Predicate and Relation Lifting for Parametric Algebraic Specifications

Hendrik Tews [1]

TU Dresden, Institut für Theoretische Informatik, D-01062 Dresden, Germany.

Abstract

Relation lifting [6] extends an endofunctor $F : \mathcal{C} \longrightarrow \mathcal{C}$ to a functor $\mathrm{Rel}(F) : \mathrm{Rel}(\mathcal{C}) \longrightarrow \mathrm{Rel}(\mathcal{C})$, where $\mathrm{Rel}(\mathcal{C})$ is a suitable category of relations over $\mathcal{C}$. The relation lifting for the functor F can be used to define the notion of bisimulation for coalgebras $X \longrightarrow F(X)$. The related notion of predicate lifting can be used to define invariants for F–coalgebras. Predicate and relation lifting can be directly defined for a rich class of polynomial functors [5,6,19]. In this paper I investigate the case where the functor F is defined as the initial semantics of a (single sorted) parametric algebraic specification.

1 Introduction

The notion of bisimulation captures the concept of behavioural equivalence. For a system described as a coalgebra $X \longrightarrow F(X)$ the definition of bisimulation is usually derived in a more or less direct way from the functor F that describes the interface of the system.

In this paper I investigate the question how to define the notion of bisimulation for functors that arise from (single sorted) parametric algebraic specifications with initial semantics. Such functors are important for specification environments such as CCSL [20] or CoCASL [10]. In these specification languages one can construct functors in an iterated way by nesting algebraic and coalgebraic specifications. One can, for instance, first define sequences on a set A as the final coalgebra for the functor $F_A^{\mathrm{Seq}}(X) = A \times X + \mathbf{1}$ and obtain a functor $\mathrm{Seq} : \mathbf{Set} \longrightarrow \mathbf{Set}$ that maps any set A to the possibly non-terminating sequences over A. One can then define trees of finite depth and infinite width with labels drawn from B as the initial algebra of $F_B^{\mathrm{Tree}}(X) = \mathrm{Seq}(X \times B)$ [5]. This defines a functor $\mathrm{Tree} : \mathbf{Set} \longrightarrow \mathbf{Set}$.

[1] Email: tews@tcs.inf.tu-dresden.de

This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

Elem : **Loose Specification**
Operation
$\mathsf{equal} : \mathsf{Elem} \times \mathsf{Elem} \longrightarrow \mathrm{bool}$
End Elem

$\mathsf{FSet}[T : \mathsf{Elem}]$: **Initial Specification**
Operations
$\mathsf{empty} : \mathsf{FSet}$
$\mathsf{add} : \mathsf{FSet} \times T \longrightarrow \mathsf{FSet}$
$\mathsf{del} : \mathsf{FSet} \times T \longrightarrow \mathsf{FSet}$
Variables
$x : \mathsf{FSet}; \quad t, t_1, t_2 : T$
Axioms
$\mathsf{add\_comm} : \mathrm{add}(\mathrm{add}(x, t_1), t_2) = \mathrm{add}(\mathrm{add}(x, t_2), t_1)$
$\mathsf{add\_idem} : \mathrm{add}(\mathrm{add}(x, t), t) = \mathrm{add}(x, t)$
$\mathsf{del\_empty} : \mathrm{del}(\mathrm{empty}, t) = \mathrm{empty}$
$\mathsf{del\_add\_eq} : \mathrm{equal}(t_1, t_2) = \mathrm{true} \Rightarrow \mathrm{del}(\mathrm{add}(x, t_1), t_2) = \mathrm{del}(x, t_2)$
$\mathsf{del\_add\_neq} : \mathrm{equal}(t_1, t_2) = \mathrm{false} \Rightarrow \mathrm{del}(\mathrm{add}(x, t_1), t_2) = \mathrm{add}(\mathrm{del}(x, t_2), t_1)$
End FSet

Fig. 1. Algebraic specification of finite sets with a delete operation

Consider a functor F that has been constructed in such a way from iterating algebraic specifications with initial semantics and coalgebraic specifications with final semantics. In [5] Hensel and Jacobs investigate the case of iterated data types, that is, where all these specification do not contain any axioms. Their approach also extends to the case in which the last step in the construction of F is a proper coalgebraic specification (i.e., contains axioms).

Assume now that the last step in the construction of F is a proper parametric algebraic specification $\mathcal{S}$. For simplicity I assume that $\mathcal{S}$ has only one parameter in the following. If this parameter is a type parameter, that is, if there are no operations (not even equality) used on the parameter, then F can be considered as a functor $\mathbf{Set} \longrightarrow \mathbf{Set}$. In this case one can use the approach of Aczel and Mendler [1] to define bisimulations on F–coalgebras.

This paper concentrates on the case where the specification $\mathcal{S}$ uses additional operations, for instance equality, on its parameter. In this case the parameter stands for a model of a parameter specification $\mathcal{T}$ (with usually loose semantics). Consequently F is then only a functor $\mathrm{Mod}(\mathcal{T}) \longrightarrow \mathbf{Set}$, where $\mathrm{Mod}(\mathcal{T})$ denotes the category of models of $\mathcal{T}$. I show in the sequel that in general for such functors the known approaches to define bisimulations fail.

Consider for example Figure 1. It contains an algebraic specification of finite sets. There are two important points: Firstly, the specification FSet is parametric in T, which stands for an arbitrary model of the specification Elem. Secondly, FSet specifies finite sets *with an additional delete operation.*

The specification of this delete operation relies on the operation equal that comes along with every model of Elem.

Admittedly, the preferred approach is to specify finite sets without the delete operation and to define it as a definitional extension. This way one could make the specification parametric in a type only (that is, without an equality operation) and avoid all the problems that result from the use of the equality relation of the parameter T.

