Algebraic and Coalgebraic Structures

(Lecture Notes for *Álgebra de Processos*)

Mestrado em Matemática Computacional

Luís Soares Barbosa

lsb@di.uminho.pt
Departamento de Informática
Universidade do Minho

Março 2005

1 Observation & Construction

1. Functions. One of the most elementary models of a computational process is that of a *function*

$$f : I \longrightarrow O$$

which specifies a transformation rule between two structures I and O. The behaviour of a function is captured by the output it produces, which is completely determined by the supplied input. In a (metaphorical) sense, this may be dubbed as the 'engineer's view' of reality: *here is a recipe (a tool, a technology) to build gnus from gnats.*

Often, however, reality is not so simple. For example, one may know how to produce 'gnus' from 'gnats' but not in all cases. This is expressed by observing the output of f in a more refined context: O is replaced by $O + \mathbf{1}$ and f is said to be a *partial* function. In other situations one may recognise that there is some environmental (or context) information about 'gnats' that, for some reason, should be hidden from input. It may be the case that such information is too extensive to be supplied to f by its user, or that it is shared by other functions as well. It might also be the case that building gnus would eventually modify the environment, thus influencing latter production of more 'gnus'. For U a denotation of such context information, the signature of f becomes

$$f : I \longrightarrow (O \times U)^U$$

In both cases f can be typed as

$$f : I \longrightarrow \mathsf{T}\, O$$

for $\mathsf{T} = \mathsf{Id} + \mathbf{1}$ and $\mathsf{T} = (\mathsf{Id} \times U)^U$, respectively. Informally, T can be thought of as a type transformer providing a *shape* for the output of f. Technically, T is a *functor* which, to facilitate composition and manipulation of such functions, is often required to be a *monad*. In this way, the 'universe' in which $f : I \longrightarrow \mathsf{T}\, O$ lives and is reasoned about is the *Kleisli category* for T. In fact, monads in functional programming offer a general technique to smoothly incorporate, and delimit, 'computational effects' of this kind without compromising the purely functional semantics of such languages, in particular, referential transparency.

2. State. A function computed within a context is often referred to as 'state-based', in the sense the word 'state' has in automaton theory — the internal memory of the automaton which both constraints and is constrained by the execution of actions. In fact, the 'nature' of $f : I \longrightarrow (O \times U)^U$ as a 'state-based function' is made more explicit by rewriting its signature as

$$f : U \longrightarrow (O \times U)^I$$

This, in turn, may suggest an alternative model for computations, which (again in a metaphorical sense) one may dub as the 'natural scientist's view'. Instead of a recipe to build 'gnus' from 'gnats', the simple awareness that *there exist gnus and gnats and that their evolution can be observed*. That *observation* may entail some form of *interference* is well known, even from Physics, and thus the underlying notion of computation is not necessarily a passive one.

The able 'natural scientist' will equip herself with the right 'lens' — that is, a tool to observe with, which necessarily entails a particular shape for observation. Also note that the emphasis is now placed on the state itself: the input and output parameters may or may not become relevant, depending on the particular kind of observation one may want to perform. In other words, one's focus becomes the 'nature', or the 'universe' or, more pragmatically, the *state space*. That we can observe 'gnus' being produced out of 'gnats' is just one, among other possible observations. The basic ingredients required to support an 'observational', or 'state-based', view of computational processes may be summarised as follows:

a *lens*:	$\bigcirc\frown\bigcirc$	a functor T
an *observation structure*:	universe $\xrightarrow{p}$ $\bigcirc\frown\bigcirc$ universe	a T-coalgebra

Formally, in Set, a *coalgebra* for a functor T is a set U, which corresponds to the object being observed (the 'universe'), and a function $p : U \longrightarrow \mathsf{T}\ U$. Such a function is often referred to as the coalgebra *dynamics*.

3. Colours. There is, of course, a great diversity of 'lenses' and, for the same 'lens', a variety of observation structures, *i.e.*, of coalgebras. Moreover, such structures can be related and compared. For this one only needs what in Universal Algebra is known as a *homomorphism*, *i.e.*, a structure-preserving map. In our case the structure to be preserved is the shape of T as an observation tool. Therefore, a T-coalgebra morphism (or *comorphism*, as abbreviated in the sequel) h between, say, coalgebras p and q is just a function between the respective carriers ('universes' or 'state-spaces') making the following diagram to commute:

$$\begin{tikzcd} U \arrow[r, "p"] \arrow[d, "h"'] & \mathsf{T}\ U \arrow[d, "\mathsf{T}\ h"] \\ V \arrow[r, "q"'] & \mathsf{T}\ V \end{tikzcd}$$

Let us consider some possible lenses. An extreme case is the opaque lens: no matter what we try to observe through it, the outcome is always the same. Formally, such a lens is the constant functor $\underline{\mathbf{1}}$ which maps every object to $\mathbf{1}$ and every morphism to the identity on $\mathbf{1}$. Since $\mathbf{1}$ is the final object in Set, all $\mathbf{1}$-coalgebras reduce to $!$.

A slightly more interesting lens is $\underline{\mathbf{2}}$, which allows states to be classified into two different classes: black or white. This makes it possible to identify *subsets* of the 'universe' U under observation, as an observation structure p for this functor will map elements of U to one or another element of $\mathbf{2}$.

Should an arbitrary set O be chosen to colour one's lens, the possible observations become more discriminating. A coalgebra for $\underline{O}$ is a 'colouring' device in the sense that elements of the universe are classified (*i.e.*, regarded as distinct) by being assigned to different elements of O. Of course, a map h between such two observation structures p and q should be a colour-preserving function, *i.e.*, equation

$$q \cdot h \;=\; p$$

must hold. This means that if two elements of the universe are grouped together by p, their images under h remain together when compared by q.

4. Interfaces. A 'colour set' as $\mathbf{1}$, $\mathbf{2}$ or O above, can be regarded as a *classifier* of the state space. Coalgebras, for such constant functors, are *pure* observers providing a limited access to the state space by mapping into the 'colour set'. In object-oriented programming they are known as *attributes*. Naturally, the same 'universe' can be observed through different attributes and, furthermore, such observations can be carried out on parallel. Thus, the shape of a 'multi-attribute' lens is

$$\bigcirc\!\frown\!\bigcirc\; U \;=\; \prod_{k \in K} O_k$$

where K is a finite set of attribute names. The corresponding observation structure, a function mapping U to a (finite) product, is defined as a K-indexed *split*

$$\langle o_k \rangle_{k \in K} : U \longrightarrow \prod_{k \in K} O_k$$

of attributes o_k from U to O_k.

A very common assumption in state-based computations is that the state itself is a 'black box': it may evolve either internally or as a reaction to external stimuli, but the only way one has to become aware of such an evolution is by observing the values of its attributes. The product of their types forms the *output interface* of the coalgebra. No direct access to the state space is possible. Under this assumption the 'transparent' lens is not particularly useful. Technically, this lens corresponds to the *identity* functor Id. An observation structure for Id amounts to a function

$$p : U \longrightarrow U$$

This means that, by using p, the state U can indeed be modified, an ability we hadn't before. But, on the other hand, the absence of attributes makes any meaningful observation impossible. The best we can say, if no direct access to U is allowed, is just that *things happen.*

A better alternative is to combine attributes with such state modifiers, or update operations, to model the 'universe' evolution. The latter will be called *actions* here; in the object paradigm they are known as *methods*. Such a combination leads to a richer stock of lens. We might consider, for example, that

1. *things happen and disappear* or *stop*, *i.e.*

   $$\bigcirc\frown\bigcirc\, U \;=\; U + \mathbf{1}$$

   The observation structures for this functor are the partial functions. Accordingly, morphisms between them consist of functions that preserve partiality.

2. *things happen and, in doing so, some of their attributes become visible*, *i.e.*, (non trivial) output is produced:

   $$\bigcirc\frown\bigcirc\, U \;=\; U \times O$$

3. *the evolution of things is triggered by some external stimulus*, *i.e.*, additional *input* is accepted:

   $$\bigcirc\frown\bigcirc\, U \;=\; U^I$$

4. *we are not completely sure about what has happened*, in the sense that the evolution of the system being observed may be nondeterministic. In this case, the lens above can be combined with

   $$\bigcirc\frown\bigcirc\, U \;=\; \mathcal{P}U$$

   where $\mathcal{P}U$ is the finite powerset of U.

In the third example, the action also has an *input interface*. Typically, actions over the same state space cannot happen simultaneously and, therefore, if more than one is specified in a particular structure, in each execution the input supplied will also select the action to be activated. In some cases, the input is there only for selection purposes: actions with trivial input (*i.e.*, $I = \mathbf{1}$) correspond to buttons that can be pressed. Then the input interface organises itself as a coproduct. A possible shape for a sophisticated lens with both attributes and actions is

$$\bigcirc\frown\bigcirc\, U \;=\; \prod_{k \in K} O_k \times U^{\sum_{j \in J} I_j}$$

whose coalgebras are

$$\langle \langle o_k \rangle_{k \in K}, menu \rangle : U \longrightarrow \bigcirc\frown\bigcirc\, U$$

where $menu$ can be specified as

$$menu \;=\; \overline{[a_j]_{j \in J} \cdot \mathsf{dr}_J}$$

where dr_J is the suitable distribution law and each $a_j : U \times I_j \longrightarrow U$ is an elementary action. There are, of course, other alternatives to combine actions and attributes into sophisticated observation structures.

