



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

ScienceDirect

Electronic Notes in  
Theoretical Computer  
Science

Electronic Notes in Theoretical Computer Science 203 (2008) 243–262

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

# Coequational Logic for Finitary Functors

Daniel Schwencke<sup>1,2</sup>

*Department of Theoretical Computer Science  
Institute of Technology,  
Braunschweig, Germany*

---

## Abstract

Coequations, which are subsets of a cofree coalgebra, can be viewed as properties of systems. In case of a polynomial functor, a logic of coequations was formulated by J. Adámek. However, the logic is more complicated for other functors than polynomial ones, and simple deduction rules can no longer be formulated. A simpler coequational logic for finitely branching labelled transition systems was later presented by the author. The current paper carries that research further: it yields a simple coequational logic for finitary functors that preserve preimages. Furthermore we prove a statement for semantical consequences of sets of coequations in the case of accessible functors.

*Keywords:* cofree coalgebra, coequation, logic, finitary functor, preimage

---

## 1 Introduction

Several authors studied logics that formalise properties of systems described as coalgebras over a set functor. On the one hand, modal logics were investigated e.g. by Lawrence Moss [8] or Dirk Pattinson [9]. On the other hand, dualizing equational presentations of algebras, Jan Rutten [10] introduced coequations as subsets of cofree coalgebras. These coequations can also be used to establish logics of system properties, see e.g. the work of Jesse Hughes [7]. Our paper deals with a coequational logical topic different to the one of Hughes. In [1] Jiri Adámek formulated a logic of coequations and proved it to be sound and complete. Since the elements of a cofree coalgebra for a fixed functor can be viewed as “behaviour patterns” of coalgebras for this functor, coequations describe certain properties of such coalgebras. Thus with a logic of coequations we are able to obtain properties of a coalgebra automatically from other properties of this coalgebra.

Coequational logic provides a deduction system to derive coequations from a given set of assumed coequations. It is particularly simple in case of polynomial

---

<sup>1</sup> The grant of DFG is gratefully acknowledged.

<sup>2</sup> Email: [schwencke@iti.cs.tu-bs.de](mailto:schwencke@iti.cs.tu-bs.de)

functors: the cofree coalgebra consists of coloured trees, and the deduction system has only two rules, the subtree rule and the recolouring rule. The logic for accessible functors is more complicated.

In this paper we focus on finitary functors, a subclass of accessible functors. Our prime example is the finite power-set functor  $\mathcal{P}_f(A \times -)$ : the corresponding systems are the finitely branching labelled transition systems (LTS), and our paper is a generalisation of the logic for LTS presented in the thesis [11]. That logic for LTS is based on the presentation of the cofree  $\mathcal{P}_f(A \times -)$ -coalgebra as the set of all “strongly extensional” trees, which were introduced by James Worrell in [12]. Coming back to an abstract finitary functor  $H$ , we show how to obtain a presentation of its cofree coalgebra as a set of trees. Using that presentation, under the relatively weak assumption that  $H$  preserves preimages we formulate a coequational logic which is nearly as simple as the one for polynomial functors and prove it to be sound and complete. This way we provide a simple coequational logic for a wide range of relevant non-polynomial functors.

A little section at the end shows a result about semantical consequences of arbitrary subsets of cofree coalgebras of accessible functors, whereas coequations are special subsets of cofree coalgebras.

## 2 Coequational logic

### 2.1 Preliminaries

Since we are working with coalgebras (sometimes also referred to as “systems”) as structured sets all functors considered will be endofunctors of **Set** in the category **Set** of sets and functions. We already mentioned some special cases in the previous section:

- A *polynomial functor* for a given signature  $\Sigma$  is a functor of the shape

$$\sum_{\sigma \in \Sigma} (-)^{n_\sigma},$$

where  $n_\sigma$  is the arity of the operation  $\sigma \in \Sigma$ . The signature is called *k-ary* if  $n_\sigma < k$  for all  $\sigma \in \Sigma$ , where  $k$  is some infinite regular cardinal.

- A *finitary functor*  $H$  is a functor preserving filtered colimits, or equivalently, it is a quotient of some polynomial functor  $H_\Sigma$  for a finitary signature  $\Sigma$ . This means there exists a natural transformation  $\epsilon : H_\Sigma \rightarrow H$  having surjective components, where the arities  $n_\sigma$  of all operations  $\sigma \in \Sigma$  are finite. For equivalent characterisations of finitary functors see [5].
- An *accessible functor* is a functor preserving  $k$ -filtered colimits for some infinite regular cardinal  $k$ , or equivalently, it is quotient of some polynomial functor  $H_\Sigma$  for a  $k$ -ary signature  $\Sigma$ . Each finitary functor is accessible ( $k = \omega$ ). For equivalent characterisations of accessible functors see [4].

For a polynomial functor  $H_\Sigma$  we shall denote the terminal coalgebra and the cofree coalgebra on some fixed countable infinite set  $C$  of colours by  $T_\Sigma$  and  $Q_\Sigma$  respectively. Their structure maps are denoted by  $\alpha_{T_\Sigma} : T_\Sigma \rightarrow H_\Sigma T_\Sigma$  and  $\alpha_{Q_\Sigma} : Q_\Sigma \rightarrow H_\Sigma Q_\Sigma$ , the couniversal colouring arrow of  $Q_\Sigma$  by  $\gamma_\Sigma : Q_\Sigma \rightarrow C$ . Analogously, for a finitary functor  $H$  we shall write  $T$  and  $Q$ ,  $\alpha_T$ ,  $\alpha_Q$  and  $\gamma$ .

All trees considered in the paper are rooted, ordered trees that arise from a signature  $\Sigma$ , i.e. every node is labelled with an operation symbol  $\sigma \in \Sigma$  and has exactly  $n_\sigma$  children, where  $n_\sigma$  is the arity of  $\sigma$ . These trees are called  $\Sigma$ -trees. We always consider trees up to isomorphism, e.g. if we take the set of all subtrees of a tree it is meant to contain no duplicates rooted at different positions.

For a polynomial functor  $H_\Sigma$  it is well-known that  $T_\Sigma$  consists of all  $\Sigma$ -trees and that  $\alpha_{T_\Sigma}$  sends a  $\Sigma$ -tree to its root label (the operation symbol) and the list of its subtrees, see e.g. [4]. Similar, the cofree  $H_\Sigma$ -coalgebra  $Q_\Sigma$  consists of all coloured  $\Sigma$ -trees; that is, every node carries two labels: an operation symbol from  $\Sigma$  and a colour from  $C$ . The map  $\alpha_{Q_\Sigma}$  sends a coloured  $\Sigma$ -tree to its root operation symbol and the list of its coloured subtrees;  $\gamma_\Sigma$  maps every coloured  $\Sigma$ -tree to its root colour.

For a finitary signature  $\Sigma$  (i.e. the  $\Sigma$ -trees from  $T_\Sigma$  are finitely branching) it holds that if  $\Sigma$  is linearly ordered, so is  $T_\Sigma$  in the following sense: we compare two trees from  $T_\Sigma$  by breadth-first search. Starting at the roots (depth  $n = 0$ ), as long as we find only equal operation symbols at the nodes of depth  $n$ , it is always assured that we can compare the nodes of depth  $n + 1$  from left to right. As soon as we find two nodes with different operation symbols, the order of the trees is given by the order of these operations. It is easy to verify that this way we obtain a linear order of all trees from  $T_\Sigma$ : the transitivity is easy to see and the completeness follows from the facts that two different trees (even those of infinite depth) have to differ within finite depth and that these trees are finitely branching, thus every finite depth can be reached by breadth-first search. Therefore, if  $\Sigma \times C$  is linearly ordered, so is  $Q_\Sigma$  in the same sense.

## 2.2 Coequations

**Definition 2.1** Given a cofree coalgebra  $Q$ , a *coequation* is a coatomic subset  $Q \setminus \{q\}$  for some  $q \in Q$ . Since it is completely described by the element  $q$ , we shall denote it by  $\boxdot q$  (read: avoid  $q$ ).

These coatomic sets are called coequations because they are the coalgebraic dual to equations in universal algebra: while the latter determine quotients of free algebras, the former determine quotients of cofree coalgebras.

**Definition 2.2** Let  $(Q, \alpha_Q, \gamma)$  be the cofree  $H$ -coalgebra on  $C$ . An  $H$ -coalgebra  $(S, \alpha_S)$  is said to *satisfy* a coequation  $\boxdot q$  if for all colourings  $f : S \rightarrow C$  and for all states  $s \in S$  we have  $f^*(s) \neq q$  for the unique homomorphism  $f^* : S \rightarrow Q$  with  $f = \gamma \circ f^*$ . More generally,  $(S, \alpha_S)$  is said to satisfy a subset  $U \subseteq Q$  of the cofree coalgebra if for all colourings  $f : S \rightarrow C$  we have  $f^*[S] \subseteq U$ .

