you pass a variable, and decreasing it by $n-1$ each time you pass an n-ary operation-symbol. Then the string belongs to FX iff the counter never falls below 1 after your start, and finishes at 1. As illustrations, we show the counter values for a number of strings where Ω is the operational type of groups and $X = \{x, y, z\}$; the first one is in FX, the other two are not.

1 2 1 2 3 3 2 1 1 0 1 2 2 2 1 0 1 0 3 3 2 3 2 2 1 0
m e m m i x y i z ; *m i i x y m z ;* *i x m e i y x .*

Theorem 1.2. (i) $F_\Omega(X)$ has an Ω-structure.

(ii) $F_\Omega(X)$ is the *free* Ω-structure generated by X; i.e., given any Ω-structure A and any function $f: X \to A$, there exists a unique homomorphism $\bar{f}: F_\Omega(X) \to A$ extending f.

Proof. The existence of the Ω-structure is immediate from clause (b) of the definition: if $\omega \in \Omega$ (with $\alpha(\omega) = n$, say) and $t_1, t_2, \ldots, t_n \in FX$, define

$$\omega_{FX}(t_1, t_2, \ldots, t_n) = \omega t_1 t_2 \ldots t_n$$

(i.e. just erase the brackets and commas).

Part (ii) is essentially a matter of putting the brackets back in. Since FX was defined inductively, we can define $\bar{f}$ inductively:

if $t = x \in X$, then $\bar{f}(t) = f(x)$;
if $t = \omega t_1 \ldots t_n$ where $\omega \in \Omega$, $\alpha(\omega) = n$ and $\bar{f}$ has already been defined at $t_1, \ldots, t_n \in FX$, then $\bar{f}(t) = \omega_A(\bar{f}(t_1), \ldots, \bar{f}(t_n))$.

It is then clear that $\bar{f}$ is a homomorphism, and that it is the unique homomorphism extending f. □

Another important (and trivial) property of free Ω-structures is

Lemma 1.3. For any X,

$$F_\Omega(X) = \bigcup \{F_\Omega(X') \mid X' \subseteq X, \quad X' \text{ finite}\}. \quad \square$$

Thus we may, for many purposes, restrict our attention to free structures generated by finite sets. Let $X_n = \{x_1, x_2, \ldots, x_n\}$ be a standard n-element set; let $t \in FX_n$, and let A be any Ω-structure.

Then we may define a function $t_A: A^n \to A$ inductively as follows:

if $t = x_i$ $(1 \leqslant i \leqslant n)$, then t_A is projection onto the ith factor;
if $t = \omega t_1 t_2 \ldots t_m$ where $\alpha(\omega) = m$, then t_A is the composite

$$A^n \xrightarrow{((t_1)_A, \ldots, (t_m)_A)} A^m \xrightarrow{\omega_A} A.$$

In particular, if t is the term $\omega x_1 x_2 \ldots x_n$, where $\alpha(\omega) = n$, then $t_A = \omega_A$. The function t_A is called (the interpretation in A of) the n-ary *derived operation* corresponding to the term t (in contrast to the 'primitive operations' which are the functions of the form ω_A). It is easy to see that a homomorphism $f: A \to B$ of Ω-structures commutes with all derived operations as well as with primitive ones.

Now let us return to the equations. If we look, for example, at the associative law for groups, we see that each side of the equation is a ternary derived operation (let us call the corresponding terms s and t); and the assertion that the associative law holds in a group G is just the assertion that the functions s_G and t_G are equal. We thus define an n-ary *equation* (in an operational type Ω) to be an expression $(s = t)$, where s and t are elements of $F_\Omega(X_n)$, and we say an equation $(s = t)$ is *satisfied* in a structure A if $s_A = t_A$. Finally, we define an *algebraic theory* to be a pair $T = (\Omega, E)$ where Ω is an operational type and E is a set of equations in Ω, and we define a *model* for T (or T-algebra) to be an Ω-structure which satisfies all the equations in E.

