ELEMENTS OF THE GENERAL THEORY OF COALGEBRAS

PRELIMINARY VERSION

H. PETER GUMM

Abstract. Data Structures arising in programming are conveniently modeled by universal algebras. State based and object oriented systems may be described in the same way, but this requires that the state is explicitly modeled as a sort. From the viewpoint of the programmer, however, it is usually intended that the state should be "hidden" with only certain features accessible through attributes and methods. States should become equal, if no external observation may distinguish them.

It has recently been discovered that state based systems such as transition systems, automata, lazy data structures and objects give rise to structures dual to universal algebra, which are called coalgebras. Equality is replaced by indistinguishability and co-induction replaces induction as proof principle. However, as it turns out, one has to look at universal algebra from a more general perspective (using elementary category theoretic notions) before the dual concept is able to capture the relevant applications.

In this note we shall give an introduction to the fascinating theory of coalgebras which is still in an early phase of development. In contrast with the standard introduction by J.J.M.M. Rutten [Rut96], we try to develop the theory as far as possible without requiring the type functor to preserve weak pullbacks. It turns out that almost all relevant parts of the theory can in fact be obtained without any assumptions on the type functor. Even for the coalgebraic version of Birkhoff's theorem (originally proven under such extra hypotheses in [Gum98a, Gum99]), all we need to require is that the functor is bounded.

Contents

1. Preface

Universal Coalgebra as a unifying theory for describing state based and object oriented systems has received increasing attention in the last few years. To many people, including the author, who had previously worked with various kinds of transition system, automata, object oriented specification and lazy functional programming languages, the abstract theory of coalgebras came like a revelation. All of the basic notions and results that had seemed to reappear in all mentioned fields found a common and general explanation. Moreover, well known theories from topology and even analysis could be seen to fit the mold of universal coalgebra.

Coalgebras, as direct duals of universal algebras, had been considered more than 30 years ago. As sets A together with a family of mappings f_i from A to its n_i-fold sum $n_i \cdot A$, they did not receive much attention – a major reason being the lack of vital examples. Only when the right generalization was achieved, by defining a *coalgebra of type* F as a map $\alpha : A \to F(A)$, where F is an arbitrary endo-functor on the category of sets, the special theories of interest could all be seen to fit the framework. Amongst those theories are all the above mentioned and more. Even though many of them had also been tackled in a universal algebraic spirit, many phenomena had remained without universal algebraic explanation. The construction of the minimal automaton, the treatment of infinite data structures and proofs of observational equivalence in state based systems are just a few examples of phenomena that are now known to result from the general structure theory of coalgebras. Even though many data structures can be modelled both algebraically and co-algebraically, when it comes to nondeterministic behaviour, it is striking how easy and naturally this fits the coalgebraic framework.

The basic introduction to universal coalgebra has been the text by J. Rutten ([Rut96]). After discussing a convincing collection of examples, he develops the general theory, largely modeled after the development of universal algebra. An important observation is that most of the practically relevant functors, even the ones modeling nondeterministic systems, satisfy an extra condition: they *preserve weak pullbacks.* Assuming this property, a number of convenient structure theoretic results can be obtained. In particular, the important concept of "behaviour" can be equated with *one-generated* subcoalgebras. Therefore, in [Rut96] the theory is developed with this particular assumption on the functor F.

There are, however, reasons for not accepting any such condition on the functor F. One being that there are viable examples of coalgebras (e.g. topological spaces) whose type functors do not obey such a restriction. The other is that the structure theory obtained seems to be too special when compared to the universal algebraic dual. For instance, subcoalgebras are closed under both unions and intersections. Universal algebras with the corresponding property are extremely special, essentially their operations are unary.

Therefore, when we set out to teach a course on the subject of coalgebras in the spring of 1999, we tried to develop the theory as far as possible without assuming that the type functor preserves weak pullbacks. After some first encouraging results we started developing the present course notes for the South African Summer school LUATCS in Johannesburg. To our surprise, it turned out that basically all of the relevant theory can be developed for arbitrary functors, even including the Birkhoff theorem which states that covarieties are the same as behavioural classes. Here behaviours are elements of cofree coalgebras, representing the coalgebraic counterpart to "equations" on the universal algebra side.

In joint work with Tobias Schröder ([GS99a]) we have also identified structure theoretic facts that do, in fact, require particular weak limit preserving properties from the type functor F, see section 8. Another result from this cooperation is the short and elementary proof in subsection 7.4 of completeness of the category of all F-coalgebras ([GS99b]).

Much of this material can be put into an even more general framework by replacing the underlying category $\mathcal{S}et$ by some other category $\mathcal{C}$. The main tools we need from the category of sets are the diagram lemmas in subsection 3.3. The category minded reader will observe that they express the fact that epis are coequalizers and monos are equalizers. Discussions with Bart Jacobs and Wilfried Hodges during LUATCS also convinced me that most invocations of the axiom of choice can be disregarded, so that indeed $\mathcal{S}et$ could be replaced by any category with the mentioned condition on monos and epis. In fact, the final version of these notes might eventually use this more general framework, at the moment, however, we are leaning towards the view that the set theoretic development is more conducive to most readers. The category minded reader will have no difficulty translating the proofs into his language. All that is needed, therefore, from category theory, is the basic language up to and including no more than the elementary notions of *functor*, *limit* and *colimit*.

These notes are still preliminary. Many examples and exercises need to be worked out and many references need to be added. We strongly welcome criticism, remarks, suggestions and corrections.

2. Data Types, Systems and Models

In this introductory section we consider abstract data type specifications from the algebraic and from the coalgebraic viewpoint. Some coalgebraic phenomena are observed in the realm of functional programming and the notion of universal coalgebra is introduced.

2.1. Modeling Data Types as Universal Algebras.

2.1.1. *Examples.* In a functional programming language, a data type `BT` might have been introduced as follows:

```
BT =
    e   :  1         --> BT
    mkT :  BT x BT   --> BT.
```

It was the intention of the programmer to introduce a data type representing *binary trees*. Hence the choice of `BT` as a name for the new type and `e` signifying empty tree. Names, however, are irrelevant, so all we can say is that the notation introduces a universal algebra $BT = (BT; e, mkT)$ of type $(0, 2)$. The line

```
    mkT :  BT x BT   --> BT
```

says that `mkT` is a binary operation and the line

```
    e   :  1         --> BT
```

which is short for $e : BT^0 \to BT$, says that `e` is a constant operation, yielding a tree. (We use "1" to denote a prototypical 1-element set, e.g. $1 := \{\emptyset\}$. Thus, what the code states is:

"A `BT` is a groupoid[1] with a distinguished element."

By accepting definitions as above, the language provides an implementation of a type `BT`.

[1]A groupoid, in universal algebra terms, is just a set with a binary operation.

2.1.2. *Constructors.* There are many groupoids, and many groupoids with a distinguished element. Which one is meant - what is the “semantics” of such a definition ? Since there are only two operations resulting in a `BT`, the first reasonable agreement should be:

Axiom 2.1. *A data type should contain only elements that can be constructed using the operations used in its definition.*

In a type definition, the operations whose result is an object of the newly defined type are called *constructors.* Thus the axiom says that each element of the new type must be the result of a constructor operation.

This axiom throws out groupoids such as e.g. $(\mathbb{R}; 0, *)$ or $(\mathbb{N}; 0, +)$, but not $(\mathbb{N} - \{0\}; 1, +)$ or $(\mathbb{Z}_{(k)}; 1, +)$ and many more. In fact there is always a trivial model for such a definition, i.e. the one-element universal algebra, in this case $(\{1\}; 1, *)$. There is nothing useful about this groupoid, so we should rather search at the opposite end of the spectrum. Therefore, we require:

Axiom 2.2. *No two elements should be identified unless they are identically constructed.*

These two rules determine uniquely the free (0-generated) universal algebra of the type given by the constructors. In the case of `BT`, its elements are

$$\{\mathtt{e},\ \mathtt{mkT}(\mathtt{e},\mathtt{e}),\ \mathtt{mkT}(\mathtt{e},\mathtt{mkT}(\mathtt{e},\mathtt{e})),\ \mathtt{mkT}(\mathtt{mkT}(\mathtt{e},\mathtt{e}),\mathtt{e}),\ \mathtt{mkT}(\mathtt{mkT}(\mathtt{e},\mathtt{e}),\mathtt{mkT}(\mathtt{e},\mathtt{e})),\ \ldots\}$$

and the operations are *syntactical*, i.e. applying mkT to the syntactical objects e_1 and e_2 yields the syntactical object $\mathtt{mkT}(e_1,e_2)$

For specifications such as the one above, a freely generated universal algebra always exists, and it is unique, so we postulate:

Axiom 2.3. *A data type specification defines the freely generated universal algebra satisfying the specification.*

With this assumption, the following data type specifies exactly the natural numbers with the successor operation.

```
Nat =
    0         :  1       --> Nat
    succ      :  Nat     --> Nat.
```

More interesting are data types which can be used as containers for other data - they lead to many-sorted types, such as e.g. stacks or trees.

```
NatStack =
    empty    : 1                  --> NatStack
    push     : Nat x NatStack     --> NatStack.
```

For trees containing other data, we have several possibilities - storing data in leaves, in inner nodes, or in both. The latter could be done with the following type definition for a many sorted universal algebra:

```
Tree =
    emptyTree: 1                  --> Tree
    mkLeaf   : Nat                --> Tree
    mkNode   : Tree x Char x Tree --> Tree.
```

2.1.3. *Predicates.* Due to the axioms, for every data object there is precisely one way to construct it. Therefore we can derive Boolean operations - called *predicates* - to answer us for every data item whether it was constructed by a specific constructor. For instance, for the above type `Tree`, there are three predicates, corresponding to the three ways to construct a tree:

```
isEmpty  : Tree     --> Boolean
isLeaf   : Tree     --> Boolean
isNode   : Tree     --> Boolean.
```

Similarly, for `NatStack` we get the predicates

```
isEmpty  : NatStack --> Boolean
isPush   : NatStack --> Boolean.
```

Naturally, in the case of just two different constructors, one predicate will suffice, since the other one is the negation of the first. Thus, `isPush` = `not isEmpty` and similarly, for `Nat`, the predicate

```
isZero   : Nat      --> Boolean
```

is enough to determine whether a number is 0 or the result of constructor `succ`.

2.1.4. *Selectors.* What are trees and stacks worth, if we cannot look inside to see what has been stored or to take them apart again. Once we know by means of which constructor a piece of data has been constructed, we can determine the components out of which it has been made. They are unique, due to axiom 2.2. For every argument of a constructor, we therefore have a function yielding the corresponding component. E.g. for the constructor

```
mkNode   : Tree x Char x Tree --> Tree
```

we get three selectors:

```
left     :: Tree   --> Tree
node     :: Tree   --> Char
right    :: Tree   --> Tree
```

Of course, these are only partial operations, as indicated by the double colon "::". They are defined on all data objects constructed with the constructor `mkNode`, i.e.:

$$dom(\mathtt{left}) = dom(\mathtt{content}) = dom(\mathtt{right}) = \{t \in Tree \mid \mathtt{isNode}(t)\}.$$

There is also a selector associated with the constructor `mkLeaf`, i.e.

```
leaf     :: Tree   --> Nat
```

but no selector corresponding to `emptyTree`, as this constructor has no argument. Similarly, for `Nat` we get a single selector:

```
pred     :: Nat    --> Nat
```

with $dom(\mathtt{pred}) = \{n \in Nat \mid not\ \mathtt{isZero}(n)\}$ and for `NatStack`:

```
top : NatStack    --> Nat
pop : NatStack    --> NatStack
```

with $dom(\mathtt{top}) = dom(\mathtt{pop}) = \{s \in \mathtt{NatStack} \mid not\ \mathtt{isEmpty}(s)\}$.

Notice that there is a simple syntactical criterion to distinguish constructors and selectors: Constructors are arrows whose target is the newly constructed type, whereas selectors have the new type at their source. The dichotomy can be resolved by combining all constructors into one map. For this, let us denote by "+" the disjoint set union. For the tree example, we then have a single constructor

```
c : 1 + Nat + (Tree x Char x Tree) --> Tree,
```

and a single selector

```
s : Tree --> 1 + Nat + (Tree x Char x Tree),
```

which is precisely the inverse of the constructor. We observe that a constructor is a map

$$f : F(X) \to X$$

where $F(-)$ is some set theoretic construction whose argument is the type to be built. In the above case, $F(X) = \{0\} + Nat + (X \times Char \times X)$.

A selector is a map in the converse direction

$$\alpha : X \to F(X).$$

This will be seen to embody the coalgebraic approach to data.

2.1.5. *Recursive definitions.* Due to the unique construction of data objects, we can define further functions, using a case analysis on how the data type was constructed. A simple example is the function giving the length of a stack

```
length(empty)     = 0
length(push(n,s)) = 1 + length(s)
```

or the factorial function

```
fact(0)       = 1
fact(succ(n)) = succ(n) * fact(n).
```

Such definitions are complete, if the left side covers all possible constructors. Using *patterns* of the constructed data with variables at the component positions is a syntactical trick to avoid the explicit use of selectors. The above definition using argument patterns is just a shorthand for:

```
if isEmpty(u) then length(u) = 0
              else length(u) = 1 + length(pop(u)).
```

As an example of a function with several arguments, take

```
append(empty,s) = s
append(push(n,s1),s2) = push(n,append(s1,s2))
```

2.1.6. *Recursive functions as homomorphisms.* It is well known that the free 0-generated universal algebra $\mathcal{D}$ is *initial* in the class of all algebras of the same type. This means that for every algebra $\mathcal{A}$ of the same type there is exactly one homomorphism $\varphi : \mathcal{D} \to \mathcal{A}$. This fact can be used to define recursive functions simply by providing a data type of the same signature. This way, the function `length`, for instance, may be defined as the unique homomorphism into the algebra $\mathcal{A} = (A_0, A_1; 0, succ)$ with sorts $A_0 = \{0\} = 1$ and $A_1 = \mathbb{N}$ and operations

$$\begin{array}{lcll} 0 & : & 1 & \to \{0\} \\ succ & : & 1 \times \mathbb{N} & \to \mathbb{N}. \end{array}$$

2.1.7. *Induction.* An important property of the free (initial) algebra is that it does not have any proper subalgebra. This is a consequence of axiom 2.1. A subalgebra is a subset closed under all operations, so the fact that $\mathcal{D}$ has no subalgebra may be phrased:

"If P is a subset of D closed under all operations, then $P = D$".

For $\mathcal{D} = Nat$, this is the familiar induction principle, i.e.

$$\frac{P(0) \ \wedge \ \forall x \in \texttt{Nat}.[P(x) \implies P(\texttt{succ}(x))]}{\forall x \in \texttt{Nat}.P(x)}.$$

For `NatStack`, the induction principle is

$$\frac{P(\texttt{empty}), \forall n \in \texttt{Nat}.\forall s \in \texttt{NatStack}.[P(s) \implies P(\texttt{push}(n,s))]}{\forall s \in \texttt{NatStack}.P(s)}$$

2.1.8. *Proofs by induction.* A frequent use of the induction principle for a data structure is for showing properties of defined functions, in particular, equality of two defined functions. A simple example is showing that

$$length(append(s_1, s_2)) = length(s_1) + length(s_2),$$

where we assume the standard recursive definition of append:

```
append(empty,s) = s
append(push(n,s1),s2) = push(n,append(s1,s2)).
```

For fixed s consider the set

$$P = \{x \in NatStack \mid length(append(x,s)) = length(x) + length(s)\}.$$

Then $\texttt{empty} \in P$ and $[x \in P \implies \texttt{push}(n,x) \in P]$. Hence the induction rule says that $P = NatStack$, i.e. the proposed rule is true for every `NatStack`.

In general, given two homomorphisms $\varphi, \psi : \mathcal{A} \to \mathcal{B}$, their *equalizer*, the set $\{a \in A \mid \varphi(a) = \psi(a)\}$ is a subalgebra. Thus, the induction principle states the uniqueness of homomorphisms from the free algebra to any other algebra $\mathcal{B}$.

2.2. **Infinite Data Objects and State Based Systems.** Not all practically relevant data structures satisfy axiom 2.1. One example is the data type of *streams*. Streams are used to model continuous input, infinite files or sequences. They are useful in programming, for instance in combination with the UNIX pipe mechanism or as *lazily evaluated* data objects in functional programming languages. As an example, consider the following functional program. Here we use the symbol ":" as an infix operation symbol to denote inserting an element into a list. The program

```
ones    = 1 : ones
```

defines `ones` as the infinite list $[1, 1, 1, \dots]$, and

```
from n  = n : from(n+1)
int     = from(0)
```

defines `int` as the infinite list $[0, 1, 2, 3, \dots]$. Such objects are very useful data objects in programming and we can define functions in the same way as with finitely constructed data objects, for instance:

```
twice h:l = (2*h): twice l
```

and

```
    add (h1:l1) (h2:l2) = (h1+h2) : add l1 l2.
```

If we now try to prove by induction that $add(l,l) = twice(l)$ for every infinite list l, we cannot use structural induction over ":" as we would do for finite lists. The reason is that axiom 2.1 is not present, that is, the induction does not "bottom out".

2.2.1. *Coalgebraic types.* When trying to define a type "infinite stream", we have the problem that there is no nullary operation to get the construction of streams started. However, a definition using the selectors `head` and `tail` is possible:

```
S =
    h    :  S  --> Nat
    t    :  S  --> S,
```

or, if we combine the two selectors into one function, we get

```
    (h,t) :  S  --> Nat x S.
```

This is an example of a so called *system* or *co-datatype*. It is defined by selectors, which in this context will often be called *observers*. The intuition is that the components of selectors return attributes or components of selectors. In such a definition, it is often useful, to consider the defined type as a set of states. The selectors are then used to either find some attributes or to change the state.

This is exactly how infinite streams may be represented in a (finite) computer. A stream is represented by a state. On demand, the function `h` yields a number, which is considered the first element of the stream and the function `t` causes a transition to a new state, representing the tail of the stream. Still, it is not clear, why the above co-datatype definition should define $\mathbb{N}^\omega$, the set of all natural number streams with selectors *head* and *tail*, since there are other systems of the same functionality:

Example 2.1. *Let* $S = \{s_0, s_1, s_2, s_3, s_4\}$ *and define* $h(s_i) = i \ mod\ 2$, *and* $t(s_0) = t(s_2) = s_1, t(s_1) = t(s_3) = s_2, t(s_4) = s_4$.

$$\begin{tikzcd} s_0 \arrow[r] & s_1 \arrow[r, bend left] & s_2 \arrow[l, bend left] & s_3 \arrow[l] & s_4 \arrow[loop above] \end{tikzcd}$$

The figure shows the effect of the transition t on S.

Assume that the base set S is not directly observable, after all it is to be introduced by the type definition. Think of a black box with only the result of the selector h visible. The operation t may also be executed, it just leads to another state without any directly observable output. Jacobs and Rutten [JR97] interpret this as a machine with two buttons h and t. One yields a number and the other changes the internal state. The state is otherwise not observable.

Can we distinguish between any two states in the example? Clearly, s_0 and s_1 can be distinguished directly by the outcome of h. To distinguish s_0 from s_4 we need to first execute t, then h. One finds that s_0 and s_2 cannot be distinguished by the outcome of h whatever sequence of operations may have been performed.

As the states are not directly observable, we have no reason to distinguish two states such as s_0 and s_2 which display the same external behavior on identical sequences of tests. By identifying two states with the same external behavior, we get an extensional notion of equality, that can be captured by the following axiom:

Axiom 2.4. *Two states are considered equal if they cannot be distinguished by (a combination of) observations.*

Let us write $u \sim v$ if the states u and v are *indistinguishable*. It is easy to see that $\sim$ ought to satisfy

$$\frac{u \sim v}{h(u) = h(v) \wedge t(u) \sim t(v)}.$$

In the above example, $\sim$ is an equivalence relation and factoring yields a minimal system,

$$\begin{tikzcd} s'_0 \arrow[r, bend left] & s'_1 \arrow[l, bend left] & s'_4 \arrow[loop right] \end{tikzcd}$$

that is a system with a minimal state set which still exhibits the same external behavior.

2.2.2. *Automata.* As a second example consider a finite automaton. Its use is to decide, whether a given string satisfies a regular expression or not. An automaton is usually given by two functions:

```
isTerminal : State        --> Boolean
delta      : State x Char --> State
```

As it stands, this is neither an algebraic type nor a system. However, the second operation can be replaced by a function

$$\delta : \mathtt{State} \to \mathtt{State}^{\mathtt{Char}}$$

so that both operations are observers. Again, we can think of having two buttons. The first answers *true* or *false*, depending whether we are in a terminal state or not and the second one takes a character input and yields a corresponding new state.

To a user, again, the state may remain hidden, it is irrelevant, as long as the automaton implements the desired regular expression. Again, two states may be identified, if they behave the same way on the same input, which is to say, if they cannot be distinguished by any observation. An observation consists of entering a sequence of characters and then pressing the `isTerminal`-button. The indistinguishability relation for this data type is easily seen to be the same as the well known "Nerode congruence". Factoring by this congruence, we obtain the *minimal automaton.* Once more we can give a rule for the relation of indistinguishability:

$$\frac{s \sim t}{isTerminal(s) = isTerminal(t) \ \wedge \ \forall c \in Char.\delta(s,c) \sim \delta(t,c)}$$

Any relation satisfying this proof rule, that is any relation ρ satisfying

$$\frac{s \ \rho \ t}{isTerminal(s) = isTerminal(t) \ \wedge \ \forall c \in Char.\delta(s,c) \ \rho \ \delta(t,c)}$$

will be called a *bisimulation*, and two states u, v related by some bisimulation are called bisimilar. It is easy to see that the union of bisimulations is again a bisimulation, so there is always a largest bisimulation. We denote it by $\sim$.

2.2.3. *Proofs by Co-Induction.* The axioms for $\sim$ may be used as proof rules. If s_1 and s_2 are streams and if ρ is a relation on streams satisfying the proof rule then $s_1 \ \rho \ s_2$ implies that s_1 and s_2 are indistinguishable.

For instance, to prove that with the above definitions of stream operations we have $twice(s) = add(s, s)$, all we need to show is that $\rho = \{(twice(l), add(l, l)) \mid l \in Stream\}$ is a bisimulation. This is easy to check:

$$hd(twice(l)) = 2 * hd(l) = hd(l) + hd(l) = hd(add(s, s)), \ \ and$$

$$tl(twice(l)) = twice(tl(s)) \rho add(tl(s), tl(s)) = tl(add(s, s)).$$

Keep in mind that we have only proven that $twice(l)$ and $add(l, l)$ are indistinguishable.

2.2.4. *Final Semantics.* Given a coalgebraic type definition, defined by its selectors, say as

```
Str =
    h : S     --> Nat
    t : S     --> S
```

is there a universal system, playing a role similar to the initial algebra for algebraic types? In standard cases, the answer is yes, although, in more general circumstances the answer may be no – largely due to cardinality problems. Nevertheless, we can say what the *final coalgebra* must be made of, if it exists: It consists of the bisimilarity classes of all possible observations emanating from a certain state.

In the above case, the final coalgebra is given by the set of all infinite streams of natural numbers. This is because any possible behaviour of such a system can be encoded as an infinite stream and any two different streams can be distinguished. In the system of example 2.1 the behaviors of state s_0 and s_2 are represented by the streams $(01)^\omega$, the behaviour of s_1 and s_3 by $(10)^\omega$ and that of s_4 by $(0)^\omega$.

This relationship will be seen to determine the unique homomorphism from the system in example 2.1 to the final system consisting of the infinite streams of natural numbers, $\mathbb{N}^\omega$ with operations *head* and *tail*.

