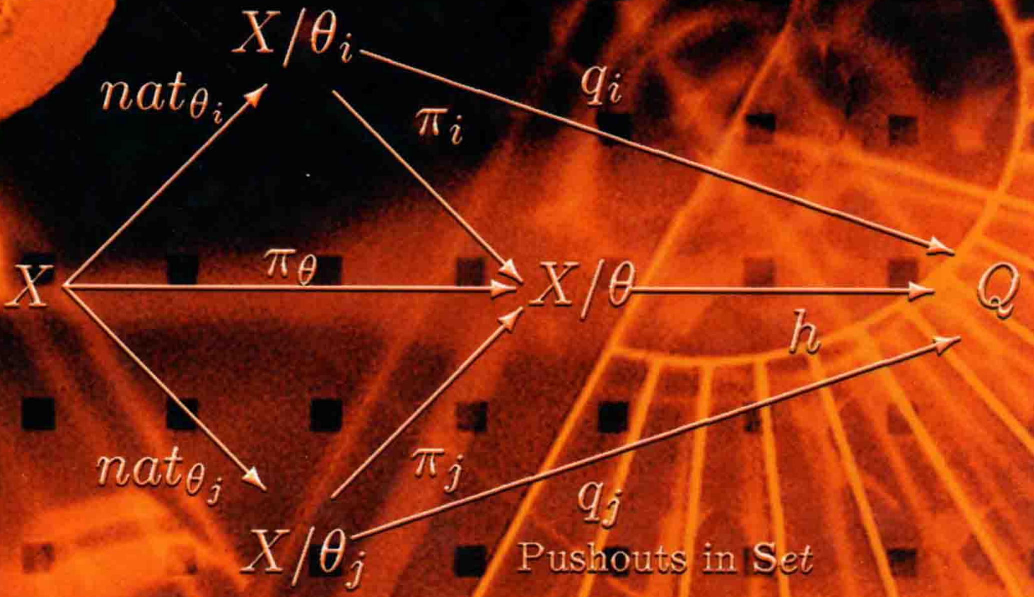


Universal Algebra and Coalgebra

Klaus Denecke
Shelly L Wismath

World Scientific

Universal Algebra and Coalgebra

Klaus Denecke
Universität Potsdam, Germany

Shelly L Wismath
University of Lethbridge, Canada

World Scientific

NEW JERSEY · LONDON · SINGAPORE · BEIJING · SHANGHAI · HONG KONG · TAIPEI · CHENNAI

Published by

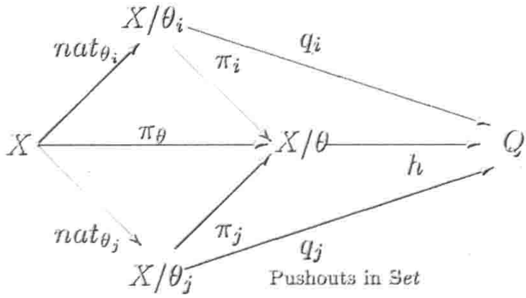
World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

UNIVERSAL ALGEBRA AND COALGEBRA

ISBN-13 978-981-283-745-5
ISBN-10 981-283-745-0

Printed in Singapore by World Scientific Printers



Pushouts in Set

Universal Algebra and Coalgebra

Les chiens aboient, les chats miaulent,
c'est leur nature;
nous nous sommes mathématicienne,
c'est la notre.

Preface

A mathematical theory is considered to be elegant if it allows us to model various notions with only a few basic concepts. From this point of view, category theory is elegant. The concepts of a category, a functor, a natural transformation and limits provide a strong expressive power. Universal coalgebra is another powerful and elegant theory, based on category theory. Categorical coalgebras are the dual objects of algebras; they are used to model several kinds of automata and more generally to model transition and dynamical systems. The basic notions of algebras, homomorphisms and congruences from universal algebra correspond to coalgebras, homomorphisms of coalgebras and bisimulations, respectively in the theory of coalgebras.