However, as the specification FSet stands in Figure 1 it represents an interesting border case. It is interesting because the semantics and the relevant notions of predicate and relation lifting are well known—namely the finite powerset functor with its liftings [7]. Still, for technical reasons one cannot use the familiar approaches to define bisimulations for $[\![\mathsf{FSet}]\!]$–coalgebras. The reason is that in general the projections π_1 and π_2 do not reflect the equality relation on the parameter. Thus $[\![\mathsf{FSet}]\!](\pi_{1/2})$ is not even well-defined!

The following section on technicalities introduces some more special categorical notions. In the second subsection it describes the semantics of algebraic specifications. This subsection contains only standard material but introduces the notation and assumptions necessary for the rest of the paper. The assumptions are summarised at the end of the subsection on page 8. Section 3 contains the definition and Section 4 investigates its properties.

Acknowledgements. I thank Horst Reichel for many discussions and his explanations of the subtleties of algebraic specifications.

2 Technicalities

2.1 Category Theory

The category of sets and total functions is denoted by **Set**. I use the standard notation for products ($\times$) and coproducts or disjoint unions ($+$) with projections $\pi_{1/2}$, tupleing $\langle f, g\rangle$, injections $\kappa_{1/2}$, and cotupleing $[f, g]$. I write $X \Rightarrow Y$ for the exponent (the function space between X and Y). A n-ary polynomial functor is a functor $\mathbf{Set}^n \longrightarrow \mathbf{Set}$ generated from the following grammar:

$$F(X_1, \ldots, X_n) \;=\; A \;\mid\; X_i \;\mid\; F_1(\cdots)\times F_2(\cdots) \;\mid\; F_1(\cdots)+F_2(\cdots) \;\mid\; A \Rightarrow F_1(\cdots)$$

The grammar for iterated polynomial functors [5,14] contains additional clauses for least and greatest fixed points:

$$F(X_1, \ldots, X_n) \;=\; \ldots \;\mid\; \mu X \,.\, F(\ldots, X, \ldots) \;\mid\; \nu X \,.\, F(\ldots, X, \ldots)$$

Here $\mu X \,.\, F(\ldots, X, \ldots)$ denotes the initial algebra for the functor $F(\ldots, -, \ldots) : \mathbf{Set} \longrightarrow \mathbf{Set}$ (where all up to one argument positions of F are fixed) and $\nu X \,.\, F(\cdots)$ its final coalgebra. When used in an iterated polynomial functor $\mu X \,.\, F(\cdots)$ and $\nu X \,.\, F(\cdots)$ denote the carrier set of the initial and final coalgebra, respectively. The mapping on morphisms is defined as follows (I show only the case $n = 2$): Let $b : F(V, B) \longrightarrow B$ be the initial $F(V,-)$ algebra and $f : U \longrightarrow V$ be an arbitrary function. Then $b \circ F(f, \mathrm{id}_B)$ has the form of an $F(U,-)$ algebra, so there is an unique algebra homomorphism with $a : F(U,A) \longrightarrow A$, the initial $F(U,-)$ algebra, as domain. The underlying function of this algebra morphism gives $\mu X \,.\, F(f,X)$. To define $\nu X \,.\, F(f,X)$ one takes the unique coalgebra morphism for $F(f,\mathrm{id}) \circ c$, where c is the final $F(V,-)$ coalgebra.

A few notions of fibred category theory are relevant for this paper. I just define them here as operations on sets or predicates, disregarding their fibred nature. See [12,8] for an introduction into fibration. The material used in this paper is also covered in [4, Chapter 3] or [20, Chapter 2].

A predicate on a set X is written as $P \subseteq X = \{x \mid x \in P\}$. The truth predicate on X is $\top_X = X \subseteq X$. Predicates form the category **Pred** (in fact they are fibred over **Set**). A morphism $P \subseteq X \longrightarrow Q \subseteq Y$ is a function $f : X \longrightarrow Y$ such that $f\,x \in Q$ for all $x \in P$. A binary relation between X and Y is written as $R \subseteq X \times Y = \{(x : X, y : Y) \mid R(x,y)\}$. The equality relation on an arbitrary set A is $\mathrm{Eq}(A) = \{(a,a) \mid a \in A\}$. Relations form the category **Rel** (which is fibred over **Set** $\times$ **Set**). Morphisms are pairs of functions that preserve the domain relation $(\forall\,(x,y) \in R \,.\, (f\,x, g\,y) \in S)$. The inverse image of a function $f : X \longrightarrow Y$ with respect to a predicate $Q \subseteq Y$ is denoted with $f^*(Q) \subseteq X = \{x \in X \mid f(x) \in Q\}$. The image of f with respect to a predicate $P \subseteq X$ is $\coprod_f (P) \subseteq Y = \{f(x) \mid x \in P\}$.

For an *arbitrary* relation $R \subseteq X \times X$ the quotient X/R is the set of equivalence classes of X with respect to the *least equivalence relation containing* R. There is a quotient map $q_R : X \longrightarrow X/R$ that sends every x to its equivalence class $[x]_R$. In a suitable setting quotients are functorial [8, Section 4.8]. The morphism component of the quotient is also important here. For two relations $R_1 \subseteq X \times X$, $R_2 \subseteq Y \times Y$ and a function $f : X \longrightarrow Y$ such that $x\,R_1\,y \Rightarrow (f\,x)\,R_2\,(f\,y)$ the morphism component of the quotient functor is denoted as $\mathcal{Q}(f) : X/R_1 \longrightarrow Y/R_2$. It sends an equivalence class $[x]_{R_1}$ to $[f\,x]_{R_2}$. The equation $\mathcal{Q}(f) \circ q_{R_1} = q_{R_2} \circ f$ holds in general (it follows from the adjunction $\mathcal{Q} \dashv \mathrm{Eq}$).