The viewpoint taken in the previous examples is opposite from the algebraic one. Instead of defining data objects by constructors and keeping them apart whenever they are differently constructed, we now are only interested in the observable behaviour of systems with some hidden state. We are given some observers and we identify two states if they cannot be distinguished by any observers. This might be called the co-algebraic view. Confronting the algebraic with the co-algebraic view, we can sum up:

	Algebraic Type	Coalgebraic Type
Data objects	constructions	observations
Equality	identical construction	indistinguishability
Proofs	induction	coinduction
Semantic domain	initial algebra	final coalgebra

2.3. **Algebras abstractly.** Traditionally, a *universal algebra* $\mathcal{A} = (A; (f_i)_{i \in I})$ is given by a set A and a collection of operations $f_i^A : A^{n_i} \to A$. The operations may be combined into a single map $f^A : \Sigma_{i \in I} A^{n_i} \to A$, so that in general a universal algebra is given by a set A and a map

$$f^A : F(A) \to A$$

where the "set theoretic construction" $F(X) = \Sigma_{i \in I} X^{n_i}$ determines the *similarity type* of $\mathcal{A}$. Given two algebras $\mathcal{A}$ and $\mathcal{B}$ of this same type, a map $\varphi : A \to B$, canonically induces a map $F(\varphi) : F(A) \to F(B)$. It is easy to check that φ is a homomorphism, just in case the following diagram commutes:

$$\begin{tikzcd} F(A) \arrow[r, "F(\varphi)"] \arrow[d, "f^{\mathcal{A}}"'] & F(B) \arrow[d, "f^{\mathcal{B}}"] \\ A \arrow[r, "\varphi"] & B \end{tikzcd}$$

2.4. **The concept of coalgebra.** A *co-algebraic type definition* is determined by a collection of selectors, which also may be combined into a single map

$$\alpha_{\mathcal{A}} : A \to F(A)$$

for some "set theoretic construction" F. Hence we define a *coalgebra of type* F to be a pair $\mathcal{A} = (A, \alpha_{\mathcal{A}})$ where $\alpha_{\mathcal{A}} : A \to F(A)$ is a map. A *homomorphism* between coalgebras $\mathcal{A} = (A, \alpha_{\mathcal{A}})$ and $\mathcal{B} = (B, \alpha_{\mathcal{B}})$ of type F will be a map $\varphi : A \to B$ for which the following diagram commutes:

$$\begin{tikzcd} A \arrow[r, "\varphi"] \arrow[d, "\alpha_A"'] & B \arrow[d, "\alpha_B"] \\ F(A) \arrow[r, "F(\varphi)"] & F(B) \end{tikzcd}$$

F has been termed vaguely as "some set theoretic construction". Firstly, we see that this construction must also act on maps. In particular it must transform a map $f : A \to B$ into a map $F(f) : F(A) \to F(B)$. If we want that id_A is always a homomorphism then we shall need to require:

$$F(id_A) = id_{F(A)}.$$

Similarly, if the composition of homomorphisms is to be a homomorphism, we need to require

$$F(f \circ g) = F(f) \circ F(g).$$

Such a "construction" will later be called a "functor". While in traditional universal algebra $F(X)$ is always of the form $\Sigma_{i \in I} X^{n_i}$ where these requirements are obviously satisfied, we shall need more general constructions in the case of coalgebras, as we have already seen with the type of automata. For automata we needed

$$F(X) = Boolean \times X^{Char},$$

and for nondeterministic automata we shall require

$$F(X) = Boolean \times \mathcal{P}(X)^{Char}$$

where $\mathcal{P}(X)$ denotes the power set $\{U \mid U \subseteq X\}$ of X. The right language to formulate the mentioned requirements on such a "construction" F and to study its properties is provided by *category theory*, so we shall have a brief look at the most fundamental notions of this mathematical discipline.

2.5. Exercises.

Exercise 2.1. *Let F be "some set theoretical construction" transforming any set X into another set $F(X)$ and any map $f : X \to Y$ into a map $F(f) : F(X) \to F(Y)$. Show that the following are equivalent:*

1. *For any coalgebras $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ of type F we have*
   (a) *id_A is a homomorphism, and*
   (b) *if $\varphi : \mathcal{A} \to \mathcal{B}$ and $\psi : \mathcal{B} \to \mathcal{C}$ are homomorphisms, then $\psi \circ \varphi : \mathcal{A} \to \mathcal{B}$ is a homomorphism*
2. *F is a "functor", that is it satisfies for all sets X, Y, and Z and all mappings $f : X \to Y$, $g : Y \to Z$:*
   (a) *$F(id_X) = id_{F(X)}$ and*
   (b) *$F(g \circ f) = F(g) \circ F(f)$*

Exercise 2.2 (Universal Algebra, abstractly)**.**

1. *Show that universal algebras $A = (A, (\cdot, ^{-1}, 1))$ of type $(2, 1, 0)$ correspond uniquely to maps*
$$f^A : (A \times A) + A + 1 \to A,$$
   where $\times$ denotes cartesian product, $+$ disjoint union and $1 = \{0\}$ a 1-element set.
2. *Show that F, defined on sets X as*
$$F(X) = (X \times X) + X + 1$$
   and on maps $g : X \to Y$ as
$$F(g)(w) = \begin{cases} (g(x_1), g(x_2)), & \text{if } w = (x_1, x_2) \text{ from the first component} \\ g(x), & \text{if } w = x \text{ from the second component} \\ 0, & \text{if } w{=}0\text{, from the third component} \end{cases}$$
   is a "functor" in the sense of exercise 2.1.
3. *Show that a map $\varphi : \mathcal{A} \to \mathcal{B}$ is a homomorphism between universal algebras $\mathcal{A} = (A; (\cdot, ^{-1}, 1))$ and $\mathcal{B} = (B; (\cdot, ^{-1}, 1))$, if and only if the following diagram, where f^A and f^B are defined as above, commutes:*
$$\begin{tikzcd} F(A) \arrow[r, "F(\varphi)"] \arrow[d, "f^A"] & F(B) \arrow[d, "f^B"] \\ A \arrow[r, "\varphi"'] & B \end{tikzcd}$$
4. *Formulate and prove the general theorem for arbitrary algebras $\mathcal{A} = (A; (f_i)_{i \in I})$ of type $(n_i)_{i \in I}$.*

3. Basic Notions of Category Theory

3.1. Categories. A category axiomatizes the abstract structural properties of sets and mappings between sets. Sets are considered as the *objects* and mappings are called the *morphisms* or *arrows* of the abstract *category of sets*. The language of category theory allows us to talk about arrows, their *sources* and *targets* and about their *composition* ($\circ$), of arrows, but not about the internal construction of sets and the nature of their elements. In particular, we cannot talk about the application "$f(x)$" of a map to an element of a set nor about the way $f(x)$ is evaluated. One might say that sets and arrows are considered atomic particles of category theory and everything that is to be said about sets and mappings must be expressed solely in terms of the notion of *composition*, *source* and *target*.

To every object A, the existence of a particular *identity* arrow id_A (sometimes written as 1_A) is postulated. Categorical language is too weak to axiomatize it using an equation such as e.g. "$id_A(x) = x$", for this refers to elements x *inside* the object A and to the *application* $f(x)$ of f to x. In categorical language rather, id_A must be characterized as an arrow satisfying:

- $source(id_A) = target(id_A) = A$
- for all morphisms f with $source(f) = A$ we have $f \circ id_A = f$, and
- for all morphisms g with $target(g) = A$ we have $id_A \circ g = g$.

Note that composition is to be read from right to left – in accordance with traditional mathematical habit.

Definition 3.1. *A* category $\mathcal{C}$ *consists of a class* $\mathcal{C}_o$ *of* objects $A, B, C, \ldots$ *and a class* $\mathcal{C}_m$ *of* morphisms *or* arrows $f, g, h, \ldots$ *between these objects together with the following operations:*

- $dom : \mathcal{C}_m \to \mathcal{C}_o$,
- $codom : \mathcal{C}_m \to \mathcal{C}_o$, *and*
- $id : \mathcal{C}_o \to \mathcal{C}_m$,

*associating with each arrow its source (*domain*), resp. its target (*codomain*), and with every object A its* identity arrow id_A. *Moreover there is a partial operation ($\circ$) of* composition *of arrows. Composition of f and g is defined whenever $codom(f) = dom(g)$. The result is a morphism $g \circ f$ with $dom(g \circ f) = dom(f)$ and $codom(g \circ f) = codom(g)$. The following laws have to be satisfied whenever the composition is defined:*

- $(h \circ g) \circ f = h \circ (g \circ f)$
- $id_A \circ f = f$ *and* $g = g \circ id_A$.

3.1.1. *Commutative Diagrams.* Many notions have their origin in the standard example, the category of sets and mappings, so we borrow notions, symbols and graphical visualizations from there. For instance, we write $f : A \to B$, if f is a morphism with $dom(f) = A$ and $codom(f) = B$. We use uppercase letters for objects and lower case letters for arrows.

It is convenient to draw objects as points and morphisms as arrows between these points. Such a representation is called a *diagram.* Often, compositions of arrows are not drawn - their presence is implied. A path of arrows represents the composition of the arrows involved. Whenever there are two different paths from an object A to an object B that enclose an area, it is often implied that their compositions are equal. One says that the diagram (or parts of it) *commutes.* To emphasize this, a circle is sometimes drawn inside the area whose bounding paths are assumed to commute.

Thus, associativity of composition and the property of the identity arrow can be expressed as commutativity of the following diagrams:

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, "g\circ f"'] & B \arrow[d, "g"] \arrow[dr, "h\circ g"] & \\
& C \arrow[r, "h"'] & d
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, "f"'] & B \arrow[d, "1_b"] \arrow[dr, "g"] & \\
& B \arrow[r, "g"'] & C
\end{tikzcd}
$$

3.1.2. *Examples.* The category $\mathcal{S}et$ whose objects are all sets and whose morphisms are all mappings between sets, forms the standard example of a category. However, there are other, less familiar examples that fulfill the definition of a category:

Example 3.2. *The category $\mathcal{R}g$ of rings, whose objects are all rings, and whose morphisms are all ring homomorphisms. Composition and identity are defined as in $\mathcal{S}et$.*

Example 3.3. *The category $\mathcal{R}el$ whose objects are sets and whose morphisms are binary relations between sets. Composition is relational product, i.e. if A, B, C are sets and $\rho : A \to B$ and $\sigma : B \to C$ morphisms, that is $\rho \subseteq A \times B$, $\sigma \subseteq B \times C$ then $\rho \circ \sigma = \{(a, c) \mid \exists b.(a, b) \in \rho, (b, c) \in \sigma\}$.*

Example 3.4. *A partially ordered set $\mathcal{P} = (P, \leq)$ can be viewed as a category $C_{\mathcal{P}}$, whose objects are the elements of P and whose morphisms are the pairs $(p, q) \in P \times P$ for which $p \leq q$. The composition is obtained from transitivity: $(p, q) \circ (q, r) = (p, r)$ and the identity morphism for $p \in P$ is (p, p).*

3.2. **Special morphisms.**

Definition 3.5. *Let $f : A \to B$ and $g : B \to A$ be morphisms. If $g \circ f = id_{\mathcal{A}}$ then g is called a* left inverse *of f and f is said to be* left-invertible. *In the same situation, g is called* right-invertible *with* right inverse *f.*

$f : A \to B$ is called an isomorphism *if f is both left and right invertible. It follows that the left-inverse and the right inverse of f agree, it is written f^{-1}.*

Two objects A and B in a category are called isomorphic *(we write $A \cong B$) if there exists an isomorphism $A \to B$.*

Isomorphy introduces an equivalence relation on the objects of a category. Isomorphic objects cannot be distinguished in the language of category theory – thus they are considered the same.

Definition 3.6. *Let $g : A \to B$ be a morphism. g is called* monomorphism, *(*mono *for short), if g is left cancellable, that is if for any other morphisms f_1 and f_2 we have $g \circ f_1 = g \circ f_2 \implies f_1 = f_2$. g is called* epimorphism *(or* epi*) if it is right cancellable, that is for any other morphisms h_1 and h_2 we have $h_1 \circ g = h_2 \circ g \implies h_1 = h_2$.*

We shall sometimes write $f : A \rightarrowtail B$ if f is mono and $g : A \twoheadrightarrow B$ if g is epi.

Lemma 3.7. *In every category we have:*

1. *every right-invertible morphism is epi*
2. *every left-invertible morphism is mono*
3. *compositions of monos are mono, compositions of epis are epi.*

Definition 3.8. *A* functor *F between categories $\mathcal{C}$ and $\mathcal{D}$ consists of two maps*

- *$F_o : \mathcal{C}_o \to \mathcal{D}_o$ between the objects, and*
- *$F_m : \mathcal{C}_m \to \mathcal{D}_m$ between the morphisms of a category,*

where F_m maps a morphism $f : A \to B$ from $\mathcal{C}_m$ to a morphism $F(f) : F(A) \to F(B)$ from $\mathcal{D}_m$ so that composition and identity are respected, that is:

- $F_m(g \circ f) = F_m(g) \circ F_m(f)$
- $F_m(id_A) = id_{F_o(A)}$.

Usually, the indices $_m$ and $_o$ to F are dropped. For our purposes it will be enough to consider functors between a category $\mathcal{C}$ and itself, such functors are called endofunctors.

Example 3.9. *The power set functor $\mathcal{P} : \mathcal{S}et \to \mathcal{S}et$ is given on objects as the power set construction*

$$\mathcal{P}(X) = 2^X = \{U \mid U \subseteq X\}$$

and on morphisms $f : X \to Y$ as

$$\mathcal{P}(f)(U) = f[U] := \{f(u) \mid u \in U\}.$$

Example 3.10. [AM89] *The functor $(-)^3_2 : \mathcal{S}et \to \mathcal{S}et$ is given on objects as*

$$A^3_2 = \{(a_1, a_2, a_3) \in A^3 \mid |\{a_1, a_2, a_3\}| \leq 2\},$$

and on maps $f : A \to B$ as

$$f^3_2(a_1, a_2, a_3) = (f(a_1), f(a_2), f(a_3)).$$

Example 3.11 (Hypersystems)**.** *$\bar{\mathcal{P}}(-)$ is defined on objects just like $2^{(-)}$, but a morphism $f : A \to B$ is assigned to $\bar{\mathcal{P}}(f) : 2^B \to 2^A$ with $\bar{\mathcal{P}}(f)(S) = f^{-1}[S]$. Unfortunately, $\bar{\mathcal{P}}$ is not really a functor, for the images of the morphisms point in the wrong direction. However, by iterating the construction, we obtain a functor $\bar{\mathcal{P}}\bar{\mathcal{P}} : \mathcal{S}et \to \mathcal{S}et$. An element of $\bar{\mathcal{P}}\bar{\mathcal{P}}(X)$ is a collection of subsets of X. A map $f : A \to B$ is assigned to $\bar{\mathcal{P}}\bar{\mathcal{P}}(f) : 2^{2^A} \to 2^{2^B}$. This maps a collection $\mathcal{G} \subseteq 2^A$ to*

$$\bar{\mathcal{P}}\bar{\mathcal{P}}(f)(\mathcal{G}) = \{V \subseteq B \mid f^{-1}[V] \in \mathcal{G}\}.$$

Example 3.12. *A* filter *on a set X is a collection $\mathcal{G}$ of subsets of X satisfying:*

1. $X \in \mathcal{G}$
2. $U_1, U_2 \in \mathcal{G} \implies U_1 \cap U_2 \in \mathcal{G}$
3. $U \in \mathcal{G}, U \subseteq V \subseteq X \implies V \in \mathcal{G}$.

The filter functor *$\mathcal{F}$ associates to every set X the set of all filters on X. For every map $f : X \to Y$, we let $\mathcal{F}(f) : \mathcal{F}(X) \to \mathcal{F}(Y)$ associate to every filter $\mathcal{G}$ on X the smallest filter on Y which contains all $f[U]$ with $U \in \mathcal{G}$.*

3.3. **The Category of Sets.** The category $\mathcal{S}et$ of sets has as objects all sets and as morphisms all mappings between sets. If $U \subseteq V$ we have the *natural inclusion map* $\subseteq^V_U : U \to V$ with domain U and codomain V mapping each element of U to itself. Whenever domain and codomain are clear from the context we drop the indices and use "hooked arrows" $U \hookrightarrow V$ to indicate the natural inclusion map.

3.3.1. *Epis and monos in $\mathcal{S}et$.*

Lemma 3.13. *In the category $\mathcal{S}et$ we have the following equivalences :*

1. *Monomorphisms are exactly the injective mappings.*
2. *Epimorphisms are exactly the surjective mappings.*
3. *Isomorphisms are exactly the bijective mappings.*

If $f : A \to B$ is a map and $U \subseteq A$, we denote by $f[U]$ the image of U under f that is $f[U] := \{f(x) \mid x \in U\}$. f may be decomposed into a surjective (i.e. epi) map $g : A \twoheadrightarrow f[A]$ and an injective (i.e. mono) map given by the natural inclusion $\subseteq: f[A] \hookrightarrow B$. We say that every morphism $f : A \to B$ in $\mathcal{S}et$ is *epi-mono-factorizable* as $f = h \circ g$ with g epi and h mono.

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, two heads, "g"'] & & B \\ & f[A] \arrow[ur, hook, "h"'] & \end{tikzcd}$$

If θ is an equivalence relation on the set A and if $a \in A$, then let

$$[a]\theta = \{x \in A \mid (x, a) \in \theta\}$$

be the *θ-class* of a. By A/θ we denote the *θ-factor* of A, i.e. $A/\theta = \{[a]\theta \mid a \in A\}$. $\pi_\theta : A \to A/\theta$ is the canonical projection defined as $\pi_\theta(a) = [a]\theta$. For any map $f : A \to B$ we define the *kernel of f* as

$$ker(f) = \{(x, y) \mid x, y \in A, f(x) = f(y)\}.$$

Lemma 3.14. *If $f : A \to B$ is an arbitrary mapping, then $ker(f)$ is an equivalence relation on A and $f[A] \cong A/ker(f)$.*

In the sequel we shall assume the axiom of choice. It can be formulated in the following way:

Axiom 3.1 (Axiom of Choice)**.** *Every epimorphism in $\mathcal{S}et$ is right-invertible.*

Thus epis and monos in $\mathcal{S}et$ can be characterized as:

Lemma 3.15. *For a morphism $f : A \to B$ in the category $\mathcal{S}et$ we have*

1. *f is epi $\iff$ f is right-invertible*
2. *f is mono $\iff$ f is left invertible or $A = \emptyset$.*

Corollary 3.16. *A functor $F : \mathcal{S}et \to \mathcal{S}et$ preserves epis and all monos with nonempty domain.*

In many relevant cases, functors $F : \mathcal{S}et \to \mathcal{S}et$ preserve inclusions, that is $F(\subseteq_U^V) = \subseteq_{F(U)}^{F(V)}$ whenever $U \subseteq V$. Such functors are called *standard*. Standard functors preserve *all* monos. An example of a functor $G : \mathcal{S}et \to \mathcal{S}et$ that is not standard is obtained by mapping the empty set to 2 (=$\{0,1\}$) and each nonempty set to 1 (=$\{0\}$). G will not preserve the monomorphism $\subseteq_\emptyset^A : \emptyset \to A$.

3.3.2. *Diagram Lemmas for $\mathcal{S}et$.*

Lemma 3.17 (First Diagram Lemma)**.** *Let $f : A \to B$ and $g : A \to C$ be mappings and $C \neq \emptyset$. There exists a map $h : B \to C$ with $h \circ f = g$ if and only if $ker(f) \subseteq ker(g)$. If f is epi then h is uniquely determined.*

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "g"'] & B \arrow[d, dashed, "h"] \\ & C \end{tikzcd}$$

The map h is said to *complete* the diagram given by f and by g. Dually, the completion of a diagram given by two arrows with identical target can similarly be characterized:

Lemma 3.18 (Second Diagram Lemma)**.** *Let $f : B \to A$ and $g : C \to A$ be maps. There exists a map $h : C \to B$ with $f \circ h = g$ if and only if $g[C] \subseteq f[B]$. If f is mono then h is uniquely determined.*

$$\begin{tikzcd} A & B \arrow[l, "f"'] \\ & C \arrow[ul, "g"] \arrow[u, dashed, "h"'] \end{tikzcd}$$

3.3.3. *Limits and Colimits.* Let $\mathcal{C}$ be a category and $\mathcal{D}$ a diagram in $\mathcal{C}$, that is $\mathcal{D}$ is a collection $(D_i)_{i \in I}$ of objects and a collection $(f_k)_{k \in K}$ of morphisms between the objects of $(D_i)_{i \in I}$.

Definition 3.19. *(Cone, Limit) Given a diagram $\mathcal{D}$, a* cone *over $\mathcal{D}$ will be a single object L together with morphisms $\pi_i : L \to D_i$ for each $i \in I$, so that for every arrow $f_k : D_i \to D_j$ we have $f_k \circ \pi_i = \pi_j$.*

A cone $(L, (\pi_i)_{i \in I})$ is called a weak limit *of $\mathcal{D}$, if for every other cone $(L', (\pi_i')_{i \in I})$ over $\mathcal{D}$ there is a morphism $\tau : L' \to L$ so that $\pi_i' = \pi_i \circ \tau$ for every $i \in I$. τ is sometimes called a* mediating morphism.

$L = (L, (\pi_i)_{i \in I})$ is called the limit *of $\mathcal{D}$, if L is a weak limit and the mediating morphism $\tau : L' \to L$ is always unique. The π_i are called the* canonical morphisms.

$$\begin{tikzcd} L' \arrow[rr, dashed, "\tau"] \arrow[dr, "\pi_i'"'] \arrow[drrr] & & L \arrow[dl] \arrow[dr, "\pi_j"] & \\ & D_i \arrow[rr, "f_k"'] & & D_j \end{tikzcd}$$

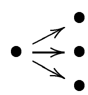
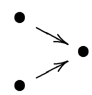
Colimits are defined *dual* to limits, that is the concept is the same when all arrows are reversed. To be precise, a *co-cone* $S = (S, (\varepsilon_i)_{i \in I})$ has arrows $\varepsilon : D_i \to S$. S is a *weak colimit* if for every competitor co-cone S' there is a mediating morphism $\sigma : S \to S'$ with $\sigma \circ \varepsilon_i = \varepsilon_i'$. S is the *colimit* of D, if for every competitor co-cone S' the mediating morphism σ is unique.

Limits and colimits, if they exist, are unique up to isomorphism. In the category $\mathcal{S}et$ all possible limits and colimits exist. $\mathcal{S}et$ is called *complete* and *cocomplete* for this reason.

Limits over a specific type of diagram often play important roles, so they are given particular names as listed in the following table. We shall consider some special types of diagrams.

diagram type	limit	colimit
empty	final object	initial object
no arrows	product	sum
source		pushout
sink	pullback	
parallel arrows	equalizer	co-equalizer

A diagram consisting of a collection of arrows, all with the same domain, is called a *source*. Its dual, a *sink*, is a collection of maps with common codomain. A source may also be considered as a cone over a diagram without arrows, a sink as a co-cone.

source :  *sink* : 

Commonly, the name "pullback" is only used for the limit of a sink with just two arrows, or of finitely many arrows – a similar remark applies to pushouts. We shall have reason to consider limits of arbitrary sinks. We are going to introduce the name κ-pullback for a pullback of κ many arrows. They are often referred to in the literature as "generalized pullbacks".