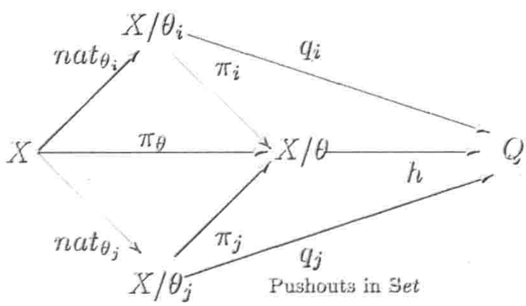
Universal algebra and coalgebra theory are studied here by a category theory approach, using F-coalgebras and the dual concept of F-algebras for a set-valued functor F. This approach allows similar ideas and proofs for the basic results for both F-algebras and F-coalgebras, but it is not as general as category theory itself. But also there are some structures which do not admit any representation as coalgebras. The concept of a clone, for instance, while one of the most important concepts of universal algebra, has no counterpart in the general theory of coalgebras. Clones appear only in a particular case, for coalgebras of type τ, as clones of co-operations. This was the motivation for our comparison of clones of operations and clones of co-operations.

Coalgebras based on a functor were first used to dualize the Eilenberg-Moore construction ([60], [3]). In [5] a coalgebraic approach was used to describe the dynamics of deterministic automata, and later coalgebraic methods were applied to infinite data structures in [6]. Aczel and Mendler suggested in [1] a way to specify semantics of processes by means of final coalgebras. This idea is expressed more explicitly in [2], and culminates in the argument that syntax is an algebraic phenomenon while semantics are coalgebraic.

Not all aspects of the theory of F-coalgebras are covered in this book: for instance, the duality between the equational theory of universal algebra and modal logic is not considered. We believe however that we have presented a clear overview of the area, from which further study may proceed. The reader is also encouraged to tackle the problems listed as exercises in each chapter, to further consolidate the theory and applications presented.

The material of this book is based on lectures and research seminars given by the first author and his students at the University of Potsdam (Germany), and the authors are grateful for the critical input of a number of students. Some of the material was also presented as a part of the DAAD-funded project "Centre of Excellence for Applications of Mathematics" at the South-West-University Blagoevgrad, Bulgaria. The work of the second author was funded by the NSERC of Canada.

K. Denecke and S. L. Wismath



Pushouts in *Set*

Universal Algebra and Coalgebra

Contents

Introduction

The purpose of this book is to study the structures needed to model objects in the two areas of algebra and theoretical computer science. Universal or general algebra is used to describe algebraic structures, while coalgebras are used to model state-based or finite-state machines in computer science.

Most mathematicians first encounter algebraic structures in the classical examples of groups, rings, fields and vector spaces. In each of these areas, common themes arise: we have sets of objects which are closed under one or more operations performed on the objects, and we are interested in subsets which inherit the structure (subgroups, subspaces, etc.), in mappings which preserve the structure (group homomorphisms, linear transformations, etc.) and construction of new structures from old, for instance by Cartesian products or quotients. We can also classify our structures according to the laws or identities they satisfy, as for instance with commutative groups or groups of order four.

In universal algebra we abstract and generalize from these examples to a core structure of *an algebra*: a set A of objects, with one or more operations defined on the set. We study substructures, homomorphisms and product algebras, and we classify algebras according to the identities they satisfy. To study such algebras we also need to know how many operation symbols our algebra has, and the arity of each one. This information is called the *type* or *signature* of the algebra. In general we assume a type indexed by some set I: for each $i \in I$ we have an operation symbol f_i, of arity $n_i \geq 0$, and we write the type as $\tau = (n_i)_{i \in I}$.

While universal algebras can be used to model most algebraic structures, they are not as useful in modelling state-based systems. The main reason for this is the following. An n_i-ary operation on set A is a mapping $f^A : A^{n_i} \mapsto A$, which combines n_i "input" elements of A into one output element. In a state-based system however, we often have the opposite situation: we need to map a single state to an output which carries several pieces of information, for instance to a state-output-symbol pair. That is, we need mappings from set A to some more complex set involving A. Originally a co-operation on set A was defined to be a mapping from A to a copower $A^{\sqcup n}$ of A, that is, a disjoint union of n copies of A for some number n which is the arity of the co-operation. A co-algebra was then a structure consisting of a set A with one or more co-operations defined on it. Again we have a type for the coalgebra, indexed by some set I, and we look at coalgebras of type τ.