Thus, for example, a group is exactly an (Ω, E)-model, where $\Omega = \{m, i, e\}$ as before and

$$E = \{(mx_1 mx_2 x_3 = mmx_1 x_2 x_3), (mex_1 = x_1), (mix_1 x_1 = e)\}.$$

[Note that, as in the third member of E above, it is not necessary for each of the variables $x_1, \ldots, x_n$ to appear explicitly on each side of an n-ary equation.]

Just as we did with operations, we may now enlarge the set E of 'primitive' equations to a larger set $\tilde{E}$ of derived equations. [For example, one proves in a first course on group theory that any Ω-structure satisfying the three equations in the particular E above also satisfies the 'right identity' and 'right inverse' equations $(mx_1 e = x_1)$, $(mx_1 ix_1 = e)$.] Once again, we give an inductive definition of $\tilde{E}$:

(a) $E \subseteq \tilde{E}$.
(b) $\tilde{E}$ is an equivalence relation on the set of terms: thus
  (i) for any term t, $(t = t) \in \tilde{E}$;

(ii) if $(s = t) \in \tilde{E}$, then $(t = s) \in \tilde{E}$;
(iii) if $(s = t)$ and $(t = u)$ are in $\tilde{E}$, then $(s = u) \in \tilde{E}$.

(c) $\tilde{E}$ is closed under substitution, in two different ways:
(i) if $(s = t) \in \tilde{E}$, x_i is a variable involved in s and/or t and u is any term, then $(s[u/x_i] = t[u/x_i]) \in \tilde{E}$, where $s[u/x_i]$ denotes the effect of replacing each occurrence of x_i in s by the term u;
(ii) if s is a term, x_i a variable involved in s and $(t = u)$ is in $\tilde{E}$, then $(s[t/x_i] = s[u/x_i]) \in \tilde{E}$.

(d) That's all.

[As before, this definition really means that $\tilde{E}$ is the smallest subset of the set of all expressions $(s = t)$ which is closed under (a), (b) and (c).]

If s and t are elements of $F_\Omega(X)$ for some X, let us write $s \sim_E t$ to mean $(s = t) \in \tilde{E}$; then by (b) above $\sim_E$ is an equivalence relation, and we can form the set $F_{(\Omega,E)}(X)$ of $\sim_E$-equivalence classes.

Theorem 1.4. (i) $F_{(\Omega,E)}(X)$ inherits an Ω-structure from $F_\Omega(X)$, and it satisfies the equations in E.
(ii) $F_{(\Omega,E)}(X)$ is the free (Ω, E)-model generated by X.

Proof. (i) Clause (c)(ii) of the definition of $\tilde{E}$ says that the interpretations in $F_\Omega(X)$ of the operations of Ω respect the equivalence relation $\sim_E$, and hence induce operations on the quotient set $F_{(\Omega,E)}(X)$. The fact that these induced operations satisfy the equations in E follows from ((a) and) (c)(i), since every element of $F_{(\Omega,E)}(X)$ is the equivalence class of some term.

(ii) Let $\hat{E}$ denote the set of expressions $(s = t)$ where s and t are elements of $F_\Omega(X)$ such that $h(s) = h(t)$ for every Ω-homomorphism h from $F_\Omega(X)$ to an (Ω, E)-model A. Then it is easily verified that $\hat{E}$ satisfies the closure properties (a), (b) and (c) [for (c), this requires the observation that $h(s[u/x_i]) = h'(s)$, where h' is the unique homomorphism sending x_i to $h(u)$ and the other elements of X to their images under h]; so $\tilde{E} \subseteq \hat{E}$, and hence every homomorphism $h: F(X) \to A$ factors through the quotient map $F_\Omega(X) \to F_{(\Omega,E)}(X)$. In particular, if $h = \bar{f}$ is the unique homomorphism extending a given map $f: X \to A$ (as in Theorem 1.2(ii)), we obtain a homomorphism $\tilde{f}: F_{(\Omega,E)}(X) \to A$, which is clearly the unique homomorphism extending f. □

Corollary 1.5. Let (Ω, E) be an algebraic theory. Then an equation $(s = t)$ belongs to $\tilde{E}$ iff it is satisfied in every (Ω, E)-model.