For predicates and relations there exists a special Cartesian closed structure. For predicates $P \subseteq X$, $Q \subseteq Y$ and relations $S \subseteq U \times V$, $R \subseteq X \times Y$ it can be defined as follows:

$$
\begin{array}{rcl}
P \times_{\mathrm{P}} Q \;\subseteq\; X \times Y & = & \big\{(x,y) \mid x \in P \wedge y \in Q\big\} \\
P +_{\mathrm{P}} Q \;\subseteq\; X + Y & = & \{\kappa_1\,x \mid x \in P\} \,\cup\, \big\{\kappa_2\,y \mid y \in Q\big\} \\
P \Rightarrow_{\mathrm{P}} Q \;\subseteq\; X \Rightarrow Y & = & \big\{f \mid x \in P \Longrightarrow (f\,x) \in Q\big\} \\
S \times_{\mathrm{R}} R \;\subseteq\; (U \times X) \times (V \times Y) & = & \big\{((u,x),(v,y)) \mid u\,S\,v \wedge x\,R\,y\big\} \\
S +_{\mathrm{R}} R \;\subseteq\; (U + X) \times (V + Y) & = & \big\{(\kappa_1\,u, \kappa_1\,v) \mid u\,S\,v\big\} \\
 & & \quad \cup \quad \big\{(\kappa_2\,x, \kappa_2\,y) \mid x\,R\,y\big\} \\
S \Rightarrow_{\mathrm{R}} R \;\subseteq\; (U \Rightarrow X) \times (V \Rightarrow Y) & = & \big\{(f,g) \mid u\,S\,v \Longrightarrow (f\,u)\,R\,(g\,y)\big\}
\end{array}
$$

Predicate lifting [6] lifts a polynomial functor F on **Set** to a functor $\mathrm{Pred}(F)$ on **Pred**, such that $\mathrm{Pred}(F)(P) \subseteq F(X)$ for $P \subseteq X$. The predicate lifting for F can be used to define the notion of invariant for F–coalgebras. Relation lifting lifts F to a functor on **Rel** with $\mathrm{Rel}(F)(R) \subseteq F(X) \times F(Y)$ for $R \subseteq X \times Y$. It can be used to define bisimulations on F–coalgebras. One obtains these liftings by substituting the cartesian closed structure of predicates or relations for the cartesian closed structure of **Set** in F. More precisely, for a polynomial functor $F : \mathbf{Set} \times \mathbf{Set} \longrightarrow \mathbf{Set}$ and two relations R_1, R_2 the relation lifting $\mathrm{Rel}(F)(R_1, R_2)$ is inductively defined as follows:

$$
\begin{array}{rcl}
\mathrm{Rel}(A)(R_1, R_2) & = & \mathrm{Eq}(A) \\
\mathrm{Rel}(X_i)(R_1, R_2) & = & R_i \\
\mathrm{Rel}(F_1 \times F_2)(R_1, R_2) & = & \mathrm{Rel}(F_1)(R_1, R_2) \times_{\mathrm{R}} \mathrm{Rel}(F_2)(R_1, R_2) \\
\mathrm{Rel}(F_1 + F_2)(R_1, R_2) & = & \mathrm{Rel}(F_1)(R_1, R_2) +_{\mathrm{R}} \mathrm{Rel}(F_2)(R_1, R_2) \\
\mathrm{Rel}(A \Rightarrow F_1)(R_1, R_2) & = & \mathrm{Eq}(A) \Rightarrow_{\mathrm{R}} \mathrm{Rel}(F_1)(R_1, R_2)
\end{array}
$$

The precise definition for predicate lifting is very similar: predicate lifting employs the cartesian closed structure of predicates and uses $\top_A$ for the case $F(\cdots) = A$.

Hensel and Jacobs extend in [5] predicate and relation lifting to iterated polynomial functors. The liftings for $\mu X \,.\, F$ and $\nu X \,.\, F$ are defined as least and greatest fixed points in the following way: Consider a relation $R \subseteq X \times Y$ and let $\alpha : F(X, A) \longrightarrow A$ and $\beta : F(Y, B) \longrightarrow B$ be the initial algebras for $F(X, -)$ and $F(Y, -)$, respectively. The relation lifting $\mathrm{Rel}(\mu X \,.\, F(-, X))(R)$ is defined as the least fixed point of $S \longmapsto \coprod_{\alpha \times \beta} \mathrm{Rel}(F)(R, S)$. For the predicate lifting $\mathrm{Pred}(\mu X \,.\, F(-, X))(P)$ one takes the least fixed point of $Q \longmapsto \coprod_{\alpha} \mathrm{Pred}(F)(P, Q)$. Similarly for $\nu X \,.\, F$: For suitable final coalgebras ρ and γ the greatest fixed points of $Q \longmapsto \rho^* \,\mathrm{Pred}(F)(P, Q)$ and $S \longmapsto (\rho \times \gamma)^* \,\mathrm{Rel}(F)(R, S)$ define predicate lifting and relation lifting, respectively. These fixed points exist for all iterated polynomial functors.

The following two technical results have been proved in PVS [11].

Lemma 2.1 *The relation lifting of an equivalence relation is an equivalence relation for all polynomial functors.*

Lemma 2.2 *Let F be a polynomial functor. There is an isomorphism between $F(X_1, \ldots, X_n)/\mathrm{Rel}(F)(R_1, \ldots, R_n)$ and $F(X_1/R_1, \ldots, X_n/R_n)$ for all relations $R_i \subseteq X_i \times X_i$. The isomorphism is given by the function that maps $[x]_{\mathrm{Rel}(F)(\cdots)}$ to $F(q_{R_1}, \ldots, q_{R_n})(x)$.*

2.2 Specifications and their Semantics

The following sections rely on several assumptions on the form and especially the semantics of algebraic specifications. This section explains these assumptions and introduces the needed notions. The assumptions are summarised at the end of this subsection on page 8.