Coalgebras of type τ are thus dual to algebras of type τ. The first papers on coalgebras (see [33], [31]) were motivated by this theoretical interest in dualizing concepts and theorems from universal algebra, and were largely forgotten. However, in the last decade coalgebras have become important as the fundamental structures needed to model state-based systems and state-based dynamics, and universal coalgebra has emerged as a general theory of such systems. The connection between algebras and coalgebras also provides a way to connect static data-oriented systems with dynamical behaviour-oriented systems. This duality has been informally known for a long time, with algebras used to describe data types and coalgebras used to describe abstract systems or machines.

In order to model dynamic state-based systems, we need to generalize the original definition of a co-algebra to allow other codomains than just copowers of A. For instance, the codomain might be a product of the form $A \times \Sigma$ where Σ is an input or output language of the machine. In general, we use some functor F to describe this structure, and consider mappings $f : A \mapsto F(A)$. This leads to the definition of an F-coalgebra, for a functor F, as a structure with a base set A together with one or more mappings from A to $F(A)$. There is an algebraic dual of this concept too: an F-algebra is a set A with one or more mappings from $F(A)$ to A.

It is evident that duality plays a key role in the study of algebras and coalgebras, and a main goal of this book is to explain this duality in a clear and accessible way. We have already mentioned the use of a functor in defining the general version of F-algebras and F-coalgebras, and indeed category theory plays a key role in formalizing the duality we need. For this reason we introduce categorical notions early in our development of the theory. Let $\mathbf{C}$ be a category, and let $F : \mathbf{C} \to \mathbf{C}$ be a functor on this category. An F-coalgebra over the category $\mathbf{C}$ is defined as an object A of $\mathbf{C}$ equipped with a morphism $\alpha_A : A \to F(A)$. When $\mathbf{C}$ is a category of sets, the morphism α_A is a mapping from the set A to the set $F(A)$. For a special set-valued functor F the mapping α_A can encode the indexed or non-indexed set of fundamental co-operations of a coalgebra $\mathcal{A}$. Dually, F-algebras are pairs consisting of a set A and a mapping $\beta_A : F(A) \to A$, where β_A encodes the indexed or non-indexed set of fundamental operations of the algebra $\mathcal{A}$. In this most general sense, the concept of an F-coalgebra generalizes that of a coalgebra of type τ, and similarly for F-algebras and algebras of type τ. It turns out that the concept of an F-coalgebra models many kinds of structures occurring in mathematics and computer science. F-coalgebras over categories of algebras are also of interest in the area of concurrency theory (see [16]).

In Chapter 1 of this book we introduce the topic of universal algebra, beginning with algebras and operations. We describe the subalgebra, homomorphic image, product and quotient constructions, and define a variety as a class of algebras closed under the formation of subalgebras, products and homomorphic images. We also define terms and identities, and an equational class as a class of algebras which all satisfy a common set of identities. The chapter culminates in Section 1.4 where we study the connection between the algebraic-structural approach of varieties and the more model-theoretical, logical approach of equational classes. The main theorem here is Birkhoff's Theorem, which tells us that the two approaches are equivalent: this theorem says that any variety is an equational class, and vice versa.

As a lead-in to the modelling of state-based systems, we show that any algebra on set A with only unary fundamental operations $f_1^A, \ldots, f_r^A$ can be regarded as an automaton without output. The set of unary fundamental operations forms the input alphabet of the automaton, while the set of states is the universe set A of the algebra. A non-empty subset A' of the set A must be chosen as the set of final states, and one designated element a_0 from the universe is used as an initial state. Then a word $f_{i_1}^A \cdots f_{i_r}^A$ is recognized if and only if the output $f_{i_1}^A(f_{i_2}^A(\ldots(f_{i_r}^A(a_0))\ldots))$ is in A'.

To model automata with output we also need structures with more than one set of objects. These are called multi-based or multi-sorted algebras. In the last section of Chapter 1 we describe such multi-based algebras. An important example of this structure is a clone, which has a disjoint set of objects for each $n \geq 1$. For example, the collection of all finitary operations on a base set A forms a clone, with the set $O_n(A)$ for each $n \geq 1$ consisting of all the n-ary operations on A.