5. Combining Lenses. The tree below depicts several combinations of three of the basic alternative lenses of §4 (nondeterministic observations are left aside for the moment):

$$
\begin{tikzcd}
 & & (O \times \mathsf{Id})^I \arrow[r] & (O \times \mathsf{Id})^I + \mathbf{1} & \text{(a)} \\
 & O \times \mathsf{Id} \arrow[ur] \arrow[dr] & & & \\
 & & (O \times \mathsf{Id}) + \mathbf{1} \arrow[r] & ((O \times \mathsf{Id}) + \mathbf{1})^I & \text{(b)} \\
 & & O \times (\mathsf{Id} + \mathbf{1}) \arrow[r] & (O \times (\mathsf{Id} + \mathbf{1}))^I & \text{(c)} \\
\mathsf{Id} \arrow[uuur] \arrow[r] \arrow[dddr] & \mathsf{Id} + \mathbf{1} \arrow[ur] \arrow[dr] & & & \\
 & & (\mathsf{Id} + \mathbf{1})^I \arrow[r] & O \times (\mathsf{Id} + \mathbf{1})^I & \text{(d)} \\
 & & O \times \mathsf{Id}^I \arrow[r] & (O \times \mathsf{Id}^I) + \mathbf{1} & \text{(e)} \\
 & \mathsf{Id}^I \arrow[ur] \arrow[dr] & & & \\
 & & \mathsf{Id}^I + \mathbf{1} \arrow[r] & O \times (\mathsf{Id}^I + \mathbf{1}) & \text{(f)}
\end{tikzcd}
$$

It is instructive to see what happens in each case if the state space 'collapses', *i.e.*, if the 'universe' becomes trivial or, formally, if U is identified with $\mathbf{1}$:

$$
\begin{tikzcd}
\text{(a)} \arrow[d, dash, dotted] & \text{(b)} \arrow[d, dash, dotted] & \text{(c)} \arrow[d, dash, dotted] & \text{(d)} \arrow[d, dash, dotted] & \text{(e)} \arrow[d, dash, dotted] & \text{(f)} \arrow[d, dash, dotted] \\
O^I + \mathbf{1} & (O + \mathbf{1})^I & (O \times \mathbf{2})^I & O \times \mathbf{2}^I & O + \mathbf{1} & O \times \mathbf{2}
\end{tikzcd}
$$

We conclude that (b) as a set is isomorphic to the space of partial functions, (e) is just a pointed set and (a) a pointed function space. On the other hand, (d) is a set and a predicate and in (f) the structure has boiled down simply to a set and a Boolean flag.

6. INTERACTION. Another way of regarding observation structures is as *transition systems* over the state space. For example, coalgebras over both Id^I and $\mathsf{Id} \times O$ can be described in terms of *transitions* of the form

$$u \xrightarrow{x} u'$$

where u, $u' \in U$ and $x \in I$ or $x \in O$ in, respectively, the former and the latter case. Depending on how this transition relation is interpreted, we may classify the system as *reactive* or *active*, respectively.

- In the first case $u \xrightarrow{x} u'$ means that, for a coalgebra p, $(p\ u)\ x = u'$. Therefore p models a *reactive* system in which the transition can be read as *state u is able to accept x and move into state u'*.
- In the second case, the transition relation reads as *generate x in state u and then become u'*. Rather than a stimulus, x is an outcome and such a system is called *active*.

This distinction may not be intrinsic to the system being observed. It is essentially a distinction on the *shape* of observation. The lenses of the two examples complement each other. Recall that, similarly, finite automaton theory deals with automata *accepting* (*recognising*) or *generating* a particular language.

7. Observation Equivalence. Given a particular lens T and a T-observation structure p, when can two states be taken as 'equivalent'? If it is not possible to access their internal structure, all we can say is that they should be identified if all the observations that can be made over one and the other exhibit the same outcome and this remains true along the whole evolution of the system. Thinking of observation structures as transition systems, the notion of *bisimulation* [Par81, Mil80] can be recalled as precisely such a relation.

Whatever is observed of a system constitutes its *behaviour*, *i.e.*, the possible patterns of interaction with the observer. Two states are said bisimilar if they have (or *generate*, or *unfold to*) the same behaviour. Thus, equivalence means *indistinguishability* under observation. What coalgebra theory offers is a notion of *bisimulation* parametric on the particular 'lens' used. In other words, bisimilarity *acquires a shape*. An important fact is that, providing lens T is 'smooth enough', there exists a *canonical* representation embodying all T-behaviours into a T-coalgebra as well. Canonical in the sense that, from every other observation structure for the same lens, there is one and just one morphism to it. Technically, such a coalgebra, usually denoted by $\omega_{\mathsf{T}} : \nu_{\mathsf{T}} \longrightarrow \mathsf{T}\ \nu_{\mathsf{T}}$, is said to be the *final* coalgebra. The unique morphism mapping any other coalgebra p to it, which unfolds p to its behaviour, is represented by $[\!(p)\!]_{\mathsf{T}}$ and is called the *anamorphism* generated by p [MFP91], the *coinductive extension* of p [TR98] or, simply, the *final semantics* of the states of p [RT94].

8. Construction Strikes Back. On the other hand, and returning to our metaphor , the 'engineer's view' emphasises the possibility of at least some (essentially finite) things being not only observed, but actually *built*. In this case, one works not with a 'lens' but with a 'toolbox'. The *assembly process* is specified in a similar (but dual) way to the one used to define observation structures. I.e, the engineer will become equipped with,

a *tool box*:		a functor T
an *assembly process*:	artifact $\xrightarrow{d}$ artifact	a T-algebra

Notice that in the picture 'artifact' has replaced 'universe', to stress that one is now dealing with 'culture' (as opposed to 'nature') and, which is far more relevant, that the arrow has been *reversed*. Formally, an *assembly process* is a T-algebra. As a function this amounts to a collection of *constructors*. Because an artifact cannot be built simultaneously in two different ways (*i.e.*, constructors), the external structure or shape of the toolbox is usually a coproduct and the algebra arises as an *either* of constructors. For example, for the binary tree 'artifact', the suitable toolbox will be

$$\overset{\Cap}{\boxed{}\!\boxed{}}\; D \;=\; \mathbf{1} + \mathit{Data} \times D \times D$$

This means that binary trees can either be built via a constant constructor, yielding the trivial, empty tree, or via the aggregation of some data to two previously constructed trees, thus building a larger one. An assembly process arises then as

$$[\mathit{empty}, \mathit{node}] : \overset{\Cap}{\boxed{}\!\boxed{}}\; D \longrightarrow D$$

Of course, 'assembly processes' can be related and compared. In fact, the notion of an algebra for a functor generalises the classical concept of an algebra; morphisms of the former also generalise classical homomorphisms as functions preserving construction. Again, if the toolbox is smooth enough, there exist a canonical representative of the assembly process: the *initial* algebra, which may be regarded as the formal analog to the 'smallest' machine able to produce all possible $\overset{\Cap}{\boxed{}\!\boxed{}}$-artifacts.

In the next two sections algebraic and coalgebraic structures will be reviewed in deeper detail. As expected, *initial* algebras turn out to be *inductive data types*, *i.e.*, abstract descriptions of data structures. Dually, *final* coalgebras entail a notion of *behaviour types*, representing the dynamics of systems. Both of these structures, referred to as *categorical data types*, may be directly used in programming.

2 Algebraic Structures

9. Functors provide a sensible abstraction for the somewhat more vague notion of a *type* of a mathematical structure. And, in particular, for the even vaguer concept of a module *interface* in programming. Following the intuitions in the previous section, they specify the kind of 'lenses' available and the contents of 'toolboxes'. There are two main reasons for this. First, the type of a structure depends normally on types of other (sub-)structures and the very definition of a functor conveys this idea of *parameterization*. Secondly, a functor characterises, not only the structure itself, but also the transformations which preserve it. In fact, the action of a functor on a morphism f applies the transformation embodied in f to all the elements 'built into' the structure without changing the shape of the structure itself.

In this section, we will review structures intended to 'store' data elements in particular configurations. It turns out that the shapes (or types) of such configurations are suitably described by functors and the structures themselves arise as *algebras* for such functors.

10. Polynomial Functors. Despite of the wide scope of the previous paragraph, we shall restrict our attention to a particularly well-behaved class of endofunctors. We start by considering the so-called *polynomial functors*. The class is inductively defined as the least collection of functors containing the identity Id and constant functors $\underline{K}$ for all objects K in the category, closed by functor composition and finite application of product and coproduct functors. The terminology arises from the fact that, in a distributive category, any polynomial functor can be written in the form

$$\mathsf{T}\,X \;=\; \sum_i C_i \times X^i$$

for i a natural number and C_i a constant coefficient.

Polynomial functors are standard in presenting algebraic signatures. In the sequel, however, we will need to extend the inductive definition above to include

$$\begin{array}{lll} \mathsf{T}\,X &=& X^A \qquad \text{(the covariant exponential functor)} \\ \mathsf{T}\,X &=& \mathcal{P}X \qquad \text{(the finite powerset functor)} \end{array}$$

Functors in this class will be referred to in the sequel as *extended polynomial*. A more general class of functors — called *regular* — is obtained by further extending the definition to include *type* functors (§**??**).