**Remark 2.3** The coequations introduced by Jan Rutten are defined more generally as subsets  $U \subseteq Q$ . As we see from the above Definition 2.2, for every coalgebra to satisfy a subset  $U \subseteq Q$  is equivalent to satisfying all coequations  $\boxtimes q$  where  $q \in Q \setminus U$ , i. e.  $U$  can be expressed via coequations by  $U = \bigcap_{q \in Q \setminus U} \boxtimes q$ . This means that when investigating subsets of cofree coalgebras, we can restrict ourselves to coequations, a fact that was first observed by Peter Gumm [6].

Since the elements of the cofree  $H$ -coalgebra can be viewed as “behaviour patterns” of  $H$ -coalgebras and the pattern  $q$  is forbidden by the corresponding coequation  $\boxtimes q$ , coequations can be regarded as certain “system properties”. Coequations stand for exactly these system properties preserved by coproducts, homomorphic images and subsystems. This follows from the Co-Birkhoff Theorem proven by Jan Rutten [10].

**Definition 2.4** A coequation  $\boxtimes q$  is said to be a *semantical consequence* of a coequation  $\boxtimes q'$  if every  $H$ -coalgebra that satisfies  $\boxtimes q'$  also satisfies  $\boxtimes q$ . We shall denote this by  $\boxtimes q' \models \boxtimes q$ . More generally, a subset  $U \subseteq Q$  is said to be a semantical consequence of a subset  $U' \subseteq Q$  if every  $H$ -coalgebra that satisfies  $U'$  also satisfies  $U$ , notation  $U' \models U$ .

### 2.3 A logic of coequations

When presenting a logic of coequations, we first define what proofs in that logic will be and then state some soundness and completeness results.

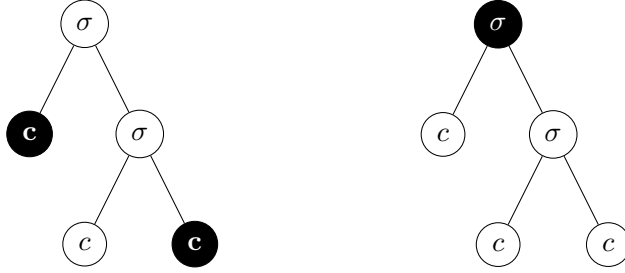
**Definition 2.5** Let  $P$  be a set of coequations (the premise). A *proof in a coequational logic* of a coequation  $\boxtimes q$  is a list of coequations  $\boxtimes c_i$ ,  $i \leq l$  of length  $l + 1$ , where  $l$  is some ordinal, where  $\boxtimes c_l = \boxtimes q$  and where for every  $i \leq l$  it holds that  $\boxtimes c_i \in P$  or  $F \models \boxtimes c_i$ , where  $F \subseteq \{\boxtimes c_j \mid j < i\}$ .

Let us consider systems of polynomial functors  $H_\Sigma$ . We call two nodes  $v, w$  of a tree  $t \in Q_\Sigma$  *equivalent*, if the coloured subtrees  $t_v$  and  $t_w$  of  $t$  rooted at  $v$  and  $w$ , respectively, are isomorphic.

**Definition 2.6** Given a cofree coalgebra  $Q_\Sigma$  of a polynomial functor, a *recolouring* of a tree  $t \in Q_\Sigma$  is a tree  $r \in Q_\Sigma$  for which there exists a homomorphism  $h : Q_\Sigma \rightarrow Q_\Sigma$  such that  $h(t) = r$ .

**Remark 2.7** As shown in [1], this is equivalent to the following:  $t$  and  $r$  have the same shape and for any two equivalent nodes in  $t$  the corresponding nodes in  $r$  are equivalent again.

**Example 2.8** Let  $H_\Sigma = (-)^2 + 1$ , i. e.  $\Sigma$  contains a binary operation  $\sigma$  and a constant  $c$ . Then  $Q_\Sigma$  consists of all coloured binary trees. Consider the following two trees from  $Q_\Sigma$ :



The right-hand tree is a recolouring of the left-hand one: both trees have the same shape and the black leaves of the left-hand one (which are the only equivalent nodes in this tree) are equivalent in the right-hand one. Note that the converse is not true since in the right-hand tree all leaves are equivalent nodes but this is not so in the left-hand one.

Before we restate the main result for coequational logic of polynomial functors, we introduce a relation on  $Q_\Sigma$  as follows: for two trees  $t, r \in Q_\Sigma$  we write  $r \sqsubseteq t$ , if  $r$  is a recolouring of a subtree of  $t$ .

**Theorem 2.9** ([1]) *For a polynomial functor  $H_\Sigma$  a coequation  $\boxtimes t$  is semantical consequence of the coequations  $\boxtimes r_i$ ,  $i \in I$ , if and only if there is some  $r_i$  that is a recolouring of some subtree of  $t$ , shortly*

$$\bigcap_{i \in I} \boxtimes r_i \models \boxtimes t \iff \exists i \in I : r_i \sqsubseteq t.$$

**Corollary 2.10** ([1]) *For systems of a polynomial functor deduction of coequations with the following rules is sound and complete:*

(i) *child rule*

$$\frac{\boxtimes t_v}{\boxtimes t} \quad \text{where } v \text{ is a child of the root of } t$$

(ii) *recolouring rule*

$$\frac{\boxtimes r}{\boxtimes t} \quad \text{where } r \text{ is a recolouring of } t$$

Now we can go one step further and consider systems as coalgebras of an accessible functor  $H$ . These are quotients of polynomial functors  $H_\Sigma$ , i.e. there is a natural transformation  $\epsilon : H_\Sigma \rightarrow H$  with surjective components. From this one can derive that the cofree  $H$ -coalgebra  $Q$  on  $C$  is quotient of the cofree  $H_\Sigma$ -coalgebra  $Q_\Sigma$  on  $C$ , see [1]. Thus  $Q$  consists of equivalence classes of coloured  $\Sigma$ -trees, which we call  $\epsilon$ -classes. Given a cofree  $H$ -coalgebra  $Q$ , we denote by  $[t] \in Q$  the  $\epsilon$ -class of a  $\Sigma$ -tree  $t \in Q_\Sigma$  from the corresponding cofree  $H_\Sigma$ -coalgebra  $Q_\Sigma$ .

**Theorem 2.11** ([1]) *For an accessible functor  $H$  a coequation  $\boxtimes q$  is a semantical consequence of the coequations  $\boxtimes q_i$ ,  $i \in I$ , if and only if for every tree  $t \in Q_\Sigma$  with  $[t] = q$  there is some tree  $r \in Q_\Sigma$  with  $[r] = q_i$  for some  $i \in I$ , so that  $r$  is recolouring of some subtree of  $t$ , shortly*

$$\bigcap_{i \in I} \boxtimes q_i \models \boxtimes q \iff \forall t. [t] = q \implies \exists i \in I \exists r. [r] = q_i : r \sqsubseteq t.$$

This logic clearly is more complicated than the one from Theorem 2.9 because one has to consider all trees from an  $\epsilon$ -class instead of just one tree. As a consequence of this, one can no longer give simple deduction rules for this logic. Nevertheless, for the (non-polynomial) functor  $\mathcal{P}_f(A \times -)$  of finitely branching labelled transition systems in [11] we found a logic similar to and nearly as simple as the one for polynomial functors. We do not state it at this time because it is a special case of the results of section 4 of this paper.

### 3 Cofree coalgebras of finitary functors

Throughout this section  $H$  denotes a finitary functor with a fixed presentation as a quotient  $\epsilon : H_\Sigma \rightarrow H$ , see section 2.1.

**Example 3.1** As an example of such a functor consider the LTS-functor  $\mathcal{P}_f(A \times -)$ . It has a presentation as a quotient of the polynomial functor

$$\sum_{n \in \mathbb{N}} (A \times -)^n = 1 + (A \times -) + (A \times -)^2 + \dots$$

whose signature contains  $|A^n|$  operations with arity  $n$  for each  $n \in \mathbb{N}$ . The functions  $\epsilon_X : \sum_{n \in \mathbb{N}} (A \times X)^n \rightarrow \mathcal{P}_f(A \times X)$  that map, for a given set  $X$ , every list of pairs  $(a, x) \in A \times X$  to the set of these pairs are easily proven to form a surjective natural transformation  $\epsilon$ .