Lemma 3.20. *Arbitrary limits and colimits exist in the category $\mathcal{S}et$. In particular:*

- **initial object:** *The empty set $0 := \emptyset$ is initial in $\mathcal{S}et$. For every object A, the empty map $\emptyset$ is the unique morphism from 0 to A.*
- **final/terminal object:** *Each one-element set, e.g. $1 = \{0\}$, is terminal. For each object A, the constant map $\{(a, 0) \mid a \in A\}$ is the unique morphism from A to 1.*
- **sums:** $\Sigma_{i \in I} A_i = \bigcup_{i \in I}(A_i \times \{i\}) = \{(a, i) \mid a \in A_i, i \in I\}$.
  The canonical morphisms are the inclusions $e_i : A_i \to \Sigma_{i\in I} A_i$ with $e_i(a_i) = (a, i)$. If I is finite, e.g. $I = \{0, \ldots, n-1\}$, we shall write $\Sigma_{i<n} A_i = A_0 + \ldots + A_{n-1}$.
- **products:** $\Pi_{i\in I} A_i = \{(a_i)_{i\in I} \mid a_i \in A_i\}$.
  The canonical morphisms are the projections $\pi_j : \Pi_{i\in I} A_i \to A_j$ defined as $\pi_j((a_i)_{i\in I}) = a_j$ for all $j \in I$. Again, $\Pi_{i<n} A_i$ is written as $A_0 \times \ldots \times A_{n-1}$.
- **pullbacks, kernels :** *The pullback of two morphisms $f : A \to C$ and $g : B \to C$ is given by the object $pb(f, g) = \{(a, b) \in A \times B \mid f(a) = g(b)\}$ with the canonical morphisms $\pi_1 : pb(f, g) \to C$ and $\pi_2 : pb(f, g) \to C$ defined as $\pi_1((a, b)) = a$, $\pi_2((a, b)) = b$.*
  The pullback of a morphism $f : A \to C$ with itself is an equivalence relation which is also known as the kernel of f, i.e.: $ker(f) = \{(x, y) \mid x, y \in A, f(x) = f(y)\}$.
- **pushouts:** *The pushout of two morphisms $f : A \to B$ and $g : A \to C$ is given by the factor $(B + C)/\theta$, where θ is the smallest equivalence relation containing all pairs $(f(a), g(a))$ with $a \in A$. The morphisms are $p_B : B \to (B + C)/\theta$ with $p_B(b) = [e_B(b)]/\theta$ and p_C analogously.*
- **equalizer:** *The equalizer of two morphisms $f, g : A \to C$ is given by $eq(f, g) = \{a \in A | f(a) = g(a)\}$ together with the embedding map into A.*

- **coequalizer:** *The equalizer of two morphisms $f, g : A \to C$ is obtained as the obvious map $\pi_\theta : A \to A/\theta$ where θ is the smallest equivalence relation on C containing all pairs $(f(a), g(a))$.*

3.4. F-Coalgebras.

Definition 3.21. *A* type *is an endofunctor on $\mathcal{S}et$, i.e. a functor $F : \mathcal{S}et \to \mathcal{S}et$.*

We shall keep F fixed for the sequel. F is going to provide the *type* of the co-algebras we are about to discuss. Other names that have been used for a type F in this context are *type constructor* or *interface.*

Definition 3.22. *Let F be a type. A coalgebra of type F, also called F-*coalgebra *or F-system, is a pair $\mathcal{A} = (A, \alpha_\mathcal{A})$, consisting of a set A and a map $\alpha_\mathcal{A} : A \to F(A)$. A is called the* base set *(or* state set*) and $\alpha_\mathcal{A}$ is the* co-operation *(or* structure map*) on A.*

$$\begin{tikzcd} A \arrow[d, "\alpha_\mathcal{A}"'] \\ F(A) \end{tikzcd}$$

Definition 3.23. *Let $\mathcal{A} = (A, \alpha_\mathcal{A})$ and $\mathcal{B} = (B, \alpha_\mathcal{B})$ be F-coalgebras. A* homomorphism from $\mathcal{A}$ to $\mathcal{B}$ *is a map $\varphi : A \to B$, for which the following diagram commutes:*

$$\begin{tikzcd} A \arrow[r, "\varphi"] \arrow[d, "\alpha_\mathcal{A}"'] & B \arrow[d, "\alpha_\mathcal{B}"] \\ F(A) \arrow[r, "F(\varphi)"] & F(B) \end{tikzcd}$$

The identity map is always a homomorphism and the composition of two homomorphisms is again a homomorphism. This is an immediate consequence of the conditions defining a functor. Thus, for a given type F, the class of all F-coalgebras forms a category which we shall denote with $\mathcal{S}et_F$. Before studying this category, we shall collect a number of examples of coalgebras.

Example 3.24 (Self maps). *Let $F = \mathcal{I}d$ be the identity functor on $\mathcal{S}et$. An $\mathcal{I}d$-coalgebra is a set A together with a self map $\alpha : A \to A$. A homomorphism from $\mathcal{A} = (A, \alpha)$ to $\mathcal{B} = (B, \beta)$ is a map φ with $\varphi \circ \alpha = \beta \circ \varphi$.*

Example 3.25 (Sets). *Let $1 = \{0\}$ be a one-element set and $F : \mathcal{S}et \to \mathcal{S}et$ the* constant functor, *which maps every set X to the one-element set $1 := \{0\}$. For each morphism $f : X \to Y$ put $F(f) = id_1$. We shall denote this functor with 1. 1-coalgebras are simply sets and 1-homomorphisms are arbitrary set maps.*

Example 3.26 (Colorings). *Let Γ be a fixed set and F the constant functor with $F(X) = \Gamma$ and $F(f) = id_\Gamma$ for arbitrary sets X, Y and maps $f : X \to Y$. Interpreting Γ as a set of colors, an F-coalgebra $\mathcal{A}$ is a Γ-coloring of A and an F-homomorphism is a color preserving map.*

Example 3.27 (Partial maps). *Let $F = 1 + \mathcal{I}d$ be the functor assigning to a set X the set $1 + X$ and to every map $f : X \to Y$ the map $F(f) = id_1 + f$. $1 + \mathcal{I}d$-coalgebras correspond precisely to the* partial self maps $\tilde{\alpha}_A :: A \to A$. *A homomorphism is a map $\varphi : A \to B$ with*

1. $x \in dom(\tilde{\alpha}_A) \iff \varphi(x) \in dom(\tilde{\alpha}_B)$ *and*
2. $\varphi \circ \tilde{\alpha}_A = \tilde{\alpha}_B \circ \varphi$.

One often writes $x \uparrow$ if $\alpha(x) \in 1$, that is $x \notin dom(\tilde{\alpha})$. Otherwise the notion $x \downarrow$ is used. The first condition can now be written as

(1.') $x \downarrow \iff \varphi(x) \downarrow$.

Example 3.28 (Infinite Σ-lists). *For a fixed set Σ choose $F = \Sigma \times \mathcal{I}d$. An F-coalgebra is a set A, together with a pair of maps $\alpha_1 : A \to \Sigma$ and $\alpha_2 : A \to A$. An important example of such a $\Sigma \times \mathcal{I}d$-coalgebra is given by $A = \Sigma^\omega$, the set of all infinite Σ-sequences where the structure map is given by $\alpha_A(\sigma) = (head(\sigma), tail(\sigma))$.*

Example 3.29 (Finite and infinite Σ-lists). *Finite and infinite Σ-lists are modeled as a coalgebra of the functor $F = \{0\} + \Sigma \times \mathcal{I}d$. Let Σ^∞ be the set of all finite and infinite sequences of elements from Σ, i.e. $\Sigma^\infty = \Sigma^\omega \cup \Sigma^*$. The co-operation is defined as $\alpha(\epsilon) = 0$ for the empty sequence ϵ and as $\alpha(\sigma) = (head(\sigma), tail(\sigma))$ for a nonempty sequence σ.*

Example 3.30 (Trees). *All imaginable variants of trees may be modeled as coalgebras:*

1. Infinite binary trees : *Coalgebras of $\mathcal{I}d \times \mathcal{I}d$.*
2. Nonempty binary trees with leaves of type Σ : *Coalgebras of the functor $\Sigma + \mathcal{I}d \times \mathcal{I}d$.*
3. Binary trees whose leaves contain data of type Σ and whose nodes contain data of type Γ : *Coalgebras of the functor $1 + \Sigma + \mathcal{I}d \times \Gamma \times \mathcal{I}d$.*
4. Finitely branching trees : *Coalgebras for $\mathcal{I}d^*$. This functor associates to a set A the set A^* of all finite lists of elements from A. On maps $f : A \to B$ the functor is defined as $f^* : A^* \to B^*$ given by:*
   (a) $f^*(\epsilon) = \epsilon$
   (b) $f^*([a : rest]) = [f(a) : f^*(rest)]$.
5. Finitely and infinitely branching trees of finite or infinite depth with leaves of type Σ : *Coalgebras for the functor $\Sigma + \mathcal{I}d^\infty$. Here $\mathcal{I}d^\infty$ is the functor, associating to a set A the set A^∞ of all finite and infinite sequences.*

Example 3.31 (Bank account). *Assume that the methods* `deposit(`*amount*`)` *and* `showBalance` *are defined for a class* `bank account`*:*

`deposit` : $X \times \mathbb{R} \to X$
`showBalance` : $X \to \mathbb{R}$.

A bank account with these methods is a coalgebra for the functor $F(X) = X^{\mathbb{R}} \times \mathbb{R}$. The structure map is of the form $\alpha : X \to X^{\mathbb{R}} \times \mathbb{R}$ with $(\pi_1 \circ \alpha)(x)(r) = \mathtt{deposit}(x, r)$ and $\pi_2 \circ \alpha = \mathtt{showBalance}$.

Example 3.32 (Deterministic transition systems). *For a fixed set Σ let $F = (-)^\Sigma$ be the functor associating with a set A the set $F(A) = A^\Sigma = \{\sigma \mid \sigma : \Sigma \to A\}$. A map $f : A \to B$ is associated to $f^\Sigma : A^\Sigma \to B^\Sigma$ where $f^\Sigma(\sigma) = f \circ \sigma$. An F-Coalgebra is a map $\alpha_A : A \to A^\Sigma$. Such a map* *corresponds uniquely to a map $\delta_A : A \times \Sigma \to A$, that is to a* deterministic transition system*:*

$$\delta_A(a,s) = \alpha_A(a)(s).$$

An F-homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ is then a map with $\varphi(\delta_A(a,s)) = \delta_B(\varphi(a),s)$.

Example 3.33 (Deterministic automata)**.** *An automaton $\mathcal{A} = (A,\Sigma,\delta,T)$ with state set A, Alphabet Σ, transition function δ and terminal states T can be viewed as F-Coalgebra for the functor $F = (-)^\Sigma \times \mathbb{B}$ where $\mathbb{B} = \{true, false\}$. An F-coalgebra consists of a set A, which at the same time carries the structure of a $(-)^\Sigma$-coalgebra and of a $\mathbb{B}$-coalgebra. A homomorphism needs to preserve both these structures.*

Example 3.34 (Automata with output)**.** *An automaton with output is a structure $\mathcal{A} = (A,\Sigma,\Gamma,\delta,\gamma,T)$, where $(A,\Sigma,\delta,\gamma,T)$ as above is an automaton and $\gamma : A \to \Gamma$ is an output function. Automata with output may therefore be modeled as coalgebras for the functor $F = (-)^\Sigma \times \mathbb{B} \times \Gamma$.*

Example 3.35 (Relations)**.** *The power set functor $2^{(-)}$ associates with a set X its power set $2^X = \{S \mid S \subseteq X\}$ and with a map $f : A \to B$ the map $2^f : 2^A \to 2^B$ with $2^f(S) = f[S]$ for each $S \subseteq A$. A $2^{(-)}$-coalgebra is just a map $\alpha : A \to 2^A$. A homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ must satisfy: $\alpha_B(\varphi(a)) = \varphi[\alpha_A(a)]$.*

The maps $\alpha : A \to 2^A$ correspond uniquely to the maps $\chi : A \times A \to 2$. Those in turn correspond uniquely to the binary relations on A. Thus we can view a $2^{(-)}$ coalgebra as a binary relation and vice versa:

$$a_1 \ \rho_A \ a_2 \iff a_2 \in \alpha_A(a_1).$$

With this translation, we get the following conditions on homomorphisms $\varphi : A \to B$:

1. $a_1 \ \rho_A \ a_2 \implies \varphi(a_1) \ \rho_B \ \varphi(a_2)$, *and*
2. $\varphi(a) \ \rho_B \ b \implies \exists_{x \in A}.(a \ \rho_A \ x \wedge \varphi(x) = b)$

It is customary to represent transitions by arrows. We write

$$a_1 \longrightarrow a_2$$

for $a_1 \ \rho_A \ a_2$. We use the same type of arrow to represent ρ_A and ρ_B. It is always clear from the context which of these relations the arrow is to represent. In this notation the homomorphism conditions become:

1. $a_1 \longrightarrow a_2 \implies \varphi(a_1) \longrightarrow \varphi(a_2)$, *and*
2. $\varphi(a) \longrightarrow b \implies \exists_{x \in A}.(a \longrightarrow x \ \wedge \ \varphi(x) = b)$.

Example 3.36 (Nondeterministic transition systems)**.** *These are sets with a family of transitions, indexed by some set Σ. For each $e \in \Sigma$ the transition ρ_e is a binary relation, describing all permitted state transitions which the system may perform on input e. A nondeterministic transition system is therefore a system $\mathcal{T} = (S,(\rho_e)_{e \in \Sigma})$. It may be viewed as a coalgebra of the functor $\mathcal{P}(-)^\Sigma$. Again, we indicate transitions by arrows. This time we label them with e for each $e \in \Sigma$. Here the homomorphism conditions are for each $e \in \Sigma$:*

1. $a_1 \xrightarrow{e} a_2 \implies \varphi(a_1) \xrightarrow{e} \varphi(a_2)$, *und*
2. $\varphi(a) \xrightarrow{e} b \implies \exists_{x \in A}.(a \xrightarrow{e} x \ \wedge \ \varphi(x) = b)$.

Example 3.37 (Nondeterministic automata). *A nondeterministic automaton with alphabet Σ is a coalgebra of the functor $\mathcal{P}(-)^{\Sigma} \times \mathbb{B}$. We define the abbreviation*

$$a \downarrow \; :\Longleftrightarrow \quad \pi_2(\alpha(a)) = true,$$

then we must add to the homomorphism conditions for transition systems the condition

3. $a \downarrow \Longleftrightarrow \varphi(a) \downarrow$.

Example 3.38 (Hypersystems). *A* hypersystem *is a coalgebra of type $\bar{\mathcal{P}}\bar{\mathcal{P}}$. With every element $a \in A$ is associated a collection $\alpha(a)$ of subsets of A.*

Example 3.39 (Topological spaces). *Special examples of such hypersystems, in fact coalgebras for the filter functor, are obtained from topological spaces. If X is a topological space then the map*

$$x \mapsto \mathcal{U}(x),$$

which associates to every point x the system of all neighborhoods of X defines an $\mathcal{F}$ coalgebra structure on X where $\mathcal{F}$ is the filter functor (see example 3.12). Homomorphism between such structures are precisely the continuous open maps, see [Gum98b].

3.5. **The category of F-coalgebras.** In this subsection and in the rest of this paper, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ will denote coalgebras on base sets A, B, and C. φ and ψ will denote homomorphisms and f, g ordinary maps.

Lemma 3.40 ([Rut96]). *Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ coalgebras, $\varphi : \mathcal{A} \to \mathcal{C}$ a homomorphism, $f : A \to B$ and $g : B \to C$ maps with $\varphi = g \circ f$.*

$$\begin{tikzcd}
A \arrow[rr, "\varphi"] \arrow[dr, "f"'] & & C \\
& B \arrow[ur, "g"'] &
\end{tikzcd}$$

1. *If f is a surjective homomorphism, then g is a homomorphism.*
2. *If g is an injective homomorphism then f is a homomorphism.*

Proof. In the diagram below, the back square commutes, for φ is a homomorphism.

$$\begin{tikzcd}
A \arrow[rr, "\varphi"] \arrow[dr, "f"'] \arrow[dd, "\alpha_A"'] & & C \arrow[dd, "\alpha_C"] \\
& B \arrow[ur, "g"'] \arrow[dd, dashed, "\alpha_B"] & \\
F(A) \arrow[rr, "F(\varphi)"] \arrow[dr, "F(f)"'] & & F(C) \\
& F(B) \arrow[ur, "F(g)"'] &
\end{tikzcd}$$

Assume that f is a homomorphism, then the left front square commutes too. A diagram chase yields:

$$\begin{aligned} \alpha_C \circ g \circ f &= \alpha_C \circ \varphi \\ &= F(\varphi) \circ \alpha_A \\ &= F(g) \circ F(f) \circ \alpha_A \\ &= F(g) \circ \alpha_B \circ f. \end{aligned}$$

If f is surjective, we can cancel it on the right and we are done. For the second statement, we first consider the case when B is empty. Otherwise, corollary 3.16 shows that $F(g)$ is injective, hence also left cancellable. A diagram chase yields

$$F(g) \circ \alpha_S \circ f = F(g) \circ F(f) \circ \alpha_A,$$

so we cancel $F(g)$ to get the desired result. □

If a homomorphism φ is bijective, then it has an inverse map φ^{-1} with $id = \varphi^{-1} \circ \varphi$, so the lemma shows that φ^{-1} is in fact a homomorphism:

Corollary 3.41. *A bijective homomorphism is an isomorphism.*

Combining lemma 3.40 with the diagram lemmas in $\mathcal{S}et$ we get the corresponding diagram lemmas lifted to $\mathcal{S}et_F$:

Lemma 3.42 (First Diagram Lemma)**.** *Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C} \neq \emptyset$ be coalgebras, $\varphi : \mathcal{A} \to \mathcal{B}$ and $\psi : \mathcal{A} \to \mathcal{C}$ homomorphisms. There is a homomorphism $\chi : \mathcal{B} \to \mathcal{C}$ with $\chi \circ \varphi = \psi$ iff* $\ker(\varphi) \subseteq ker(\psi)$*. If φ is epi then χ is unique.*

$$\begin{tikzcd} A \arrow[r, two heads, "\varphi"] \arrow[dr, "\psi"'] & B \arrow[d, dashed, "\chi"] \\ & C \end{tikzcd}$$

Lemma 3.43 (Second Diagram Lemma)**.** *Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ be coalgebras, $\varphi : \mathcal{B} \to \mathcal{A}$ and $\psi : \mathcal{C} \to \mathcal{A}$ homomorphisms. There is a unique homomorphism $\chi : \mathcal{C} \to \mathcal{B}$ with $\varphi \circ \chi = \psi$ iff $\psi[C] \subseteq \varphi[B]$. If φ is mono then χ is uniquely determined.*

$$\begin{tikzcd} A & B \arrow[l, tail, "\varphi"'] \\ & C \arrow[ul, "\psi"] \arrow[u, dashed, "\chi"'] \end{tikzcd}$$

Given a coalgebra $\mathcal{A}$ and a set map $f : A \to S$ we shall use f to construct a coalgebra on the set S. We emphasize that the construction will *not* turn f into a homomorphism. Rather, whenever g is a map so that $g \circ f$ is a homomorphism, then g must be a homomorphism too.

Lemma 3.44 (Image Construction)**.** *Let $\mathcal{A}$ be a coalgebra, S a set and $f : A \twoheadrightarrow S$ an onto mapping. Then there exists a coalgebra structure α_S on S, so that for every coalgebra $\mathcal{B}$ and every map $g : S \to B$ we have: If $\varphi = g \circ f$ is a homomorphism, then so is g.*

$$
\begin{tikzcd}
A \arrow[rr, "\varphi"] \arrow[dr, "f"'] \arrow[dd, "\alpha_A"'] & & B \arrow[dd, "\alpha_B"] \\
& S \arrow[ur, "g"'] \arrow[dd, dashed, "\alpha_S"] & \\
F(A) \arrow[rr, "F(\varphi)"] \arrow[dr, "F(f)"'] & & F(B) \\
& F(S) \arrow[ur, "F(g)"'] &
\end{tikzcd}
$$

Proof. f is onto, hence it has a right-inverse f^- in $\mathcal{S}et$ with $f \circ f^- = id_S$. Define $\alpha_S = F(f) \circ \alpha_A \circ f^-$. Since f is epi in $\mathcal{S}et$, it is enough to show: $F(g) \circ \alpha_S \circ f = \alpha_B \circ g \circ f$.

$$
\begin{aligned}
F(g) \circ \alpha_S \circ f &= F(g) \circ F(f) \circ \alpha_A \circ f^- \circ f \\
&= F(\varphi) \circ \alpha_A \circ f^- \circ f \\
&= \alpha_B \circ \varphi \circ f^- \circ f \\
&= \alpha_B \circ g \circ f \circ f^- \circ f \\
&= \alpha_B \circ g \circ f.
\end{aligned}
$$

□

The following lemma is dual to the above. After a separate consideration of the case when $S = \emptyset$, the proof proceeds analogously:

Lemma 3.45 (Pre-image construction)**.** *Let $\mathcal{B}$ be a coalgebra, S a set and $g : S \rightarrowtail B$ an injective mapping. There is a coalgebra structure on S, so that for every coalgebra $\mathcal{A}$ and for every map $f : A \to S$ we have : If $\varphi = g \circ f$ is a homomorphism, then so is f.*

Lemma 3.46. *Let $\varphi : \mathcal{A} \to \mathcal{B}$ be a homomorphism of coalgebras. Let $\varphi = g \circ f$ be an epi-mono factorization of φ in $\mathcal{S}et$ with $f : A \twoheadrightarrow S$ and $g : S \rightarrowtail B$. Then there is a unique coalgebra structure γ on S so that both f and g are homomorphisms.*

Proof. On S we find structures γ, making f a homomorphism and a structure γ' making g a homomorphism. It suffices to show that necessarily $\gamma = \gamma'$.

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "\alpha_A"'] & S \arrow[r, "g"] \arrow[d, shift right, "\gamma"'] \arrow[d, shift left, "\gamma'"] & B \arrow[d, "\alpha_B"] \\
F(A) \arrow[r, "F(f)"] & F(S) \arrow[r, "F(g)"] & F(B)
\end{tikzcd}
$$

For $S = \emptyset$ the claim is trivial, otherwise we calculate:

$$
\begin{aligned}
F(g) \circ \gamma \circ f &= F(g) \circ F(f) \circ \alpha_A \\
&= F(g \circ f) \circ \alpha_A \\
&= \alpha_A \circ g \circ f \\
&= F(g) \circ \gamma' \circ f
\end{aligned}
$$

We can cancel $F(g)$ on the left and f on the right to obtain $\gamma = \gamma'$. $\square$

The following lemma shows that different epi-mono factorizations of the same morphism will produce isomorphic coalgebras, so we may speak of *the* epi-mono factorization of a given homomorphism:

Lemma 3.47. *Let* $A \overset{\varphi_1}{\twoheadrightarrow} S_1 \overset{\psi_1}{\rightarrowtail} B$ *and* $A \overset{\varphi_2}{\twoheadrightarrow} S_2 \overset{\psi_2}{\rightarrowtail} B$ *be epi-mono factorizations in* $\mathcal{S}et$ *of the same homomorphism. Then the coalgebras defined on* S_1 *and* S_2 *with respect to which all maps are homomorphisms are isomorphic.*

Proof. The First Diagram Lemma yields a homomorphism $\tau : S_1 \to S_2$ so that $\tau \circ \varphi_1 = \varphi_2$, the Second Diagram Lemma yields a homomorphism $\delta : S_1 \to S_2$ with $\psi_2 \circ \delta = \psi_1$.

$$\begin{tikzcd}
 & S_1 \arrow[dd, dotted, shift right, "\tau"'] \arrow[dd, dotted, shift left, "\delta"] \arrow[dr, "\psi_1"] & \\
A \arrow[ur, "\varphi_1"] \arrow[dr, "\varphi_2"'] & & B \\
 & S_2 \arrow[ur, "\psi_2"'] &
\end{tikzcd}$$

Clearly, τ must be onto and δ must be injective. A diagram chase yields

$$\begin{aligned}
\psi_2 \circ \tau \circ \varphi_1 &= \psi_2 \circ \varphi_2 \\
&= \psi_1 \circ \varphi_1 \\
&= \psi_2 \circ \delta \circ \varphi_1
\end{aligned}$$

We can cancel φ_1 on the right and ψ_2 on the left and get $\tau = \delta$, a bijective homomorphism, i.e. an isomorphism.

$\square$

Theorem 3.48. *Every homomorphism* $\varphi : \mathcal{A} \to \mathcal{B}$ *in* $\mathcal{S}et_F$ *has a unique epi-mono factorization in* $\mathcal{S}et_F$ *as* $\mathcal{A} \twoheadrightarrow \varphi[\mathcal{A}] \subseteq B$. *The coalgebra* $\varphi[\mathcal{A}]$ *is called the* image *of* φ.

3.6. **Exercises.**

Exercise 3.1 (Preimage construction). *Prove lemma 3.45.*

Exercise 3.2 (Sums in a category). *If in a category* $\mathcal{C}$ *the sum* $\Sigma_{i\in I}\ A_i$ *of a family of objects* $(A_i)_{i\in I}$ *exists, then the injections* $e_i : A_i \to \Sigma_{i\in I}\ A_i$ *are* jointly epi, *that is to say: If* $(\Sigma_{i\in I}\ A_i) \underset{g}{\overset{f}{\rightrightarrows}} B$ *and if for all* $i \in I$ *we have* $f \circ e_i = g \circ e_i$, *then* $f = g$.

Exercise 3.3 (Characterization of epis).