The main purpose of Chapter 2 is to illustrate through a number of examples how the notion of coalgebra models a large class of state-based systems. In general, state-based systems satisfy the following criteria:

- the behaviour of the system depends on internal states, which are not visible to the user of the system;
- the system is reactive, interacts with its environment, and is not necessarily terminating;
- the interaction is performed by a set of operations.

For each example presented of a state-based system, we show how it may be regarded as a coalgebra, how to express structure-preserving maps or homomorphisms between such systems as coalgebra homomorphisms, and how to model behavioural equivalence of states.

As noted above, our general approach to the duality of algebras and coalgebras is based on categorical ideas, and in Chapter 3 we give a brief introduction to and overview of category theory. But our approach is limited mainly to the category of sets: we consider the interpretation of each abstractly defined categorical notion in the category of sets, and we consider mainly what are called concrete categories, where the objects are sets equipped with some structure such as operations, co-operations, relations, or topologies.

In Chapter 4 we define F-coalgebras and their homomorphisms, and study the properties of the category of all F-coalgebras. The main question is the categorical one of whether there exist coproducts, coequalizers and pushouts, or colimits generally, in this category. It turns out that some properties, dual to well-known and easy to prove results in universal algebra, work for F-coalgebras only if we restrict to special kinds of functors F. These are what are called standard functors, functors which preserve weak pullbacks. In this chapter we also define the dual to a variety of algebras, the concept of a covariety of coalgebras, as a class of F-coalgebras which is closed under the formation of homomorphic images, subcoalgebras and coproducts. We also introduce the concept of a bisimulation, a relation which models behavioural equivalence of state-based systems; this is a concept not studied in universal algebra.

Noting that coalgebras are dual to algebras, and having generalized coalgebras to F-coalgebras in Chapter 4, we turn in Chapter 5 to the question of the dual of F-coalgebras. Here we construct F-algebras as generalizations of algebras of type τ.

Chapter 6 takes a further step in abstraction to a single structure which encompasses both F-algebras and F-coalgebras. This can be done by using two functors F_1 and F_2, instead of a single functor F. A functorial system or (F_1, F_2)-structure ((F_1, F_2)-coalgebra), for functors F_1 and F_2, consists of a set A and mappings $f : F_1(A) \mapsto F_2(A)$. This expresses both F-algebras, by taking $F_1 = F$ and F_2 to be the identity functor, and F-coalgebras, when $F_2 = F$ and F_1 is the identity functor. But this concept also models other interesting algebraic structures as well, such as power algebras (also called hyperstructures) and power coalgebras, and tree automata. This structure thus allows us to unify results from the two different areas of algebra and theoretical computer science. We also investigate in this chapter conditions on F_1 and F_2 under which limits and colimits in the category of all (F_1, F_2)-coalgebras exist.

A key concept in the study of algebras of type τ is the free algebra of the type over a countably infinite set of generators. This algebra has a universal mapping property, expressed in categorical terms by the fact that it is an initial object in the category $Alg(\tau)$ of all algebras of type τ. In Chapter 7 we consider the dual concept in the category of F-coalgebras, a terminal F-coalgebra. Elements of a terminal coalgebra can be viewed as interpreting the behaviour of state-based systems.

As mentioned in Chapter 1, a key theorem in universal algebra is Birkhoff's result that varieties are the same as equational classes. The varietal approach to coalgebras is discussed here in Chapter 4, and in Chapter 8 we study the equational approach. We introduce coequations (as patterns of behaviour) and prove a dual of Birkhoff's Theorem.

To consider the dual of terms and identities, we turn in Chapter 9 to the study of coalgebras of type τ, as F-coalgebras for a particular functor F. In this setting we can define coterms, coidentities and covarieties, analogous to the terms, identities and varieties of universal algebra. The set of all coterms of type τ induces a clone of co-operations.

The study of clones is another important feature of this work. Clones occur in universal algebra as clones of term operations or polynomial operations of an algebra or variety. Also well known is the clone of all operations on a base set A, and its lattice of subclones. When A has cardinality three or more, this lattice is uncountably infinite. The remaining case, when A is a two-element set, is usually called the Boolean case, and the countably infinite lattice of all Boolean clones has been well-studied, particularly by Post. In Chapter 10 we compare Post's lattice of all Boolean clones of operations with the lattice of all clones of co-operations defined on a two-element set, which turns out to be a finite lattice.