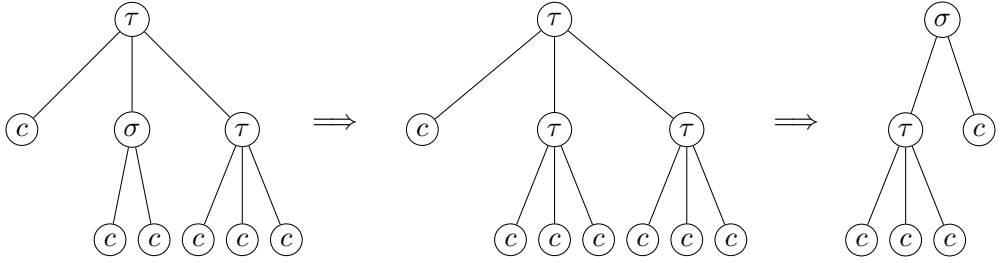
**Definition 3.2** Let  $X$  be a set (of variables). For two operations  $\sigma$  ( $n$ -ary) and  $\tau$  ( $k$ -ary) from  $\Sigma$  an expression of the form

$$\sigma(x_1, \dots, x_n) = \tau(y_1, \dots, y_k) \quad \text{where } x_i \text{ and } y_j \text{ in } X \quad (3.1)$$

is called an  $\epsilon$ -equation in case  $\epsilon_X : H_\Sigma X \rightarrow HX$  merges the two sides, i. e. they are equal under  $\epsilon_X$ . In that expression both sides are elements of  $H_\Sigma X$ , so-called *flat terms* in variables from  $X$ .

As already mentioned,  $T_\Sigma$  is the set of all  $\Sigma$ -trees. One can *apply the  $\epsilon$ -equation (3.1)* to such a tree at a node  $v$  from left to right, whenever  $v$  is labelled by  $\sigma$  and for two variables that are equal ( $x_i = x_{i'}$ ) the corresponding child nodes of  $v$  are equivalent: the tree is rearranged by relabelling  $v$  by  $\tau$  and choosing arbitrary  $\Sigma$ -trees for variables  $y_j$  that are not found on the left-hand side. If a  $\Sigma$ -tree  $t'$  can be obtained from a  $\Sigma$ -tree  $t$  by an application of finitely many  $\epsilon$ -equations, we shall denote this by  $t \sim t'$ .

**Example 3.3** Let  $\Sigma = \{c, \sigma, \tau\}$  contain a constant  $c$ , a binary operation  $\sigma$  and a ternary operation  $\tau$ . We consider the  $\epsilon$ -equation  $\sigma(x, y) = \tau(y, x, x)$  and give an example of its application to  $\Sigma$ -trees.



Each arrow  $\Rightarrow$  indicates one application of the  $\epsilon$ -equation: first it is applied from left to right to the center child of the left-hand tree's root, second it is applied from right to left to the root of the center tree. Looking at that second application note that not only the root label has changed from  $\tau$  to  $\sigma$ , but also how the child trees of the root are rearranged according to the variables of the  $\epsilon$ -equation. Also note that the first application is necessary to do the second one—an application to the root is not allowed in the left-hand tree.

**Definition 3.4** Given a  $\Sigma$ -tree  $t$ , we denote by  $\partial_k t$  (read: the *cutting* of  $t$  at level  $k$ ) the tree we obtain by cutting  $t$  at level  $k \in \mathbb{N}$  and then labelling every leaf of depth  $k$  with the constant symbol  $\perp$ .

**Definition 3.5** ([3]) Given two  $\Sigma$ -trees  $t, t'$ , we say that  $t'$  can be obtained from  $t$  by (possibly infinitely many) *applications of  $\epsilon$ -equations* provided that

$$\partial_k t \sim \partial_k t' \text{ for every } k \in \mathbb{N}.$$

**Theorem 3.6** ([3]) *The terminal  $H$ -coalgebra of a finitary functor is the quotient of the terminal  $H_\Sigma$ -coalgebra  $T_\Sigma$  modulo the congruence of applications of  $\epsilon$ -equations.*

**Example 3.7** Coming back to the above presentation of the LTS-functor  $\mathcal{P}_f(A \times -)$  from Example 3.1, the terminal coalgebra of the polynomial functor  $\sum_{n \in \mathbb{N}} (A \times -)^n$  is the set of all finitely branching trees whose edges are labelled with elements from  $A$ . The  $\epsilon$ -equations that determine the terminal  $\mathcal{P}_f(A \times -)$ -coalgebra as a quotient of that set can be derived from the surjective natural transformation  $\epsilon$  described above (see [3]): these are all  $\epsilon$ -equations

$$\sigma(x_1, \dots, x_n) = \tau(y_1, \dots, y_m)$$

for which we have  $\{(a_1, x_1), \dots, (a_n, x_n)\} = \{(b_1, y_1), \dots, (b_m, y_m)\}$ , where  $a_1, \dots, a_n$  and  $b_1, \dots, b_m$  are the lists of edge labels from  $A$  associated with the operations  $\sigma$  and  $\tau$  respectively. As  $\epsilon$  does by mapping two lists to the same set, they equate two list of pairs from  $A \times -$  whenever the sets of these pairs are equal. Finally, the terminal  $\mathcal{P}_f(A \times -)$ -coalgebra consists of equivalence classes of finitely branching edge-labelled trees that can be obtained from each other by applications of these  $\epsilon$ -equations. Since the  $\epsilon$ -equations allow arbitrary reordering, duplication of child trees and elimination of such duplicates (always together with the associated edge labels), we get a picture of how these equivalence classes look like.

Analogously to Theorem 3.6, the cofree  $H$ -coalgebra  $Q$  can be described in

terms of application of equations. We denote the corresponding quotient map by  $\gamma_\Sigma^* : Q_\Sigma \rightarrow Q$ . It is the unique  $H$ -homomorphism satisfying  $\gamma_\Sigma = \gamma \circ \gamma_\Sigma^*$ , where  $Q_\Sigma$  is viewed as an  $H$ -coalgebra using  $\epsilon_{Q_\Sigma} : H_\Sigma Q_\Sigma \rightarrow H Q_\Sigma$ , see [1].

We can view  $Q_\Sigma$  and  $Q$  as terminal coalgebras of  $H_\Sigma \times C$  and  $H \times C$  respectively and from  $\epsilon$  we get a surjective natural transformation  $\epsilon \times id : H_\Sigma \times C \rightarrow H \times C$ . This way we see that  $Q$  is a congruence of  $Q_\Sigma$  modulo the application of  $(\epsilon \times id)$ -equations, which we will call  $\epsilon$ -equations, too. The terms forming these  $\epsilon$ -equations consist of a colour from  $C$  and an operation from  $\Sigma$  on variables from  $Q_\Sigma$ . Since the congruence fulfills  $\gamma_\Sigma = \gamma \circ \gamma_\Sigma^*$ , the colours must be the same on both sides of any  $\epsilon$ -equation. The  $\epsilon$ -equations describing  $Q$  as a congruence of  $Q_\Sigma$  are the same as the ones describing  $T$  as a the congruence of  $T_\Sigma$ , except that the variables now stand for coloured trees from  $Q_\Sigma$  and the  $\epsilon$ -equations can be applied whatever the colour of the node is, but they have to preserve it.

**Corollary 3.8** *Two coloured trees from  $Q_\Sigma$  belong to the same  $\epsilon$ -class from  $Q$  if and only if they can be obtained from each other by an application of  $\epsilon$ -equations.*

**Definition 3.9** An  $\epsilon$ -equation is called *regular* provided the sets of variables used on both sides are equal. A presentation of a functor via a polynomial functor together with  $\epsilon$ -equations is called *regular* if all  $\epsilon$ -equations are regular.

**Remark 3.10** As proven by Adámek, Lücke and Milius ([2]), for finitary functors to have a regular presentation is equivalent to preserving preimages.

Again coming back to LTS we see that in the above presentation all  $\epsilon$ -equations are regular: this follows from the equality of the sets of pairs as seen in Example 3.7. Thus the functor  $\mathcal{P}_f(A \times -)$  has a regular presentation. And indeed it does preserve preimages.

## 4 A simpler logic

In the present section we establish a simplification of the coequational logic of Theorem 2.11. This section is a generalisation of our approach for finitely branching labelled transition systems (LTS) in [11]. More concretely, we will show that all finitary functors that are represented by regular  $\epsilon$ -equations allow a simple logic similar to the one we found for the LTS-functor  $\mathcal{P}_f(A \times -)$ . In this section we assume  $H$  to be a finitary functor that preserves preimages and to be a quotient of the polynomial functor  $H_\Sigma$  via regular  $\epsilon$ -equations.

### 4.1 $\epsilon$ -classes and subtrees

Let us consider trees  $t, t' \in Q_\Sigma$  where  $t'$  arises from  $t$  by application of one regular  $\epsilon$ -equation at the node  $v$  (the corresponding node in  $t'$  is called  $v'$ ). It is easy to prove that the set of  $\epsilon$ -classes of subtrees of  $t'$  is the same as the set of  $\epsilon$ -classes of subtrees of  $t$ : one can do that by case distinction between subtrees that are rooted

- (i) on the path from the root of  $t$  to  $v$  / on the path from the root of  $t'$  to  $v'$



- (ii) somewhere below  $v$  / somewhere below  $v'$ , where “below” means that  $v$  or  $v'$  are inner nodes of the path from the root to the node
- (iii) somewhere else.