In this paper I only consider single sorted specifications with at most one parameter. The restriction to one sort is somewhat essential because I use morphisms $F(X) \longrightarrow X$ as semantics of algebraic specifications. Moreover, the proposed liftings for specifications rely on the liftings of Hensel and Jacobs for data types [5]. The restriction to one sort is also present in their work.

The restriction to one parameter is for convenience only. For more parameters one only has to add more arguments to the involved functors and liftings.

Throughout this paper I consider two specifications $\mathcal{T}$ and $\mathcal{S}$ like in Figure 1. The specification $\mathcal{T}$ is the parameter specification. The specification $\mathcal{S}$ is a single sorted algebraic specification with initial semantics. It is parametric in $\mathcal{T}$. Therefore the notions of signature, signature model, and model for $\mathcal{S}$ are also parametrised.

The parameter specification can be an arbitrary specification (not necessarily algebraic) with loose semantics, like the Elem specification in Figure 1. I neglect all syntactic issues and assume that the semantics of the parameter specification $\mathcal{T}$ is given as a category $\mathrm{Mod}(\mathcal{T})$ of models of $\mathcal{T}$. Objects in this category are tuples $T = \langle |T|, (f_i) \rangle$ consisting of a carrier set $|T|$ and a family of functions (f_i) for the operations declared in $\mathcal{T}$. Morphisms are functions between the carriers that are compatible with the operations. There is an obvious forgetful functor $U : \mathrm{Mod}(\mathcal{T}) \longrightarrow \mathbf{Set}$.

Example 2.3 Let *bool* be the set $\{\mathit{true}, \mathit{false}\}$ of boolean values. Models of the Elem specification in Figure 1 are pairs $\langle E, \mathit{eq} \rangle$, where E is a set and eq is a function $E \times E \longrightarrow \mathrm{bool}$. A function $f : E_1 \longrightarrow E_2$ is a morphism $\langle E_1, \mathit{eq}_1 \rangle \longrightarrow \langle E_2, \mathit{eq}_2 \rangle$ if it is compatible with the operation equal, that is, if $\mathit{eq}_2(f\, e_1, f\, e_2) = \mathit{eq}_1(e_1, e_2)$ for all $e_1, e_2 \in E_1$. Note that f is required to preserve *and reflect* the equal's predicate. This forms the category $\mathrm{Mod}(\mathsf{Elem})$ of models of Elem. ■

For the algebraic specification $\mathcal{S}$ I would also like to ignore syntactic issues. I assume that the signature of $\mathcal{S}$ is given as an iterated polynomial functor $\mathit{Sig} : \mathbf{Set} \times \mathbf{Set} \longrightarrow \mathbf{Set}$. This functor combines all operations of $\mathcal{S}$ into one. The first argument reflects the dependency on the parameter $\mathcal{T}$. A first argument from $\mathbf{Set}$ (in contrast to $\mathrm{Mod}(\mathcal{T})$) is sufficient, because (the semantics of) the signature depends only on the carrier of the parameter. A signature model is an interpretation of all operations in $\mathcal{S}$, which does not necessarily fulfil the axioms of $\mathcal{S}$. It can be given as an algebra $\mathit{Sig}(U, X) \longrightarrow X$, where U is an arbitrary set, possibly forming the carrier of a model of $\mathcal{T}$.

The algebras $\mathit{Sig}(U, X) \longrightarrow X$ (for arbitrary U) are the objects of the category $\mathrm{SigMod}(\mathcal{S})$. A Morphism in this category between an algebra $a : \mathit{Sig}(U, X) \longrightarrow X$ and an algebra $b : \mathit{Sig}(V, Y) \longrightarrow Y$ consists of a pair of functions $(f : U \longrightarrow V, g : X \longrightarrow Y)$ such that $g \circ a = b \circ \mathit{Sig}(f, g)$. There is an obvious forgetful functor $\mathcal{U} : \mathrm{SigMod}(\mathcal{S}) \longrightarrow \mathbf{Set}$ that sends the algebra a to U and the morphism (f, g) to f. The functor $\mu X \,.\, \mathit{Sig}(-, X)$ that maps a set T to the initial algebra of the functor $\mathit{Sig}(T, -)$ is left adjoint to $\mathcal{U}$. There is a second forgetful functor, denoted by $|-|$, that sends every algebra $\mathit{Sig}(U, X) \longrightarrow X$ to its carrier set X.

In order to deal with the axioms of the specification $\mathcal{S}$ I assume that for every model T of $\mathcal{T}$ and every signature model $a : \mathit{Sig}(|T|, X) \longrightarrow X$ it is possible to check if the axioms of $\mathcal{S}$ hold on a. If the axioms do hold the pair $\langle T, a\rangle$ is called a model of $\mathcal{S}$ for T.

The category $\mathrm{Mod}(\mathcal{S})$ has all models of $\mathcal{S}$ (for all possible T) as objects. Morphisms are pairs (f, g) of functions such that f is a morphism in $\mathrm{Mod}(\mathcal{T})$ and (f, g) is a morphism between the underlying signature models in $\mathrm{SigMod}(\mathcal{S})$. For every model T of $\mathcal{T}$ there is the subcategory $\mathrm{Mod}_T(\mathcal{S})$ of $\mathrm{Mod}(\mathcal{S})$ that contains just the models for T. Morphisms in $\mathrm{Mod}_T(\mathcal{S})$ are those that have the form $(\mathrm{id}_{|T|}, g)$. Again, there are two forgetful functors $\mathcal{U} : \mathrm{Mod}(\mathcal{S}) \longrightarrow \mathrm{Mod}(\mathcal{T})$ and $|-| : \mathrm{Mod}(\mathcal{S}) \longrightarrow \mathbf{Set}$.