1. *Show that in an arbitrary category* $\mathcal{C}$ *a morphism* $f : A \to B$ *is* epi *if and only if the following diagram is a pushout:*

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "f"'] & B \arrow[d, "id_B"] \\
B \arrow[r, "id_B"'] & B
\end{tikzcd}$$

2. *Show that in every category $\mathcal{C}$ pushouts of epis are epi, i.e. if in the following pushout-diagram f is epi, then so is p_1.*

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "p_2"] \\ C \arrow[r, "p_1"'] & P \end{tikzcd}$$

3. *Show that in the category of Sets pushouts of monos are mono, that is, if in the above diagram f is mono, then so is p_1.*

Exercise 3.4. *Show that the arrows in a limit cone are* jointly mono*, that is: If $L = (L, (\pi_i)_{i\in I})$ is a limit, and if $f, g : A \to L$ are morphisms with $\pi_i \circ f = \pi_i \circ g$ for all $i \in I$, then $f = g$.*

Exercise 3.5. *Show that the first diagram lemma in $\mathcal{S}et$ is equivalent to the statement: "Every morphism in $\mathcal{S}et$ is a coequalizer".*

Exercise 3.6 (Homomorphisms)**.** *Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{N}^\omega$ the set of all countable sequences of natural numbers. Consider $F_\mathbb{N}$ defined on sets as*

$$F_\mathbb{N}(X) = \mathbb{N} \times X,$$

and for every map $f : X \to Y$ as

$$F_\mathbb{N}(f)(n, x) = (n, f(x)) \text{ for } x \in X, n \in \mathbb{N}.$$

1. *Show that $F_\mathbb{N}$ is a functor.*
2. *Show that $\mathbb{N}^\omega = (\mathbb{N}^\omega, (h \times t))$ where $(h \times t)((a_i)_{i\in\omega}) := (a_0, ((a_{i+1})_{i\in\omega}))$ and $\mathcal{N} = (\mathbb{N}, \alpha)$ with $\alpha(n) = (n, n+1)$ are $F_\mathbb{N}$-coalgebras.*
3. *Given $F_\mathbb{N}$-coalgebras $\mathcal{A} = (A, \alpha_A)$ and $\mathcal{B} = (B, \alpha_B)$, and a map $\varphi : A \to B$, when is φ a homomorphism?*
4. *Describe the unique homomorphism $\varphi : \mathcal{N} \to \mathbb{N}^\omega$ as a functional program.*
5. *Show that for every $F_\mathbb{N}$-coalgebra $\mathcal{A}$ there is precisely one homomorphism $\varphi : \mathcal{A} \to \mathbb{N}^\omega$.*

4. Derived structures

4.1. **Sums.** We start constructing the sum of two coalgebras $\mathcal{A}$ and $\mathcal{B}$. Consider the following diagram, where $A + B$ denotes the disjoint union of A and B, e_A and e_B the canonical embeddings:

$$\begin{tikzcd} A \arrow[r, "e_1"] \arrow[d, "\alpha_A"] & A + B \arrow[d, dashed, "\alpha_{A+B}"] & B \arrow[l, "e_2"'] \arrow[d, "\alpha_B"] \\ F(A) \arrow[r, "F(e_i)"] & F(A+B) & F(B) \arrow[l, "F(e_2)"'] \end{tikzcd}$$

The map α_{A+B} is uniquely given by the universal property of the sum $A+B$ in $\mathcal{S}et$. Thus there is a unique structure map α_{A+B} on $A + B$ turning e_1 and e_2 into homomorphisms.

The same construction works for an arbitrary family of coalgebras and the resulting coalgebra is in fact the sum in the category $\mathcal{S}et_F$([Rut96]):

Lemma 4.1. *For every family $(A_i, \alpha_i)_{i\in I}$ of coalgebras there exists the sum $\Sigma_{i\in I}\mathcal{A}_i$. Its carrier set is the disjoint union $\Sigma_{i\in I}A_i = \bigcup_{i\in I}\{(a,i) \mid a \in A_i\}$ of the A_i. Its structure map $\alpha : \Sigma_{i\in I}A_i \to F(\Sigma_{i\in I}A_i)$ is given by $\alpha(a,i) = F(e_i)(\alpha_i(a))$ where the $e_i : A_i \to \Sigma_{i\in I}A_i$ are the canonical embeddings given by $e_i(a) = (a,i)$*

Proof. First, α is constructed as before. It remains to show that the constructed coalgebra is indeed the sum of the (A_i, α_i) in the category $\mathcal{S}et_F$. Let $\mathcal{B}$ with morphisms $\varphi_i : \mathcal{A}_i \to \mathcal{B}$ be a "competitor" of the sum, then there is exactly one mapping $\psi : \Sigma_{i\in I}A_i \to B$ with $\psi \circ e_i = \varphi_i$. It remains to show that ψ is a homomorphism. In the following diagram all triangles and all squares of solid arrows commute:

$$\begin{tikzcd}
A_i \arrow[rr, "e_i"] \arrow[dr, "\varphi_i"] \arrow[dd, "\alpha_i"'] & & \Sigma A_i \arrow[dl, dashed, "\psi"] \arrow[dd, "\alpha"] \\
& B \arrow[dd, "\alpha_B"] & \\
F(A_i) \arrow[rr, "F(e_i)"] \arrow[dr, "F(\varphi_i)"'] & & F(\Sigma A_i) \arrow[dl, dashed, "F(\psi)"] \\
& F(B) &
\end{tikzcd}$$

If we consider the family of maps $F(\varphi_i) \circ \alpha_i : A_i \to F(B)$ there must exist precisely one map $h : \Sigma_{i\in I}A_i \to F(B)$ with $F(\varphi_i) \circ \alpha_i = h \circ e_i$. With $\alpha_B \circ \psi$ and $F(\psi) \circ \alpha$ we have two candidates for h, so they must be equal. Consequently, ψ is a homomorphism. ☐

4.2. **Colimits.** Sums are special colimits. Other important colimits are pushouts and coequalizers. All colimits in $\mathcal{S}et_F$ have an easy representation:

Theorem 4.2 ([Bar93])**.** *Every colimit exists in $\mathcal{S}et_F$. Base set and morphisms are identical with the corresponding colimit (in $\mathcal{S}et$) of the underlying sets.*

Let $\mathcal{D}$ be a diagram in $\mathcal{S}et_F$. Let C with maps $(\varepsilon_i)_{i\in I} : D_i \to C$ be the colimit of $\mathcal{D}$ in $\mathcal{S}et$. For every morphism $\varphi : \mathcal{D}_i \to \mathcal{D}_k$ in $\mathcal{D}$ we have $\varepsilon_k \circ \varphi = \varepsilon_i$. $F(C)$ with maps $F(\varepsilon_i) \circ \alpha_i$ is a competitor for the colimit C in $\mathcal{S}et$, thus there is a unique structure map $\alpha_C : C \to F(C)$ turning all ε_i into homomorphisms.

$$
\begin{tikzcd}
 & D_i \arrow[dl, "\varepsilon_i"'] \arrow[dr, dotted, "\varphi"] \arrow[dd, "\alpha_i"] & \\
C \arrow[dd, dashed, "\alpha_C"'] & & D_k \arrow[ll, "\varepsilon_k"] \arrow[dd, "\alpha_k"] \\
 & F(D_i) \arrow[dl, "F(\varepsilon_i)"'] \arrow[dr, dotted, "F(\varphi)"] & \\
F(C) & & F(D_k) \arrow[ll, "F(\varepsilon_k)"]
\end{tikzcd}
$$

The so constructed coalgebra $\mathcal{C} = (C, \alpha_{\mathcal{C}})$ together with homomorphisms ε_i is in fact the colimit of $\mathcal{D}$ in $\mathcal{S}et_F$. To see this, consider an object $\mathcal{E}$ with homomorphisms $\lambda_i : \mathcal{D}_i \to \mathcal{E}$ in $\mathcal{S}et_F$. The colimit property of C in $\mathcal{S}et$ yields a unique map $\tau : C \to E$ with $\tau \circ \varepsilon_i = \lambda_i$. Since $F(E)$ with maps $\alpha_E \circ \lambda_i$ is another co-cone over $\mathcal{D}$ and both $\alpha_E \circ \tau$ and $F(\tau) \circ \alpha_C$ are universal arrows from $\mathcal{C}$ to $F(E)$, they must be identical. This means that τ is a homomorphism.

$$
\begin{tikzcd}
 & D_i \arrow[rr, "\lambda_i"] \arrow[dl, "\varepsilon_i"'] \arrow[dr, dotted, "\varphi"] \arrow[dd, "\alpha_i"] & & E \arrow[dd, "\alpha_E"] \\
C \arrow[urrr, dashed, "\tau"] \arrow[dd, dashed, "\alpha_C"'] & & D_k \arrow[ll, "\varepsilon_k"] \arrow[ur, "\lambda_k"'] \arrow[dd, "\alpha_k"] & \\
 & F(D_i) \arrow[rr, "F(\lambda_i)"] \arrow[dl, "F(\varepsilon_i)"'] \arrow[dr, dotted] & & F(E) \\
F(C) \arrow[urrr, dashed, "F(\tau)"] & & F(D_k) \arrow[ll, "F(\varepsilon_k)"] \arrow[ur, "F(\lambda_k)"'] &
\end{tikzcd}
$$

4.2.1. *Pushouts.* One particular colimit will be of importance in the sequel. Let $\varphi_i : \mathcal{A} \to \mathcal{B}_i$ be homomorphisms for $i = 1, 2$. The colimit of this diagram is called the *pushout* of the φ_i. It can be constructed as follows: On the disjoint union $B_1 + B_2$ consider the smallest equivalence relation containing all pairs $(\varphi_1(a), \varphi_2(a))$ with $a \in A$. Now $P := (B_1 + B_2)/\theta$ together with the maps $e_i : B_i \to \mathcal{P}$ defined by $e_i(a) = [a]\theta$ is the pushout of φ_1 and φ_2.

4.2.2. *Coequalizers.* Coequalizers of two parallel homomorphisms Let $\varphi, \psi : \mathcal{A} \to \mathcal{B}$ are constructed similar to their pushout, except that the equivalence relation θ, generated by $\{(\varphi(a), \psi(a)) \mid a \in A\}$ is constructed directly on B, rather than on $B + B$. The pushout is, as in $\mathcal{S}et$, given by the map $\pi_\theta : B \to B/\theta$ with $\pi_\theta(b) = [b]\theta$.

4.3. **Substructures.**

Definition 4.3. *A coalgebra* $\mathcal{S} = (S, \alpha_{\mathcal{S}})$ *is called* subcoalgebra *(or* substructure*) of* $\mathcal{A} = (A, \alpha_{\mathcal{A}})$*, if* $S \subseteq A$ *and the canonical inclusion map* $S \hookrightarrow A$ *is a homomorphism. We write*

$$\mathcal{S} \leq \mathcal{A}$$

if $\mathcal{S}$ is a subcoalgebra of $\mathcal{A}$.

If S is any subset of the carrier set of the coalgebra $\mathcal{A}$, then a structure map $\alpha_S : S \to F(S)$ making (S, α_S) a subcoalgebra of $\mathcal{A}$ is required complete the following diagram:

$$\begin{tikzcd} S \arrow[r, "\subseteq"] \arrow[d, dashed, "\alpha_S"'] & A \arrow[d, "\alpha_A"] \\ F(S) \arrow[r, "F(\subseteq)"] & F(A) \end{tikzcd}$$

If $S = \emptyset$ then α_S is the empty map. Otherwise, $F(\subseteq)$ is injective (Corollary 3.16), so it follows from the Second Diagram Lemma for $\mathcal{S}et$ (3.18), that α_S, if it exists, is uniquely determined. Therefore, the structure map on a subcoalgebra is uniquely determined by its base set.

Still, not every subset of the carrier set of a coalgebra $\mathcal{A}$ qualifies as carrier set of a subcoalgebra. We therefore call a subset $S \subseteq A$ *closed* if there exists a structure map α_S so that $(S, \alpha_{\mathcal{S}}) \leq (A, \alpha_{\mathcal{A}})$.

The empty set $\emptyset$ is clearly closed. For $S \neq \emptyset$, we read from the above diagram, using the Second Diagram Lemma:

Lemma 4.4. *A subset S of a coalgebra $\mathcal{A} = (A, \alpha_A)$ is closed iff for every $s \in S$ there is some $u \in F(S)$ with $\alpha_A(s) = F(\subseteq_S^A)(u)$.*

In the case where the functor F is *standard* (see page 17), the criterion simplifies to

$$S \text{ is closed } \iff \alpha_A[S] \subseteq F(S).$$

Theorem 3.48 yields immediately:

Lemma 4.5. *A subset $S \subseteq A$ of the coalgebra $\mathcal{A}$ is closed if and only if there is a coalgebra $\mathcal{P}$ and a homomorphism $\varphi : \mathcal{P} \to \mathcal{A}$ with $S = \varphi[P]$.*

Since a closed set uniquely specifies a subcoalgebra, we shall often just use the term "subcoalgebra" in place of "closed subset". The following result is originally proven in [Rut96] for functors preserving weak pullbacks(see page 53). Using the above lemmas, we can prove it for arbitrary functors:

Lemma 4.6. *Let $\varphi : \mathcal{A} \to \mathcal{B}$ be a homomorphism and $\mathcal{S} \leq \mathcal{A}$ a subcoalgebra of $\mathcal{A}$. Then $\varphi[\mathcal{S}]$, the image of $\mathcal{S}$ under φ is a subcoalgebra of $\mathcal{B}$.*

Proof. The epi-mono factorization of the homomorphism $\varphi \circ \leq$ in $\mathcal{S}et$ as $\subseteq \circ \varphi_{\restriction S}$ together with theorem 3.48 yields that $\varphi[S]$ is a subcoalgebra of $\mathcal{B}$.

$$\begin{tikzcd} A \arrow[r, "\varphi"] & B \\ S \arrow[u, hook, "\leq"] \arrow[r, "\varphi_{\restriction S}"] & \varphi[S] \arrow[u, hook, "\leq"'] \end{tikzcd}$$

$\square$

If $\varphi : \mathcal{A} \to \mathcal{B}$ is an injective homomorphism, then $\mathcal{A} \cong \varphi[\mathcal{A}]$, thus $\mathcal{A}$ is isomorphic to a subcoalgebra of $\mathcal{B}$. Since we don't distinguish between isomorphic structures we shall simply say that $\mathcal{A}$ *is* a subcoalgebra of $\mathcal{B}$. An injective homomorphisms is therefore called an *embedding.* In particular, in a sum $\Sigma_{i \in I} \mathcal{A}_i$ every summand $\mathcal{A}_i$ is a subcoalgebra of the sum. The fact that sums in $\mathcal{S}et$ and in $\mathcal{S}et_F$ agree is responsible for the following important fact:

Theorem 4.7. *The union of a family of subcoalgebras is a subcoalgebra.*

Proof. Let $(\mathcal{S}_i)_{i \in I}$ be a family of subcoalgebras. From the sum $\Sigma_{i \in I} \mathcal{S}_i$ there is a unique homomorphism φ to $\mathcal{A}$ so that $\varphi \circ e_i \ = \leq_i$. But $\Sigma_{i \in I} \mathcal{S}_i$ is at the same time the sum of the S_i in $\mathcal{S}et$, so φ, as a map, must agree with the unique map in $\mathcal{S}et$ with the above equations, that is $\varphi((s,i)) \ = \leq_i (s)$. Hence $\varphi[S] = \cup_{i \in I} S_i$.

$$\begin{tikzcd}
\Sigma S_i \arrow[r] \arrow[dr, "\varphi"] & \cup S_i \arrow[d, hook] \\
S_i \arrow[u, "e_i"] \arrow[r, hook, "\leq_i"'] & A
\end{tikzcd}$$

□

Corollary 4.8. *Let $\mathcal{A}$ be a coalgebra and S a subset of A. Then there is a largest subcoalgebra of $\mathcal{A}$ contained in S. It is denoted by $[S]$ and called the subcoalgebra of $\mathcal{A}$* co-generated *by S.*

Corollary 4.9. *The substructures of a coalgebra $\mathcal{A}$ form a complete lattice $Sub(\mathcal{A})$. For a family $(S_i)_{i \in I}$ of subcoalgebras of $\mathcal{A}$ their supremum $\bigvee S_i$ and their infimum $\bigwedge S_i$ exist and*
$\bigvee S_i = \bigcup S_i$
$\bigwedge S_i = [\bigcap S_i]$.

4.3.1. *Glued sums.* Let $\mathcal{A}$ and $\mathcal{B}$ be coalgebras with a common subcoalgebra $\mathcal{S}$. We form the pushout $\mathcal{A} +_{\mathcal{S}} \mathcal{B}$ of the embeddings $e_1 : \mathcal{S} \to \mathcal{A}$ and $e_2 : \mathcal{S} \to \mathcal{B}$. It is easy to see that the pushout in $\mathcal{S}et$ can be formed by "glueing" a copy of A and of B over the common subset S, that is $A +_S B = (A - S) + S + (B - S)$. Theorem 4.2 implies that the maps $p_1 : A \to A +_S B$ and $p_2 : B \to A +_S B$ are embeddings. Thus $\mathcal{A} +_S \mathcal{B}$ is a coalgebra with subcoalgebras $\mathcal{A}$ and $\mathcal{B}$, whose union is $A +_S B$ and whose intersection is S.

4.4. **Homomorphic images, congruences, factors.**

Definition 4.10. *A coalgebra $\mathcal{B}$ is called a* homomorphic image *of a coalgebra $\mathcal{A}$, if there is a surjective homomorphism $\varphi : \mathcal{A} \twoheadrightarrow \mathcal{B}$.*

We have seen that the unique epi-mono factorization of an arbitrary homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ yields a unique homomorphic image $\varphi[\mathcal{A}]$ which is at the same time a subcoalgebra of $\mathcal{B}$.

In the category $\mathcal{S}et$, the homomorphic image $f[S]$ of S can be obtained by factoring S by the kernel of f, i.e. $f[S] \cong S/ker(f)$, see lemma 3.14. Kernels of coalgebra homomorphisms are special equivalence relations. We define:

Definition 4.11. *An equivalence relation θ on A is called a* congruence relation, *if it is the kernel of a homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ for some coalgebra $\mathcal{B}$.*

Lemma 4.12. *For an equivalence relation θ on a coalgebra $\mathcal{A}$, the following are equivalent:*

1. *θ is a congruence relation*
2. *There is a (unique) structure α_θ on $\mathcal{A}/\theta$ for which $\pi_\theta : A \to A/\theta$ defined as $\pi_\theta(a) := [a]\theta$ is a homomorphism.*
3. *$\theta \subseteq ker(F(\pi_\theta) \circ \alpha_A)$*

Proof. Let θ be the kernel of $\varphi : \mathcal{A} \to \mathcal{B}$. Factor φ into $\tilde{\varphi} : \mathcal{A} \twoheadrightarrow \varphi[\mathcal{A}] \leq \mathcal{B}$. There is a bijection $f : A/\theta \to \varphi[A]$. With the image construction (lemma 3.44) we equip A/θ with a (unique) structure so that f becomes an isomorphism. Hence $\pi_\theta = f^{-1} \circ \tilde{\varphi}$ is an epimorphism.

Similarly, all other claims are direct consequences of the first diagram lemma 3.42. ☐

Another direct consequence of the First Diagram Lemma is:

Lemma 4.13. *If θ and ϕ are congruences, then $\theta \subseteq \phi$ iff there exists a homomorphism $A/\theta \to A/\phi$ with $\varphi \circ \pi_\theta = \pi_\phi$.*

Definition 4.14. *Let θ be a congruence relation on the coalgebra $\mathcal{A}$. The factor A/θ together with the unique structure map α_θ is called the* factor coalgebra *$\mathcal{A}/\theta$. The surjective homomorphism $\pi_\theta : A \to A/\theta$ is called the* canonical projection *from $\mathcal{A}$ onto $\mathcal{A}/\theta$. The (unique) homomorphism of the previous lemma is usually denoted as $\pi_{\theta/\phi}$.*

Theorem 4.15 (Isomorphism Theorem)**.** *Every homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ can be decomposed in $\mathcal{S}et_F$ as $\varphi = \leq \circ\, \psi \circ\, \pi_\theta$ where $\theta = ker(\varphi)$ and ψ is an isomorphism. In particular, every homomorphic image is isomorphic to a factor.*

$$\begin{tikzcd}
\mathcal{A} \arrow[rr, "\varphi"] \arrow[dr, "\varphi'"] \arrow[ddr, bend right, "\pi_{ker(\varphi)}"'] & & \mathcal{B} \\
& f[A] \arrow[ur, "\leq"] & \\
& \mathcal{A}/ker(\varphi) \arrow[u, "\cong"'] \arrow[uur, bend right, "\mu"'] &
\end{tikzcd}$$

Thus we can get an overview of all possible homomorphic images of a coalgebra $\mathcal{A}$ by determining all congruences on $\mathcal{A}$, using e.g. lemma 4.12.

The set of all congruence relations on a coalgebra $\mathcal{A}$ is ordered by set inclusion. In fact we get:

Lemma 4.16. *Let $(\theta_i)_{i \in I}$ be a nonempty family of congruences on $\mathcal{A}$. Then the supremum of the θ_i exists and it is given by*

$$\bigvee_{i \in I} \theta_i := (\bigcup_{i \in I} \theta_i)^*,$$

the transitive closure of the union of all θ_i.

Proof. $\Phi := (\bigcup_{i \in I} \theta_i)^*$ is the smallest equivalence relation containing all θ_i. The canonical maps $\pi_{\theta_i/\Phi} : A/\theta_i \to A/\Phi$ constitute the pushout of the π_{θ_i} in $\mathcal{S}et$. Since the π_{θ_i} are homomorphisms, the pushout is the same in $\mathcal{S}et_F$, so $\Psi = \ker(\pi_{\theta_i/\Phi} \circ \pi_{\theta_i})$ is a congruence too. It is easy to check that it is, in fact, the supremum. ☐

The supremum over the empty index set, i.e. the smallest congruence, exists too, it is $\Delta_A = ker(id_{\mathcal{A}})$. For any reflexive relation $R \subseteq A \times A$ we can therefore form the supremum over all congruence relations θ_i contained in R. It will be called the *congruence co-generated by* R and denoted $Con[R]$. We now have:

Theorem 4.17. *The set of all congruences on a coalgebra $\mathcal{A}$ is a complete lattice. The supremum is given by*
$\bigvee_{i \in I} \theta_i = (\bigcup_{i \in I} \theta_i)^*$, *and the infimum by*
$\bigwedge_{i \in I} \theta_i = Con[\bigcap_{i \in I} \theta_i)]$.

The smallest element of the lattice of all congruences on $\mathcal{A}$ is trivial, i.e. Δ_A. The largest element, however, will in general be a proper subset of $A \times A$. Its factor has the following uniqueness property:

Theorem 4.18. *If θ is the largest congruence on $\mathcal{A}$ then for every coalgebra $\mathcal{B}$ there is at most one homomorphism $\varphi : B \to A/\theta$.*

Proof. Assume there were two different homomorphisms $\varphi_1, \varphi_2 : \mathcal{B} \to \mathcal{A}/\theta$. Let $\psi : \mathcal{A}/\theta \to \mathcal{C}$ be their coequalizer. Then the kernel of $(\psi \circ \pi_\theta) : A \to C$ properly contains θ. ☐

4.5. **Exercises.**

Exercise 4.1. *Let $\phi : \mathcal{A} \to \mathcal{B}$ be a homomorphism that is not onto. Find two homomorphisms ψ_1 and ψ_2 with $\psi_1 \circ \varphi = \psi_1 \circ \varphi$ but $\psi_1 \neq \psi_2$.*

5. Bisimulations and κ-simulations

Relations compatible with the coalgebra structure are called *bisimulations*. The name goes back to the special case of transition systems. If a state s_1 of a transition system S_1 can simulate a state s_2 in another transition system S_2 and vice versa, then the pair (s_1, s_2) is called *bisimilar*. A *bisimulation* is defined as a particular collection of such bisimilar pairs:

Definition 5.1. *Let $\mathcal{A}$ and $\mathcal{B}$ be coalgebras. A relation $R \subseteq A \times B$ is called a* bisimulation between $\mathcal{A}$ and $\mathcal{B}$, *if there exists a structure map $\gamma : R \to F(R)$ so that the projections $\pi_1 : R \to A$ and $\pi_2 : R \to B$ are homomorphisms with respect to the structure (R, γ). A* bisimulation on $\mathcal{A}$ *is a bisimulation between $\mathcal{A}$ and $\mathcal{A}$.*

Thus, a bisimulation between $\mathcal{A}$ and $\mathcal{B}$ is a binary relation $R \subseteq A \times B$, for which there exists a map γ, making the following diagram commutative:

$$\begin{tikzcd} A \arrow[d, "\alpha_A"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, dashed, "\gamma"] & B \arrow[d, "\alpha_B"] \\ F(A) & F(R) \arrow[l, "F(\pi_1)"'] \arrow[r, "F(\pi_2)"] & F(B) \end{tikzcd}$$

5.0.1. *κ-sources and κ-simulations.* A bisimulation R between $\mathcal{A}$ and $\mathcal{B}$ gives rise to a coalgebra $\mathcal{R}$ and two homomorphisms $\mathcal{A} \xleftarrow{\varphi_1} \mathcal{R} \xrightarrow{\varphi_2} \mathcal{B}$. Such a diagram will be called a *2-source.* In general we define for any ordinal κ:

Definition 5.2. *A* κ-source *is a coalgebra $\mathcal{P}$ together with a family $(\varphi_k)_{k\in\kappa}$ of homomorphisms $\varphi_k : \mathcal{P} \to \mathcal{A}_k$. A* κ-simulation *between coalgebras $(\mathcal{A}_k)_{k\in\kappa}$ is a subset $R \subseteq \Pi_{k\in\kappa} A_k$ of the cartesian product of the A_k, on which a coalgebra structure can be defined so that all projections $\pi_k : R \to A_k$ become homomorphisms.*

Clearly, 2-simulations are just bisimulations. Slightly more interesting is the observation that 1-simulations are just closed subsets (i.e. subcoalgebras). Therefore we might consider bisimulations as 2-dimensional versions of subcoalgebras. Indeed, the relevant properties found for subcoalgebras carry through, in particular:

Theorem 5.3. *Let $(\mathcal{P}, (\varphi_k)_{k\in\kappa})$ be a κ-source, then $\Pi\varphi_k[P] := \{(\varphi_k(p))_{k\in\kappa} \mid p \in P\}$ is a κ-simulation.*

The set $\Pi\varphi_k[P]$ from the above theorem will be called the *canonical κ-simulation* for the κ-source $(\mathcal{P}, (\varphi_k)_{k\in\kappa})$.