The equality of the sets of  $\epsilon$ -classes of the subtrees is proven by Corollary 3.8 for (i), by regularity of the  $\epsilon$ -equation for (ii) and for (iii) there is nothing to prove. By transitivity of equality of sets, the statement proven this way is still valid if  $t'$  arises from  $t$  by application of finitely many regular  $\epsilon$ -equations.

We do not know how to use this result in order to prove that the statement is still valid in the case where infinitely many applications are made (which nevertheless is true, as we will see): an induction proof on the depth of the trees seems to be impossible because an infinite sequence of  $\epsilon$ -equations might contain infinitely many applications of  $\epsilon$ -equations at greater depths which are necessary for a following application in lesser depth to be executed. The problem can be worked around by restricting  $t$  to have special properties (to be a “first representative tree”, which is explained in section 4.2), and under this assumption an induction proof can be given. Since this restriction is unnecessary, we give a different proof for the infinite case.

**Proposition 4.1** *Given a regular presentation, for trees  $t, t' \in Q_\Sigma$  of the same  $\epsilon$ -class the  $\epsilon$ -classes of their subtrees are also the same, shortly (where  $v$  and  $w$  are nodes of  $t$  and  $t'$  respectively)*

$$[t] = [t'] \implies \forall v \exists w : [t_v] = [t'_w].$$

**Proof.** Since  $\gamma_\Sigma^* : Q_\Sigma \rightarrow Q$  is an  $H$ -homomorphism, we have the following commutative diagram on the left-hand side which can be completed to the one on the right-hand side since  $\gamma_\Sigma = \gamma \circ \gamma_\Sigma^*$ :

$$\begin{array}{ccc}
 Q_\Sigma & \xrightarrow{\gamma_\Sigma^*} & Q \\
 \alpha_{Q_\Sigma} \downarrow & & \downarrow \alpha_Q \\
 H_\Sigma Q_\Sigma & & \\
 \epsilon_{Q_\Sigma} \downarrow & & \\
 HQ_\Sigma & \xrightarrow{H\gamma_\Sigma^*} & HQ
 \end{array}
 \qquad
 \begin{array}{ccc}
 Q_\Sigma & \xrightarrow{\gamma_\Sigma^*} & Q \\
 \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle \downarrow & & \downarrow \langle \alpha_Q, \gamma \rangle \\
 H_\Sigma Q_\Sigma \times C & & \\
 \epsilon_{Q_\Sigma} \times id_C \downarrow & & \\
 HQ_\Sigma \times C & \xrightarrow{H\gamma_\Sigma^* \times id_C} & HQ \times C
 \end{array}$$

We know by Lambek’s Lemma that  $\langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle$  and  $\langle \alpha_Q, \gamma \rangle$  are isomorphisms since these are the structure maps of the terminal coalgebras of the functors  $H_\Sigma \times C$  and  $H \times C$  respectively. Thus  $\gamma_\Sigma^*$  merges two trees from  $Q_\Sigma$  if and only if the composition of the two surjective maps  $\epsilon_{Q_\Sigma} \times id_C$  and  $H\gamma_\Sigma^* \times id_C$  merges their presentations from  $H_\Sigma Q_\Sigma \times C$ . Recall that a tree  $r \in Q_\Sigma$  is presented in  $H_\Sigma Q_\Sigma \times C$  by  $\langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle$  in terms of its root operation symbol  $\sigma$  and the list of its child trees, which form an element  $\sigma(r_{v_1}, \dots, r_{v_n})$ , and its root colour.

Let us consider the first surjective map  $\epsilon_{Q_\Sigma} \times id_C$ . From Definition 3.2 we know that  $\epsilon_{Q_\Sigma} \times id_C$  merges two such tree presentations if and only if there is

a corresponding  $\epsilon$ -equation, i. e. one tree can be obtained from the other one via an application of this  $\epsilon$ -equation at the root. Since we assumed the equational presentation of  $H$  to be regular,  $\epsilon_{Q_\Sigma}$  only merges presentations for which the sets of all child trees are the same. This particularly means that for a pair of trees from  $\ker((\epsilon_{Q_\Sigma} \times id_C) \circ \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle)$  the  $\epsilon$ -classes of the child trees are the same.

Now let us consider the second surjective map  $H\gamma_\Sigma^* \times id_C$ . Since  $\langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle$  is an isomorphism and  $\epsilon_{Q_\Sigma} \times id_C$  is surjective, its domain  $HQ_\Sigma \times C$  is isomorphic to the quotient set of  $Q_\Sigma$  modulo the equivalence  $\ker((\epsilon_{Q_\Sigma} \times id_C) \circ \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle)$ . And since  $H\gamma_\Sigma^* \times id_C$  again is surjective,  $HQ$  is isomorphic to the quotient set of  $HQ_\Sigma \times C$  modulo the equivalence  $\ker(H\gamma_\Sigma^* \times id_C)$ . Taken together  $HQ$  (or equivalently,  $Q$ ) is isomorphic to the quotient set of  $Q_\Sigma$  modulo some equivalence  $R$ .

If we can prove that  $R$  only contains pairs of trees whose child trees have the same  $\epsilon$ -classes, we easily get our proposition: since  $[t] = [t']$ , we have  $\gamma_\Sigma^*(t) = \gamma_\Sigma^*(t')$ . We conclude from the above commutative diagram that the composition of  $(\epsilon_{Q_\Sigma} \times id_C) \circ \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle$  and  $\langle \alpha_Q, \gamma \rangle^{-1} \circ (H\gamma_\Sigma^* \times id_C)$  merges  $t$  and  $t'$ . This is equivalent for  $t$  and  $t'$  to form a pair in  $R$ , which implies that the child trees of  $t$  and  $t'$  have the same  $\epsilon$ -classes. By structural induction on the trees this is also true for arbitrary subtrees of  $t$  and  $t'$ .

In order to prove the desired property of  $R$ , we claim that

$$R = (\ker((\epsilon_{Q_\Sigma} \times id_C) \circ \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle) \cup M)^*,$$

where

$$M := \{(r, s) \mid \alpha_{Q_\Sigma}(r) = \sigma(r_{v_1}, \dots, r_{v_n}), \alpha_{Q_\Sigma}(s) = \sigma(s_{v_1}, \dots, s_{v_n}), \\ [r_{v_i}] = [s_{v_i}] \text{ for } i = 1 \dots n \text{ and } \gamma_\Sigma(r) = \gamma_\Sigma(s)\}.$$

The star operation is the reflexive transitive closure and the set  $M$  contains all pairs  $(r, s)$  of trees where the root labels (operation symbol and colour) of  $r$  and  $s$  are equal and all their child trees at the same positions have the same  $\epsilon$ -classes. We prove our claim by showing the quotient set of  $Q_\Sigma$  modulo  $P := (\ker((\epsilon_{Q_\Sigma} \times id_C) \circ \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle) \cup M)^*$  to be isomorphic to the codomain  $Q$  of  $\gamma_\Sigma^*$ .

We first prove that all pairs of trees merged by  $\gamma_\Sigma^*$  are contained in  $P$ : to be merged by  $\gamma_\Sigma^*$  means that the trees can be obtained from each other by an application of  $\epsilon$ -equations. But each sequence of applications of  $\epsilon$ -equations is covered by the construction of  $P$ : after applications of  $\epsilon$ -equations at the root the pair is contained in  $\ker((\epsilon_{Q_\Sigma} \times id_C) \circ \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle) \subseteq P$ . After (possibly infinitely many) applications of  $\epsilon$ -equations at depths  $\geq 1$  in a tree the corresponding pair of trees lies in  $P$  because it is contained in  $M$ . And an arbitrary combination of both is possible due to the reflexive and transitive closure, thus the pair of the original tree and the resulting tree will always lie in  $P$ .

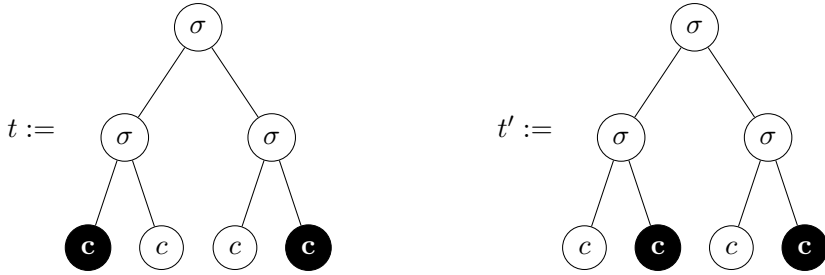
Conversely, assume a pair of trees to lie in  $P$ : this means that the pair arose by transitivity from some pairs from  $\ker((\epsilon_{Q_\Sigma} \times id_C) \circ \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle)$  and from  $M$ . But for these pairs we know that  $\gamma_\Sigma^*$  merges the trees, and by transitivity of “having the same  $\epsilon$ -class” we conclude that  $\gamma_\Sigma^*$  always merges the trees from a pair in  $P$ .