There are two important assumptions on the specification $\mathcal{S}$: First, the forgetful functor $\mathcal{U}$ must have a left adjoint, denoted as $[\![\mathcal{S}]\!]$, such that $\mathcal{U} \circ [\![\mathcal{S}]\!]$ is the identity on $\mathrm{Mod}(\mathcal{T})$. As one might guess from the notation, the left adjoint $[\![\mathcal{S}]\!]$ is called the semantics of $\mathcal{S}$. It maps any model T to the initial object in $\mathrm{Mod}_T(\mathcal{S})$. The requirement that $\mathcal{U}([\![\mathcal{S}]\!](T)) = T$ amounts to the fact that the axioms of $\mathcal{S}$ place no restrictions on the parameter specification.

The second important assumption is that the semantics $[\![\mathcal{S}]\!]$ can be constructed as quotients of the initial algebras $\mu X \,.\, \mathit{Sig}(-, X)$. This quotient construction works as follows (consider also the diagram below): Let T be an arbitrary parameter model and let $\alpha = \mu X \,.\, \mathit{Sig}(|T|, X)$. The axioms of $\mathcal{S}$ determine a congruence relation (with respect to α) $Q_T \subseteq A \times A$ on the carrier of α such that the quotient A/Q_T is the carrier of $b = [\![\mathcal{S}]\!](T)$. Because Q_T is a congruence relation α is a morphism $\mathrm{Rel}(\mathit{Sig})(\mathrm{Eq}(|T|), \mathcal{Q}_T) \longrightarrow \mathcal{Q}_T$ between relations, so $\mathcal{Q}(\alpha)$ is as depicted in the diagram below. The operations on $A/\mathcal{Q}_T$ are defined via the composite $b = \mathcal{Q}(\alpha) \circ i^{-1}$, where i is the isomorphism from Lemma 2.2. The following diagram illustrates this construction (in the diagram I write $\mathrm{Rel}(\mathit{Sig})$ for $\mathrm{Rel}(\mathit{Sig})(\mathrm{Eq}(|T|), Q_T))$:

$$\begin{tikzcd}
\mathit{Sig}(|T|, A) \arrow[rrr, "\alpha"] \arrow[drr, "q_{\mathrm{Rel}(\mathit{Sig})}"'] & & & A \arrow[dr, "q_{Q_T}"] & \\
& & \mathit{Sig}(|T|, A)/\mathrm{Rel}(\mathit{Sig}) \arrow[rr, "\mathcal{Q}(\alpha)"] & & A/Q_T \\
& \mathit{Sig}(|T|, A/Q_T) \arrow[ur, "i^{-1}"] \arrow[urrr, "b"'] & & &
\end{tikzcd}$$

The morphism component of $[\![\mathcal{S}]\!]$ must also be constructed as a quotient: Consider a morphism $f : T \longrightarrow V$ in $\mathrm{Mod}(\mathcal{T})$. Then $\mu X \,.\, \mathit{Sig}(f, X)$ is an algebra morphism $\alpha \longrightarrow \mu X \,.\, \mathit{Sig}(|V|, X)$ and we can set $[\![\mathcal{S}]\!](f) = \mathcal{Q}(\mu X \,.\, \mathit{Sig}(f, X))$.

The assumption about the construction of $[\![\mathcal{S}]\!]$ is not too restrictive: The initial model of $\mathcal{S}$ can be obtained in the depicted way if the axioms of $\mathcal{S}$ are equational Horn formulas [9]. The origin of the relation Q_T is not relevant for this paper (only its existence). Meinke and Tucker construct Q_T as the intersection of the kernels of all algebra morphisms from α to $\mathrm{Mod}_T(\mathcal{S})$. If the axioms of $\mathcal{S}$ are positive Horn formulas one can construct Q directly from the axioms, as the following example suggests.

Example 2.4 The signature functor for the specification FSet from Figure 1 is $F(T, X) = \mathbf{1} + (X \times T) + (X \times T)$. From a signature model $\gamma : F(T, X) \longrightarrow X$ one obtains the single operations by precomposing an injection, for instance $\mathsf{add}_\gamma = \gamma \circ \kappa_2$. The lists over bool $\times$ $|T|$ form the initial signature model for T. The additional boolean in the cons cells is used to distinguish add's and del's. The equations of $\mathcal{S}$ have a simple structure. Therefore one can define the semantics of each axiom as a relation on X. For the last axiom this is for instance:

$$
\begin{aligned}
x \, [\![\mathsf{del\_add\_neq}]\!] \, y \quad \Longleftrightarrow & \\
& \exists t_1, t_2 \in |T|, z \in X \,.\, \neg \mathsf{eq}_T(t_1, t_2) \quad \wedge \\
& \qquad x = \mathsf{del}_\gamma(\mathsf{add}_\gamma(z, t_1), t_2) \quad \wedge \quad y = \mathsf{add}_\gamma(\mathsf{del}_\gamma(z, t_2), t_1)
\end{aligned}
$$

One obtains the semantics $Q_T^\gamma \subseteq X \times X$ as the union of the semantics of all axioms. The algebra γ is in $\mathrm{Mod}_T(\mathsf{FSel})$ if $Q_T^\gamma \subseteq \mathrm{Eq}(|T|)$.