11. Algebras. Syntactically, a data structure is described by a set of operations which specify how its values are to be produced. A *sequence*, for example, is either empty or built by adding an element to the front of a pre-existing sequence. A binary tree signature includes an empty constant and a node constructor whereby data and two other trees are aggregated to become the root node of a new tree, and so on. Notice that these two examples can be modeled by polynomial (§10) functors, which are basically n-ary sums (of alternatives) of m-ary products (of information associated to each alternative). For example,

$$\begin{array}{ll} \mathsf{T}_{\mathsf{Nat}}\,X = \mathbf{1} + X & \textit{natural numbers} \\ \mathsf{T}_{\mathsf{Seq}}\,X = \mathbf{1} + \mathit{Data} \times X & \textit{sequences} \\ \mathsf{T}_{\mathsf{Bin}}\,X = \mathbf{1} + \mathit{Data} \times X \times X & \textit{binary trees} \\ \mathsf{T}_{\mathsf{Lef}}\,X = \mathit{Data} + X \times X & \textit{leaf trees} \end{array}$$

All constructors of a given type can be grouped together into a single operation. For example, the constructors of a sequence are

$$[\mathsf{nil}, \mathsf{cons}] : \mathbf{1} + \mathit{Data} \times X \longrightarrow X$$

In general, if the shape of one of these structures is specified by a functor T, the structure itself is given as a map

$$d : \mathsf{T}\,D \longrightarrow D$$

i.e., as a T-*algebra*. Concrete structures are, therefore, obtained by specifying both the carrier set D and map d. Formally, we define,

12. Definition. For a given endofunctor T, a T-*algebra* is a pair $\langle D, d\rangle$ consisting of an object D, referred to as the *carrier* of d, and a map $d : \mathsf{T}\,D \longrightarrow D$. A T-algebra *morphism*, or simply, a T-*morphism*, between two T-algebras d and e is a map h between their carriers such that the following diagram commutes,

$$\begin{tikzcd} \mathsf{T}\,D \arrow[r, "d"] \arrow[d, "\mathsf{T}\,h"'] & D \arrow[d, "h"] \\ \mathsf{T}\,E \arrow[r, "e"] & E \end{tikzcd}$$

T-algebras and T-morphisms form a category C^{T} where both composition and identities are inherited from C. In the sequel, unless explicitly mentioned, we will be working on $\mathsf{Set}^{\mathsf{T}}$.

13. COMPATIBLE RELATIONS. A basic relation that can be established between two T-algebras is one that preserves their shape. Formally, given two T-algebras $\langle D, d\rangle$ and $\langle E, e\rangle$, a *compatible relation* R is a relation on $D \times E$ that can itself be extended to a T-algebra ρ such that the canonical projections become T-algebra morphisms. This may be expressed by the commutativity of the following diagram:

$$\begin{tikzcd}
\mathsf{T}\, D \arrow[d, "d"'] & \mathsf{T}\, R \arrow[l, "\mathsf{T}\, \pi_1"'] \arrow[r, "\mathsf{T}\, \pi_2"] \arrow[d, "\rho"'] & \mathsf{T}\, E \arrow[d, "e"] \\
D & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & E
\end{tikzcd}$$

It is well known that both the *kernel* and the *graph* of a T-algebra morphism are compatible relations. Conversely, if the graph of a map h between the carriers of two algebras is a compatible relation, then h is a T-algebra morphism. Should such a relation be also an equivalence, it would be called a (T-)*congruence*, a concept which plays a fundamental role in algebraic specification.

14. INITIAL ALGEBRAS. As we have mentioned in §8, there is a particular T-algebra which is canonical in the sense that only equally constructed elements can be identified. This is, of course, the *term* algebra, which happens to be the initial object in C^{T}. Notice, by the way, that the final object in this category always exists (if C itself has a final object $\mathbf{1}$): the trivial T-algebra $\langle \mathbf{1}, ! : \mathsf{T}\, \mathbf{1} \longrightarrow \mathbf{1}\rangle$. The initial algebra will be denoted by

$$\alpha_{\mathsf{T}} : \mathsf{T}\, \mu_{\mathsf{T}} \longrightarrow \mu_{\mathsf{T}}$$

Being initial means that there exists a *unique* T-algebra morphism from α_{T} to each other T-algebra $\langle D, d\rangle$. This morphism depends, of course, on d and is said to be the *inductive extension* of d [TR98] or the *catamorphism* generated by d [MFP91]. It is written as $(\!|d|\!)_{\mathsf{T}}$ or, simply, $(\!|d|\!)$, if the functor is clear from the context. As any other universal construction, it is unique up to isomorphism, which justifies the particle *the* used above.

Formally, a catamorphism is characterised as the unique T-morphism making the following diagram to commute

$$\begin{tikzcd}
\mathsf{T}\, D \arrow[r, "d"] & D \\
\mathsf{T}\, \mu_{\mathsf{T}} \arrow[u, "\mathsf{T}\, (\!|d|\!)_{\mathsf{T}}"] \arrow[r, "\alpha_{\mathsf{T}}"] & \mu_{\mathsf{T}} \arrow[u, "(\!|u|\!)_{\mathsf{T}}"']
\end{tikzcd}$$

or, alternatively, by the following universal property:

$$k = (\!|d|\!)_{\mathsf{T}} \;\Leftrightarrow\; k \cdot \alpha_{\mathsf{T}} = d \cdot \mathsf{T}\, k \tag{1}$$

15. CATA LAWS. From the universal property of catamorphisms (1) the following results are easily derived:

$$\begin{aligned} (\!|d|\!) \cdot \alpha_{\mathsf{T}} &= d \cdot \mathsf{T}\,(\!|d|\!) && (2)\\ (\!|\alpha_{\mathsf{T}}|\!) &= \mathsf{id}_{\mu_{\mathsf{T}}} && (3)\\ h \cdot (\!|d|\!) &= (\!|e|\!) \quad \text{if} \quad h \cdot d \;=\; e \cdot \mathsf{T}\,h && (4) \end{aligned}$$

Equations (2), (3) and (4) above are usually referred to as, respectively, the *cancellation*, *reflection* and *fusion* laws for catamorphisms.

16. Induction. Like any other universal construction, a catamorphism entails an *existence* property and an *uniqueness* one. Existence gives us a *definition* principle: to define a (circular) function from an initial algebra amounts to equip the target set with an algebra structure as well. In other words, a function is defined by providing a specification of its output for each of the constructors.

Uniqueness, on the other hand, gives a *proof* principle. Suppose we want to prove a predicate P over an initial algebra, *i.e.*, that the set defined by P coincides with μ_{T}. If it is possible to prove that the inclusion $i : P \hookrightarrow \mu_{\mathsf{T}}$ is a T-algebra morphism, we are done. In fact, we have equipped P with a T-algebra structure and, therefore, the composite of the corresponding catamorphism with i is unique and, necessarily, coincides with $\mathsf{id}_{\mu_{\mathsf{T}}}$. More simply, this amounts to prove that P is *closed* under the algebra constructors, which is easily recognised as the familiar *structural induction* principle.

Another way of stating this is to note that initial algebras have no proper sub-algebras, which again is a direct consequence of initiality. Some intuition on the (quite special) nature of initial algebras is gathered from the following result:

17. Lemma. The equality relation $\Delta_{\mu_{\mathsf{T}}}$ is the least T-compatible relation definable on the initial algebra α_{T}. That is, 'equality equals compatibility'.

Proof. Let $\langle R \subseteq \mu_{\mathsf{T}} \times \mu_{\mathsf{T}}, \rho \rangle$ be a compatible relation. Its projections are T-morphisms, but, by initiality,

$$\pi_1 \cdot (\!|\rho|\!)_{\mathsf{T}} \;=\; \pi_2 \cdot (\!|\rho|\!)_{\mathsf{T}} \;=\; \mathsf{id}_{\mu_{\mathsf{T}}}$$

Therefore, $\Delta_{\mu_{\mathsf{T}}} \subseteq R$, for any R. The result follows from the (trivial) fact that $\Delta_{\mu_{\mathsf{T}}}$ is itself a T-compatible relation.