Finally it is an easy observation that  $R$  only contains pairs of trees whose child trees have the same  $\epsilon$ -classes: we already stated that for  $\ker((\epsilon_{Q_\Sigma} \times id_C) \circ \langle \alpha_{Q_\Sigma}, \gamma_\Sigma \rangle)$  and by definition it is also true for  $M$  and consequently for the union of both. Finally the reflexive and transitive closure only adds pairs with the desired property because “having the same set of child tree  $\epsilon$ -classes” is transitive.  $\square$

#### 4.2 $\epsilon$ -classes and recolourings

Now we consider recolourings of trees. Unfortunately it is not true that for two trees of the same  $\epsilon$ -class the sets of all their recolourings are equal. To see this, we give the following

**Example 4.2** Let  $H$  be the finitary functor that is given as a quotient of  $H_\Sigma = (-)^2 + 1$  modulo the  $\epsilon$ -equation  $\sigma(x, y) = \sigma(y, x)$  ( $\Sigma = \{\sigma, c\}$ , where  $\sigma$  is binary and  $c$  nullary). Since this equation is regular,  $H$  preserves preimages. Let  $C$  contain the colours white and black. We consider the following coloured trees  $t$  and  $t'$  from  $Q_\Sigma$ :



They obviously have the same  $\epsilon$ -class  $[t] = [t']$ . We obtain a recolouring  $r$  of  $t$  by colouring the left-hand child of the root of  $t$  black. But now it is clear that there is no recolouring of  $t'$  that lies in the  $\epsilon$ -class of  $r$ : in every recolouring of  $t'$  both children of the root must have the same colour because they are equivalent nodes in  $t'$ . But in every tree of the  $\epsilon$ -class of  $r$  they have different colours, namely black and white, because  $\epsilon$ -equations preserve the colours of the nodes they are applied to and the colourings of all subtrees of that node.

Nevertheless we can state a similar proposition for recolourings as we had for subtrees (Proposition 4.1) if we restrict  $t$  to be a “first representative tree”, as we shall see.

**Definition 4.3** Given a finitary functor  $H$  as a quotient of the polynomial functor  $H_\Sigma$  and a linear order on  $\Sigma \times C$ , we have a linear order  $(Q_\Sigma, \leq)$  as mentioned in section 2.1. Then for every  $\epsilon$ -class  $q \in Q$  we denote by  $t^q \in Q_\Sigma$  the *first representative tree* of  $q$ . It is defined by  $[t^q] = q$  and  $[t] = q \Rightarrow t^q \leq t$  for all  $t \in Q_\Sigma$ .

**Remark 4.4** Given a linear order on  $\Sigma \times C$ , the subset  $R \subseteq Q_\Sigma$  of first representative trees contains exactly one tree from each  $\epsilon$ -class, which is clear from the definition. Additionally,  $R$  is closed under subtrees: we easily see that indeed every subtree of such a first representative tree is a first representative tree of an  $\epsilon$ -class again. If there were some  $\epsilon$ -equations that could be applied to a subtree such that

the result was a previous tree in the order, these applications of  $\epsilon$ -equations had shifted the whole tree to a previous position in the order.

For the rest of the paper when working with first representative trees we assume  $\Sigma \times C$  to be linearly ordered.

**Example 4.5** In the case of LTS we have the presentation from Example 3.1 via the polynomial functor  $\sum_{n \in \mathbb{N}} (A \times -)^n$ . We can find a linear order where  $(\sigma_n, k) < (\sigma_m, k)$  for every colour  $k$  and every two operations with arities  $n < m$ . The first representative trees we obtain are the strongly extensional trees as studied by James Worrell [12].

Now let us come back to the above Example 4.2. The reason that there is no recolouring of  $t'$  in the  $\epsilon$ -class of  $r$  is that the recolouring makes two subtrees of  $t$  of the same  $\epsilon$ -class become subtrees of  $r$  of different  $\epsilon$ -classes. We can avoid this by choosing a linear order on  $\Sigma \times C$  and restricting  $t$  to be the first representative tree  $t^q$  of its  $\epsilon$ -class  $q$ . In this case it follows from being the first representative tree (cf. Remark 4.4) that for any two subtrees of  $t^q$  of the same  $\epsilon$ -class their root nodes are equivalent. Since recolouring a tree preserves equivalent nodes, recolourings of first representative trees have the special property that any two subtrees of the same  $\epsilon$ -class in the original tree become subtrees of the same  $\epsilon$ -class in the recolouring again. But note that recolourings of first representative trees in general are no longer first representative trees.

**Proposition 4.6** *Given some  $q \in Q$ , for every tree  $t' \in Q_\Sigma$  such that  $[t'] = q$  and every recolouring  $r$  of  $t^q$  there is a recolouring  $r'$  of  $t'$  such that  $[r] = [r']$ .*

**Proof.** Given  $q$  and a recolouring  $r$  of  $t^q$ , we will prove that

- (1) a homomorphism  $f^* : Q_\Sigma \rightarrow Q_\Sigma$  can be defined such that  $r = f^*(t^q)$  and
- (2)  $[f^*(t^q)] = [f^*(t')]$  for all  $t'$  with  $[t'] = q$ .

(1) Definition of  $f^*$ . The given recolouring  $r$  of  $t^q$  yields a colour from  $C$  for every subtree of  $t^q$ , namely the colour  $r$  assigns to its root. Since  $t^q$  is the first representative tree, all of its subtrees are first representative trees again. Thus there is at most one subtree of  $t^q$  from a given  $\epsilon$ -class, namely the first representative tree of that  $\epsilon$ -class. Furthermore, for possibly existing isomorphic subtrees the assignment always yields the same colour, see Remark 2.7. Then the colours of the subtrees of  $t^q$  given by  $r$  can be assigned to the  $\epsilon$ -classes of these subtrees of  $t^q$  in a unique way. More formally, let  $S^{t^q} \subseteq Q$  be the set of all  $\epsilon$ -classes of subtrees of  $t^q$ . Then  $r$  uniquely determines a  $C$ -colouring  $f'' : S^{t^q} \rightarrow C$  of  $S^{t^q}$ .

Choosing arbitrary colours for the remaining  $\epsilon$ -classes, we can extend  $f''$  to a  $C$ -colouring  $f' : Q \rightarrow C$  of  $Q$ . Using the map  $\gamma_\Sigma^* : Q_\Sigma \rightarrow Q$  that assigns to every tree its  $\epsilon$ -class, we extend  $f'$  to a  $C$ -colouring  $f : Q_\Sigma \rightarrow C$  by defining  $f := f' \circ \gamma_\Sigma^*$ . We obtain the homomorphism  $f^* : Q_\Sigma \rightarrow Q_\Sigma$  as the unique colour-compatible homomorphism ( $\gamma_\Sigma \circ f^* = f$ ) into the cofree  $H_\Sigma$ -coalgebra on  $C$ . We obviously have  $f^*(t^q) = r$  because for every subtree  $s$  of  $t^q$  we have  $s \in \gamma_\Sigma^{*-1}(S^{t^q})$  and the root of  $f^*(s)$  is coloured with the colour  $f''$  assigns to the  $\epsilon$ -class of  $s$ .

(2) Proof of  $[f^*(t^q)] = [f^*(t')]$ . We consider a tree  $t'$  with  $[t'] = q = [t^q]$ . Recall from Theorem 3.6 that this is equivalent to  $\partial_k t^q \sim \partial_k t'$  for all  $k \in \mathbb{N}$ . We want to show by induction on the depth  $k$  of the cutting that  $\partial_k(f^*(t^q)) \sim \partial_k(f^*(t'))$ . Unfortunately we do not know how to prove that directly—what we actually do is to prove by induction on  $k$  that for all  $k \in \mathbb{N}$  there is a tree  $t_k$  for which the following five statements hold.