The relation Q_T that can be used for factoring the initial F–algebra α is the least congruence relation that contains Q_T^α. The last two axioms of $\mathcal{S}$ are rewrite rules. For any list of add and del operations they permit to compute an equivalent list without del's. The relation Q_T relates two list, precisely if there del-free equivalents contain the same elements of T (disregarding order and multiple occurrences). So every equivalence class of Q_T describes a finite set of elements of $|T|$ (with respect to the equality eq_T). Finally, the initial algebra in $\mathrm{Mod}^T(\mathsf{FSet})$ is isomorphic to the finite sets with the obvious interpretation of the operations empty, add, and del. ∎

To summarise, there are the following assumptions for this paper:

- $\mathcal{T}$ is a single sorted parameter specification with loose semantics
- $\mathcal{S}$ is a single sorted algebraic specification that is parametric in $\mathcal{T}$ with initial semantics.
- the semantics $[\![\mathcal{S}]\!]$ of $\mathcal{S}$ is a functor $\mathrm{Mod}(\mathcal{T}) \longrightarrow \mathrm{Mod}(\mathcal{S})$ from the category of models of $\mathcal{T}$ to the one of models of $\mathcal{S}$. In particular, $[\![\mathcal{S}]\!]$ maps any function $f : U \longrightarrow V$ that is compatible with the operations of the parameter specification to a function between the initial models of $\mathcal{S}$ for U and V.
- The functor $[\![\mathcal{S}]\!]$ is left adjoint to the forgetful functor $\mathcal{U}$ and $\mathcal{U} \circ [\![\mathcal{S}]\!] = \mathrm{Id}_{\mathrm{Mod}(\mathcal{T})}$. Further, $[\![\mathcal{S}]\!]$ is constructed as quotient of the initial algebras for the signature of $\mathcal{S}$.

These assumptions are true if the axioms of $\mathcal{S}$ are equational Horn formulas [9] and if they do not place any restrictions on the parameter $\mathcal{T}$.

2.3 $[\![S]\!]$–Coalgebras

In general one could extend the definition of polynomial functors to admit ingredient functors of the form $[\![\mathcal{S}]\!]$ for parametric algebraic specifications $\mathcal{S}$ with initial semantics (and also for parametric coalgebraic specifications with final semantics). There are, however, two technical complications: Firstly, $[\![\mathcal{S}]\!]$ takes models of the parameter specification as arguments. So before applying $[\![\mathcal{S}]\!]$ to a set X one has to equip X with suitable operations (f_i). Of course in general the result of $[\![\mathcal{S}]\!](\langle X, (f_i)\rangle)$ depends on the choice of the operations. The second complication is that one has to use $|[\![\mathcal{S}]\!](\langle X, (f_i)\rangle)|$ to extract the carrier set of the model of $\mathcal{S}$. In this paper I would like to ignore these technical issues. I write $[\![\mathcal{S}]\!](X)$ to denote, depending on the context, both the model of $\mathcal{S}$ and its carrier set. The operations on X will be clear from the context.

In summary, the definition of a coalgebra $X \longrightarrow [\![\mathcal{S}]\!](X)$ is a four stage process: One has to define the carrier set X, equip it with suitable operations (f_i), form the application $[\![\mathcal{S}]\!](X)$, and finally define the coalgebra itself. Note that for such a coalgebra the notion of coalgebra morphisms is only meaningful for functions g that are actually morphisms in $\mathrm{Mod}(\mathcal{T})$ (thus preserving the operations (f_i)) because otherwise the application $[\![\mathcal{S}]\!](g)$ is not well formed.

In this paper I consider only coalgebras of the form $X \longrightarrow [\![\mathcal{S}]\!](X)$, where the signature functor of $\mathcal{S}$ does *not* contain an ingredient derived from an algebraic specification.

2.4 Bisimulations for Coalgebras

There are at least three approaches to define the bisimulations between two coalgebras $c : X \longrightarrow F(X)$ and $d : Y \longrightarrow F(Y)$:

(i) Aczel and Mendler suggest the following diagram as defining property [1]:

$$\begin{tikzcd} X \arrow[d, "c"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d] & Y \arrow[d, "d"] \\ F(X) & F(R) \arrow[l, "F(\pi_1)"'] \arrow[r, "F(\pi_2)"] & F(Y) \end{tikzcd}$$

(ii) Hermida and Jacobs [6] suggest to use the relation lifting $\mathrm{Rel}(F)$ for F and define bisimulations as relations R that fulfil

$$\forall x, y \,.\, x\, R\, y \quad \Longrightarrow \quad (c\, x)\mathrm{Rel}(F)(R)(d\, y)$$

(iii) One can also define bisimulations as those relations whose elements are made equal by a pair of coalgebra morphisms. More precisely, R is a bisimulation if there exists a coalgebra $z : Z \longrightarrow F(Z)$ and coalgebra morphisms $f : X \longrightarrow Z, g : Y \longrightarrow Z$ such that $x\,R\,y \Leftrightarrow f\,x = g\,y$.

For functors that preserve weak pullbacks all three approaches are equivalent [18].

Consider now FSet–coalgebras. Note that neither the first nor the second approach can be used to define bisimulations for FSet–coalgebras. The first approach fails because it is usually impossible to turn a relation R into a model of Elem such that the projections are morphism in Mod(Elem). The problem with the second approach is that relation lifting must be explicitly defined for each functor. The generic property that links the relation lifting of F with F itself, namely the equation $\mathrm{Rel}(F)(R) = \coprod_{\langle F(\pi_1), F(\pi_2)\rangle} F(R)$, appearing in [17,2], cannot be used for FSet, again because of the projections.

The third approach can be used for FSet–coalgebras, but in general it yields an unsatisfactory notion of bisimulation as the following example shows.

Example 2.5 Consider the specification of finite sets from Figure 1 with the following changes: Add a new operation $\mathsf{color} : \mathsf{Elem} \longrightarrow \mathsf{bool}$ to the parameter specification Elem together with the axiom $\forall t_1, t_2 : \mathsf{Elem} \,.\, \mathsf{color}(t_1) = \mathsf{color}(t_2)$. This implies that all elements of one model have the same colour. However, models of Elem can be coloured either with true or false.