Proof. In the category $\mathcal{S}et$, the product of the sets A_k exist, it is the cartesian product $\Pi_{k\in\kappa} A_k$ together with the canonical projections π_k. The κ-source is a competitor of this product, and $\Pi\varphi_k : P \to \Pi A_k$ is the unique mediating map into the product. $\Pi\varphi_k[P]$ is just the image of this map. By the image construction (lemma 3.44) we find a coalgebra structure γ on this image so that all projections become homomorphisms, i.e. $\Pi\varphi_k[P]$ is a κ-simulation.

$$\begin{tikzcd}
P \arrow[r, "\Pi\varphi_k"] \arrow[dr, "\varphi_k"'] & \Pi\varphi_k[P] \arrow[r, hook, "\subseteq"] \arrow[d, dashed, "\pi_k"] & \Pi_k A_k \arrow[dl] \\
& A_k &
\end{tikzcd}$$

□

Theorem 5.4. *A subset $S \subseteq \Pi A_k$ is a κ-simulation if and only if there is a κ-source $(P, (\varphi_k)_{k\in\kappa})$ with $S = (\Pi\varphi_k)[P]$.*

Corollary 5.5 ([Rut96]). *If R is a bisimulation between $\mathcal{A}$ and $\mathcal{B}$ then R^- is a bisimulation between $\mathcal{B}$ and $\mathcal{A}$.*

The above theorem is the κ-dimensional analog of lemma 4.5. The corresponding analog of theorem 4.7 is:

Theorem 5.6. *The union of a family of κ-simulations is a κ-simulation.*

Proof. Let $R_i \subseteq \Pi_{k\in\kappa} A_k$ be a κ-simulation for each $i \in I$. Then each R_i can be given a coalgebra structure so that $(\mathcal{R}_i, (\pi_k^i)_{k\in\kappa})$ is a κ-source. Consider the sum of these coalgebras, $\Sigma_{i\in I}\mathcal{R}_i$. For every $k \in \kappa$ there is a unique homomorphism $\varphi_k : \Sigma_{i\in I}\mathcal{R}_i \to \mathcal{A}_k$ with $\varphi_k \circ e_i = \pi_k^i$ for every $i \in I$. From the sum $\Sigma_{i\in I} R_i$ to the product (in $\mathcal{S}et$) $\Pi_{k\in\kappa} A_k$ we have firstly the product map $\psi := \Pi_{k\in\kappa}\varphi_k$, whose image is a κ-simulation by theorem 5.3 and secondly the sum map $\sigma = \Sigma_{i\in I} \subseteq_i$ whose image is the union $\cup_{i\in I} R_i$ of the R_i. All that is left to do is showing the equality of these two maps:

$$\begin{tikzcd}
\Sigma R_i \arrow[r] \arrow[dr, shift left, "\psi"] \arrow[dr, shift right, "\sigma"'] \arrow[dd, bend right=60, "\varphi_k"'] & \bullet \arrow[d, hook, "\subseteq"] \\
R_i \arrow[u, "e_i"] \arrow[r, hook, "\subseteq_i"'] \arrow[d, "\pi_k^i"] & \Pi A_k \arrow[dl, "\pi_k"] \\
A_k &
\end{tikzcd}$$

For every $i \in I$ and every $k \in \kappa$ we have:

$$\begin{aligned}
\pi_k \circ \psi \circ e_i &= \varphi_k \circ e_i \\
&= \pi_k^i \\
&= \pi_k \circ \subseteq_i \\
&= \pi_k \circ \sigma \circ e_i.
\end{aligned}$$

We are done now, since we can cancel the π_k-s on the left (see exercise 3.4) and the e_i-s on the right. $\square$

Corollary 5.7. *Let $(\mathcal{A}_i)_{i<\kappa}$ be a family of coalgebras and $S \subseteq \Pi_{i<\kappa} A_i$. There is a largest κ-simulation contained in S. We call it the κ-simulation* cogenerated *by S and denote it as $[S]$.*

Since the empty set is always a κ-simulation, and the set of all κ-simulations is union-closed, we get:

Corollary 5.8. *For a family $(\mathcal{A}_k)_{k\in\kappa}$ of coalgebras the set of all κ-simulations between the $(\mathcal{A}_k)_{k\in\kappa}$ forms a complete lattice with*
$\bigvee_{i\in I} R_i = \bigcup_{i\in I} R_i$, *and*
$\bigwedge_{i\in I} R_i = [\bigcap_{i\in I} R_i]$.

5.1. **Bisimulations and homomorphisms.** On the one hand, bisimulations are 2-dimensional versions of subcoalgebras, but at the same time, they are also generalizations of homomorphisms as the following theorem states:

Theorem 5.9 ([Rut96])**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be coalgebras and $f : A \to B$ a map. f is a homomorphism if and only if the graph of f, i.e. $G(f) = \{(x, f(x)) \mid x \in A\}$ is a bisimulation.*

Proof. Let $\varphi : \mathcal{A} \to \mathcal{B}$ be a homomorphism. For the 2-source $\mathcal{A} \xleftarrow{id_A} \mathcal{A} \xrightarrow{\varphi} \mathcal{B}$, the canonical bisimulation is $(id \times \varphi)[A] = \{(a, \varphi(a)) \mid a \in A\}$, which is the graph of φ. Conversely, if the graph $G(f)$ of a map $f : A \to B$ is a bisimulation, then $\pi_1 : G(f) \to \mathcal{A}$ is bijective, hence an isomorphism and $f = \pi_2 \circ \pi_1^{-1}$ must be a homomorphism. $\square$

Corollary 5.10. *For every coalgebra A, the diagonal $\Delta_A = \{(a, a) \mid a \in A\}$ is a bisimulation. $S \subseteq A$ is closed iff Δ_S is a bisimulation on A.*

According to corollary 5.7, for every pair $\mathcal{A}, \mathcal{B}$ of coalgebras there is a largest bisimulation between $\mathcal{A}$ and $\mathcal{B}$. We denote it by $\sim_{\mathcal{A},\mathcal{B}}$, or by $\sim_{\mathcal{A}}$, in the case $\mathcal{A} = \mathcal{B}$. From the above lemmas it follows immediately, that $\sim_{\mathcal{A}}$ is reflexive and symmetric. We shall see that it is not necessarily transitive.

The following theorem is a useful characterization of the largest bisimulation $\sim_{\mathcal{A},\mathcal{B}}$ between two coalgebras $\mathcal{A}$ and $\mathcal{B}$:

Theorem 5.11. *$(x, y) \in \sim_{\mathcal{A},\mathcal{B}}$ iff there is a coalgebra $\mathcal{P}$ and homomorphisms $\varphi_1 : \mathcal{P} \to \mathcal{A}$ and $\varphi_2 : \mathcal{P} \to \mathcal{B}$ so that for some $p \in P$ we have: $x = \varphi_1(p)$ and $y = \varphi_2(p)$.*

5.2. **Bisimulations and congruences.** Sub-coalgebras are 1-simulations, homomorphisms are special bisimulations. Congruences, as subsets of the cartesian product, should be related to bisimulations too.

Definition 5.12. *A* bisimulation equivalence *on a coalgebra $\mathcal{A}$ is a bisimulation between $\mathcal{A}$ and $\mathcal{A}$, which is at the same time an equivalence relation.*

Theorem 5.13. *Every bisimulation equivalence is a congruence relation.*

Proof. Let θ be an equivalence relation on A. The coequalizer (in $\mathcal{S}et$) of the projections $\pi_1, \pi_2 : \theta \to A$ is just A/θ with the map $\pi_\theta : A \to A/\theta$ defined by $\pi_\theta(a) = [a]\theta$. If θ is a bisimulation, then π_1 and π_2 are homomorphisms, so by theorem 4.2 the same map $\pi_\theta : A \to A/\theta$ is the coequalizer of π_1 and π_1 in $\mathcal{S}et_F$. Its kernel is θ, so θ is a congruence relation. □

The converse of this theorem is not true in general, that is, not every congruence relation needs to be a bisimulation. We shall give a characterization of functors F whose coalgebras satisfy this extra property in section 8. For now we produce a counterexample from the functor $F = (-)^3_2$, see 3.10:

Example 5.14. *Consider the $(-)^3_2$-coalgebra on the set $A = \{0, 1\}$ with $\alpha(0) = (0, 0, 1)$ and $\alpha(1) = (0, 1, 0)$. Apparently, $A \times A$ is a congruence relation, but not a bisimulation, for the structure map γ on $A \times A$ would have to map $(0, 1)$ to a triple $((x_1, y_1), (x_2, y_2), (x_3, y_3))$ so that $(x_1, x_2, x_3) = F(\pi_1)((x_1, y_1), (x_2, y_2), (x_3, y_3)) = \alpha(0) = (0, 0, 1)$ and at the same time $(y_1, y_2, y_3) = (0, 1, 0)$. It follows immediately that $((x_1, y_1), (x_2, y_2), (x_3, y_3)) = ((0, 0), (0, 1), (1, 0))$, but this element is not in $F(A \times A)$.*

We have seen that on every coalgebra $\mathcal{A}$ there is a largest bisimulation $\sim_\mathcal{A}$ and a largest congruence relation $con[A \times A]$. What is their relationship? The answer is given by the following lemma:

Lemma 5.15. *For every bisimulation R there is a smallest congruence relation $\langle R \rangle$ containing R.*

Proof. Consider R as a coalgebra with projections $\pi_1, \pi_2 : \mathcal{R} \to \mathcal{A}$. Let φ be the coequalizer of π_1 and π_2. Then $ker(\varphi)$ contains R and it is obviously the smallest congruence relation with this property. □

5.3. **Epis and monos in $\mathcal{S}et_F$.** A morphism which is epi (mono) in $\mathcal{S}et$ is trivially also epi (mono) in $\mathcal{S}et_F$. For epis the converse also holds. In fact this could be seen already from exercise 4.1, but we shall give the standard proof. The story is different for monos, and we shall see that monos in $\mathcal{S}et_F$ need not be injective.

Theorem 5.16 ([Rut96]). *Let $\varphi : \mathcal{A} \to \mathcal{B}$ be a homomorphism.*

1. *φ is epi in $\mathcal{S}et_F$ iff it is epi in $\mathcal{S}et$, i.e. surjective.*
2. *If φ is mono in $\mathcal{S}et$, it is mono in $\mathcal{S}et_F$.*

Proof. In each category it is the case that $\varphi : A \to B$ is epi iff the following diagram is a pushout. According to theorem 4.2 this is the case if and only if it is a pushout in $\mathcal{S}et$.

$$\begin{tikzcd} A \arrow[r, "\varphi"] \arrow[d, "\varphi"'] & B \arrow[d, "id_B"] \\ B \arrow[r, "id_B"] & B \end{tikzcd}$$

□

Indeed, monos need not be injective. The following theorem points out the difference :

Theorem 5.17. [GS99a] *A homomorphism φ is mono if and only if $[ker(\varphi)] = \Delta_{\mathcal{A}}$.*

Proof. Assume that $\varphi : \mathcal{A} \to \mathcal{B}$ is mono. Let $\pi_1, \pi_2 : ker(\varphi) \to A$ be the canonical projection maps. Let $\tilde{\pi}_1, \tilde{\pi}_2 : [\mathrm{ker}(\varphi)] \to A$ be their restrictions to the coalgebra $[ker(\varphi)]$. Now $\tilde{\pi}_1$ and $\tilde{\pi}_2$ are homomorphisms and $\varphi \circ \tilde{\pi}_1 = \varphi \circ \tilde{\pi}_2$, hence it follows $\tilde{\pi}_1 = \tilde{\pi}_2$, that is, $[ker(\varphi)] = \Delta_A$.

Conversely, assume that $[ker(\varphi)] = \Delta_A$ and assume that there are homomorphisms $\kappa_1, \kappa_2 : \mathcal{P} \to \mathcal{A}$ with $\varphi \circ \kappa_1 = \varphi \circ \kappa_2$. For every element $p \in P$ we have $\kappa_1(p) \sim \kappa_2(p)$, hence $(\kappa_1(p), \kappa_2(p)) \in [ker(\varphi)]$, so $\kappa_1 = \kappa_2$ and φ is mono. □

Corollary 5.18. *A homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ is an injective map iff φ is mono and $ker(\varphi)$ is a bisimulation.*

5.4. **Exercises.**

Exercise 5.1 (Automata as coalgebras). *For fixed sets Σ and Γ consider the functor*

$$F_\Gamma^\Sigma(X) = X^\Sigma \times \Gamma^\Sigma.$$

1. *Explain how F_Γ^Σ acts on maps $f : X \to Y$ and show that F_Γ^Σ becomes a functor.*
2. *What is the correspondence between F_Γ^Σ-coalgebras and deterministic automata with output.*
3. *We write $a \xrightarrow{e,g} b$, if on input e the automaton moves from state a to state b and outputs g. Express the conditions under which $\varphi : \mathcal{A} \to \mathcal{B}$ is a homomorphism and describe them in terms of $\longrightarrow$.*
4. *Characterize* subcoalgebras *and* congruence relations *of F_Γ^Σ-coalgebras.*

Exercise 5.2 (Nondeterministic Automata). *We consider nondeterministic automata (Example 2.28).*

1. *Prove the homomorphism conditions as stated there.*
2. *Characterize bisimulations between coalgebras $\mathcal{A}$ and $\mathcal{B}$ in terms of $\xrightarrow{e}$.*

Exercise 5.3. *For the functor $F = (-)^3_2$, find two coalgebras $\mathcal{A}$ and $\mathcal{B}$ and a homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ which is both epi and mono, but not an isomorphism.*

6. Existence of Limits

For limits the situation is not as easy as with colimits. Limits may exist, but their carrier set and their morphisms in general will not come from the corresponding limits in $\mathcal{S}et$.

6.1. **Equalizers.** We start with a positive result that essentially goes back to Worrell([Wor98]):

Theorem 6.1. *The equalizer of parallel morphisms* $\mathcal{A} \underset{\psi}{\overset{\varphi}{\rightrightarrows}} \mathcal{B}$ *exists in* $\mathcal{S}et_F$. *It is given as* $\mathcal{E}_{\varphi,\psi} := [\ \{a \in A \mid \varphi(a) = \psi(a)\}\ ]$.

Proof. Let $\mathcal{D}$ with morphism $\delta : \mathcal{D} \to \mathcal{A}$ be a competitor of $\mathcal{E}_{\varphi,\psi}$. Then $\delta[D] \subseteq \{a \in A \mid \varphi(a) = \psi(a)\}$ and $\delta[\mathcal{D}]$ is a subcoalgebra of $\mathcal{A}$, hence $\delta[D] \leq \mathcal{E}_{\varphi,\psi}$. The Second Diagram Lemma yields a mediating morphism from $\mathcal{D}$ to $\mathcal{E}_{\varphi,\psi}$.

$$\begin{tikzcd}
E_{\varphi,\psi} \arrow[r, "\leq"] & A \arrow[r, shift left, "\varphi"] \arrow[r, shift right, "\psi"'] & B \\
D \arrow[u, dashed] \arrow[ur, "\delta"'] & &
\end{tikzcd}$$

□

Thus, in general the carrier set of the equalizers of φ and ψ can be a proper subset of $\{a \in A \mid \varphi(a) = \psi(a)\}$, which would be their equalizer in $\mathcal{S}et$.

6.2. **Products.** Worrell shows in [Wor98] that the product of a family $(\mathcal{A}_i)_{i \in I}$ of coalgebras exists, provided that the functor F is "bounded" and "preserves weak pullbacks". We shall define and study these conditions in the later sections. A careful analysis of his proof, which mainly rest on a result in [Bor94], shows that "preservation of weak pullbacks" is not really required. Putting together all necessary ingredients from the proof in [Bor94] becomes extremely complicated and involved. In [GS99b], see also page 48, we offer an elementary proof of this result, of which the following is the main observation:

Theorem 6.2 ([GS99b])**.** *Let the product* $\mathcal{A}$ *of a family* $(\mathcal{A}_i)_{i \in I}$ *of coalgebras exist, and let* $\mathcal{B}_i \leq \mathcal{A}_i$ *for each* $i \in I$. *Then the product of the* $\mathcal{B}_i$ *exists and it is a subcoalgebra of* $\mathcal{A}$.

Proof. Let $\mathcal{A}$ with projections $\eta_i : \mathcal{A} \to \mathcal{A}_i$ be the product in $\mathcal{S}et_F$ of the $\mathcal{A}_i$. Put

$$B := \bigcup \{S \mid \mathcal{S} \leq \mathcal{A}, \forall_{i \in I}.\eta_i[S] \subseteq B_i\}.$$

It is straightforward to check that $\mathcal{B}$ with the restrictions of the η_i to B, that is $\tilde{\eta}_i = \eta_i \circ \leq_B$, is the product of the $\mathcal{B}_i$. □

6.3. **Limits preserved by functors.** The situation is even easier, if the functor F preserves a particular type of limit. In that case the corresponding limit exists and is formed as in $\mathcal{S}et$.

For instance, the functor $F = (-)^{\Sigma}$ is easily seen to preserve products, for $F(A_1 \times A_2) \cong F(A_1) \times F(A_2)$ and for $\pi_i : A_1 \times A_2 \to A_i$ we have $F(\pi_{A_i}) = \pi_{F(A_i)}$. In this case, the product of the $F-coalgebras$, i.e. the transition systems (A_1, α_1) and (A_2, α_2) has as base set $A_1 \times A_2$. From the fact that π_1 and π_2 are homomorphisms it follows that the structure map is given as α_1 in the first component and α_2 in the second.

Theorem 6.3 ([Rut96]). *$\mathcal{S}et_F$ has all limits which are preserved by F. The base set and the canonical morphisms agree with those obtained in $\mathcal{S}et$ by forming the corresponding limit of the base sets.*

Proof. Let $\mathcal{D}$ be a diagram in $\mathcal{S}et_F$. Let $\mathcal{C}$ with maps $(\varepsilon_i)_{i\in I}$ be the limit of $\mathcal{D}$ in $\mathcal{S}et$. Since F preserves the limits of diagram $\mathcal{D}$, we get that $F(C)$ with morphisms $F(\varepsilon_i) : F(C) \to F(D_i)$ is the limit of $F(\mathcal{D})$. Now $\mathcal{C}$ with morphisms $\alpha_i \circ \varepsilon_i$ is a competitor for $F(C)$. This yields a unique structure map $\alpha_C : C \to F(C)$, with respect to which all ε_i become homomorphisms.

$$\begin{tikzcd}
& D_i \arrow[dr, dotted, "\varphi"] \arrow[dd, "\alpha_i"] & \\
C \arrow[ur, "\varepsilon_i"] \arrow[rr, "\varepsilon_k"'] \arrow[dd, dashed, "\alpha_C"'] & & D_k \arrow[dd, "\alpha_k"] \\
& F(D_i) \arrow[dr, dotted, "F(\varphi)"] & \\
F(C) \arrow[ur, "F(\varepsilon_i)"] \arrow[rr, "F(\varepsilon_k)"'] & & F(D_k)
\end{tikzcd}$$

The constructed coalgebra (C, α_C) with homomorphisms ε_i is in fact the limit of $\mathcal{D}$ in $\mathcal{S}et_F$. To see this, consider an object E with homomorphisms $\lambda_i : E \to D_i$ in $\mathcal{S}et_F$. According to the limit property of $\mathcal{C}$ in $\mathcal{S}et$ there is exactly one map $\tau : E \to C$ with $\varepsilon_i \circ \tau = \lambda_i$.

Since E with maps $\alpha_i \circ \lambda_i$ is also a cone over $F(\mathcal{D})$ and since both $\alpha_C \circ \tau$ and $F(\tau) \circ \alpha_E$ are universal arrows from E to $F(C)$, they must be identical. This means that τ is a homomorphism.

$$\begin{tikzcd}
& D_i \arrow[dr, dotted, "\varphi"] \arrow[dd, "\alpha_i"] & & E \arrow[ll, "\lambda_i"'] \arrow[dl, "\lambda_k"] \arrow[dlll, dashed, "\tau"'] \arrow[dd, "\alpha_E"] \\
C \arrow[ur, "\varepsilon_i"] \arrow[rr, "\varepsilon_k"'] \arrow[dd, "\alpha_C"'] & & D_k \arrow[dd, "\alpha_k"] & \\
& F(D_i) \arrow[dr, dotted] & & F(E) \arrow[ll, "F(\lambda_i)"'] \arrow[dl, "F(\lambda_k)"] \arrow[dlll, dashed, "F(\tau)"'] \\
F(C) \arrow[ur, "F(\varepsilon_i)"] \arrow[rr, "F(\varepsilon_k)"'] & & F(D_k) &
\end{tikzcd}$$

$\square$

6.4. **Final coalgebras.** An object P in a category $\mathcal{C}$ is called *final* (or *terminal*), if for every object A in $\mathcal{C}_o$ there is precisely one morphism $\varphi_A : A \to P$.

A final object, if it exists, is the limit of the empty diagram i.e. the product over the empty index set. Analogously, an object I is called *initial*, if for every object A there is precisely one morphism $\psi_A : I \to A$. An initial object, if it exists, is the sum over the empty index set.

In $\mathcal{S}et$, the final object is the one-element set 1. When F is the power set functor, however, one can easily see that there is no way to define a transition structure on 1 turning this set into the final object in the category $\mathcal{S}et_F$.

This argument does not yet exclude $\mathcal{S}et_F$ from *having* a final object. So far we only know that it would need to have more than one element. In fact, the same argument is true for $\mathcal{P}_\omega$, the functor, assigning to every set S the set of all finite subsets of S. On mappings, $\mathcal{P}_\omega$ acts just like the power set functor 2^-. The category $\mathcal{S}et_{\mathcal{P}_\omega}$ does possess a final object, whereas $\mathcal{S}et_{2^{(-)}}$ does not.

A good intuition as to the nature of elements in the final coalgebra $\mathcal{P}$ is given by the following theorem. It states that for every $a \in \mathcal{A} \in \mathcal{S}et_F$ there is exactly one bisimilar element in $\mathcal{P}$:

Theorem 6.4. *If the final coalgebra exists, then for every $\mathcal{A} \in \mathcal{S}et_F$ and every $a \in A$ there exists precisely one element $\tau(a)$ in the final coalgebra with $a \sim \tau(a)$.*

Proof. Given $\mathcal{A}$, there is a (unique) homomorphism $\tau : A \to \mathcal{P}$, so $a \sim \tau(a)$. Let $p \in P$ be any other element with $a \sim p$. Then there exists some $\mathcal{Q} \in \mathcal{S}et_F$, homomorphisms $\varphi : \mathcal{Q} \to \mathcal{A}$ and $\psi : \mathcal{Q} \to \mathcal{P}$, and an element $q \in Q$ with $\varphi(q) = a$ and $\psi(q) = p$. Now both $\tau \circ \varphi$ and ψ are homomorphisms from $\mathcal{Q}$ to $\mathcal{P}$, so they must be identical, in particular, $\tau(a) = (\tau \circ \varphi)(q) = \psi(q) = p$. $\square$

Corollary 6.5. *No two different elements x and y in the final coalgebra $\mathcal{P}$ are bisimilar, that is $\sim_{\mathcal{P}} = \Delta_{\mathcal{P}}$.*

This can also be written as a proof principle which will be the basis for the proof method known as *co-induction*:

"Whenever two elements are bisimilar, they must be equal."

$$\frac{x \sim y}{x = y}$$

We shall discuss this principle in connection with the notion of *simple coalgebras*. First, we shall look at some examples of final coalgebras.

6.4.1. *Examples of final coalgebras.* In $\mathcal{S}et$, the empty set $0 = \emptyset$ is initial and the one-element set $1 = \{0\}$ is final. In $\mathcal{S}et_F$, the initial object always exists, it is the empty coalgebra $\emptyset = (\emptyset, \emptyset)$. The final coalgebra need not exist.