- (i)  $\partial_k(t_k) = \partial_k(t')$
- (ii)  $t_k$  is obtained from  $t^q$  by finitely many applications of  $\epsilon$ -equations at depths  $< k$
- (iii) all subtrees of  $t_k$  rooted at depths  $\leq k$  have the same  $\epsilon$ -classes as the corresponding subtrees of  $t'$
- (iv)  $f^*(t_k)$  is obtained from  $f^*(t^q)$  by application of the same  $\epsilon$ -equations as the ones used in (ii)
- (v)  $\partial_k(f^*(t_k)) = \partial_k(f^*(t'))$

Then for every  $k$  we obtain  $\partial_k(f^*(t^q)) \sim \partial_k(f^*(t'))$  from (iv) and (v) as follows: (iv) implies  $f^*(t^q) \sim f^*(t_k)$ . Since the  $\epsilon$ -equations were only applied at depths  $< k$  and cutting at depth  $k$  preserves equivalent subtrees of depths  $< k$ , the same  $\epsilon$ -equations witness  $\partial_k(f^*(t^q)) \sim \partial_k(f^*(t_k))$ . Together with (v) we obtain  $\partial_k(f^*(t^q)) \sim \partial_k(f^*(t'))$ . Thus  $[f^*(t')] = [f^*(t^q)]$ , and  $f^*(t')$  is the required recolouring  $r'$  of  $t'$ . Furthermore the induction proof is applicable to every tree  $t'$  such that  $[t'] = [t^q]$ , and we finally obtain the proposition.

Now we prove by induction on  $k$  that for all  $k \in \mathbb{N}$  there is a tree  $t_k$  for which the statements (i) to (v) hold.

Base clause. For  $k = 0$  we define  $t_0 := t^q$ . Then (ii) is directly clear from the definition (zero applications of  $\epsilon$ -equations). We have (i)  $\partial_0(t_0) = \partial_0(t')$ , because these are the root nodes of the trees  $t^q$  and  $t'$  respectively, which are labelled by the same constant  $\perp$  and moreover have the same colour: by assumption they can be obtained from each other by finitely many applications of (colour preserving)  $\epsilon$ -equations. For (iii), there is nothing to prove. (iv) again is clear from the definition of  $t_0$  (again zero applications of  $\epsilon$ -equations). We prove (v)  $\partial_0(f^*(t_0)) = \partial_0(f^*(t'))$ . Since  $\partial_0$  takes the root nodes of the trees and labels them with the constant  $\perp$ , we only have to show that the root colours of  $f^*(t_0)$  and  $f^*(t')$  are equal. But because of  $[t_0] = [t^q] = [t']$ , according to its definition  $f^*$  colours both roots with the same colour.

Induction step from  $k$  to  $k + 1$ . We assume that there is a tree  $t_k$  such that the statements (i)<sub>k</sub> to (v)<sub>k</sub> hold. We then prove that these statements also hold for  $k + 1$  instead of  $k$ , referred to as (i)<sub>k+1</sub> to (v)<sub>k+1</sub>.

We claim that we can obtain a tree  $t_{k+1}$  which fulfills (i)<sub>k+1</sub> and (iii)<sub>k+1</sub> by application of (finitely many)  $\epsilon$ -equations to  $t_k$  at depth  $k$ . Then from (ii)<sub>k</sub> we directly get (ii)<sub>k+1</sub>. In fact one can obtain such a tree  $t_{k+1}$  that way:

From (i)<sub>k</sub> and (iii)<sub>k</sub> we know that we can obtain  $t'$  by applications of (possibly infinitely many)  $\epsilon$ -equations to  $t_k$  at depths  $\geq k$ . Additionally we know that, in

order to do so, we can first apply the (finitely many)  $\epsilon$ -equations at depth  $k$  (without applying equations in greater depths first) because any two nodes of  $t_k$  at depth  $k+1$  that could be made equivalent (by application of  $\epsilon$ -equations in greater depths), already are equivalent—this is so because  $t^q$  as a first representative tree has only first representative trees as subtrees and the application of regular  $\epsilon$ -equations at depths  $< k$  does not change subtrees rooted at depths  $\geq k$ . After the application of all the  $\epsilon$ -equations at depth  $k$  we can be sure that the subtrees of  $t_{k+1}$  rooted at depth  $k+1$  have the same  $\epsilon$ -classes as the corresponding subtrees of  $t'$ :  $\epsilon$ -equations that were possibly applied at depths  $\geq k+1$  before an  $\epsilon$ -equation was applied at depth  $k$  in the original chain of  $\epsilon$ -equations did not change the  $\epsilon$ -classes of subtrees rooted at depth  $k+1$ , so looking only at the  $\epsilon$ -classes, the  $\epsilon$ -equation applied at depth  $k$  rearranges subtrees at depth  $k+1$  in exactly the same way no matter if applied first or later. This proves (iii) $_{k+1}$ . This also ensures that the nodes of  $t_{k+1}$  at depth  $k+1$  have the same colours as the corresponding nodes of  $t'$ , because trees of the same  $\epsilon$ -class always have the same root colour. And since the resulting operation symbols at depth  $k$  are independent of when the  $\epsilon$ -equations at depth  $k$  are applied, we have (i) $_{k+1}$ .

Next we prove (iv) $_{k+1}$ . It is sufficient to prove the commutativity of the following diagram, then together with (iv) $_k$  we obtain (iv) $_{k+1}$ .

$$\begin{array}{ccc} t_k & \xrightarrow{\epsilon\text{-equations}} & t_{k+1} \\ f^* \downarrow & & \downarrow f^* \\ f^*(t_k) & \xrightarrow{\text{same } \epsilon\text{-equ.}} & f^*(t_{k+1}) \end{array}$$

To do so, we first have to assert that the  $\epsilon$ -equations used to obtain  $t_{k+1}$  from  $t_k$  are still applicable to the recolouring  $f^*(t_k)$ . This is indeed the case because a recolouring does not change the tree shape and equivalent nodes of  $t_k$  become equivalent nodes in the recoloured tree again. Second, both paths in the diagram result in trees of the same shape: this is true because the  $\epsilon$ -equations reshape  $t_k$  and  $f^*(t_k)$  the same way and the recolouring  $f^*$  has no impact on the shape. Finally, both paths in the diagram lead to the same colouring (and thus the same tree): let us look at the nodes of depth  $k$ . The application of  $\epsilon$ -equations to  $t_k$  at depth  $k$  preserves the  $\epsilon$ -classes of the trees that are rooted at this depth and by its definition  $f^*$  consequently colours their roots the same way. Since the application of the same  $\epsilon$ -equations to  $f^*(t_k)$  at depth  $k$  preserves colours at this depth, at depth  $k$  we get the same colours. The same argument can be used for nodes at depths  $< k$ . The remaining nodes are these of depths  $> k$ , i.e. these of the subtrees rooted at depth  $k+1$ . To them it does not matter whether they are rearranged by applications of regular  $\epsilon$ -equations at depth  $k$  first and then recoloured by  $f^*$  or recoloured by  $f^*$  first and then rearranged in the same way.

Item (v) $_{k+1}$  remains to be proven. By (i) $_{k+1}$  we know that  $t_{k+1}$  and  $t'$  have the same shape down to depth  $k+1$  and by (iii) $_{k+1}$  we know that all the  $\epsilon$ -classes of subtrees rooted at depth  $\leq k+1$  are the same for corresponding nodes of  $t_{k+1}$  and  $t'$ . Thus down to depth  $k+1$  corresponding nodes are recoloured by  $f^*$  with the

same colour and we obtain  $\partial_{k+1}(f^*(t_{k+1})) = \partial_{k+1}(f^*(t'))$ .  $\square$

### 4.3 A simple coequational logic

**Theorem 4.7** *Let  $H$  be a finitary functor preserving preimages. For every linear order on  $\Sigma \times C$  the coequational logic of  $H$  has the following form: a coequation  $\boxtimes q$  is semantical consequence of the coequations  $\boxtimes q_i$ ,  $i \in I$ , if and only if there is some tree  $\bar{r}$  with  $[\bar{r}] = q_i$  for some  $i \in I$ , so that  $\bar{r}$  is a recolouring of some subtree of  $t^q$ , shortly*

$$\bigcap_{i \in I} \boxtimes q_i \models \boxtimes q \iff \exists i \in I \exists \bar{r}. [\bar{r}] = q_i : \bar{r} \sqsubseteq t^q. \quad (4.1)$$

**Proof.** Since every finitary functor is accessible, from Theorem 2.11 we know that the left-hand side of (4.1) is equivalent to

$$\forall t. [t] = q \iff \exists i \in I \exists r. [r] = q_i : r \sqsubseteq t. \quad (4.2)$$

We prove (4.2) to be equivalent to the right-hand side of (4.1).

$\Rightarrow$  Since  $[t^q] = q$  we can look at (4.2) in the special case of  $t = t^q$ . It yields the existence of a tree  $r$  such that  $[r] = q_i$  for some  $i \in I$  and  $r \sqsubseteq t^q$ . Renaming  $r$  to  $\bar{r}$ , this already is the right-hand side of (4.1).