$\square$

18. 'Algebras' are algebras. An expected, but fundamental, observation is that the notion of a T-algebra over Set subsumes the its classical homonym in Universal Algebra. Classically, an algebra is defined as a set plus a (finite) collection of constructors op_i : $X \times X \times ... \times X \longrightarrow X$. Denoting by ar_i the arity of operator op_i, this can be described as an algebra for functor

$$\mathsf{T}\,X \;=\; \sum_{i \in I} X^{\mathsf{ar}_i}$$

which captures its *signature*. Clearly, a T-morphism is just a classical homomorphism for the given class of algebras. Then the free term algebra over a set V of variables arises simply as the initial algebra, not for T, but for $\mathsf{T}' = \mathsf{T} + V$. T'-algebras have the form $[d, f] : \mathsf{T}\ D + V \longrightarrow D$, for any carrier D. Let

$$\langle W_V, \alpha_{\mathsf{T}'}\rangle \quad \text{with} \quad \alpha_{\mathsf{T}'} \;=\; [\sigma, i_V]$$

be the initial T'-algebra, where W_V is, as usual, the set of terms with variables taken from V, i_V the inclusion of variables from V as terms, and $\sigma : \mathsf{T}\ W_V \longrightarrow W_V$ the term formation operation, obviously a T-algebra. Given another T-algebra $d : \mathsf{T}\ D \longrightarrow D$ and a valuation function $f : V \longrightarrow D$ on the carrier of d, the induced unique morphism from the free algebra arises as catamorphism $h = (\!|[d, f]|\!)_{\mathsf{T}'}$. Therefore,

$$
\begin{array}{cl}
 & h \;=\; (\!|[d, f]|\!)_{\mathsf{T}'} \\
\equiv & \qquad \{ \text{ law (1) } \} \\
 & h \cdot \alpha_{\mathsf{T}'} \;=\; [d, f] \cdot \mathsf{T}'\ h \\
\equiv & \qquad \{ \text{ definitions } \} \\
 & h \cdot [\sigma, i_V] \;=\; [d, f] \cdot (\mathsf{T}\ h + \mathsf{id}) \\
\equiv & \qquad \{ \text{ + fusion and absorption } \} \\
 & [h \cdot \sigma, h \cdot i_V] \;=\; [d \cdot \mathsf{T}\ h, f]
\end{array}
$$

which (omitting variable injection operators and the superscripts in $op_i^{\alpha_{\mathsf{T}'}}$, the interpretation of constructor op_i in $\alpha_{\mathsf{T}'}$) can be re-written in the more familiar format:

$$
\begin{aligned}
h\ \ op_i\langle t_1, ..., t_n\rangle \;&=\; op_i^d\langle h\ t_1, ..., h\ t_n\rangle \\
h\ \ v \;&=\; f\ v
\end{aligned}
$$

where op_i^d stands for the interpretation in d of constructor op_i. It follows that, for a fixed d, every valuation function f corresponds bijectively to a T-morphism from $\langle W_V, \sigma\rangle$ to $\langle D, d\rangle$. And this for every algebra d. Moreover, this bijection is *natural* both in V and $\langle D, d\rangle$. In other words, and recalling, the forgetful functor $\mathsf{U} : \mathsf{C}^{\mathsf{T}} \longrightarrow \mathsf{C}$, which sends an algebra to its carrier, has a left adjoint $\mathsf{Free}^{\mathsf{T}} : \mathsf{C} \longrightarrow \mathsf{C}^{\mathsf{T}}$ mapping each set V into $\langle W_V, \sigma\rangle$. That is,

$$\mathsf{Free}^{\mathsf{T}} \;\vdash\; \mathsf{U}$$

As left adjoints preserve colimits, it turns out that, if C has an initial object and $\mathsf{Free}^{\mathsf{T}}$ exists, $\mathsf{Free}^{\mathsf{T}}\ \emptyset$ is the initial object in C^{T}, *i.e.*, the initial algebra. In Set, its carrier is, of course, $W_\emptyset$, the set of closed terms.

3 Coalgebraic Structures

19. From Algebras to Coalgebras. Once known how to build a data structure, one can reverse its 'assembly process'. Taking, as in §11, the simple case of sequences, such a decomposition is performed by the familiar *selectors* hd and tl, for nonempty sequences, which can be joined together in

$$\langle \mathsf{hd}, \mathsf{tl}\rangle : X \longrightarrow Data \times X$$

This reversal of our point of view (recall the introductory discussion in section 1) yields a different understanding of what X may stand for. First notice that what is structured in $\langle \mathsf{hd}, \mathsf{tl} \rangle$ is its target, instead of its source as before. Target product $Data \times X$ captures the fact that both the head and the tail of a sequence may be selected (or *accessed*, or *observed*) simultaneously. The emphasis on observation opens a broader understanding of the structure being defined. In fact, once one is no longer focused on how to construct X, but simply on what can be observed from it, finiteness is no longer required. Therefore, X can be more accurately thought of as a state space of a machine generating an infinite sequence of values of type $Data$. Elements of X can no longer be distinguished by construction, but should rather be identified when generating the same infinite sequence. That is, when it becomes impossible to distinguish them by the observations allowed by the shape (or 'lens') T.

Infinite sequences are common in programming. In practice they are represented by a particular *state* in a particular *state machine*. Formally, by an element of the carrier of a particular *coalgebra*, as described next.

20. Definition. Given an endofunctor T, a T-*coalgebra* is a pair $\langle U, p \rangle$ consisting of an object U, referred to as the *carrier* of p, and a map $p : U \longrightarrow \mathsf{T}\ U$. A T-coalgebra *morphism*, or simply, a T-*comorphism* between two T-coalgebras, $\langle U, p \rangle$ and $\langle V, q \rangle$, is a map h between carriers U and V such that the following diagram commutes:

$$\begin{tikzcd} U \arrow[r, "p"] \arrow[d, "h"'] & \mathsf{T}\ U \arrow[d, "\mathsf{T}\ h"] \\ V \arrow[r, "q"] & \mathsf{T}\ V \end{tikzcd}$$

T-coalgebras and T-comorphisms form a category C_T where both composition and identities are inherited from C.

21. Universal Coalgebra. The study of coalgebras along the lines of Universal Algebra, initiated by J. Rutten in [Rut95] and [Rut96] (of which a revised version [Rut00] appeared recently), assumes coalgebra carriers' are sets, and, therefore, constitutes an exploration of Set_T, for different Set endofunctors T. It should be mentioned, however, that both the study of concrete coalgebras over different base categories [TR98, Mon00] and the development of Set-independent, *i.e.*, purely categorical, presentations of coalgebra theory (see, among others [TR98, PW98, GS98] and chapter 1 in A. Kurz thesis [Kur01]) have emerged as active research areas.

22. Bisimulation. The role of bisimulations in coalgebra theory is similar to that of compatible relations (§13) in algebras. Informally, two states of a T-coalgebra (or of two different T-coalgebras) are related by a bisimulation if their observation produces equal results and this is maintained along all possible transitions. I. e., each one can mimic the other's evolution. The notion was introduced in process calculi by [Par81] and [Mil80] to capture a particular kind of observational equivalence. Later [AM88] gave a categorical definition of bisimulation which applies, not only to the kind of transition systems underlying the operational semantics of process calculi, but also to arbitrary coalgebras. As anticipated in §7, bisimulation *acquired a shape*. Formally,

23. Definition. Given two T-coalgebras $\langle U, p \rangle$ and $\langle V, q \rangle$, for a Set endofunctor T, a T-*bisimulation* is a relation $R \subseteq U \times V$ which can be extended to a coalgebra ρ such that projections π_1 and π_2 lift to T-comorphisms, as expressed by the commutativity of the following diagram:

$$\begin{tikzcd}
U \arrow[d, "p"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\rho"'] & V \arrow[d, "q"] \\
\mathsf{T}\, U & \mathsf{T}\, R \arrow[l, "\mathsf{T}\ \pi_1"'] \arrow[r, "\mathsf{T}\ \pi_2"] & \mathsf{T}\, V
\end{tikzcd}$$

The definition generalises to an arbitrary base category C, replacing relation R by a monic span $\langle R, r_1, r_2 \rangle$ in C whose legs lift to T-coalgebra morphisms, or, in other words, such that there is a T-coalgebra structure $\rho : R \longrightarrow \mathsf{T}R$ making the diagram below to commute:

$$\begin{tikzcd}
 & R \arrow[dl, "r_1"'] \arrow[dr, "r_2"] \arrow[dd, "\rho"] & \\
U \arrow[dd, "p"'] & & V \arrow[dd, "q"] \\
 & \mathsf{T}R \arrow[dl, "\mathsf{T}r_1"'] \arrow[dr, "\mathsf{T}r_2"] & \\
\mathsf{T}U & & \mathsf{T}V
\end{tikzcd}$$

Notice that a span $\langle R, r_1, r_2 \rangle$ is *monic* iff, for any arrows $f, g : X \longrightarrow R$, $r_1 \cdot f = r_1 \cdot g$ and $r_2 \cdot f = r_2 \cdot g$ implies $f = g$. Should C have binary products, being a monic span equivales to require split $\langle r_1, r_2 \rangle : X \longrightarrow U \times V$ to be monic as well.

24. Bisimulations And Comorphisms. Bisimulations provide a 'relational' view of comorphisms. In fact, the graph — $\mathsf{graph}\, h$ — of a T-comorphism $h : p \longrightarrow q$ is a T-bisimulation [Rut00]. An immediate, but fundamental, corollary of this result is the fact that, in every coalgebra $\langle U, p \rangle$, the diagonal Δ_U is a bisimulation. This follows from $\Delta_U = \mathsf{graph}\, \mathsf{id}_U$, the identity id_U being trivially a comorphism.

25. Properties. [Rut00] proves several results on bisimulations in $\mathsf{Set}_{\mathsf{T}}$. In particular, it is shown that the *converse* of a bisimulation and, if T preserves weak pullbacks, the *composition* of two bisimulations are still bisimulations. As a corollary, the (relational) *direct* and *inverse* images of a bisimulation, as well as the *kernel* of a comorphism, are bisimulations as well. To see this, it is enough to look at their definitions, where $\circ$ denotes relational composition,

$$\begin{array}{rcl}
h\ [R] & = & (\mathsf{graph}\, h)^\circ \circ R \circ \mathsf{graph}\, h \\
h^\circ\ [R] & = & \mathsf{graph}\ h \circ R \circ (\mathsf{graph}\, h)^\circ \\
\mathsf{ker}\, h & = & \mathsf{graph}\ h \circ (\mathsf{graph}\, h)^\circ
\end{array}$$

and conclude by the property above and §24. Moreover, as $\mathsf{ker}\, h$ is transitive, it gives rise to an equivalence bisimulation.