Example 6.6. *For the identity-functor $\mathcal{I}d$ we have: $1 = (1, id_1)$ is final in $\mathcal{S}et_{\mathcal{I}d}$.*

Example 6.7. *Let Σ be a fixed set. For the functor $\Sigma \times \mathcal{I}d$, the coalgebra $\Sigma^\omega = (\Sigma^\omega, hd \times tl)$ is final. Here Σ^ω is the set of all infinite sequences of elements of Σ. The structure map associates to every sequence $\sigma = (s_i)_{i\in\omega}$ the pair consisting of its head s_0 and its tail $(s_{i+1})_{i\in\omega}$.*

Example 6.8. *For the functor $\Sigma \times \mathcal{I}d + 1$, the final coalgebra consists of Σ^∞, the set of all finite and infinite sequences. The structure is defined as $\alpha(\sigma) = (hd(\sigma), tl(\sigma))$, if $\sigma \neq ()$ and $\alpha(\sigma) = 0$ otherwise.*

Example 6.9. *For $\mathcal{I}d\times\mathcal{I}d$, the final coalgebra is the one-element structure, but for $\mathcal{I}d \times \Sigma \times \mathcal{I}d$, the final coalgebra consists of all infinite binary trees with nodes from Σ.*

The following example goes back to Reichel ([Rei95]). A thorough treatment has recently been given by Rutten in [Rut98]:

Example 6.10. *An automaton $A = (A, \Sigma, \delta, T)$ with terminal states T over the alphabet Σ can be considered as a coalgebra for the functor $\mathcal{I}d^{\Sigma} \times \mathbb{B}$. As final coalgebra one obtains the coalgebra of all languages over Σ. The base set of this coalgebra is $\mathcal{P}(\Sigma^\star)$, the set of all languages over Σ. The structure map is given as $\alpha : \mathcal{P}(\Sigma^\star) \to \mathcal{P}(\Sigma^\star)^{\Sigma} \times \mathbb{B}$ with $\pi_1 \circ \alpha(L)(e) = L_e = \{w \in \Sigma^\star \mid e \cdot w \in L\}$ and $\pi_2 \circ \alpha(L) = true$ iff $\varepsilon \in L$.*

Example 6.11. *For the finite power set functor $\mathcal{P}_\omega$, the final object consists of the bisimilarity classes of all finitely branching trees. The transition structure assigns to every tree the set of its immediate subtrees (up to bisimilarity).*

6.5. **Simple coalgebras.** For many purposes, it will be enough to consider coalgebras which are not final, but satisfy the weaker condition of *simplicity*:

Definition 6.12. *A coalgebra $\mathcal{S}$ is called* simple, *if Δ_S is the largest bisimulation on $\mathcal{S}$.*

A structure theoretic characterization of simple coalgebras is given by the following theorem:

Theorem 6.13. *For a coalgebra $\mathcal{S}$ the following are equivalent:*

1. *$\mathcal{S}$ is simple*
2. *Every homomorphism φ with domain $\mathcal{S}$ is mono.*
3. *For every $A \in \mathcal{S}et_F$ there is at most one homomorphism $\psi : \mathcal{A} \to \mathcal{S}$.*

Proof. 1. $\to$ 2.: Let $\varphi : \mathcal{S} \to \mathcal{B}$ be a homomorphism, then $[ker(\varphi)]$ is a bisimulation, hence $[ker(\varphi)] = \Delta_{\mathcal{S}}$. As a consequence of 5.17, φ is mono. 2. $\to$ 3.: Given two homomorphisms $\psi_1, \psi_2 : \mathcal{A} \to \mathcal{S}$, we consider their co-equalizer $\mathcal{C}$ with canonical morphism $\varepsilon : \mathcal{S} \to \mathcal{C}$. It must be mono, whence $\psi_1 = \psi_2$. 3. $\to$ 1.: Assuming there is a pair $(x, y) \in \sim_{\mathcal{S}}$, we know from theorem 5.11 that there is a coalgebra $\mathcal{A}$, an element $a \in A$ and homomorphisms $\varphi_1, \varphi_2 : A \to S$ with $\varphi_1(a) = x$ and $\varphi_2(a) = y$. From the hypothesis, $\varphi_1 = \varphi_2$, so $x = y$. $\square$

6.5.1. *Proofs by coinduction.* The definition of simplicity is just a reformulation of the *coinduction principle*:

$$\frac{x \sim y}{x = y}.$$

This is a convenient proof principle. In order to show that two elements x and y are equal it is enough to find a bisimulation R with $(x, y) \in R$. This principle makes simple coalgebras, in particular the final coalgebra $\mathcal{P}$, if it exists, into a useful semantical domain for programming. We shall demonstrate this at an example:

Example 6.14. *Consider once more $\Sigma \times \mathcal{I}d$-coalgebras. The final $\Sigma \times \mathcal{I}d$-coalgebra is the set Σ^ω of all infinite Σ-lists. Consider the following functional program:*

```
sec(k)     =  [k : sec(k+2)]
average(s) =  [(hd(s)+hd(tl(s)))/2 : average(tl(s))].
```

We wish to show the following claim:

$$average(sec(k)) = sec(k+1).$$

For this we consider the relation $R = \{(average(sec(k)), sec(k{+}1)) \mid k \in \mathcal{N}\}$ and show that it is a bisimulation. So assume that $(average(sec(k)), sec(k+1)) \in R$, then

$$\begin{array}{rcl} hd(average(sec(k))) & = & (hd(sec(k)) + hd(tl(sec(k))))/2 \\ & = & (k + (k+2))/2 \\ & = & k+1 \\ & = & hd(sec(k+1)),\ \ and \end{array}$$

$$\begin{array}{rcl} tl(average(sec(k))) & = & average(tl(sec(k))) \\ & = & average(sec(k+2)) \\ & R & sec(k+3) \\ & = & tl(sec(k+1)). \end{array}$$

Hence R is a bisimulation, consequently $R = id$, which proves the claim.

6.5.2. *Strong simplicity.* The notion of "simple coalgebra" was coined after the corresponding notion in universal algebra. A universal algebra is called simple, if it does not have any nontrivial homomorphic image. This is the same as saying that there is no nontrivial congruence relation. In the field of coalgebras, the interest was initially focused on coalgebras for functors F that "preserve weak pullbacks". (We shall explain this concept later in section 8). In that context, indeed, a coalgebra is simple if and only if it does not have a nontrivial homomorphic image. In general, though, the two notions do not agree. Therefore, we define:

Definition 6.15. *A coalgebra is called* strongly simple *if it does not possess any nontrivial congruence relation.*

By lemma 5.15, every bisimulation R generates a congruence $\langle R\rangle$, containing R, so a strongly simple coalgebra is also simple. If the final coalgebra $\mathcal{P}$ exists, then strongly simple coalgebras have an easy description:

Lemma 6.16. *If the final coalgebra $\mathcal{P}$ exists, then the strongly simple coalgebras are precisely the subcoalgebras of $\mathcal{P}$.*

6.6. **Existence of final coalgebras.** Not for each functor F does there exist a final coalgebra. An example is given by the power-set functor $2^{(-)}$. To see this, we need the following observation, due to Lambek (see also [Bor94]):

Theorem 6.17. *If $\mathcal{P} = (P, \pi)$ is a final coalgebra, then the structure map π is an isomorphism.*

Proof. Let $\mathcal{P} = (P, \pi)$ be the final F-coalgebra. Since $(F(P), F(\pi))$ is an F-coalgebra too, there exists precisely one homomorphism $\varphi : F(P) \to P$. The composition $\varphi \circ \pi$ is now a homomorphism from $\mathcal{P}$ to $\mathcal{P}$, hence $\varphi \circ \pi = id_P$. Considering that φ is a homomorphism one gets: $\pi \circ \varphi = F(\varphi) \circ F(\pi) = F(\varphi \circ \pi) = F(id_P) = id_{F(P)}$. Hence φ is inverse to π.

$$\begin{tikzcd}
P \arrow[r, "\pi"] \arrow[d, "\pi"'] & F(P) \arrow[r, "\varphi"] \arrow[d, "F(\pi)"] & P \arrow[d, "\pi"] \\
F(P) \arrow[r, "F(\pi)"'] & F(F(P)) \arrow[r, "F(\varphi)"'] & F(P)
\end{tikzcd}$$

□

This theorem implies that there cannot be a final $2^{(-)}$-coalgebra. We would need a bijection between its base set A and its power set 2^A. Cantor has shown, that this is impossible. [2]

The final coalgebra, if it exists, must subsume all possible behaviours. This is a consequence of theorem 6.4. Usually, therefore, we shall expect this to be a rather large object - so large in fact, that it cannot possibly be a proper set. However, there is a useful condition guaranteeing its existence. We only need a collection of coalgebras, ample enough to generate all possible "local behaviours":

Definition 6.18. *A collection of coalgebras $(G_i)_{i \in I}$ is called a* set of generators, *if for every $a \in A \in \mathcal{S}et_F$ there exists some G_i which is isomorphic to a subcoalgebra $U \leq A$ with $a \in U$.*

The existence of a set of generators is usually formulated as a smallness condition on the functor F. For that we define:

Definition 6.19. *A functor F is* bounded, *if there is some cardinality κ so that for every F-coalgebra $\mathcal{A}$ and every $a \in A$ one can find a subcoalgebra U_a of $\mathcal{A}$ with $a \in U_a$ and $|\mathcal{U}_a| \leq \kappa$.*

Lemma 6.20. *F is bounded if and only if $\mathcal{S}et_F$ has a set of generators.*

The following theorem could be obtained as an application of the "Special Adjoint Functor Theorem"([Lan71]). In the given context the proof simplifies considerably:

Theorem 6.21. *If F is bounded, then the final F-coalgebra exists.*

Proof. Let $\mathcal{G} = \Sigma_{i \in I} G_i$ be the sum of all generators and let Ξ be the largest congruence relation on $\mathcal{G}$. We claim that $\mathcal{P} = \mathcal{G}/\Xi$ is final.

Given any coalgebra $\mathcal{A}$, then for every $a \in A$ there is a $U_a \leq A$ which is isomorphic to one of the G_i. There is an onto homomorphism $\varphi : \Sigma_{a \in A} U_a \to A$ and a canonical homomorphism ψ to $\Sigma_{i \in I} G_i$. Let π_Ξ be the canonical factor homomorphism from $\Sigma_{i \in I} G_i$ to P. We need to find a homomorphism from $\mathcal{A}$ to $\mathcal{P}$.

Form the pushout $\mathcal{Q}$ of φ and ψ. This results in a map $\tilde{\psi} : \mathcal{A} \to \mathcal{Q}$ and an epi $\tilde{\varphi} : \mathcal{G} \to Q$. Surely, $ker(\tilde{\varphi}) \subseteq \Xi$, so there is a homomorphism $\chi : \mathcal{Q} \to \mathcal{P}$

[2] Assume that $f : 2^A \to A$ is bijective with inverse f^{-1}. Consider the set $G = \{x \mid x \notin f^{-1}(x)\}$, then we get the contradiction $f(G) \in G \Leftrightarrow f(G) \notin G$.

and therefore $\chi \circ \tilde{\psi} : A \to P$. It remains to show that this homomorphism from $\mathcal{A}$ to $\mathcal{P}$ is unique.

$$
\begin{tikzcd}
A \arrow[r, "\tilde{\psi}"] & Q \arrow[r, "\chi"] & P \\
U_a \arrow[u, hook] \arrow[r] \arrow[rr, bend right] & \Sigma_{a\in A} U_a \arrow[ul, "\varphi"'] \arrow[r, "\psi"] & \Sigma_{i\in I} G_i \arrow[ul, "\tilde{\varphi}"'] \arrow[u, "\pi_\Xi"']
\end{tikzcd}
$$

So, assume that there is an $\mathcal{A}$ with two homomorphisms $\phi_1, \phi_2 : \mathcal{A} \to \mathcal{P}$. Consider the coequalizer $\vartheta : \mathcal{P} \to \mathcal{R}$ of ϕ_1 and ϕ_2. Again, $ker(\vartheta \circ \pi_\Xi) = ker(\pi_\Xi)$, so it follows that $ker(\vartheta) = \Delta_{\mathcal{P}}$. This implies $\phi_1 = \phi_2$.

$$
\begin{tikzcd}
A \arrow[r, shift left, "\phi_1"] \arrow[r, shift right, "\phi_2"'] & P \arrow[r, "\vartheta"] & R \\
& \Sigma_{i\in I} G_i \arrow[u, "\pi_\Xi"'] \arrow[ur] &
\end{tikzcd}
$$

□

6.7. **Exercises.**

Exercise 6.1 (Equalizers)**.** *Find two coalgebras $\mathcal{A}$ and $\mathcal{B}$ and homomorphisms $\varphi, \psi : \mathcal{A} \to \mathcal{B}$ whose equalizer in $\mathcal{S}et_F$ is a proper subset of their equalizer in $\mathcal{S}et$.*

Exercise 6.2 (Final coalgebra)**.** *Prove that the final coalgebra for the functor $\Sigma \times \mathcal{I}d + 1$ is given by Σ^∞, the set of all finite and infinite sequences of elements of Σ (Example 5.8 in the notes).*

Exercise 6.3 (Strongly simple)**.** *Prove the statement of lemma 5.14, i.e. if the final coalgebra $\mathcal{P}$ exists, then the strongly simple coalgebras are just the subcoalgebras of $\mathcal{P}$.*

7. Co-varieties and cofree coalgebras

7.1. **Co-varieties.**

Definition 7.1. *Let $\mathcal{K}$ be a class of F-coalgebras. We define the following classes:*

1. *$\mathcal{H}(\mathcal{K})$: the class of all homomorphic images of objects from $\mathcal{K}$,*
2. *$\mathcal{S}(\mathcal{K})$: the class of all subcoalgebras of objects from $\mathcal{K}$,*
3. *$\Sigma(\mathcal{K})$: the class of all sums of objects from $\mathcal{K}$.*

A class $\mathcal{K}$ is called closed *under $\mathcal{H}$, S, or Σ, provided that $\mathcal{H}(\mathcal{K}) \subseteq \mathcal{K}$, $\mathcal{S}(\mathcal{K}) \subseteq \mathcal{K}$, or $\Sigma(\mathcal{K}) \subseteq \mathcal{K}$.*

Lemma 7.2. *$\mathcal{S}$, $\mathcal{H}$ and Σ are* closure operators*, that is for arbitrary classes $\mathcal{K}$, $\mathcal{K}_1$, $\mathcal{K}_2$ of coalgebras and any operator $\mathcal{O} \in \{\mathcal{S}, \mathcal{H}, \Sigma\}$ we have:*

1. $\mathcal{K} \subseteq \mathcal{O}(\mathcal{K})$
2. $\mathcal{K}_1 \subseteq \mathcal{K}_2 \implies \mathcal{O}(\mathcal{K}_1) \subseteq \mathcal{O}(\mathcal{K}_2)$
3. $\mathcal{O}(\mathcal{K}) = \mathcal{O}(\mathcal{O}(\mathcal{K}))$

Lemma 7.3. *For an arbitrary class $\mathcal{K}$ of coalgebras we have:*

1. $\mathcal{H}\mathcal{S}(\mathcal{K})) \subseteq \mathcal{S}\mathcal{H}(\mathcal{K}))$

2. $\Sigma\mathcal{S}(\mathcal{K}) \subseteq \mathcal{S}\Sigma(\mathcal{K})$
3. $\Sigma\mathcal{H}(\mathcal{K}) \subseteq \mathcal{H}\Sigma(\mathcal{K})$

Proof. Let $\mathcal{A} \in \mathcal{K}$ und $\mathcal{A} \geq \mathcal{B} \twoheadrightarrow \mathcal{C}$. Let $\mathcal{Q}$ with morphisms $p_1 : \mathcal{A} \to \mathcal{Q}$ and $p_2 : \mathcal{C} \to \mathcal{Q}$ be the pushout of this diagram. p_1 is epi, hence $\mathcal{Q} \in \mathcal{H}(\mathcal{K})$. Since $\mathcal{Q}$ with p_1 and p_2 is also the pushout in $\mathcal{S}et$ (theorem 4.2), it follows that p_2 is injective. Consequently, $\mathcal{C}$ is isomorphic to a subcoalgebra of $\mathcal{Q}$, hence $\mathcal{C} \in \mathcal{SH}(\mathcal{K}))$. The other two cases are analogous. $\square$

Definition 7.4. *A* co-variety *is a class $\mathcal{K}$ of coalgebras, which is closed under $\mathcal{S}$, $\mathcal{H}$ and Σ. A co-quasivariety is a class closed under $\mathcal{H}$ and Σ.*

From the preceding lemma we obtain immediately:

Theorem 7.5. [GS98] *Let $\mathcal{K}$ be a class of F-coalgebras. Then $\mathcal{SH}\Sigma(\mathcal{K})$ is the smallest co-variety containing $\mathcal{K}$.*

Definition 7.6. *Let $\mathcal{U} \leq \mathcal{A}$ be coalgebras. Let $\mathcal{Q}(\mathcal{A},\mathcal{U})$ be the class of all those coalgebras $\mathcal{C}$, for which every homomorphism $\varphi : \mathcal{C} \to \mathcal{A}$ factors through $\mathcal{U}$, i.e. for which there exists a homomorphism $\hat{\varphi} : \mathcal{C} \to \mathcal{U}$ with $\leq \circ\, \hat{\varphi} = \varphi$.*

The following theorem was proved in [GS98] under additional assumptions:

Theorem 7.7. *$\mathcal{Q}(\mathcal{A},\mathcal{U})$ is closed under homomorphic images and under sums, i.e. a co-quasivariety.*

Proof. (Sums) Consider $\mathcal{A}_i \in \mathcal{Q}(\mathcal{A},\mathcal{U})$ and $\varphi : \Sigma\mathcal{A}_i \to \mathcal{A}$. The composition with the canonical embeddings $e_i : \mathcal{A}_i \to \Sigma\mathcal{A}_i$ factors through $\mathcal{U}$, hence there exist $\gamma_i : \mathcal{A}_i \to \mathcal{U}$ with $\varphi \circ e_i = \leq \circ\, \gamma_i$. The sum property yields a unique morphism $\delta : \Sigma\mathcal{A}_i \to \mathcal{U}$ with $\delta \circ e_i = \gamma_i$. For all e_i we have now: $\varphi \circ e_i = \leq \circ\, \delta \circ e_i$, hence $\varphi = \leq \circ\, \delta$.

$$\begin{tikzcd}
\Sigma A_i \arrow[r, "\varphi"] \arrow[dr, dashed, "\delta"] & A \\
A_i \arrow[u, "e_i"] \arrow[r, "\gamma_i"'] & U \arrow[u, "\leq"']
\end{tikzcd}$$

(Homomorphic images) Let $\mathcal{B} \in \mathcal{Q}(\mathcal{A},\mathcal{U})$ and $\psi : \mathcal{B} \to \mathcal{C}$ an epimorphism. Let $\varphi : \mathcal{C} \to \mathcal{A}$ be a homomorphism. The composition $\varphi\circ\psi$ factors through $\mathcal{U}$ by way of γ, hence $\varphi\circ\psi = \leq \circ\, \gamma$. It follows that $ker(\psi) \subseteq ker(\varphi\circ\psi) = ker(\leq \circ\, \gamma) = ker(\gamma)$. Consequently, there is a unique homomorphism $\delta : \mathcal{C} \to \mathcal{U}$ with $\delta \circ \psi = \gamma$. We conclude that $\leq \circ\, \delta \circ \psi = \varphi \circ \psi$ and, since ψ is epi, $\leq \circ\, \delta = \varphi$.

$$\begin{tikzcd}
C \arrow[r, "\varphi"] \arrow[dr, dashed, "\delta"] & A \\
B \arrow[u, two heads, "\psi"] \arrow[r, "\gamma"'] & U \arrow[u, "\leq"']
\end{tikzcd}$$

$\square$

7.2. **Conjunct Sums.**

Definition 7.8. *Let $(\mathcal{G}_i)_{i\in I}$ be a family of coalgebras. A coalgebra $\mathcal{A}$ is called a* conjunct sum of the *$\mathcal{G}_i$, if for every $a \in A$ there is an $i \in I$ and an injective homomorphism $\varphi_i : \mathcal{G}_i \to \mathcal{A}$ with $a \in \varphi_i[G_i]$. A* conjunct representation of *$\mathcal{A}$* by the *$\mathcal{G}_i$ is a family of embeddings $(\varphi_k : \mathcal{G}_{i_k} \to \mathcal{A})_{k\in K}$ of some of the $\mathcal{G}_i$, so that $\bigcup_{k\in K} f[G_{i_k}] = A$.*

Thus, if $\mathcal{A}$ is a conjunct sum of the $\mathcal{G}_i$, then for every $a \in A$ we an find a subcoalgebra $\mathcal{U} \leq \mathcal{A}$ which contains a and which is isomorphic to one of the $\mathcal{G}_i$. $\mathcal{A}$ is therefore a "glued sum" of some of the $\mathcal{G}_i$. This means that the $\mathcal{G}_i$ are building blocks from which $\mathcal{A}$ is glued together(see page 31). The blocks used in the glueing are allowed to overlap.

Let $\mathcal{K}$ be a class of coalgebras. With $\Sigma_C(\mathcal{K})$ we denote the class of all coalgebras which are isomorphic to a conjunct sum of some objects from $\mathcal{K}$.

Lemma 7.9. *Σ_C is a closure operator and $\Sigma_C(\mathcal{K}) \subseteq \mathcal{H}\Sigma(\mathcal{K})$.*

Proof. The first part of the claim follows directly from the definition. Let $(\varphi_i : \mathcal{G}_i \to \mathcal{A})_{i\in I}$ be a conjunct representation of $\mathcal{A}$. Then there is a canonical homomorphism $\varphi : \Sigma_{i\in I}\mathcal{G}_i :\to \mathcal{A}$. Since $A \subseteq \bigcup \varphi_i[G_i]$ it follows that φ is epi. $\square$

For each coalgebra we have the trivial, but rather useless, conjunct representation $id_A : \mathcal{A} \to \mathcal{A}$. If a coalgebra does not possess a better representation, it must be used as an irreducible building block, we call it *conjunctly irreducible.* The precise definition is:

Definition 7.10. *A coalgebra $\mathcal{A}$ is called* conjunctly irreducible*, if in each conjunct representation $(\varphi_i : \mathcal{G}_i \to \mathcal{A})_{i\in I}$ of $\mathcal{A}$ one of the φ_i is onto, i.e. an isomorphism.*

Example 7.11. *A coalgebra structure on an $\mathcal{I}d$-coalgebra $\mathcal{A}$ is just a self map of $\mathcal{A}$. We represent it by arrows indicating the transition structure. The coalgebra*

$$\begin{tikzcd} \circ \arrow[r] & \circ \arrow[r] & \circ \arrow[r, bend left] & \circ \arrow[l, bend left] & \circ \arrow[l] \end{tikzcd}$$

is a conjunct sum of

$$\begin{tikzcd} \circ \arrow[r] & \circ \arrow[r] & \circ \arrow[r, bend left] & \circ \arrow[l, bend left] \end{tikzcd} \text{ and of } \begin{tikzcd} \circ \arrow[r, bend left] & \circ \arrow[l, bend left] & \circ \arrow[l] \end{tikzcd} .$$

The latter ones are conjunctly irreducible.

Definition 7.12. *A coalgebra $\mathcal{A}$ is called* one-generated*, if there is an $a \in A$, so that $\mathcal{A}$ is the only subcoalgebra of $\mathcal{A}$ which contains a. In short:*

$$a \in \mathcal{U} \leq \mathcal{A} \implies U = A$$

Theorem 7.13. *A coalgebra is conjunctly irreducible, iff it is 1-generated.*

Proof. Let $\mathcal{A}$ be conjunctly irreducible. If $\mathcal{A}$ was not one-generated, then to every $a \in A$ we would have a proper subcoalgebra $\mathcal{U}_a \leq A$ with $a \in \mathcal{U}_a$. The $\mathcal{U}_a$ yield a nontrivial conjunct representation of $\mathcal{A}$. Conversely, let $\mathcal{A}$ be one-generated, then there is an $a \in A$, so that for all closed subsets $U \subseteq A$ we have: $a \in U \implies U = A$. For every conjunct representation $(\varphi_i : \mathcal{G}_i \to \mathcal{A})_{i\in I}$ there must be an $i \in I$ with $a \in \varphi_i[G_i]$. Since $a \in \varphi_i[G_i] \leq A$ we conclude $\varphi_i[G_i] = A$, consequently the representation was trivial. $\square$

This theorem begs the question whether an analog to the first Birkhoff's theorem might be true: Is every coalgebra a conjunct sum of conjunctly irreducibles ?