$\Leftarrow$  From the right-hand side of (4.1) we have a tree  $\bar{r}$  such that  $[\bar{r}] = q_i$  for some  $i \in I$  and  $\bar{r} \sqsubseteq t^q$ . The latter means that  $\bar{r}$  is a recolouring of some subtree  $\bar{s}$  of  $t^q$ . Proposition 4.1 yields for every tree  $t \in Q_\Sigma$  with  $[t] = q = [t^q]$  the existence of a subtree  $s$  of  $t$  with  $[s] = [\bar{s}]$ . Since  $\bar{s}$  as a subtree of a first representative tree is again a first representative tree, we obtain from Proposition 4.6 a recolouring  $r$  of  $s$  with  $[r] = [\bar{r}]$ . Altogether this means  $r \sqsubseteq t$  and  $[r] = q_i$ , which is exactly (4.2).  $\square$

Since for a semantical consequence of coequations the logic of Theorem 4.7 only requires the existence of *one tree*  $\bar{r}$  with some properties, it is much simpler than the logic for accessible functors of Theorem 2.11, which *for each tree* from a whole  $\epsilon$ -class (such a class may contain infinitely many trees) requires the existence of a tree  $r$  with these properties. Because of this (and using the fact that a subtree of a first representative tree always is a first representative tree again) we are able to formulate two simple deduction rules similar to the case of polynomial functors.

**Corollary 4.8** *For systems of finitary functors that preserve preimages deduction of coequations with the following rules is sound and complete:*

(i) *child rule*

$$\frac{\boxtimes q'}{\boxtimes q} \quad \text{where } t^{q'} \text{ is a child tree of } t^q$$

(ii) *recolouring rule*

$$\frac{\boxtimes q'}{\boxtimes q} \quad \text{where } [\bar{r}] = q' \text{ for some recolouring } \bar{r} \text{ of } t^q$$

For an example of a logic given by Theorem 4.7, one may instantiate it with the LTS-functor  $\mathcal{P}_f(A \times -)$  and an order as given in Example 4.5.

## 5 Finitary functors not preserving preimages

The first direction of Theorem 4.7 (completeness) was easily proven: nothing else but the existence of first representative trees was needed, which is guaranteed for every finitary functor. But for the other way round (soundness) we needed some more effort, especially in Propositions 4.1 and 4.6, which are the core of the simpler logic: they guarantee that the simplification of section 4.3 works. In these propositions we needed the assumption that the functor preserves preimages. We will now prove the necessity of this assumption for Theorem 4.7.

**Theorem 5.1** *A finitary functor  $H$  with  $|T| > 1$  preserves preimages if and only if for every linear order on  $\Sigma \times C$  and any coequations  $\boxtimes q$  and  $\boxtimes q_i$ ,  $i \in I$ , we have:*

$$\bigcap_{i \in I} \boxtimes q_i \models \boxtimes q \iff \exists i \in I \exists \bar{r}. [\bar{r}] = q_i : \bar{r} \sqsubseteq t^q. \quad (5.1)$$

**Proof.** In Theorem 4.7 for finitary functors preserving preimages it was shown that for every linear order on  $\Sigma \times C$  and any coequations (5.1) is true. Now we assume a finitary functor  $H$  with  $|T| > 1$  not to preserve preimages and find an order on  $\Sigma \times C$  and coequations, for which (5.1) is not true. For this we do not need different colours, so in the following assume all trees to be coloured completely with the same colour (or equivalently, think of uncoloured trees).

Since  $H$  does not preserve preimages, in every presentation of  $H$  there is an  $\epsilon$ -equation which is not regular, i. e. of the form

$$\sigma(x_1, \dots, x_n) = \tau(y_1, \dots, y_m) \quad (5.2)$$

where we have a non-empty set

$$Y := \{y_i \mid i \leq m \wedge y_i \neq x_j \forall j \leq n\}$$

of variables that do not appear on the left hand side. Moreover, we choose (5.2) such that these new variables  $y_i \in Y$  appear as soon as possible when reading the  $\tau$ -term from left to right. More precisely, we compare the  $\tau$ -terms in their variables from left to right asking whether the variables are contained in  $\{x_1, \dots, x_n\}$ . As soon as a difference occurs, we choose the term that has the variable not contained in that set. Note that  $\sigma$  and  $\tau$  may be the same operation symbol.

Now we denote by  $t$  the tree whose nodes are all labelled by  $\sigma$ , thus it has the property that all its subtrees are isomorphic to  $t$  itself. If the arity  $n \geq 1$  then  $t$  is a tree where every path is infinite, if  $n = 0$  ( $\sigma$  is a constant) then  $t$  is a singleton tree. In the  $\epsilon$ -class  $[t]$  we can find all trees whose nodes are only labelled by the operations  $\sigma$  and  $\tau$ —this is because of (5.2): we can apply it to  $t$ , starting at the root and continuing until an arbitrary depth, exactly where we want a node to be labelled



by  $\tau$  and we do not apply any  $\epsilon$ -equation, where we want to keep the  $\sigma$ -labels. For every application of (5.2) we choose  $t$  for the variables from  $Y$  and ensure this way that in greater depths we will always meet nodes labelled by  $\sigma$ .

Since the terminal coalgebra was assumed to have at least two elements, there is a tree from an  $\epsilon$ -class different from  $[t]$  (and this is not due to different colours). Consequently this tree has a node labelled by an operation  $\rho$  different from  $\sigma$  and  $\tau$ . This operation  $\rho$  cannot be obtained at a node labelled with  $\sigma$  or  $\tau$  applying  $\epsilon$ -equations (in particular there are no  $\epsilon$ -equations that equate  $\rho$ - and  $\sigma$ - or  $\rho$ - and  $\tau$ -terms): if we could obtain  $\rho$  from  $\sigma$  or  $\tau$  via (possibly many)  $\epsilon$ -equations, we could state that directly in one  $\epsilon$ -equation  $\rho(\dots) = \sigma(\dots)$  or  $\rho(\dots) = \tau(\dots)$  because of the transitivity of  $\epsilon$ -equations. But from this we derive analogously to the above situation for the  $\epsilon$ -equation (5.2) that all trees with node labels from  $\{\rho, \tau, \sigma\}$  belong to the same  $\epsilon$ -class  $[t]$ . But then our assumption of an  $\epsilon$ -class different from  $[t]$  would yield the existence of another operation different from  $\rho$ ,  $\tau$  and  $\sigma$  and so on; that process only stops if we find an operation that cannot be obtained from one of the former operations by application of  $\epsilon$ -equations.

Now we consider a linear order of the operations in  $\Sigma$  such that the first ones are  $\rho < \tau \leq \sigma < \dots$  (which is turned into the beginning of an order of  $\Sigma \times C$  by adding the single colour we are using in this proof). We consider the tree  $\bar{r}$  whose nodes are all labelled by  $\rho$ .  $\bar{r}$  is the first representative tree of its  $\epsilon$ -class because  $\rho$  is the first operation in the order. For its  $\epsilon$ -class we have  $[\bar{r}] \neq [t]$  because there are no  $\epsilon$ -equations that could change  $\rho$  into  $\sigma$ . Now we evaluate (5.1) in the case  $q := [t]$  and  $q_0 := [\bar{r}]$  ( $I = \{0\}$ ).

We first show that the right-hand side of (5.1) is true: we find the tree  $t^q$  from the  $\epsilon$ -class  $q$  that is obtained from  $t$  by (in case  $n \geq 1$  infinitely many, but at every level finitely many) applications of (5.2) from left to right, starting at the root of  $t$  and proceeding level by level applying it at every node that is root of a tree isomorphic to  $t$ . For variables from  $Y$  we use the tree  $\bar{r}$ , and since  $Y$  is non-empty,  $\bar{r}$  is subtree of  $t^q$  of class  $q_0$ . We show that  $t^q$  is the first representative tree of  $q$ : as mentioned there are no  $\epsilon$ -equations whose application could replace  $\tau$  by  $\rho$  and as a consequence of the above assumption on (5.2) it is impossible to obtain a permutation of a node's children such that  $\rho$ -labelled nodes shift to the left. These both facts together with the chosen order of the operations ensures  $t^q$  to be the first representative tree.

Second, we show that the left-hand side of (5.1) is not true: we just consider the tree  $t$  from the  $\epsilon$ -class  $q$ , whose nodes are all labelled by  $\sigma$ . It only has subtrees isomorphic to  $t$  which have the  $\epsilon$ -class  $q \neq q_0$ ; and recolourings of these subtrees would have a different root colour than the single one we are using all the time and which is the root colour of all the trees from the  $\epsilon$ -class  $q_0$ . This means there is no recolouring of a subtree of  $t$  from the  $\epsilon$ -class  $q_0$ .  $\square$

### 5.1 Discussion

The intuition of the simpler logic is that the formation of  $\epsilon$ -classes by  $\gamma_\Sigma^*$  is compatible with subtrees and recolourings. We had to deal with both items in sections

4.1 and 4.2 respectively.