Proof. One direction is easy: the set of equations satisfied in a given (Ω, E)-model (and hence, the set of equations satisfied in every (Ω, E)-model) has the closure properties (a), (b) and (c), and so contains $\tilde{E}$. Conversely, if $(s = t)$ is satisfied in every (Ω, E)-model, then it is satisfied in $F_{(\Omega,E)}(X_n)$ for any n; in particular (assuming for notational convenience that both s and t involve exactly the variables $x_1, x_2, \ldots, x_n$), we have

$$s_{F_{(\Omega,E)}(X_n)}([x_1], \ldots, [x_n]) = t_{F_{(\Omega,E)}(X_n)}([x_1], \ldots, [x_n]) \qquad (*)$$

(where the square brackets denote $\sim_E$-equivalence classes). But by definition we have

$$s_{F_{(\Omega,E)}(X_n)}([x_1], \ldots, [x_n]) = [s_{F_\Omega(X_n)}(x_1, \ldots, x_n)] = [s],$$

and similarly the right-hand side of $(*)$ equals $[t]$; so $[s] = [t]$, i.e. $(s = t) \in \tilde{E}$. □

Corollary 1.5 is our first example of a *completeness theorem*, i.e. a theorem asserting (for some class of theories) that the things which are *true* (i.e. are satisfied in every model of a given theory) coincide with the things which are *provable* (i.e. are derivable from the postulates of the theory – in this case, the primitive equations – by a specified deduction process – in this case, the closure properties (b) and (c)). Clearly, the acid test of any formal deduction-system is whether we can prove a completeness theorem for it. The existence of free models, as we have seen, makes the completeness theorem for algebraic theories comparatively easy to prove; in the next two chapters we shall prove completeness theorems in other contexts where we have to do a good deal more work to show that every true statement is provable.

However, even for algebraic theories not everything in the garden is rosy. In contrast to the situation for terms, there is in general no algorithm for determining whether or not a given equation $(s = t)$ is derivable from a given theory. For some particular theories (e.g. that of groups – see Exercise 1.6) we can find such an algorithm; but in Chapter 4 we shall give an explicit example of an algebraic theory for which we can prove that no such algorithm exists. The problem of finding such an algorithm, for a given T, is called the *word problem* for T. ['Word' is an old-fashioned synonym for 'term'.]

There is one case where the word problem always has a trivial

solution. Let Ω be an operational type, and let $\mathscr{A} = \{A_1, A_2, \ldots, A_n\}$ be a finite set of finite Ω-structures. Then if we define E to be the set of all equations which are satisfied in every A_i, it is clear that we already have $E = \tilde{E}$; and so to determine whether $(s = t)$ is a (derived) equation of this theory it suffices to compute s_{A_i} and t_{A_i} for each i – which is a finite process since each A_i is finite.

An important example of a theory of this kind is the theory of *Boolean algebras*, which may be loosely described as 'everything you can say about a two-element set' (that is, if you confine yourself to the language of universal algebra). There are various ways of presenting this theory: a highly generous one uses two nullary operations $\top$ (true) and $\bot$ (false), a unary operation $\neg$ (not), and four binary operations $\wedge$ (and), $\vee$ (or), $\Rightarrow$ (implies) and $\Leftrightarrow$ (iff). The set $2 = \{0, 1\}$ is given a structure for this operational type by setting

$$\begin{aligned} \top_2 &= 1 \\ \bot_2 &= 0 \\ \neg_2(a) &= 1 - a \\ \wedge_2(a, b) &= \min\{a, b\} \\ \vee_2(a, b) &= \max\{a, b\} \\ \Rightarrow_2(a, b) &= 0 \text{ iff } a = 1 \text{ and } b = 0 \\ \Leftrightarrow_2(a, b) &= 1 \text{ iff } a = b. \end{aligned}$$