26. Remark. Some constructions in C_{T} depend on extra properties of functor T. As just mentioned, *preservation of weak pullbacks* is one of them. Recall that a *weak* universal is a construction which shares all the properties of the standard one but uniqueness. Its role is crucial namely in the proofs that composition of bisimulations and kernels of comorphisms are still bisimulations as well as to state the existence of a greatest bisimulation (§28). Also notice its use in the proof of the 'full abstraction lemma' in §35.

Another important property, in $\mathsf{Set}_{\mathsf{T}}$, is *boundedness*. This essentially means that there exists a set B such that, for any coalgebra $\langle U, p\rangle$, the size of any sub-coalgebra (§30) of p generated by any single element $u \in U$, is bounded by the size of B. This is a rather technical condition to avoid cardinality problems, namely, when discussing existence conditions for final coalgebras (§33). Both conditions hold for the extended polynomial functors defined in §10 [Rut00].

27. A CATEGORY OF BISIMULATIONS. Bisimulations on coalgebras $\langle U, p\rangle$ and $\langle V, q\rangle$ form a category $\mathsf{Bs}(p,q)$ whose arrows $m : \langle R, r_1, r_2\rangle \longrightarrow \langle S, s_1, s_2\rangle$ are morphisms $m : R \longrightarrow S$, in the base category C, such that $r_1 = s_1 \cdot m$ and $r_2 = s_2 \cdot m$, *i.e.*, the following diagram commutes:

$$\begin{tikzcd} & R \arrow[dl, "r_1"'] \arrow[dr, "r_2"] \arrow[dd, "m"] & \\ U & & V \\ & S \arrow[ul, "s_1"] \arrow[ur, "s_2"'] & \end{tikzcd}$$

28. T-BISIMILARITY. Clearly, every morphism m in $\mathsf{Bs}(p,q)$ is a monic in C.

Proof. Let m as in §27 and f, g stand for two arbitrary C morphisms with codomain R. Then,

$$\begin{array}{ll} & m \cdot f = m \cdot g \\ = & \qquad \{ \text{ Leibniz } \} \\ & s_1 \cdot m \cdot f = s_1 \cdot m \cdot g \;\text{ and }\; s_2 \cdot m \cdot f = s_2 \cdot m \cdot g \\ = & \qquad \{ \; m \text{ is a } \mathsf{Bs}(p,q) \text{ morphism } \} \\ & r_1 \cdot f = r_1 \cdot g \;\text{ and }\; r_2 \cdot f = r_2 \cdot g \\ \Rightarrow & \qquad \{ \; \langle R, r_1, r_2\rangle \text{ is a monic span } \} \\ & f = g \end{array}$$

□

That bisimulations are ordered by C-monics, as proved above, implies the existence of at most one arrow between any two bisimulations and generalises the well known fact that Set bisimulations are partially ordered by inclusion.

In Set the *union* of two bisimulations is also a bisimulation. Furthermore the set of all bisimulations between two coalgebras forms a complete lattice. The *greatest bisimulation* on, say, coalgebras p and q, is an *equivalence* relation, written as

$$\sim_{\langle p,q\rangle}$$

or simply, $\sim$ if the context is clear. Whenever the dependence on functor T is to be stressed, the notation $\sim^{\mathsf{T}}_{\langle p,q\rangle}$ (or $\sim^{\mathsf{T}}$) will be adopted.

29. Remark In general, coalgebra ρ in the definition of bisimulation (§23) is not uniquely determined — a counter example for the finite powerset functor is given in [Rut00]. Uniqueness is achieved, however, for polynomial endofunctors in Set and, more generally, for functors preserving pullbacks (which, therefore, preserve monic spans). Such a lack of uniqueness makes difficult the definition of constructions like the union of bisimulations or the greatest bisimulation: some conditions on either C or T are needed to show the existence of not uniquely defined structures. For example, to obtain greatest bisimulations one may recur to either the above mentioned preservation of weak pullbacks by T or the fact that all epis are split in C (see [Rut00, Kur01]).

This explains why, in more generic approaches to coalgebra theory, some alternatives to the notion of a bisimulation have been proposed. A beautiful one consists of replacing, in the definition of bisimulation, 'monic spans' by 'epi cospans', a cospan being simply a pair of arrows with a common codomain. This leads to the definition of what is called a *cocongruence* in [Kur01] or a *compatible correlation* in [Wol00] between two coalgebras $\langle U,p\rangle$ and $\langle V,q\rangle$. It is given by an epi copsan $\langle R,r_1,r_2\rangle$ in C whose legs lift to comorphisms as expressed by the commutativity of

$$\begin{tikzcd}
 & R \arrow[dd, "\rho"] & \\
U \arrow[ur, "r_1"] \arrow[dd, "p"'] & & V \arrow[ul, "r_2"'] \arrow[dd, "q"] \\
 & \mathsf{T}R & \\
\mathsf{T}U \arrow[ur, "\mathsf{T}r_1"] & & \mathsf{T}V \arrow[ul, "\mathsf{T}r_2"']
\end{tikzcd}$$

The interest of this definition is that ρ above is uniquely determined and, moreover, the greatest cocongruence on two coalgebras exists under rather general conditions [Kur01]. Intuitively, cocongruences capture behavioural equivalence because, given two states $u \in U$ and $v \in V$, $r_1\ u = r_2\ v$ only if the behaviour they unfold to is the same, as comorphisms preserve behaviour (§24).

Such a relation E on $U \times V$ determined by a particular cocongruence (*i.e.*, $\langle u,v\rangle \in E$ iff $r_1\ u = r_2\ v$) is a bisimulation if T preserves weak pullbacks, but seems more appropriate to capture behavioural equivalence on more general situations — see [AM88] and, mainly, [Kur01] for a full discussion.

As we shall restrict ourselves to a particularly well behaved class of Set endofunctors (§10), bisimulations provide all we need and this paragraph can be taken just as a curiosity. We would like to stress, however, the 'beauty' of the definition above which is the true dual of that of compatible relations on algebras mentioned in §13 (compare the diagrams!). As [Kur01] notes, the two crucial tools in algebra and coalgebra — compatible relations and correlations — can be simply and dually defined as, respectively, monic spans in C^{T} and epi cospans in C_{T}.

30. Sub-coalgebra. Let $\langle U,p\rangle$ be a T-coalgebra. A subset $i : U' \hookrightarrow U$ generates a *sub-coalgebra* of p, if the inclusion i lifts to a T-comorphism. Note that the coalgebraic structure associated to i is uniquely determined.

Proof. Consider the following diagram:

$$\begin{tikzcd} U \arrow[r, "p"] & \mathsf{T}\, U \\ U' \arrow[u, hook, "i"] \arrow[r, "q"] & \mathsf{T}\, U' \arrow[u, hook, "\mathsf{T}\, i"'] \end{tikzcd}$$

Note that $\mathsf{T}\ i$ is either the empty map, if $U' = \emptyset$, and we are done as q arises as the initial coalgebra, or a mono in all other cases, because functors preserve split monos with non empty domain (incidentally, they also preserve split epis). Now notice that q, as a Set-arrow, factorizes $p \cdot i$ through a mono (*i.e.*, $\mathsf{T}\ i$). The result follows from the fact that, in such conditions, the factorisation is unique.

□

Sub-coalgebras are related to bisimulations through the following result: a subset U' of U generates a sub-coalgebra iff $\Delta_{U'}$ is a bisimulation. Note also the following characterisation of monos and epis in Set:

31. Epis, Monos, Isos. Both epi and monomorphisms in Set lift to epi and monomorphisms in Set_T. Consequently, isomorphisms lift as well. The converse holds for epis, but for monomorphisms in Set_T to be also monomorphisms as Set-arrows, it is required that T preserves weak pullbacks (see [Rut00] for a proof).

32. Natural transformations. Given endofunctors T and R, any natural transformation $\sigma : \mathsf{T} \Longrightarrow \mathsf{R}$ provides a way to 'translate' T to R-coalgebras [Rut00]. In fact, σ induces a functor from C_T to C_R which maps a T-coalgebra $\langle U, p\rangle$ into a R-coalgebra $\langle U, \sigma_U \cdot p\rangle$, and is the identity on morphisms. Moreover, this functor preserves bisimulation, *i.e.*,

$$u \sim_p^\mathsf{T} u' \;\equiv\; u \sim_{\sigma_U \cdot p}^\mathsf{R} u' \tag{5}$$

33. Final Coalgebras. Successive observations of (or experiments with) a T-coalgebra p reveals its behavioural patterns. These are defined in terms of the results of the observers as recorded in the shape T. Then, just as the initial algebra is canonnically defined over the terms generated by successive application of constructors, it is also possible to define a canonical coalgebra in terms of such 'pure' observations. Such a coalgebra is the final object in C_T, if it exists, and will be denoted by ω_T.