Let us first recall from Proposition 4.1 what compatibility of  $\epsilon$ -classes and subtrees means: whenever two  $\Sigma$ -trees are merged in one  $\epsilon$ -class, their direct subtrees are merged in  $\epsilon$ -classes again, and recursively for all their subtrees. The consequence of this is that with knowing one tree of an  $\epsilon$ -class one already knows the  $\epsilon$ -classes of the subtrees of all other trees of this class, what is captured by the quantifiers (exists / for all) in the above formulas. Finitary functors  $H$  that preserve preimages ensure that compatibility of  $\epsilon$ -classes and subtrees by their homomorphisms ( $\gamma_\Sigma^*$  is an  $H$ -homomorphism); all other finitary functors do not: as seen in the proof of Theorem 5.1 trees can be merged in one  $\epsilon$ -class without their subtrees having the same  $\epsilon$ -classes.

Now let us have a look at compatibility of  $\epsilon$ -classes and recolourings. These concepts just are not compatible: having  $\epsilon$ -classes means having different trees in these  $\epsilon$ -classes. But as long as subtrees of a tree are not isomorphic, what would be respected by recolourings, arbitrary recolourings can be applied to them taking them to different  $\epsilon$ -classes. And since we surely have some trees with and some trees without isomorphic subtrees, more or less recolourings can be applied and more or less  $\epsilon$ -classes can be obtained as seen in Example 4.2. To solve these problems we hid the complexity of  $\epsilon$ -classes by their first representative trees. In these first representative trees “as much subtrees as possible” are isomorphic and this way the recolourings of a first representative tree form a “minimal” set of  $\epsilon$ -classes. Under the additional assumption of a preimage preserving functor it turns out in Proposition 4.6 that for every tree of the  $\epsilon$ -class of a given first representative tree the recolourings form a superset of this set. And this is enough compatibility of  $\epsilon$ -classes and recolourings to state our simple logic, now working with first representative trees.

## 6 Deduction of Sets

For semantical consequences of arbitrary subsets of the cofree coalgebra of a polynomial functor there is the following statement.

**Theorem 6.1** ([1]) *Let  $U, U' \subseteq Q_\Sigma$  be arbitrary subsets of the cofree  $H_\Sigma$ -coalgebra of a polynomial functor  $H_\Sigma$ . The following holds:  $U$  is semantical consequence of  $U'$ , if and only if every  $t \in Q_\Sigma$ , for which all recolourings of all subtrees lie in  $U'$ , is contained in  $U$ , shortly*

$$U' \models U \iff \forall t \in Q_\Sigma : (\forall r.r \sqsubseteq t : r \in U') \Rightarrow t \in U.$$

Analogously to that statement we can prove one for the case of accessible functors although this was not expected by [1].

**Theorem 6.2** *Let  $U', U \subseteq Q$  be arbitrary subsets of the cofree  $H$ -coalgebra of an accessible functor  $H$ . The following holds:  $U$  is semantical consequence of  $U'$ , if and only if every  $\epsilon$ -class  $q \in Q$ , in which there exists a tree  $t \in Q_\Sigma$ , for which the*

$\epsilon$ -classes of all recolourings of subtrees of  $t$  lie in  $U'$ , lies in  $U$ , shortly

$$U' \models U \iff \forall q \in Q : (\exists t.[t] = q \ \forall r.r \sqsubseteq t : [r] \in U') \Rightarrow q \in U.$$

**Proof.** We give a proof analogously to the one of [1] for Theorem 6.1. First we reduce semantical consequences of sets to semantical consequences of coequations proving

$$U' \models U \iff \forall q \in Q : (U \models \boxtimes q) \Rightarrow (U' \models \boxtimes q).$$

To do so, recall from section 2.2 that  $U \subseteq Q$  is logically equivalent to a set of coequations  $\boxtimes q_i$ ,  $i \in I$  where  $\bigcup_{i \in I} q_i$  is the complement of  $U$ . From  $U' \models U$  and  $U \models \boxtimes q$  we immediately have  $U' \models U \models \boxtimes q$  and consequently  $U' \models \boxtimes q$  due to transitivity of  $\models$ . On the other hand, if we assume  $(U \models \boxtimes q) \Rightarrow (U' \models \boxtimes q)$  for all  $q \in Q$ , in particular we have  $(U \models \boxtimes q_i) \Rightarrow (U' \models \boxtimes q_i)$  for all  $i \in I$ . Since  $U \models \boxtimes q_i$  is always true for all  $i \in I$ , it holds that  $U' \models \boxtimes q_i$  for all  $i \in I$ . But this just means  $U' \models U$ .

Now we can apply Theorem 2.11 in order to get an expression for semantical consequences of sets in terms of the relation  $\sqsubseteq$ . Afterwards we use the logical equivalence  $a \Rightarrow b \equiv \neg b \Rightarrow \neg a$  to rewrite that expression:

$$\begin{aligned} & (U \models \boxtimes q) \Rightarrow (U' \models \boxtimes q) \\ \iff & (\forall t \exists r \exists q_i \in Q \setminus U : [t] = q \Rightarrow [r] = q_i \wedge r \sqsubseteq t) \\ & \Rightarrow (\forall t \exists r \exists q_i \in Q \setminus U' : [t] = q \Rightarrow [r] = q_i \wedge r \sqsubseteq t) \\ \iff & (\exists t \forall r \forall q_i \in Q \setminus U' : [t] = q \wedge ([r] \neq q_i \vee r \not\sqsubseteq t)) \\ & \Rightarrow (\exists t \forall r \forall q_i \in Q \setminus U : [t] = q \wedge ([r] \neq q_i \vee r \not\sqsubseteq t)) \end{aligned}$$

By slightly rewriting the premise of the latter formula (line before last) we see that it is exactly the premise in the theorem because  $\forall q_i \in Q \setminus U' : [r] \neq q_i$  just means  $[r] \in U'$ . But the conclusion of the formula (last line) is the conclusion in the theorem, too: in the special case  $r = t$  we get  $\exists t \forall q_i \in Q \setminus U : [t] = q \wedge [t] \neq q_i$ , what means  $q \in U$ . On the other hand, we cannot conclude anything else from the formula but  $q' \in U$  for some  $\epsilon$ -class  $q'$ ; this is covered by the for all quantifier for  $q$  in the theorem.  $\square$

## 7 Conclusion and future work

In the present paper we formulated a simple coequational logic for preimage preserving finitary functors and proved it to be sound and complete. We also proved that in general we cannot obtain this simple coequational logic for finitary functors that do not preserve preimages. Finally we proved a theorem for accessible functors that makes explicit what are semantical consequences of arbitrary subsets of a cofree coalgebra.

It would be of interest to look at infinitary accessible functors. In this case functors may preserve preimages but have no regular presentation, and the question arises which of both is the property needed to establish a simple logic (if this is

still possible at all). We conjecture that the regular presentation is the important property since in our paper we never needed to consider a preimage.

## References

- [1] Adámek, J., *A logic of coequations*, in: L. Ong, editor, *Computer Science Logic – Proceedings of the 19th International Workshop, LNCS 3634* (2005), pp. 70–86.
- [2] Adámek, J., D. Lücke and S. Milius, *Recursive coalgebras of finitary functors*, Theoretical Informatics and Applications (2007).
- [3] Adámek, J. and S. Milius, *Terminal coalgebras and free iterative theories*, Information and Computation **204** (2006), pp. 1139–1172.
- [4] Adámek, J. and H.-E. Porst, *On tree coalgebras and coalgebra presentations*, Theoretical Computer Science **311** (2004), pp. 257–283.
- [5] Adámek, J. and V. Trnková, “Automata and Algebras in Categories,” Kluwer Academic Publishers, 1990.
- [6] Gumm, H. P., *Elements of the general theory of coalgebras*, LUATCS '99, Rand Afrikaans University, Johannesburg, South Africa (1999).
- [7] Hughes, J., *Modal operators for coequations*, in: *Coalgebraic Methods in Computer Science (CMCS'01)*, Electronical Notes in Theoretical Computer Science **44.1**, 2001, pp. 204–225.
- [8] Moss, L. S., *Coalgebraic logic*, Annals of Pure and Applied Logic **96** (1999), pp. 177–317.
- [9] Pattinson, D., *Coalgebraic modal logic: soundness, completeness and decidability*, Theoretical Computer Science **309** (2003), pp. 177–193.
- [10] Rutten, J. J. R. R., *Universal coalgebra: a theory of systems*, Theoretical Computer Science **249** (2000), pp. 3–80.
- [11] Schwencke, D., “Coalgebraische Logik von markierten Transitionssystemen,” Master’s thesis, Technical University, Braunschweig (2007), unpublished.
- [12] Worrell, J., *On the final sequence of a finitary set functor*, Theoretical Computer Science **338** (2005), pp. 184–199.