In general the answer is no. If F is the filter functor and (X, τ) a topological space then we get an $\mathcal{F}$-coalgebra on X by defining $\alpha_X(x) = \mathcal{U}(x)$ where $\mathcal{U}(x)$ is the set of all neighborhoods of x. Subcoalgebras of (X, α_X) are precisely the open sets in τ. Clearly, it is easy to find examples of topologies where there is no smallest open set containing a given point, that is for which there are no one-generated subcoalgebras.

In contrast, for functors "weakly preserving pullbacks" (section 8), such a representation theorem can be proven, see [GS98].

7.3. **Cofree coalgebras.** Let F be a functor and X a set. We consider the elements of X as "colors" and every map $f : A \to X$ as *coloring*. A *cofree F-coalgebra over X* consists of an F-coalgebra S_X together with a coloring $\epsilon_X : S_X \to X$, so that for every F-coalgebra $\mathcal{A}$ and for every coloring $f : A \to X$ there is precisely one F-homomorphism $\tilde{f} : \mathcal{A} \to \mathcal{S}_X$ with $\epsilon \circ \tilde{f} = f$.

$$\begin{tikzcd} & X \\ A \arrow[ur, "f"] \arrow[r, dashed, "\tilde{f}"'] & S_X \arrow[u, "\epsilon_X"'] \end{tikzcd}$$

Theorem 7.14. *The F-coalgebra (S, α) with coloring $\epsilon : S \to X$ is cofree over X, iff $(S, (\epsilon_X, \alpha))$ is the final $X \times F(-)$-coalgebra.*

It is easy to see that if the functor F is bounded then the functor $X \times F(-)$ must be bounded too for any set X, that is:

Corollary 7.15. *If F is bounded then for every set X the cofree coalgebra $\mathcal{S}_X$ exist.*

Example 7.16. *The set of all streams of elements of X can be considered as cofree coalgebra over X for the identity functor $\mathcal{I}d$. The $\mathcal{I}d$-coalgebra structure is given by the map $tl : X^\omega \to X^\omega$. The coloring is $hd : X^\omega \to X$.*

Theorem 7.17. *If the cofree coalgebra S_X exists for every set X, then a functor from $\mathcal{S}et$ to $\mathcal{S}et_F$ can be defined on objects as $X \mapsto S_X$ and on maps as $f \mapsto map(f)$.*

With the aid of cofree coalgebras we can now describe maps between streams over different base sets.

Lemma 7.18. *Let $g : X \to Y$ be a set map. If (S_X, ϵ_X) is cofree over X and (S_Y, ϵ_Y) is cofree over Y, then there is a unique F-homomorphism $map(g) : S_X \to S_Y$ with $g \circ \epsilon_X = \epsilon_Y \circ map(g)$. If g is left invertible (right invertible), then $map(g)$ is left- (right-) invertible in $\mathcal{S}et_F$.*

Proof. Put $map(g) = \widetilde{g \circ \epsilon_X}$. If g is left invertible, i.e. $id_X = g^- \circ g$ for a certain g^-, then $id_{S_X} = map(id_X) = map(g^- \circ g) = map(g^-) \circ map(g)$.

$$\begin{tikzcd} X \arrow[r, "g"'] & Y \\ S_X \arrow[u, "\epsilon_X"'] \arrow[r, dashed, "map(g)"'] & S_Y \arrow[u, "\epsilon_Y"'] \end{tikzcd}$$

□

Lemma 7.19. *Let S_X be cofree over X and let $\mathcal{A} \leq \mathcal{B}$ be any F-coalgebras. Every homomorphism $\varphi : A \to S_X$ can be extended to a homomorphism $\psi : B \to S_X$ with $\varphi = \psi \circ \leq$.*

Proof. $\epsilon_X \circ \varphi$ is a coloring of A. In $\mathcal{S}et$ it can be extended to a coloring $f : B \to X$. Let $\psi = \tilde{f} : B \to S_X$ be the homomorphic extension of f. Then $f = \epsilon_X \circ \psi$ and $\epsilon_X \circ \psi \circ \leq\, = f \circ \leq\, = \epsilon_X \circ \varphi$. From the uniqueness of the extension we get $\psi \circ \leq = \varphi$.

$$\begin{tikzcd} B \arrow[r, "f"] \arrow[dr, dashed, "\psi"] & X \\ A \arrow[u, "\leq"] \arrow[r, "\varphi"'] & S_X \arrow[u, "\epsilon_X"'] \end{tikzcd}$$

□

Corollary 7.20. *If $\mathcal{S}_X$ is cofree and $\mathcal{U} \leq \mathcal{S}_X$ then $\mathcal{Q}(\mathcal{S}_X,\mathcal{U})$ is a co-variety.*

The co-varieties (resp. co-quasivarieties) $\mathcal{Q}(\mathcal{S}_X,\mathcal{U})$ are in no way special, if the functor F is bounded. In particular, we have:

Theorem 7.21. *If F is bounded, then for every co-variety $\mathcal{K}$ there is a set X and a subcoalgebra $\mathcal{U} \leq \mathcal{S}_X$ so that $\mathcal{K} = \mathcal{Q}(\mathcal{S}_X,\mathcal{U})$.*

Proof. Since F is bounded, cofree coalgebras exist for every set of colors. Moreover, there is a set X so that for every F-coalgebra $\mathcal{A}$ and every $a \in A$ there is a subcoalgebra $\mathcal{U}_a \leq \mathcal{A}$ with $a \in U_a$ and $|U_a| \leq |X|$. For any such U_a we have an injective mapping $g_a : U_a \to X$, which extends to an embedding $\tilde{g_a} : \mathcal{U}_a \to \mathcal{S}_X$. Consequently, every F-coalgebra $\mathcal{A}$ is a conjunct sum of subcoalgebras of $\mathcal{S}_X$.

Let now $\mathcal{K}$ be a co-variety, let $U \subseteq S_X$ be the union of all images $\tilde{g}[A]$ where $\mathcal{A} \in \mathcal{K}$ and $g : A \to X$. Clearly, $\mathcal{U}$ is a subcoalgebra of $\mathcal{S}_X$ and $\mathcal{K} \subseteq \mathcal{Q}(\mathcal{S}_X,\mathcal{U})$. Moreover, $\mathcal{U} \in \Sigma_C\mathcal{H}(\mathcal{K})$, hence $\mathcal{U} \in \mathcal{K}$. Let now $\mathcal{B} \in \mathcal{Q}(\mathcal{S}_X,\mathcal{U})$, we have to show that $\mathcal{B} \in \mathcal{K}$. B is a conjunct sum of subcoalgebras of $\mathcal{S}_X$. Since $\mathcal{B} \in \mathcal{Q}(\mathcal{S}_X,\mathcal{U})$ and $\mathcal{Q}(\mathcal{S}_X,\mathcal{U})$ is closed under subcoalgebras, every summand of $\mathcal{B}$ is in $\mathcal{Q}(\mathcal{S}_X,\mathcal{U})$. But a subcoalgebra of $\mathcal{S}_X$ which at the same time is in $\mathcal{Q}(\mathcal{S}_X,\mathcal{U})$, is clearly a subcoalgebra of $\mathcal{U}$. It follows that $\mathcal{B}$ is a conjunct sum of subcoalgebras of $\mathcal{U}$, thus $\mathcal{B} \in \mathcal{K}$. □

7.4. **Completeness of $\mathcal{S}et_F$.** A category $\mathcal{C}$ is called *complete* if limits of arbitrary diagrams $\mathcal{D}$ exist. It is called *co-complete*, if all possible co-limits exist. We have already seen in theorem 4.2 that $\mathcal{S}et_F$ is co-complete. A standard exercise in category theory shows that $\mathcal{C}$ is complete if and only if equalizers and arbitrary products exist. According to theorem 6.1, equalizers exist in $\mathcal{S}et_F$, so we only need to concentrate on products.

To show the existence of products, we shall need to assume that the cofree coalgebra $\mathcal{S}_X$ exists for every set X. We have seen that for this it is enough to assume that the functor F is bounded, so we shall conveniently use this assumption.

In constructing products of coalgebras, we start with a special case, the product of a family of cofree coalgebras $\mathcal{S}_{X_i}$, $i \in I$. The following lemma is easy to check directly from the definitions of products and co-freeness:

Lemma 7.22. *If F is bounded, then the product of any family of cofree coalgebras exists, in fact*

$$\Pi_{i\in I}\mathcal{S}_{X_i} \cong \mathcal{S}_{\Pi_{i\in I}X_i}.$$

Again assuming boundedness of F, every coalgebra $\mathcal{A}$ can be embedded in some cofree coalgebra – take $\mathcal{S}_A$ with the homomorphism $\tilde{id_A} : \mathcal{A} \to \mathcal{S}_A$ which extends the coloring map $id_A : A \to A$. Therefore, we get from theorem 6.2:

Theorem 7.23 ([Wor98],[GS99b]). *If the functor F is bounded then $\mathcal{S}et_F$ is complete.*

A careful inspection of the ingredients of the proof reveals that in fact the same result holds for an arbitrary covariety in place of $\mathcal{S}et_F$.

7.5. **Forbidden behaviours and Birkhoff's theorem.** In universal algebra varieties are definable by equations. The famous Birkhoff theorem states that a class of universal algebras can be defined by a set of equations if and only if this class is a variety. What is the right co-algebraic notion to replace equations? Traditional universal algebra uses comparatively well behaved type functors. They are "polynomial functors" of the form $F(-) = \Sigma_{i\in I}(-)^{n_i}$. In particular, these functors are bounded, so the (absolutely) free algebra $\mathcal{F}(X)$ exist, for every set X. Moreover, its elements have a "syntactical" description, derived from the construction of F out of sums and powers. In the co-algebraic approach, we shall also need to require that the functor F is bounded. As a consequence, cofree coalgebras exist for every set Y. But more importantly, there is one set X large enough so that we really need only consider one cofree coalgebra, $\mathcal{S}_X$. We shall keep this set X fixed for the rest of this section. If F is bounded by κ (see page 43), then it suffices to take any fixed set X of cardinality at least κ. The material of this section had previously been proven only for functors preserving weak pullbacks in [Gum98a, Gum99].

The cofree coalgebra over the 1-element set $\{x\}$ is nothing but the final coalgebra. Its elements can be identified with all possible behaviours, up to bisimulation. Cofree coalgebras over larger color sets, can by the same token, be identified with some parameterized behaviour, or some *behaviour pattern.*

We could now interpret the previous theorem in such a way that a covariety $\mathcal{K}$ is always specified by a set $\mathcal{U}$ of permissible behaviours as $\mathcal{K} = \mathcal{Q}(S_X, \mathcal{U})$. To get local validity of a behaviour u at a point $a \in A$ in some coalgebra $\mathcal{A}$ we could define:

$$\begin{aligned} u \text{ holds at } a \in A \quad &:\Leftrightarrow \quad \exists_{g:A\to X}.\tilde{g}(a) = u \\ u \text{ holds in } \mathcal{A} \quad &:\Leftrightarrow \quad \exists_{a\in A}.u \text{ holds at } a. \end{aligned}$$

A covariety would then be defined by a disjunction of behaviours, $\bigvee_{u \in U} u$.

It is a matter of taste that we opt for the negated version of the above notion of validity. Our choice is dictated, perhaps, by a dislike of infinite disjunction, but, in a positive sense, by the experience that in many branches of mathematics classes of models are conveniently defined by the *exclusion of particular configurations.* Modular lattices, for instance, are characterized by the exclusion of a certain five-element lattice, planar graphs by the exclusion of two small graphs, K_5 and $K_{3,3}$, etc. With such examples in mind, we define:

Definition 7.24. *Let F be a bounded functor, $\mathcal{A}$ a coalgebra and $a \in A$. A* behaviour pattern *is an element v of any cofree coalgebra $\mathcal{S}_Y$. We say that v* holds at *a, if for every coloring $g : A \to Y$ we have that $\tilde{g}(a) \neq v$. We say that $\mathcal{A}$* satisfies *v, and we write*

$$\mathcal{A} \models v,$$

if v holds at every $a \in A$. For a set V of behaviours, we say $\mathcal{A} \models V$, provided $\mathcal{A} \models v$ for every $v \in V$.

Let now $V \subseteq S_X$ be any set of behaviour patterns and $\mathcal{K}$ a class of coalgebras. We define:

$$\begin{aligned} \mathcal{M}od(V) &:= \{\mathcal{A} \in \mathcal{S}et_F \mid \forall v \in V. \mathcal{A} \models v\}, \text{ and} \\ \mathcal{B}eh(\mathcal{K}) &:= \{v \in S_X \mid \forall \mathcal{A} \in \mathcal{K}. \mathcal{A} \models v\}. \end{aligned}$$

Then we have :

Lemma 7.25. *$\mathcal{M}od(V)$ is a co-variety for any set of behaviour patterns V.*

Proof. Put $U := [S_X - V]$, the largest subcoalgebra of $\mathcal{S}_X$ which is contained in the complement of V. Obviously, $\mathcal{A} \models V$ iff $\mathcal{A} \in \mathcal{Q}(S_X, U)$. $\square$

Lemma 7.26. *$S_X - Beh(\mathcal{K})$ is a subcoalgebra of S_X and it is contained in $\Sigma_C \mathcal{H}(\mathcal{K})$.*

Proof. By definition, $S_X - Beh(\mathcal{K}) = \bigcup\{\tilde{g}[A] \mid g \in X^A, \mathcal{A} \in \mathcal{K}\}$. This is a union of subcoalgebras of $\mathcal{S}_X$, each of which is a homomorphic image of some $\mathcal{A} \in \mathcal{K}$. Let $\mathcal{U}$ be the subcoalgebra of $\mathcal{S}_X$ with base set $S_X - Beh(\mathcal{K})$, then $\mathcal{U}$ is a a conjunct sum of homomorphic images of coalgebras in $\mathcal{K}$, i.e. $\mathcal{U} \in \Sigma_C \mathcal{H}(\mathcal{K})$. $\square$

Finally, the following is the coalgebraic counterpart of the famous Birkhoff theorem from universal algebra:

Theorem 7.27 (Coalgebraic Birkhoff Theorem)**.** *Every co-variety can be specified by a set of behaviour patterns, that is for an arbitrary class $\mathcal{K}$ of coalgebras we have*

$$\mathcal{SH}\Sigma(\mathcal{K}) = \mathcal{M}od(\mathcal{B}eh(\mathcal{K})).$$

Proof. By definition, $\mathcal{M}od(\mathcal{B}eh(\mathcal{K})) = \mathcal{Q}(S_X, U)$ where $U = \bigcup\{\tilde{g}[A] \mid g \in X^A, \mathcal{A} \in \mathcal{K}\}$ and $\mathcal{B}eh(\mathcal{K}) = \mathcal{S}_X - U$. Consequently, $\mathcal{M}od(\mathcal{B}eh(\mathcal{K}))$ is a covariety, which contains $\mathcal{K}$, so $\mathcal{SH}\Sigma(\mathcal{K}) \subseteq \mathcal{M}od(\mathcal{B}eh(\mathcal{K}))$. For the converse, let $\mathcal{B} \in \mathcal{M}od(\mathcal{B}eh(\mathcal{K}))$, then for every $b \in B$ there is a subcoalgebra $\mathcal{U}_b \leq \mathcal{B}$ so that $b \in U_b$ and $|U_b| \leq |X|$. Hence $\mathcal{U}_b \cong \mathcal{G}_b$ for some subcoalgebra $\mathcal{G}_b$ of

$\mathcal{S}_X$. We must have $\mathcal{G}_b \in \mathcal{M}od(\mathcal{B}eh(\mathcal{K}))$ too, so $\mathcal{G}_b \leq \mathcal{U}$. But $\mathcal{U} \in \Sigma_C \mathcal{H}(\mathcal{K})$, each $\mathcal{G}_b \in \mathcal{S}\Sigma_C(\mathcal{K})$, and $\mathcal{B}$ is a conjunct sum of the $\mathcal{G}_b$, so $\mathcal{B} \in \Sigma_C \mathcal{S}\Sigma_C(\mathcal{K}) \subseteq \mathcal{S}\mathcal{H}\Sigma(\mathcal{K})$. ☐

7.6. **Exercises.**

Exercise 7.1 (Covarieties). *Complete the proof of lemma 6.2.*

Exercise 7.2 (Injectives). *A coalgebra $\mathcal{C}$ is called* injective, *if the following holds: For all coalgebras $\mathcal{B}$ and every subcoalgebra $\mathcal{A}$ of $\mathcal{B}$ and every homomorphism $\varphi : \mathcal{A} \to C$ there exists at least one homomorphism $\tilde{\varphi} : \mathcal{A} \to \mathcal{C}$ with $\tilde{\varphi} \circ \ \leq = \ \varphi$.*

$$\begin{tikzcd} B \arrow[dr, dashed, "\tilde{\varphi}"] & \\ A \arrow[u, hook, "\leq"] \arrow[r, "\varphi"'] & C \end{tikzcd}$$

1. *Show that injective coalgebras are weakly final, i.e. if $\mathcal{C}$ is injective, then for every coalgebra $\mathcal{A} \in \mathcal{S}et_F$ there is at least one homomorphism $\varphi : \mathcal{A} \to \mathcal{C}$.*
2. *Show that every cofree coalgebra is injective.*

Exercise 7.3 (Injectives are retracts). *A* retraction *is a pair of morphisms $\delta : A \to B$ and $\kappa : B \to A$ so that $\kappa \circ \delta = id_A$. In this case, A is called a* retract *of B.*

Assume that the functor F is bounded. Show that an F-coalgebra $\mathcal{A}$ is injective iff $\mathcal{A}$ is a retract of some cofree coalgebra.

Exercise 7.4 (Products of cofree coalgebras). *Assume that F is bounded. Let $(X_i)_{i \in I}$ be a family of sets. Show that the product $\Pi_{i \in I} \mathcal{S}_{X_i}$ in $\mathcal{S}et_F$ of the cofree coalgebras $\mathcal{S}_{X_i}$ is given by $\mathcal{S}_{\Pi_{i \in I} X_i}$, i.e. the cofree coalgebra over the product in $\mathcal{S}et$ of the X_i.*

Exercise 7.5 (Relatively cofree coalgebras). *Let $\mathcal{K}$ be a class of coalgebras and X a set. A coalgebra $\mathcal{S}_X^{\mathcal{K}}$ together with a map (a coloring) $\varepsilon : \mathcal{S}_X^{\mathcal{K}} \to X$ is called* cofree over X with respect to $\mathcal{K}$, *if for every $\mathcal{A} \in \mathcal{K}$ and for every coloring $f : A \to X$ there is exactly one homomorphism $\tilde{f} : \mathcal{A} \to \mathcal{S}_X^{\mathcal{K}}$ with $\varepsilon \circ \tilde{f} = f$.*

$$\begin{tikzcd} & X \\ A \arrow[ur, "f"] \arrow[r, dashed, "\tilde{f}"'] & S_X^{\mathcal{K}} \arrow[u, "\varepsilon"'] \end{tikzcd}$$

Let F be bounded and $\mathcal{K}$ a class of coalgebras. Show that

1. *$\mathcal{S}_X^{\mathcal{K}}$ exists,*
2. *if $\mathcal{K}$ is a co-quasivariety (see definition 6.2) then $\mathcal{S}_X^{\mathcal{K}} \in \mathcal{K}$.*

8. Special Functors

Most functors which we have considered, share an extra property that we have not yet made use of: They *preserve weak pullbacks*. This property will be seen to have far reaching consequences for their structure theory. We shall study this condition and a collection of somewhat weaker requirements on F. The material of this section is from [GS99b].

8.1. **Split epis.**

Definition 8.1. *Amongst the objects of a category $\mathcal{C}$ we can define a natural relation $\preceq$ as follows:*

$$A \preceq B :\Leftrightarrow \text{there are morphisms } \tau : B \to A \text{ and } \delta : A \to B \text{ with } \tau \circ \delta = id_A$$

It follows that δ is mono and τ is epi. Such a τ is also called *split epi.* The relation $\preceq$ is a quasi ordering, i.e. $\preceq$ is

- reflexive:
  $\forall A \in \mathcal{C}_o.A \preceq A,$
- transitive:
  $\forall A, B, C \in \mathcal{C}_o.A \preceq B, B \preceq C \implies A \preceq C.$

In the category of sets we have $A \preceq B \iff |A| \leq |B|$ therefore, according to the Schröder-Bernstein theorem [3] $\preceq$ is

- anti-symmetric:
  $\forall A, B \in \mathcal{C}_o.A \preceq B, B \preceq A \implies A \cong B.$

The relation $\preceq$ is preserved by any functor F:

- Monotony:
  $\forall A, B \in \mathcal{C}_o.A \preceq B \implies F(A) \preceq F(B).$

Every split epi is an epi. In $\mathcal{S}et$ we have the converse, in other categories this need not be the case.

8.2. **Weak limit preservation.**

Definition 8.2. *Let $F : \mathcal{S}et \to \mathcal{S}et$ be a functor and D a diagram. We say that*

- F weakly preserves D-limits, *if F transforms every limit cone over D into a weak limit cone over $F(D)$, i.e. for every limit $(L, (\nu_i)_{i \in I})$ of diagram D we get that $(F(L), (F(\nu_i))_{i \in I})$ is a weak limit over the transformed diagram $F(D)$*
- F preserves weak D-limits *if it transforms every weak limit cone over D into a weak limit cone over $F(D)$,*

Fortunately, the fine linguistic difference between "F preserves weak limits" and "F weakly preserves limits" are easily seen to disappear in every category where all D-limits exist. This is an easy consequence of the following observation:

Lemma 8.3. *Let D be a diagram in an arbitrary category $\mathcal{C}$.*

1. *If $(W, (w_i)_{i \in I})$ is a weak limit of D and $W \preceq W'$ with split epi $\tau : W' \to W$, then $(W', (w_i \circ \tau)_{i \in I})$ is also a weak limit of D.*
2. *If the limit $(L, (p_i)_{i \in I})$ of D exists, then $(W, (w_i)_{i \in I})$ is a weak limit of D if and only if $w_i = p_i \circ \tau$ for some split epi $\tau : W \to L$, in particular, $L \preceq W$.*

[3]Given sets A and B and injective maps $f : A \to B$ and $g : B \to A$ then there is a bijection between A and B

Thus, with respect to the order $\preceq$, introduced on the objects of a category, limits, if they exist, are just the infima of all weak limits.

Corollary 8.4. *Let $\mathcal{C}$ be a category and D a diagram so that every D-limit exists in $\mathcal{C}$. Then F preserves weak D-limits if and only if F weakly preserves D-limits.*

Theorem 8.5. *Let C be a category in which every D-limit exists, then an endo-functor F preserves weak D-limits if and only if for every D-limit $(L,(p_i)_{i\in I})$ there exists a morphism δ from the limit $(Q,(q_i)_{i\in I})$ of $F(D)$ to $F(L)$, so that $F(p_i)\circ\delta = q_i$ for all i.*

Proof. Let $(L,(p_i)_{i\in I})$ be the limit of D. We wish to show that $(F(L),(F(p_i))_{i\in I})$ is a weak limit of the transformed diagram $F(D)$. Consider the limit $(Q,(q_i)_{i\in I})$ of $F(D)$. There is a (unique) morphism $\tau : F(L)\to Q$ with $F(p_i) = q_i\circ\tau$ for all $i\in I$. If $F(L)$ is to be a weak limit then there must be at least one morphism $\delta : Q\to F(L)$ with $F(p_i)\circ\delta = q_i$.

$$
\begin{tikzcd}
L \arrow[dr, "p_i"] & \\
& D_i
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
F(L) \arrow[d, bend left, "\tau"] \arrow[dr, "F(p_i)"] & \\
Q \arrow[u, dashed, bend left, "\delta"] \arrow[r, "q_i"'] & F(D_i)
\end{tikzcd}
$$

Conversely, if there is such a morphism δ then $q_i\circ\tau\circ\delta = q_i\circ id_Q$, so $\tau\circ\delta = id_Q$. Hence $\tau : F(L)\to Q$ is split epi, so $F(L)$ is a weak limit from lemma 8.3. $\square$

8.3. **Weak κ-pullbacks and their preservation.** We illustrate the above situation in the case of a pullback (L,p_1,p_2) of two morphisms $f : A\to C$ and $g : B\to C$.

$$
\begin{tikzcd}
L \arrow[r, "p_1"] \arrow[d, "p_2"'] & A \arrow[d, "f"] \\
B \arrow[r, "g"'] & C
\end{tikzcd}
$$

Let Q the limit of the transformed diagram, then F weakly preserves pullbacks iff there is a map $\delta : Q\to F(L)$ with $F(p_i)\circ\delta = q_i$ for all i.

$$
\begin{tikzcd}
Q \arrow[dr, shift left, "\delta"] \arrow[drr, bend left, "q_1"] \arrow[ddr, bend right, "q_2"'] & & \\
& F(L) \arrow[ul, shift left, "\tau"] \arrow[r, "F(p_1)"'] \arrow[d, "F(p_2)"] & F(A) \arrow[d, "F(f)"] \\
& F(B) \arrow[r, "F(g)"'] & F(C)
\end{tikzcd}
$$

The pullback of two morphisms $f : A\to C$ and $g : B\to C$ in $\mathcal{S}et$ has as object the set $pb(f,g) = \{(a,b) \mid f(a) = g(b)\}$ and as morphisms the canonical projections π_1 and π_2. From this we get an easy criterion for a $\mathcal{S}et$-endofunctor to preserve weak pullbacks:

Theorem 8.6. *A functor $F : \mathcal{S}et \to \mathcal{S}et$ preserves weak pullbacks, iff for all maps $f : A \to C$ and $g : B \to C$ we have: Given u and v with $F(f)(u) = F(g)(v)$ then there is a $w \in F(\{(x,y) \mid f(x) = g(y)\})$ with $F(\pi_1)(w) = u$ and $F(\pi_2)(w) = v$.*

Corollary 8.7. *F weakly preserves the pullback of f and g iff the map $F(\pi_1) \times F(\pi_2) : F(pb(f,g)) \to pb(F(f), F(g))$ is onto.*

We will need to generalize the notion of pullback to consider pullbacks of a family of arrows with common codomain.