Being *final* means that there exists a *unique* T-comorphism to ω_T from each other coalgebra $\langle U, p\rangle$. This is called the *coinductive extension* of p [TR98] or the *anamorphism* generated by p [MFP91], and written as $[\!(p)\!]_\mathsf{T}$ or, simply, $[\!(p)\!]$, if the functor is clear from context. In other words, an anamorphism is defined as the unique comorphism making the following diagram to commute:

$$\begin{tikzcd} \nu_\mathsf{T} \arrow[r, "\omega_\mathsf{T}"] & \mathsf{T}\ \nu_\mathsf{T} \\ U \arrow[u, "{[\!(p)\!]_\mathsf{T}}"] \arrow[r, "p"] & \mathsf{T}\, U \arrow[u, "{\mathsf{T}\ [\!(p)\!]_\mathsf{T}}"'] \end{tikzcd}$$

or, alternatively, by the following universal law:

$$k = [\![p]\!]_{\mathsf{T}} \ \Leftrightarrow\ \omega_{\mathsf{T}} \cdot k = \mathsf{T}\ k \cdot p \tag{6}$$

For each $u \in U$, $[\![p]\!]_{\mathsf{T}}\ u$ can be thought of as the (observable) behaviour of a sequence of p transitions starting at state u. This explains yet another alternative designation for an anamorphism: *unfold*. On its turn, u in $[\![p]\!]_{\mathsf{T}}\ u$, is called the *seed* of the anamorphism.

As in the algebraic case, the *existence* part of the universal property provides a *definition* principle for (circular) functions to the final coalgebra which amounts to equip their source with a coalgebraic structure specifying the 'one-step' dynamics. Then the corresponding anamorphism gives the rest. In other words, such functions are defined by specifying their output under all different observers. The *uniqueness* part, on the other hand, offers a powerful *proof* principle — *coinduction* — discussed in §36.

34. Ana Laws. The following laws follow from the universal property of anamorphisms — equation (6). Comparing with §15, one easily recognises them as the *cancellation*, *reflection* and *fusion* result for anamorphisms, respectively.

$$\omega_{\mathsf{T}} \cdot [\![p]\!] \ = \ \mathsf{T}\ [\![p]\!] \cdot p \tag{7}$$

$$[\![\omega_{\mathsf{T}}]\!] \ = \ \mathsf{id}_{\nu_{\mathsf{T}}} \tag{8}$$

$$[\![p]\!] \cdot h \ = \ [\![q]\!] \quad \text{if} \quad p \cdot h \ = \ \mathsf{T}\ h \cdot q \tag{9}$$

35. Lemma. We are ready to state and prove the fundamental characterisation result on final coalgebras, referred to, in [TR98], as the *full abstraction theorem* for final semantics. Let T preserve weak pullbacks. Given two T-coalgebras $\langle U, p\rangle$ and $\langle V, q\rangle$, any two states $u \in U$ and $v \in V$ satisfy

$$u \sim^{\mathsf{T}}_{\langle p,q\rangle} v \ \ \equiv\ \ [\![p]\!]_{\mathsf{T}}\ u = [\![q]\!]_{\mathsf{T}}\ v \tag{10}$$

Proof.

$\Rightarrow$

Let $\langle R \subseteq U \times V, \rho\rangle$ be a bisimulation such that $\langle u, v\rangle \in R$. Then, projections π_1 and π_2 lift to T-comorphisms, *i.e.*, $\pi_1 : \langle R, \rho\rangle \longrightarrow \langle U, p\rangle$ and $\pi_2 : \langle R, \rho\rangle \longrightarrow \langle V, q\rangle$. Now, composites $[\![p]\!]_{\mathsf{T}} \cdot \pi_1$ and $[\![q]\!]_{\mathsf{T}} \cdot \pi_2$ are comorphisms to the final coalgebra with identical source. By finality, they coincide.

$\Leftarrow$

We begin by defining relation $R = \{\langle u, v\rangle \in U \times V |\ [\![p]\!]_{\mathsf{T}}\ u = [\![q]\!]_{\mathsf{T}}\ v\}$. All we have to do is to prove that R lifts to a bisimulation, *i.e.*, we have to equip R with a coalgebraic structure ρ such that projections lift to comorphisms. Relation R is, in fact, the pullback (in Set) of $[\![p]\!]_{\mathsf{T}}$ and $[\![q]\!]_{\mathsf{T}}$. As T preserves weak pullbacks (§31), the following diagram is also, at least, a weak pullback:

$$\begin{tikzcd}
\mathsf{T}\ R \arrow[r, "\mathsf{T}\ \pi_1"] \arrow[d, "\mathsf{T}\ \pi_2"'] & \mathsf{T}\ U \arrow[d, "\mathsf{T}\ [\![p]\!]_{\mathsf{T}}"] \\
\mathsf{T}\ V \arrow[r, "\mathsf{T}\ [\![q]\!]_{\mathsf{T}}"] & \mathsf{T}\ \nu_{\mathsf{T}}
\end{tikzcd}$$

Now notice that R can be made a cone over the diagram for which $\mathsf{T}\,R$ is a weak pullback:

$$\begin{tikzcd}
R \arrow[dr, dotted, "\rho"] \arrow[drr, bend left, "p \cdot \pi_1"] \arrow[ddr, bend right, "q \cdot \pi_2"'] & & \\
& \mathsf{T}\,R \arrow[r, "\mathsf{T}\,\pi_1"] \arrow[d, "\mathsf{T}\,\pi_2"] & \mathsf{T}\,U \arrow[d, "\mathsf{T}\,[\!(p)\!]_{\mathsf{T}}"] \\
& \mathsf{T}\,V \arrow[r, "\mathsf{T}\,[\!(q)\!]_{\mathsf{T}}"] & \mathsf{T}\,\nu_{\mathsf{T}}
\end{tikzcd}$$

Here ρ plays the role of a mediating morphism to the weak pullback. It is, of course, not necessarily unique, but this is not required in the definition of a bisimulation. Moreover, combining this with the first part of the theorem, we conclude that $\langle R, \rho\rangle$ is, not only, a bisimulation, but the greatest one, *i.e.*, it coincides with $\sim^{\mathsf{T}}_{\langle p,,q\rangle}$.

□

Finally notice that, although the proof was presented in Set, the argument extends to an arbitrary category C. It suffices to recall that, in general, a T-bisimulation is defined as a span whose legs lift to comorphisms (§23) and define R directly as the pullback of the two anamorphisms. The general result stating that if T preserves weak pullbacks, then any pullback in C lifts to a bisimulation, is due to Aczel and Mendler [AM88].

36. Coinduction. The previous result shows that the final coalgebra ω_{T} acts as a state *classifier* for any other T-coalgebra and, moreover, that bisimulation provides a *local* proof theory for behavioural equality. This is exactly the core of the *coinduction* principle which may be stated, for every bisimulation R on a coalgebra $\langle U, p\rangle$ satisfying it,

$$R \;\subseteq\; \Delta_{\mathsf{id}_U}$$

Therefore, in such a coalgebra, to prove the equality of two states, it is enough to find a bisimulation containing them. Coalgebras that satisfy the *coinduction* principle are called *simple*. An alternative, equivalent, characterisation stresses the fact that Δ_{id_U} is the greatest bisimulation in such a coalgebra. This is indeed the case of the final coalgebra, since the projections of any bisimulation $\langle R \subseteq \nu_{\mathsf{T}} \times \nu_{\mathsf{T}}, \rho\rangle$ are comorphisms to ω_{T} and, therefore, by finality, $\pi_1 = [\!(\rho)\!]_{\mathsf{T}} = \pi_2$.

37. Cofree Coalgebras. Let us open a parenthesis to investigate the coalgebraic dual to *free* algebras discussed in §18. Free algebras are initial algebras with additional terms to represent variables. Dually, a T-*cofree* coalgebra [Jac95] is a final T-coalgebra in which states (thought of as behaviours) are *coloured* by elements of a set V (recall the informal discussion on 'lenses'). Formally, they are final coalgebras for functor $\mathsf{T}' = \mathsf{T} \times V$, *i.e.*,

$$\langle \sigma, \epsilon\rangle : K_V \longrightarrow \mathsf{T}\,K_V \times V$$

where $\epsilon : K_V \longrightarrow V$ is the state colouring morphism. Suppose, now, we are given a T-coalgebra $p : U \longrightarrow \mathsf{T}\,U$ and a 'colour assignment' $f : U \longrightarrow V$ from the carrier of p. The induced unique morphism to the cofree coalgebra is anamorphism $h = [\!(\langle p, f\rangle)\!]_{\mathsf{T}'}$. This is, in fact, an extension of the 'colour assignment' to behaviours generated from p, as a simple calculation shows,

$$
\begin{aligned}
& h \;=\; [\![(\langle p, f\rangle)]\!]_{\mathsf{T}'} \\
\equiv \quad & \qquad \{ \text{ ana universal (6) } \} \\
& \omega_{\mathsf{T}'} \cdot h \;=\; \mathsf{T}'\, h \cdot \langle p, f\rangle \\
\equiv \quad & \qquad \{ \text{ definitions } \} \\
& \langle \sigma, \epsilon\rangle \cdot h \;=\; (h \times \mathsf{id}_)) \cdot \langle p, f\rangle \\
\equiv \quad & \qquad \{ \text{ } \times\text{-fusion and absorption } \} \\
& \langle \sigma \cdot h, \epsilon \cdot h\rangle \;=\; \langle h \cdot p, f\rangle
\end{aligned}
$$