We then define a *Boolean algebra* to be an (Ω, E)-model, where Ω is as above and E is the set of all equations satisfied in 2. [Note: henceforth we shall generally revert to the more familiar 'algebraic' way of writing binary operations: $(x \wedge y)$ instead of $\wedge xy$, etc.] Of course, the above presentation is highly inefficient, because E contains a good many equations which tell us that some of the seven primitive operations are definable in terms of the others. For example, it is easy to verify that E contains

$$\begin{aligned} (\top &= (\bot \Rightarrow \bot)) \\ (\neg x &= (x \Rightarrow \bot)) \\ ((x \vee y) &= (\neg x \Rightarrow y)) \\ ((x \wedge y) &= \neg(\neg x \vee \neg y)) \end{aligned}$$

and $$((x \Leftrightarrow y) = ((x \Rightarrow y) \wedge (y \Rightarrow x))),$$

so that every Ω-term is $\sim_E$-equivalent to one involving only the

primitive operations $\bot$ and $\Rightarrow$. Henceforth, we shall regard $\bot$ and $\Rightarrow$ as the only primitive operations in the theory of Boolean algebras, and regard the above equations as *defining* $\top$, $\neg$, $\vee$, $\wedge$ and $\Leftrightarrow$ as (shorthand for) certain derived operations. There are many other ways of reducing the number of primitive operations; this one has the (small) merit that it gets the number down to the least possible (see Exercise 1.10).

This reduction has not exhausted all the equations in E; there are still others that we need to consider. We note, however, that $(s = t)$ belongs to E iff $((s \Leftrightarrow t) = \top)$ does; therefore we can restrict our attention to equations of the form $(t = \top)$. We say a term t is a *tautology* if $(t = \top)$ is in E (equivalently, if t_2 is the constant function $2^n \to 2$ with value 1, where n is the number of variables in t). It is easy to verify that the following are tautologies:

(a) $(x \Rightarrow (y \Rightarrow x))$,
(b) $((x \Rightarrow (y \Rightarrow z)) \Rightarrow ((x \Rightarrow y) \Rightarrow (x \Rightarrow z)))$,
(c) $(((x \Rightarrow \bot) \Rightarrow \bot) \Rightarrow x)$.

(c) looks more familiar if we write it as $(\neg\neg x \Rightarrow x)$; but we wanted to emphasize that $\Rightarrow$ and $\bot$ are now our only primitive operations. We shall be meeting these three tautologies quite frequently in future.

Exercises

1.1. Let $\Omega = \{t, b, u, c\}$ with $\alpha(t) = 3$, $\alpha(b) = 2$, $\alpha(u) = 1$, $\alpha(c) = 0$, and let x, y, z be variables. Which of the following are Ω-terms?
(i) *ttxbucyzzz* (ii) *xubytcz* (iii) *tcucbucc*
(iv) *bbbxbybyybxbzbyyy* (v) *bxytczuz* (vi) *tbxxxxx*.

1.2. Show that the following definition of the derived operation induced by a term is equivalent to the one given in the text:
'If $t \in F_\Omega(X_n)$ and $a_1, \ldots, a_n$ are elements of an Ω-structure A, then $t_A(a_1, \ldots, a_n) = \bar{f}(t)$, where $\bar{f}: F_\Omega(X_n) \to A$ is the unique homomorphism extending the map $f: X_n \to A$ with $f(x_i) = a_i$ $(1 \leqslant i \leqslant n)$.'

1.3. Let s, t and u be Ω-terms (for some fixed Ω), and let x_i and x_j be distinct variables. We write $s[t, u/x_i, x_j]$ for the term obtained from s on simultaneously replacing each occurrence of x_i by t and each occurrence of x_j by u. Show that $s[t, u/x_i, x_j]$ is not in general the same as $s[t/x_i][u/x_j]$, but that it is the same as $s[t[x_n/x_j]/x_i][u/x_j][x_j/x_n]$,

provided n is chosen so large that x_n does not occur anywhere in s, t or u. Hence show that if $(s = s')$, $(t = t')$ and $(u = u')$ are all derived equations of some theory (Ω, E), so is $(s[t, u/x_i, x_j] = s'[t', u'/x_i, x_j])$.