Consider now the following two coalgebras (I omit the operation equal because it is not relevant):

$$
\begin{array}{rcl}
X & = & \{x\} \\
\mathsf{color}(x) & = & \mathsf{true} \\
c & : & X \longrightarrow [\![\mathsf{FSet}]\!](X) \\
c(x) & = & \mathsf{empty}
\end{array}
\qquad\qquad
\begin{array}{rcl}
Y & = & \{y\} \\
\mathsf{color}(y) & = & \mathsf{false} \\
d & : & Y \longrightarrow [\![\mathsf{FSet}]\!](Y) \\
d(y) & = & \mathsf{empty}
\end{array}
$$

Intuitively both states x and y show the same behaviour (with respect to the coalgebra), so there should be a bisimulation relating them. However, as a morphism of Elem, every coalgebra morphism has to preserve the colour of the state. There is no model of the changed Elem specification with two different colours. Therefore it is impossible to find a pair of morphisms that map x and y to the same element. ■

3 Definition of Predicate and Relation Lifting

This section presents a solution for the problem of defining bisimulations for FSet–coalgebras: A definition of predicate and relation lifting that relies on the liftings for $\mu X \,.\, Sig(-, X)$ of Hensel and Jacobs [5]. The straightforward approach, namely to factor the liftings of the iterated polynomial signature functor by the congruence relation $\mathcal{Q}_T$ (which provides the semantics of the axioms) does *not* work, see Example 3.2 below.

Definition 3.1 Let $\mathcal{S}$ be an algebraic specification that is parametrised by a specification $\mathcal{T}$. Assume that the semantics $[\![\mathcal{S}]\!] : \mathrm{Mod}(\mathcal{T}) \longrightarrow \mathrm{Mod}(\mathcal{S})$ is constructed as described in Section 2.2.

(i) A set of canonical forms is an indexed set $(C_T \subseteq \mu X\,.\,\mathit{Sig}(|T|, X))_{T \in |\mathrm{Mod}(\mathcal{T})|}$ such that for all $x \in \mu X\,.\,\mathit{Sig}(|T|, X)$ we have $[x] \cap C_T \neq \emptyset$.

(ii) Let T and V be models of $\mathcal{T}$. Assume sets of canonical forms C_T and C_V. For a predicate $P \subseteq |T|$ and a relation $R \subseteq |T| \times |V|$ the predicate and relation lifting of $[\![\mathcal{S}]\!]$ is defined as

$$\begin{array}{rcl} \mathrm{Pred}([\![\mathcal{S}]\!])(P) & = & \coprod_{q_{Q_T}} \big(\mathrm{Pred}(\mu X\,.\,\mathit{Sig}(-, X))(P) \;\cap C_T\big) \\ \mathrm{Rel}([\![\mathcal{S}]\!])(R) & = & \coprod_{q_{Q_T} \times q_{Q_V}} \big(\mathrm{Rel}(\mu X\,.\,\mathit{Sig}(-, X))(R) \;\cap C_T \times C_V\big) \end{array}$$

Here, Sig is the signature functor of $\mathcal{S}$, $\mu X\,.\,\mathit{Sig}(-, X)$ denotes the functor that maps any set A to the initial $\mathit{Sig}(A, -)$–algebra, $\mathrm{Pred}(-)$ and $\mathrm{Rel}(-)$ are the predicate and relation lifting for this functor as described by Hensel and Jacobs [5], and q_{Q_T} is the quotient map for the quotient with respect to Q_T, the relation that is used in the construction of $[\![\mathcal{S}]\!](T)$.

The condition on the canonical forms ensures that in each equivalence class of terms there is at least one canonical term. Let me first show an example before I further elaborate on the canonical forms.

Example 3.2 Consider again the specification of Figure 1 (without changes). Example 2.4 describes the signature functor F and the semantics of FSet.

Let T and V be models of Elem and consider a predicate $P \subseteq |T|$ and a relation $R \subseteq |T| \times |V|$. Hensel and Jacobs define predicate and relation lifting of $\mu x\,.\,F(-, X)$ as the least fixed points of the following equations:

$$\begin{array}{rcl} \mathrm{Pred}(\mu X\,.\,F(-, X))(P) & = & \{\mathsf{empty}\} \cup \\ & & \{\mathsf{add}(l, t) \mid t \in P \wedge l \in \mathrm{Pred}(\mu X\,.\,F(-, X))(P)\} \cup \\ & & \{\mathsf{del}(l, t) \mid t \in P \wedge l \in \mathrm{Pred}(\mu X\,.\,F(-, X))(P)\} \end{array}$$

$$\begin{array}{rcl} \mathrm{Rel}(\mu X\,.\,F(-, X))(R) & = & \{(\mathsf{empty}, \mathsf{empty})\} \cup \\ & & \{(\mathsf{add}(l_1, t_1), \mathsf{add}(l_2, t_2)) \mid t_1 \mathrel{R} t_2 \wedge (l_1, l_2) \in \mathrm{Rel}(\mu X\,.\,F(-, X))(R)\} \cup \\ & & \{(\mathsf{del}(l_1, t_1), \mathsf{del}(l_2, t_2)) \mid t_1 \mathrel{R} t_2 \wedge (l_1, l_2) \in \mathrm{Rel}(\mu X\,.\,F(-, X))(R)\} \end{array}$$

The lifted predicate $\mathrm{Pred}(\mu X\,.\,F(-, X))(P)$ holds for a list l if all elements from T that occur in l are in P. The relation $\mathrm{Rel}(\mu X\,.\,F(-, X))(R)$ relates two lists l_1 and l_2 if they contain the same sequence of add's and del's (in the same order) such that the elements of T and V at corresponding positions are related by R.

The equivalence classes with respect to Q_T correspond to finite sets over $|T|$ (with respect to the equality eq_T). The predicate lifting $\mathrm{Pred}([\![\mathsf{FSet}]\!])(P)$ holds for such a set s, precisely if there exists a representation of this set as a canonical form $l \in C_T$ of add and del operations such that $l \in$

$\mathrm{Pred}(\mu X\,.\,F(-,X))(P)$. The canonical forms are mainly important for relation lifting. Here we see that, in case C_T contains enough terms (for instance if $C_T = |\mu X\,.\,Sig(|T|,X)|$), one easily derives $s \in \mathrm{Pred}([\![\mathsf{FSet}]\!])(P) \Leftrightarrow s \subseteq P$.