This can be re-written as a *coinductive definition* of h:

$$
\begin{aligned}
\sigma\ (h\ u) &= h\ (p\ u) \\
\epsilon\ (h\ u) &= f\ u
\end{aligned}
$$

This is an example of definition by coinduction: the function being defined, h, is specified by analysing its value under each observer. Considering the common case in which the most external shape of T is a product with n factors, the first clause unfolds to a collection of equations, one for each observer ob_i, with $i \in n$,

$$
ob_i\ (h\ u) \;=\; h\ ({ob_i}^p\ u)
$$

Similarly to what happens in the free algebra case, it turns out that there exists a bijective correspondence between arrows $f : U \longrightarrow V$ in C and T-comorphisms from $\langle U, p\rangle$ to $\langle K_V, \sigma\rangle$, the last being obviously a T-coalgebra. Being natural both in V and $\langle U, p\rangle$, this bijection witnesses an adjunction

$$
\mathsf{U} \;\vdash\; \mathsf{CoFree}^{\mathsf{T}}
$$

between the forgetful functor to C and the functor which associates each V to coalgebra $\langle K_V, \sigma\rangle$. As right adjoints preserve limits, it turns out that, if C has a final object $\mathbf{1}$ and $\mathsf{CoFree}^{\mathsf{T}}$ exists, $\mathsf{CoFree}^{\mathsf{T}}\ \mathbf{1}$ is the final object in C_{T}, *i.e.*, the final coalgebra. This leads us to the question of existence of final coalgebras.

38. Existence. In most cases being aware of the existence of a final coalgebra is all one needs to know. In fact, like any other universal, the use of the final coalgebra is completely determined by the universal property rather than by the internal structure of its carrier. Note, however, that this may be contrasted with a common and fruitful procedure used in coalgebraic reasoning which consists of exhibiting the underlying final coalgebra of some mathematical objects, such as streams or languages, to apply coinduction in the study of their properties (see, for example, [Jac96a, Rut98] or [Rut01]).

The next few paragraphs discuss briefly existence of final coalgebras, to conclude that they do exist for all functors considered in this text. Some concrete examples of final coalgebras are mentioned in §§43 and 44. Prior to that we shall recall a well known result which characterises both final coalgebras and initial algebras as fixpoints of functors.

39. Lambek Lemma. The final object in C_{T}, if it exists, is an isomorphism.

Proof. Let $\omega_{\mathsf{T}} : \nu_{\mathsf{T}} \longrightarrow \mathsf{T}\ \nu_{\mathsf{T}}$ be the final T-coalgebra. Because $\mathsf{T}\ \omega_{\mathsf{T}} : \mathsf{T}\ \nu_{\mathsf{T}} \longrightarrow \mathsf{T}\ \mathsf{T}\ \nu_{\mathsf{T}}$ is a T-coalgebra as well anamorphism $[\![(\mathsf{T}\ \omega_{\mathsf{T}})]\!]_{\mathsf{T}}$ exists. The composite $[\![(\mathsf{T}\ \omega_{\mathsf{T}})]\!]_{\mathsf{T}} \cdot \omega_{\mathsf{T}} : \nu_{\mathsf{T}} \longrightarrow \nu_{\mathsf{T}}$

is also a comorphism and, by finality, coincides with $\mathsf{id}_{\nu_\mathsf{T}}$. So $[\!(\mathsf{T}\ \omega_\mathsf{T})\!]_\mathsf{T} \cdot \omega_\mathsf{T} = \mathsf{id}_{\nu_\mathsf{T}}$ and the proof is half done. For the other half, note

$$
\begin{array}{ll}
 & \omega_\mathsf{T} \cdot [\!(\mathsf{T}\ \omega_\mathsf{T})\!]_\mathsf{T} \\
= & \qquad \{ \text{ comorphism } \} \\
 & \mathsf{T}\ [\!(\mathsf{T}\ \omega_\mathsf{T})\!]_\mathsf{T} \cdot \mathsf{T}\ \omega_\mathsf{T} \\
= & \qquad \{ \ \mathsf{T} \text{ functor } \} \\
 & \mathsf{T}\ ([\!(\mathsf{T}\ \omega_\mathsf{T})\!]_\mathsf{T} \cdot \omega_\mathsf{T}) \\
= & \qquad \{ \text{ just proved } \} \\
 & \mathsf{T}\ \ \mathsf{id}_{\nu_\mathsf{T}} \\
= & \qquad \{ \ \mathsf{T} \text{ functor } \} \\
 & \mathsf{id}_{(\mathsf{T}\ \mathsf{id}_{\nu_\mathsf{T}})}
\end{array}
$$

$\square$

As isomorphisms are self-dual, this also entails that the initial algebra of a functor, if it exists, is an isomorphism. Such was the original statement of the lemma in [Bar70], where it is credited to J. Lambek. As a corollary, notice that $\omega_\mathsf{T}^\circ = [\!(\mathsf{T}\ \omega_\mathsf{T})\!]_\mathsf{T}$.

40. FIXPOINTS. Lambek Lemma characterises both initial algebras and final coalgebras for a functor T as fixpoints of the equation

$$X \ \cong\ \mathsf{T}\, X$$

The initial algebra is said to be the *least* fixpoint of T, up to isomorphism, and the final coalgebra the *greatest*. The terminology arises from an analogy with what happens in a partial order $\langle P, \leq \rangle$ seen as a category. A functor, in such a setting, is just a monotone function and, therefore a coalgebra (respectively, an algebra) is an element x of P such that $x \leq \mathsf{T}\ x$ (respectively, $\mathsf{T}\ x \leq x$). The final coalgebra is, then, an element $m \leq \mathsf{T}\ m$ such that, for all $x \in P$, $x \leq \mathsf{T}\ x \ \Rightarrow\ x \leq m$. By Tarski's theorem, [Tar55], this is the greatest fixpoint of T with respect to $\leq$. Dually, the initial algebra arises as the least fixpoint. By analogy, least and greatest fixpoints of an endofunctor in an arbitrary category are defined as the initial algebra and the final coalgebra, respectively. As [MA86] remarks

> *This defines what we mean by least and greatest in the general case.*
> *There is no pre-established order.*

41. REMARK. The replacement of isomorphism by strict equality in an universe of sets raises foundational problems. In particular, the strict greatest fixpoint of most functors T would violate the *foundation axiom* of classical set theory which states the well-foundedness of the membership relation, *i.e.*, the no existence of infinitely descending chains $\ldots \in x_2 \in x_1 \in x$. This observation lead to the development of *non-well-founded* set theory [Acz88], in which the axiom is replaced by an 'anti-foundation axiom'. Such non standard set theory arose in the work of P. Aczel to provide a final semantics for CCS processes as elements of strict final coalgebras for functor $\mathcal{P}(A \times \mathsf{Id})$ in the category Class of large sets (or classes), where $\mathcal{P}$ stands for the (unrestricted) powerset functor.

From the point of view of category theory, such a 'foundational shift' seems avoidable, to a great extent, as final coalgebras do exist in Set and Class with classical foundations, as Aczel himself proved in [Acz88, AM88]. M. Barr [Bar92], in a polemic with J. Bairwise, illustrated this point of view in the following terms:

> *It is unfortunate that such solutions* [resorting to anti-foundation] *exist, for their main effect is to avoid giving serious consideration to the real problem: the irrelevance of actual elements in mathematics.*

The theory of non-well-founded sets has, however, an interest in its own, namely on research of set theoretic foundations for corecursion; see [BM96] where several modelling applications are given, in particular in the area of artificial intelligence.

42. Construction of Final Coalgebras. There is, however, a fundamental restriction which may prevent the existence of final coalgebras: *cardinality.* In particular, Lambek lemma implies that a final coalgebra for the unrestricted powerset functor in Set cannot exist, as it would violate Cantor's theorem. The problem is avoided by moving to the large category Class, as [AM88] does, and is, of course, non existent for the finite powerset functor we have been considering. In general, cardinality restrictions are avoided if we require T to be bounded (§26). [Rut00] proves that all extended polynomial functors (§10) are indeed bounded. In this paragraph we shall briefly review the construction of final coalgebras in Set.

The general method for building final coalgebras for polynomial functors is a generalisation of Kleene's theorem for finding fixpoints in complete partial orders. Basically, a fixpoint arises as the limit of a descending chain

$$\mathbf{1} \xleftarrow{\;!\;} \mathsf{T}\,\mathbf{1} \xleftarrow{\;\mathsf{T}\,!\;} \mathsf{T}^2\,\mathbf{1} \xleftarrow{\;\mathsf{T}^2\,!\;} \cdots$$

where $\mathsf{T}^n = \mathsf{T} \cdot \mathsf{T}^{n-1}$. More concretely, this can be seen as the set

$$\{\langle x_0, x_1, x_2, \cdots\rangle \mid x_n \in \mathsf{T}^n\,\mathbf{1} \;\wedge\; (\mathsf{T}^n\,!)\;x_{n+1} = x_n \quad \text{for all } n\}$$

This method requires T to preserve limits of descending chains, a condition usually known as ω-*continuity*, which is indeed the case of polynomial functors and the covariant exponential functor. A dual requirement, and a dual procedure, computes initial algebras. Recall that the original Kleene theorem can also be used to compute both least and greatest fixpoints. See [SP82] for an early reference and [MA86] for a detailed proof and examples.