1.4. Let T be an algebraic theory. Show that the one-element set $\{0\}$ has a unique T-model structure, and that the empty set has a T-model structure iff T contains no nullary operations.

1.5. Let $\Omega = \{m, i, \bar{e}\}$ with $\alpha(m) = 2$, $\alpha(i) = \alpha(\bar{e}) = 1$, and let E consist of the four equations $(mxmyz = mmxyz)$, $(\bar{e}x = \bar{e}y)$, $(m\bar{e}xx = x)$ and $(mixx = \bar{e}x)$. Show that every group is an (Ω, E)-model in a natural way. Is the converse true?

1.6. Let Ω be the operational type of groups. We say that an Ω-term is *reduced* if it is *either* the single symbol e *or* of the form $mm \ldots mw$, where w is a string of symbols involving only variables and the operation i, and not including any substring of the form ii, ixx or xix (*except* as part of a substring $ixix$).
(i) Describe an algorithm which, given an arbitrary Ω-term t, produces a reduced term $\bar{t}$ for which $(t = \bar{t})$ is a derived equation of the theory of groups.
(ii) Show that the set of all reduced terms in a given set X of variables can be made into a group RX containing X as a subset. By considering the induced homomorphism $FX \to RX$, where FX is the free group generated by X (defined as in Theorem 1.4), show that if s and t are reduced terms for which $(s = t)$ is a derived equation, then s and t are identical.
(iii) Use (i) and (ii) to solve the word problem for groups.
[Feel free to use everything you know about group theory in answering this question.]

1.7. (i) Let T be an algebraic theory, and suppose T contains a ternary (possibly derived) operation p for which

$$(pxyy = x) \quad \text{and} \quad (pxxy = y) \qquad (*)$$

are (possibly derived) equations of T. Let A be a T-model, and let R be a sub-T-model of $A \times A$ which contains $\{(a, a) \mid a \in A\}$ (i.e., considered as a binary relation on A, R is reflexive). Show that R is also symmetric and transitive.
(ii) Conversely, if T is an algebraic theory such that every reflexive submodel of the square of a T-model is also symmetric, show that T contains a ternary operation satisfying $(*)$. [Hint: let F be the free T-model generated by $\{x, y\}$, and consider the sub-T-model of $F \times F$ generated by $\{(x, x), (x, y), (y, y)\}$.]

(iii) Give an example of an operation p satisfying (*) when T is the theory of groups, but show that there is no such operation in the theory of semigroups (i.e. the theory obtained from that of groups by deleting the operation i and the equation in which i occurs).

1.8. (i) Let $\Omega = \{e, m\}$ with $\alpha(e) = 0$, $\alpha(m) = 2$, and let E consist of the two equations $(mex = x)$ and $(mxe = x)$. Suppose a set A has two (Ω, E)-model structures (e_1, m_1) and (e_2, m_2) such that the operations of the second structure are Ω-homomorphisms $1 \to A$ and $A \times A \to A$ for the first structure. Show that A satisfies the equations $(e_1 = e_2)$ and $(m_1 m_2 xzm_2 yt = m_2 m_1 xym_1 zt)$, and deduce that $m_1 = m_2$ and that m_1 is commutative and associative.
(ii) Ask an algebraic topologist to explain what this has to do with the result that the fundamental group of a topological group is abelian.

1.9. Let $2 = \{0, 1\}$ with its usual Boolean algebra structure, and let n be a natural number. Show that every function $2^n \to 2$ is (the interpretation of) an n-ary derived operation of the theory of Boolean algebras. [Hint: use induction on n.] Deduce that the free Boolean algebra on n generators has 2^{2^n} elements.

1.10. Let B be the theory of Boolean algebras, and let $\downarrow$ be the (derived) binary operation of B defined by

$$(x \downarrow y) = \neg(x \wedge y).$$

Show that the subtheory B_0 of B generated by $\downarrow$ (i.e. the set of all operations derivable from $\downarrow$) contains all of B except the two constants. Show also that no single operation can generate the whole of B; and that B cannot be generated by either $\wedge$ or $\vee$ plus one other operation.