Definition 8.8. *Let κ be an ordinal. A κ-*sink *is a family $(f_i : A_i \to A)_{i\in\kappa}$ of morphisms with common codomain. A κ-pullback is the limit $(P, (\pi_i)_{i\in\kappa})$ of a κ-sink. A* weak κ-pullback *is a weak limit of a κ-sink.*

Thus, a 2-pullback is just an ordinary pullback. There has occasionally been slight confusion in the literature concerning the notion of "preservation of weak pullbacks". There are functors preserving weak pullbacks, which do not preserve weak κ-pullbacks for $\kappa \geq \omega$. An example is the *filter functor*, see [Gum98b].

The category of sets is *complete* (and *cocomplete*), which is to say that all limits (and all colimits) exist. κ-pullbacks, in particular, are constructed similar as in the finite case:

Proposition 8.9. *Let $(f_k : A_k \to A)_{k\in\kappa}$ be a κ-sink. Then*

$$pb((f_k)_{k\in\kappa}) = \{(a_k)_{k\in\kappa} \mid \forall i,j \in \kappa. f_i(a_i) = f_j(a_j)\}$$

with canonical projections π_j defined as $\pi_j((a_k)_{k\in\kappa}) = a_j$ is the κ-pullback of the f_k in $\mathcal{S}et$. A functor F preserves weak κ-pullbacks iff for every family $(u_k)_{k\in\kappa} \in pb(F((f_k)_{k\in\kappa}))$ there exists an element $w \in F(pb((f_k)_{k\in\kappa}))$ with $F(\pi_k)(w) = u_k$ for all $k \in \kappa$.

Example 8.10. *Most functors considered so far preserve weak κ-pullbacks for arbitrary κ. Amongst those are:*

1. *The constant functor $F(X) = A$ for a fixed set A.*
2. *The identity functor $\mathcal{I}d$.*
3. *$F(X) = A + X$ for a fixed set A.*
4. *$F(X) = X^\Sigma$ for a fixed set Σ.*
5. *The power set functor $2^{(-)}$.*

The following lemma allows us to combine the above examples. Most of the practically relevant coalgebras have a type which arises in such a way.

Lemma 8.11. *If functors F and G preserve weak κ-pullbacks for some κ, then so do $F \circ G$ and $F \times G$.*

Example 8.12. *The filter functor (see page 16) can be shown to preserve weak κ-pullbacks if and only if $\kappa < \omega$ (see* [Gum98b]*).*

Example 8.13. *The functor $(-)^3_2$ does not preserve weak pullbacks. However, if at least one of $f : A \to C$, $g : B \to C$ is injective, then the functor weakly preserves the pullback of f and g. We say, that $(-)^3_2$ weakly preserves pullbacks* along injective maps.

Proof. Given $f : A \to C$, $g : B \to C$, $(u_1, u_2, u_3) \in A_2^3, and (v_1, v_2, v_3) \in B_2^3$ with $f(u_i) = g(v_i)$. We must find $w \in \{(x,y) \mid f(x) = g(y)\}_2^3$ so that $(\pi_1)_2^3(w) = u$ and $(\pi_2)_2^3(w) = v$. The only possibility is $w = ((u_1, v_1), (u_2, v_2), (u_3, v_3))$. It remains to show that $w \in (A \times B)_2^3$, i.e. that two of the components of w are equal. We may assume w.l.o.g. that $u_1 = u_2$. If $v_1 = v_2$ then we are done. The other case is, again w.l.o.g, that $v_2 = v_3$. If f is injective then $f(u_2) = g(v_2) = g(v_3) = f(u_3)$, so $(u_2, v_2) = (u_3, v_3)$, similarly, if g is injective then $(u_1, v_1) = (u_2, v_2)$.

To see that $(-)_2^3$ does not preserve weak pullbacks of arbitrary maps, choose $f = g$ the constant map $\{0, 1\} \to \{0\}$. For $u = (0, 0, 1)$ and $v = (1, 0, 0)$ we cannot find the required w. □

8.4. **Preservation theorems.** In the following we shall consider, and characterize, several particular properties which the functor F might have in regard to the weak preservation of particular types of pullbacks. It is not clear yet whether all these properties are really distinct. The material of this section is from [GS99a].

Clearly, if F weakly preserves pullbacks, then

- F weakly preserves kernel pairs,
- F weakly preserves pullbacks along injective maps,
- F weakly preserves pullbacks of injective maps.

We shall study these conditions and characterize them by way of their structure theoretical consequences.

8.4.1. *F weakly preserving pullbacks.* The following theorem characterizes functors weakly preserving pullbacks. The implications $1. \to 2 \to 3.$ are due to Rutten [Rut96].

Theorem 8.14. *For a functor F the following are equivalent:*

1. *F weakly preserves pullbacks.*
2. *The pullback of two homomorphisms is a bisimulation*
3. *The relational product $R \circ S$ of two bisimulations is a bisimulation.*

The key in proving the reverse direction of this and some later theorems is:

Lemma 8.15. *Let $F : \mathcal{S}et \to \mathcal{S}et$ be a functor and $f : A \to C$ and $g : B \to C$ be maps. Then the following are equivalent:*

1. *F weakly preserves the pullback of f and g.*
2. *$\{(a, b) \mid f(a) = g(b)\}$ is a bisimulation for each coalgebra structure on A, B, and C for which f and g are homomorphisms .*
3. *For each pair $(x, y) \in pb(f, g)$ and for all coalgebra structures on A, B, and C making f and g homomorphisms there is a 2-source (Q, p_1, p_2) and an element $q \in Q$ with $f \circ p_1 = g \circ p_2$, $p_1(q) = x$ and $p_1(q) = y$.*

Proof. The equivalence of 2. and 3. is rather straightforward: If $pb(f, g)$ is a bisimulation, then $(pb(f, g), \pi_1, \pi_2)$ is an appropriate 2-source. Conversely, from a 2-source commuting with f and g we get the standard bisimulation as a subset of $pb(f, g)$. The extra condition guarantees that the sum of all such 2-sources has precisely $pb(f, g)$ as standard bisimulation.

The implication (1. $\to$ 2.) is due to Rutten([Rut96]): Let (P, π_1, π_2) be the pullback of the homomorphisms $f : A \to C$ and $g : B \to C$. Then $(F(P), F(\pi_1), F(\pi_2))$, by assumption, is a weak pullback of $F(f) : F(A) \to F(C)$ and $F(g) : F(B) \to F(C)$. Since f and g are homomorphisms, we calculate

$$\begin{aligned} F(f) \circ \alpha_A \circ \pi_1 &= \alpha_C \circ f \circ \pi_1 \\ &= \alpha_C \circ g \circ \pi_2 \\ &= F(g) \circ \alpha_B \circ \pi_2 \end{aligned}$$

$$\begin{tikzcd}
 & B \arrow[rr, "g"] \arrow[dd] & & C \arrow[dd, "\alpha_C"] \\
P \arrow[ur, "\pi_2"] \arrow[rr, "\pi_1"] \arrow[dd, dashed, "\alpha_P"] & & A \arrow[ur, "f"'] \arrow[dd, "\alpha_A"] & \\
 & F(B) \arrow[rr, "F(g)"] & & F(C) \\
F(P) \arrow[ur, "F(\pi_2)"] \arrow[rr, "F(\pi_1)"'] & & F(A) \arrow[ur, "F(f)"'] &
\end{tikzcd}$$

This makes $(P, \alpha_A \circ \pi_1, \alpha_B \circ \pi_2)$ a competitor of the weak limit $(F(P), F(\pi_1), F(\pi_2))$ from which we get the desired mediating map α_P with $\alpha_A \circ \pi_1 = F(\pi_1) \circ \alpha_P$ and $\alpha_B \circ \pi_2 = F(\pi_2) \circ \alpha_P$.

2. $\to$ 1.: Let $f : A \to C$ and $g : B \to C$ be maps and (P, π_1, π_2) their pullback. Consider the pullback (Q, p_1, p_2) of $F(f)$ and $F(g)$. We need to construct a map $\delta : Q \to F(P)$ with $F(\pi_i) \circ \delta = p_i$.

Take any $q \in Q$ and define structure maps α_A^q, α_B^q, and α_C^q on $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ as the constant functions

$$\begin{aligned} \alpha_A^q &= \lambda a.p_1(q), \\ \alpha_B^q &= \lambda b.p_2(q), \text{ and} \\ \alpha_C^q &= \lambda c.F(f)(p_1(q)) = \lambda c.F(f)(p_2(q)). \end{aligned}$$

It is easy to see that f and g are homomorphisms with respect to these structures. We therefore find a structure map α_P^q on P turning P into a bisimulation. We finally set

$$\delta(q) = \alpha_P(p)$$

for an arbitrary $p \in P$. By construction,

$$\begin{aligned} F(\pi_1)(\delta(q)) &= F(\pi_1)(\alpha_P^q(p)) \\ &= \alpha_A^q(\pi_1(p)) \\ &= p_1(q). \end{aligned}$$

Similarly, $F(\pi_2) \circ \delta(q) = p_2(q)$, hence $p_i = F(\pi_i) \circ \delta$ as required.

$$
\begin{tikzcd}
 & & B \arrow[rr, "g"] \arrow[dd] & & C \arrow[dd, "\alpha^q_C"] \\
 & P \arrow[ur, "\pi_2"] \arrow[rr, "\pi_1"] \arrow[dd, dashed, "\alpha^q_P"] & & A \arrow[ur, "f"] \arrow[dd, "\alpha^q_A"] & \\
 & & F(B) \arrow[rr, "F(g)"] & & F(C) \\
Q \arrow[r, dotted, "\delta"] \arrow[urr, bend left, "p_2"] \arrow[rrr, bend right, "p_1"'] & F(P) \arrow[ur, "F(\pi_2)"] \arrow[rr, "F(\pi_1)"'] & & F(A) \arrow[ur, "F(f)"] &
\end{tikzcd}
$$

□

We now finish the proof of the theorem 8.14. The equivalence of 1. and 2. follows directly from the above lemma. Let now R, resp. S, be bisimulations between $\mathcal{A}$ and $\mathcal{B}$, resp. $\mathcal{B}$ and $\mathcal{C}$. The pullback of the projections $\pi_2^R : R \to B$ and $\pi_1^S : S \to B$ is $R \bowtie S := \{((a,b)(b,c)) \mid (a,b) \in R, (b,c) \in S\}$. By assumption, this can be equipped with a coalgebra structure, making $\pi_1^R \circ \pi_1$ and $\pi_2^S \circ \pi_2$ into homomorphisms. $R \circ S$ is just the canonical bisimulation for this 2-source.

$$
\begin{tikzcd}
 & & R \bowtie S \arrow[rr, dashed, two heads] \arrow[dl, "\pi_1"'] \arrow[dr] & & R \circ S \arrow[dd] \arrow[ddllll, bend right] \\
 & R \arrow[dl, "\pi_1^R"'] \arrow[dr] & & S \arrow[dl] \arrow[dr] & \\
A & & B & & C
\end{tikzcd}
$$

Conversely, assume that the relational product of any two bisimulations is a bisimulation. Let $\varphi : A \to C$ and $\psi : B \to C$ be homomorphisms. $G(\varphi)$ and $G(\psi)^-$ are bisimulations, hence also $G(\varphi) \circ G(\psi)^-$. This is nothing but $pb(\varphi, \psi)$, the pullback of φ and ψ in $\mathcal{S}et$.

8.4.2. *F weakly preserving kernels.* Recall that a *kernel* is the pullback of a morphism $f : A \to B$ with itself.

Theorem 8.16 ([GS99a]). *F weakly preserves kernels if and only if every congruence relation is a bisimulation.*

Proof. Every congruence relation θ is the kernel of a homomorphism $\pi_\theta : A \to A/\theta$, that is $\theta = pb(\pi_\theta, \pi_\theta)$. If F weakly preserves this pullback, then θ is a bisimulation by lemma 8.15.

Conversely, for every structure on A and B for which $\varphi : A \to B$ is a homomorphism, its pullback $pb(\varphi, \varphi)$ is a congruence relation, hence a bisimulation. By lemma 8.15, F weakly preserves the pullback of φ with itself, i.e. the kernel (in $\mathcal{S}et$) of φ. □

Lemma 8.17. *If F weakly preserves kernels then the largest bisimulation $\sim_A$ on a coalgebra $\mathcal{A}$ is transitive, in fact it agrees with the largest congruence relation on $\mathcal{A}$.*

Proof. Theorem 8.16 implies that every congruence θ is contained in $\sim$. On the other hand, lemma 5.15 shows that there is a congruence $\langle\sim\rangle$ containing $\sim$, consequently, $\sim = \langle\sim\rangle$. □

Corollary 8.18. *If F weakly preserves kernels then every mono in $\mathcal{S}et_F$ is injective.*

Proof. If F weakly preserves kernels, then $ker(\varphi) = [ker(\varphi)]$, hence if $\varphi : A \to C$ is mono we have $ker(\varphi) = [ker(\varphi)] = \Delta_A$. □

8.4.3. *F weakly preserving pullbacks along injectives.* F is said to *weakly preserve pullbacks along monos* if F weakly preserves pullbacks of $f, g : A \to B$ whenever f or g is mono.

This condition on F is properly weaker than the condition of preserving arbitrary pullbacks, for we have seen that the functor $(-)^3_2$ has this property yet it does not weakly preserve pullbacks of two arbitrary maps.

Theorem 8.19 ([GS99a]). *Let F be a $\mathcal{S}et$-endofunctor, then the following are equivalent:*

1. *F weakly preserves pullbacks along monos.*
2. *If $\mathcal{U} \leq \mathcal{B}$ and R is a bisimulation between $\mathcal{A}$ and $\mathcal{B}$, then $R^-(U) := \{a \in A \mid \exists u \in U.(a,u) \in R\}$ is a subcoalgebra of $\mathcal{A}$.*
3. *If $\varphi : \mathcal{A} \to \mathcal{B}$ is a homomorphism and $\mathcal{U} \leq \mathcal{B}$ is a subcoalgebra, then $\varphi^{-1}(U)$, the pre-image of $\mathcal{U}$ under φ, is a subcoalgebra of $\mathcal{A}$.*

Proof. 1. $\to$ 2.: Assume that F preserves pullbacks along monos and $R \subseteq A \times B$ is a bisimulation. The pullback in $\mathcal{S}et$ of $\leq : U \to B$ and $\pi_2 : R \to B$ is $Q = \{(u,(a,u)) \mid (a,u) \in R\}$. By assumption, Q must be a bisimulation, hence, there is a structure map on Q so that $\pi_2^Q : Q \to R$, and consequently, $\pi_1^R \circ \pi_2^Q : Q \to A$ a homomorphism. Its image, which is just $R^-(U)$, is a subcoalgebra of $\mathcal{B}$.

2. $\to$ 3.: This is a specialization with $R = G(\varphi)$.

3. $\to$ 1.: Let $\varphi : \mathcal{A} \to \mathcal{C}$ and $\psi : \mathcal{B} \to \mathcal{C}$ be homomorphisms with ψ injective. The epi-mono-factorization of ψ yields a subcoalgebra $\mathcal{B}'$ of $\mathcal{C}$ and an isomorphism $\tilde{\psi} : \mathcal{B} \to \mathcal{B}'$ so that $\psi = \leq_{B'} \circ \tilde{\psi}$. $(\varphi^-(B'), \leq, \varphi)$ is a source in $\mathcal{S}et_F$. Finally, $(\varphi^-(B'), \leq \tilde{\psi}^- \circ \varphi)$ is a source in $\mathcal{S}et_F$ whose canonical bisimulation is precisely the pullback in $\mathcal{S}et$ of φ and ψ. □

8.4.4. *F weakly preserving κ-pullbacks of injective maps.* The only-if direction of the following characterization is again due to Rutten([Rut96]), the other one is from [GS99a]:

Theorem 8.20. *F preserves weak κ-pullbacks of injective maps if and only if the intersection of a κ-family of subcoalgebras is a subcoalgebra.*

Proof. The intersection of a family $(\mathcal{U}_k)_{k \in \kappa}$ of subcoalgebras of $\mathcal{A}$ is just the pullback of their embedding. Thus, if F preserves weak pullbacks of these embeddings, the pullback is a bisimulation, so in this case, the intersection is a subcoalgebra.

Conversely, suppose that the intersection of a κ-family of subcoalgebras is a subcoalgebra. We shall present the proof for $\kappa = 2$, the general case is proven the same way. Let $\varphi : \mathcal{A} \to \mathcal{C}$ and $\psi : \mathcal{B} \to \mathcal{C}$ be injective coalgebra homomorphisms. We try to fulfill the condition of lemma 8.15 For any (x,y) with $\varphi(x) = \psi(y)$, we need to find a 2-source (Q, p_1, p_2) and a $q \in Q$ with $\varphi \circ p_1 = \psi \circ p_2$ and $p_1(q) = x$, $p_2(p) = y$.

We start with the epi-mono-factorizations $\leq \circ \tilde{\varphi} = \varphi$ and $\leq \circ \tilde{\psi} = \psi$. Then $\tilde{\varphi}$ and $\tilde{\psi}$ are isomorphisms with $\mathcal{A} \cong \mathcal{A}' = \tilde{\varphi}[A]$ and $\mathcal{B} \cong \mathcal{B}' = \tilde{\psi}[B]$. The pullback, in $\mathcal{S}et$, of φ and ψ can be taken stepwise, as indicated in the following diagram. Since $\mathcal{A}' \cap \mathcal{B}'$ is a subcoalgebra, the innermost pullback is also a pullback in $\mathcal{S}et_F$. Continuing to the outside, we are taking pullbacks along isomorphisms, which always exist in $\mathcal{S}et_F$. Thus $(\mathcal{Q}, \pi_1, \pi_2)$ is a source as required, i.e. $\varphi \circ \pi_1 = \psi \circ \pi_2$, and $\pi_1(q) = x$, $\pi_2(p) = y$.

$$
\begin{tikzcd}
A \arrow[r, "\tilde{\varphi}"] & A' \arrow[r, hook] & C \\
R \arrow[u] \arrow[r] & A' \cap B' \arrow[u, hook] \arrow[r, hook] & B' \arrow[u, hook] \\
Q \arrow[u] \arrow[r] & S \arrow[u] \arrow[r] & B \arrow[u, "\tilde{\psi}"']
\end{tikzcd}
$$

□

If the collection of all subcoalgebras is closed under intersection then for every subset $U \subseteq A$ there is a smallest subcoalgebra of $\mathcal{A}$ containing U. This will be denoted by $< U >$ and called the *subcoalgebra generated by* U. For an element $a \in A$ we write $< a >$ as a shorthand for $< \{a\} >$.

We have seen before that conjunctly irreducible coalgebras are just the one-generated ones. If subcoalgebras are closed under intersection then "one-generated" is indeed the same as generated by a 1-element set, i.e. $< a >$. Hence, by theorem 7.13, the collection of all $< a >$ with $a \in A$ forms a conjunct representation of $\mathcal{A}$ by conjunctly irreducibles. We thus get:

Theorem 8.21. [GS98] *If the functor F weakly preserves κ-pullbacks of injective maps for any κ, then every coalgebra is a conjunct sum of conjunctly irreducibles.*

Again the filter functor may serve as an example of a functor which preserves κ-pullbacks of injectives if and only if $\kappa < \omega$.

Corollary 8.22. *If F weakly preserves 2-pullbacks then the subcoalgebras of an F-coalgebra form a topological space. Every homomorphism between F-coalgebras is continuous and open with respect to the corresponding topologies. Conversely, every topological space arises as the collection of subcoalgebras of type $\mathcal{F}$, where $\mathcal{F}$ is the filter functor.*

In a certain sense, the converse is also true, see [Gum98b]. On every topological space (X, τ) we can define a coalgebra $\mathcal{A}_\tau = (X, \alpha_X)$ so that the open sets of τ become exactly the subcoalgebras of $\mathcal{A}$ and the continuous open maps between (X, τ) and (Y, σ) are exactly the homomorphisms between $\mathcal{A} = (X, \alpha_X)$ and $\mathcal{B} = (Y, \alpha_Y)$. The type F is given by the "filter functor" which associates to any set X the set $\mathcal{F}(X)$ of all filters on X. For a given topological space (X, τ) we define the structure map α by

$$\alpha(x) := \mathcal{U}(x)$$

where $\mathcal{U}(x)$ is the neighborhood filter of the point x.

References

[AM89] P. Aczel and N. Mendler, *A final coalgebra theorem*, Proceedings category theory and computer science (D.H. Pitt et al, ed.), Lecture Notes in Computer Science, Springer, 1989, pp. 357–365.

[Bar93] M. Barr, *Terminal coalgebras in well-founded set theory*, Theoretical Computer Science (1993), no. 114(2), 299–315.

[Bor94] F. Borceux, *Handbook of categorical algebra 1: basic category theory*, Cambrigde University Press, 1994.

[GS98] H.P. Gumm and T. Schröder, *Covarieties and complete covarieties*, Coalgebraic Methods in Computer Science (B. Jacobs et al, ed.), Electronic Notes in Theoretical Computer Science, vol. 11, Elsevier Science, 1998, pp. 43–56.

[GS99a] H.P. Gumm and T. Schröder, *Coalgebraic structure from weak limit preserving functors*, Preprint, 1999.

[GS99b] H.P. Gumm and T. Schröder, *On products and bisimulations*, Preprint, 1999.

[Gum98a] H.P. Gumm, *Equational and implicational classes of coalgebras*, Relational Methods in Logic, Algebra and Computer Science (A. Szalas E. Orlowska, ed.), Banach Center, Warsaw, 1998, Extended abstract.

[Gum98b] H.P. Gumm, *Functors for coalgebras*, Algebra Universalis, to appear., 1998.

[Gum99] H.P. Gumm, *Equational and implicational classes of coalgebras*, Theoretical Computer Science, to appear, 1999.

[JR97] B. Jacobs and J.J.M.M. Rutten, *A tutorial on (co)algebras and (co)induction*, EATCS Bulletin (1997), no. 62, 222–259.

[Lan71] S. Mac Lane, *Categories. for the working mathematician*, Graduate Texts in Mathematics, vol. 5, Springer Verlag, 1971.

[Rei95] H. Reichel, *An approach to object semantics based on terminal co-algebras*, Math. Struct. in Comp. Sci. (1995), no. 5, 129–152.

[Rut96] J.J.M.M. Rutten, *Universal coalgebra: a theory of systems*, Tech. report, Centrum voor Wiskunde en Informatica, 1996, Theoretical Computer Science, to appear.

[Rut98] J.J.M.M. Rutten, *Automata and coinduction (an exercise in coalgebra)*, Tech. report, Centrum voor Wiskunde en Informatica, 1998.

[Wor98] J. Worrell, *Toposes of coalgebras and hidden algebras*, Coalgebraic Methods in Computer Science (B. Jacobs et al, ed.), Electronic Notes in Theoretical Computer Science, vol. 11, Elsevier Science, 1998, pp. 215–233.

Philipps-Universität Marburg, 35032 Marburg, Germany

E-mail address: `gumm@mathematik.uni-marburg.de`