43. Examples. For the cases covered by the Kleene method we can obtain concrete descriptions of the final coalgebras. Moreover, they arise as completions of the corresponding initial algebras. Let us see some examples in Set.

- The trivial example is the final coalgebra for the identity functor Id: it is, as expected, $\langle \mathbf{1}, \mathsf{id}_{\mathbf{1}} \rangle$.

- For functor $\mathsf{T}\,X = O \times X$ the carrier of the final coalgebra is the set O^ω of infinite sequences of O, with $\langle \mathsf{hd}, \mathsf{tl} \rangle$ as the dynamics. This extends to $O^\infty = O^* \cup O^\omega$, for the usual 'list' functor $\mathsf{T}\,X = \mathbf{1} + O \times X$.

- Functor $\mathsf{T}\,X = O \times X^I$, which is the type of (deterministic) systems with an observer (or attribute) o and a parametrized action a, has as final coalgebra

$$\langle \mathsf{o}, \mathsf{a} \rangle : O^{I^*} \longrightarrow O \times O^{I^*\,I}$$

where

$$\begin{aligned} \mathsf{o}\,m &= m\ \mathsf{nil} \\ \mathsf{a}\,m &= \lambda\,i\ \lambda\,s\ .\ m\ (s \frown \langle i \rangle) \end{aligned}$$

which amounts to infinite trees whose branches are labelled by sequences of inputs and leafs by values of O.

- The final coalgebra for the more general shape

$$\mathsf{T}\,X \;=\; \prod_j (O_j + P_j \times X)^{I_j}$$

is constructed in [Jac96b]. Its carrier is the space of functions from $\sum_j I_j$ to $(\sum_j O_j + \sum_j P_j)$, subject to an invariant that assures type compatibility (*i.e.*, an input on I_j will produce output in O_j and C_j) and completion (in the sense that when a node labelled by an output value of type O_j, for any j, is reached, the tree is completed by an infinite tree whose nodes are all labelled by that same value). Again branches are labelled by inputs and nodes by values from $\sum_j O_j + \sum_j P_j$. The root, however, is not labelled.

44. Powerset. Kleene's theorem does not apply to the (finite) powerset functor. In this case, existence of final coalgebras has been proved by M. Barr [Bar93]. Roughly, the intuition is to take the coproduct of all T-coalgebras and, then, quotienting it by the greatest bisimulation. Because such coproduct may not exist, the argument is reformulated in terms of a set of 'generators'.

For the common case $\mathsf{T}\,X = \mathcal{P}(O \times X)$, this yields, as one would expect from the semantics of process calculi, the set of rooted finitely branching trees, with branches labelled by O, quotiented by the greatest bisimulation.

45. The Structure of C_T. What M. Barr actually proved in [Bar93] is that the forgetful functor $U : \mathsf{Set}_\mathcal{P} \longrightarrow \mathsf{Set}$ has a right adjoint. This is, of course, $\mathsf{CoFree}^\mathcal{P}$ (§37) and, as Set has a final object, $\mathsf{CoFree}^\mathcal{P}\ \mathbf{1}$ gives the final coalgebra. Furthermore, the paper unveils much of the structure of Set_T for an arbitrary functor T. In particular, it is shown that coproducts and coequalizers exist and their carrier coincides with the same construction in Set. [Rut00] shows a similar result for all limits that are preserved by T. The structure of C_T, in the general case, has been studied by a number of people ([Rut00, GS98, Wor98, PW98, Ada00], among others).

References

[Acz88] P. Aczel. *Non-Well-Founded Sets*. CSLI Lecture Notes (14), Stanford, 1988.

[Ada00] J. Adamek. Final colagebras as ideal completions of initial algebras. Talk at the MFIT summer school on algebraic and coalgebraic methods in mathematics of program construction, Lincoln College, Oxford University, April 2000.

[AM88] P. Aczel and N. Mendler. A final coalgebra theorem. In D. Pitt, D. Rydeheard, P. Dybjer, A. Pitts, and A. Poigne, editors, *Proc. Category Theory and Computer Science*, pages 357–365. Springer Lect. Notes Comp. Sci. (389), 1988.

[Bar70] M. Barr. Coequalizers and cofree cotriples. *Mathematische Zeitschrift*, 166:307–322, 1970.

[Bar92] L. S. Barbosa. Sobre a especificação matemática de sistemas concorrentes. PAPCC Thesis DI-LSB-92:9:1, DI (U. Minho), September 1992. (in portuguese).

[Bar93] M. Barr. Terminal coalgebras in well-founded set theory. *Theor. Comp. Sci.*, 114(2):299–315, 1993.

[BM96] J. Bairwise and P. Moss. *Vicious Circles*. CSLI Lecture Notes (59), Stanford, 1996.

[GS98] H. P. Gumm and T. Schroeder. Covarieties and complete covarieties. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *CMCS'98, Elect. Notes in Theor. Comp. Sci.*, volume 11. Elsevier, March 1998.

[Jac95] B. Jacobs. Mongruences and cofree coalgebras. In V.S. Alagar and M. Nivat, editors, *Algebraic Methodology and Software Technology (AMAST)*, pages 245–260. Springer Lect. Notes Comp. Sci. (936), 1995.

[Jac96a] B. Jacobs. Object-oriented hybrid systems of coalgebras plus monoid actions. In M. Wirsing and M. Nivat, editors, *Algebraic Methodology and Software Technology (AMAST)*, pages 520–535. Springer Lect. Notes Comp. Sci. (1101), 1996.

[Jac96b] B. Jacobs. Objects and classes, co-algebraically. In C. Lengauer B. Freitag, C.B. Jones and H.-J. Schek, editors, *Object-Orientation with Parallelism and Persistence*, pages 83–103. Kluwer Academic Publishers, 1996.

[Kur01] A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. Ph.D. Thesis, Fakultat fur Mathematik, Ludwig-Maximilians Univ., Muenchen, 2001.

[MA86] E. Manes and A. Arbib. *Algebraic Approaches to Program Semantics*. Texts and Monographs in Computer Science. Springer Verlag, 1986.

[MFP91] E. Meijer, M. Fokkinga, and R. Paterson. Functional programming with bananas, lenses, envelopes and barbed wire. In J. Hughes, editor, *Proceedings of the 1991 ACM Conference on Functional Programming Languages and Computer Architecture*, pages 124–144. Springer Lect. Notes Comp. Sci. (523), 1991.

[Mil80] R. Milner. *A Calculus of Communicating Systems*. Springer Lect. Notes Comp. Sci. (92), 1980.

[Mon00] L. Monteiro. Observation systems. In H. Reichel, editor, *CMCS'00 - Workshop on Coalgebraic Methods in Computer Science*. ENTCS, volume 33, Elsevier, 2000.

[Par81] D. Park. Concurrency and automata on infinite sequences. pages 561–572. Springer Lect. Notes Comp. Sci. (104), 1981.

[PW98] J. Power and H. Watanabe. An axiomatics for categories of coalgebras. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *CMCS'98, Elect. Notes in Theor. Comp. Sci.*, volume 11. Elsevier, March 1998.

[RT94] J. Rutten and D. Turi. Initial algebra and final co-algebra semantics for concurrency. In *Proc. REX School: A Decade of Concurrency*, pages 530–582. Springer Lect. Notes Comp. Sci. (803), 1994.

[Rut95] J. Rutten. A calculus of transition systems (towards universal co-algebra). In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra, A Bisimulation Perspective*, CSLI Lecture Notes (53), pages 231–256. CSLI Publications, Stanford, 1995.

[Rut96] J. Rutten. Universal coalgebra: A theory of systems. Technical report, CWI, Amsterdam, 1996.

[Rut98] J. Rutten. Automata and coinduction (an exercise in coalgebra). In *Proc. CONCUR' 98*, pages 194–218. Springer Lect. Notes Comp. Sci. (1466), 1998.

[Rut00] J. Rutten. Universal coalgebra: A theory of systems. *Theor. Comp. Sci.*, 249(1):3–80, 2000. (Revised version of CWI Techn. Rep. CS-R9652, 1996).

[Rut01] J. Rutten. Elements of stream calculus (an extensive exercise in coinduction). Technical report, CWI, Amsterdam, 2001.

[SP82] M. Smyth and G. Plotkin. The category theoretic solution of recursive domain equations. *SIAM Journ. Comput.*, 4(11):761–783, 1982.

[Tar55] A. Tarski. A lattice–theoretic fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[TR98] D. Turi and J. Rutten. On the foundations of final coalgebra semantics: non-well-founded sets, partial orders, metric spaces. *Math. Struct. in Comp. Sci.*, 8(5):481–540, 1998.

[Wol00] U. Wolter. On corelations, cokernels and coequations. In H. Reichel, editor, *CMCS'00 - Workshop on Coalgebraic Methods in Computer Science*, pages 347–366. ENTCS, volume 33, Elsevier, 2000.

[Wor98] J. Worrell. A topos of hidden algebras. In *CMCS'98 - Workshop on Coalgebraic Methods in Computer Science, Lisbon*. ENTCS, volume 11, Elsevier, March 1998.