Turn now to relation lifting. Consider a set s over $|T|$ and a set p over $|V|$. They are related by $\mathrm{Rel}([\![\mathsf{FSet}]\!])(R)$ if there exist canonical representations $l_s \in C_T$ and $l_p \in C_V$ such that l_s and l_p are related by $\mathrm{Rel}(\mu X\,.\,F(-,X))(R)$.

The problem here is (and that is where the canonical forms are needed) that completely different sets might have representations that are related by $\mathrm{Rel}(\mu X\,.\,F(-,X))$. Consider for instance the terms $t_1 = \mathsf{del}(\mathsf{add}(\mathsf{empty},a),a)$ and $t_2 = \mathsf{del}(\mathsf{add}(\mathsf{empty},b),c)$ (for arbitrary $a \in T, b, c \in V$). From the axioms of FSet we know that $t_1 = \mathsf{empty}$ and that $t_2 = \mathsf{add}(\mathsf{empty},b)$. Thus t_1 is a representation for the empty set, while t_2 represents the set $\{b\}$. Nevertheless, if R contains both (a,b) and (a,c) then $(t_1,t_2) \in \mathrm{Rel}(\mu X\,.\,F(-,X))(R)$. In case both terms are canonical ($t_1 \in C_T$ and $t_2 \in C_V$), the relation lifting for $[\![\mathsf{FSet}]\!]$ relates the empty set with the set $\{b\}$, which is highly undesirable.

To avoid this problem one defines the set of canonical terms to be those terms that contain no del-operation. With this choice it is easy to see that the relation lifting for $[\![\mathsf{FSet}]\!]$ relates two sets s and p precisely if for every $t \in s$ there is a $v \in p$ such that $t\,R\,v$ and vice versa. ■

Remark 3.3 The set of canonical forms has to be defined during the development of the specification. The liftings depend in a crucial way on the canonical forms. So they should be chosen with care. If the specification provides normal forms then one can take the singleton sets of normal forms as canonical forms. However, the preceding example shows that normal forms are not necessary.

The canonical forms are needed because the axioms of the specification $\mathcal{S}$ (when read as rewrite rules) might introduce or delete elements of the parameter specification (like $\mathsf{del\_empty}$ in the specification FSet). If there are no such axioms one can use the set of all terms as canonical forms.

The impression that arises from the FSet-example, that the canonical forms can be generated by a subsignature is misleading. It merely indicates that the FSet-example is somewhat contrived: The operation del should be specified as a definitional extension.

Remark 3.4 The liftings of the preceding definition are almost functors on **Pred** and **Rel**, respectively. To be precise, the liftings are endofunctors on the categories $\mathbf{Pred}^{\mathcal{T}}$ and $\mathbf{Rel}^{\mathcal{T}}$ of predicates and relations over models of $\mathcal{T}$. These categories are fibred over $\mathrm{Mod}(\mathcal{T})$ and $\mathrm{Mod}(\mathcal{T}) \times \mathrm{Mod}(\mathcal{T})$, respectively. Objects of $\mathbf{Pred}^{\mathcal{T}}$ are pairs (P,T) where T is a model of $\mathcal{T}$ and $P \subseteq |T|$. Similarly for $\mathbf{Rel}^{\mathcal{T}}$.

The following proposition shows that Definition 3.1 is reasonable: The definition yields the expected liftings for the specification FSet.

Proposition 3.5 *The liftings for* FSet *as described in Definition 3.1 coincide* *with the liftings for the finite powerset functor defined in [7] if one takes the* **del***–free terms as canonical forms.*

More precisely, let $\mathcal{P}_{fin}(T)$ denote the initial model in $\mathrm{Mod}_T(\mathcal{S})$ that consists of the finitary powerset of $|T|$ with the obvious interpretation of the three operations of $\mathcal{S}$. Let i be the algebra isomorphism between $\mathcal{P}_{fin}(P)$ and $[\![\mathcal{S}]\!](T)$. Then $i^*\,(\mathrm{Pred}([\![\mathcal{S}]\!])(P)) = \mathrm{Pred}(\mathcal{P}_{fin})(P)$ for all suitable P. A similar equation holds for the relation lifting.

Proof. Obvious, after the considerations of Example 3.2. Nevertheless, this proof has been formalised in PVS [11]. In PVS the proof is surprisingly difficult, because it proceeds by induction and involves the construction of suitable sets and lists. $\square$

4 Properties of the new Liftings

In this section I investigate some properties of the proposed liftings. There is often a tight connection between these properties and the ones of bisimulations and invariants. For instance, if the relation lifting for F commutes with intersection then one immediately has that the intersection of two F–bisimulations is a F bisimulation again (see [19] or [20, Section 2.6] for a collection of interesting properties).

Obviously, the properties of the liftings for $[\![\mathcal{S}]\!]$ depend in a crucial way on the properties of the liftings of the iterated polynomial functor $\mu X\,.\,Sig(-,X)$. Unfortunately there are only few results available for the liftings of iterated polynomial functors. Hensel and Jacobs prove that these liftings are fibred and preserve truth and equality [5]. Surprisingly there are a number of results for which the choice of canonical forms is not relevant.

Throughout this section I assume that $\mathcal{S}$ is an algebraic specification with a parameter specification $\mathcal{T}$ such that Definition 3.1.ii is applicable to $\mathcal{S}$.

Lemma 4.1 (Monotonicity) *The liftings for $[\![\mathcal{S}]\!]$ are monotone: Assume $P_1 \subseteq P_2$ and $R_1 \subseteq R_2$ for suitable predicates P_1, P_2 and relations R_1, R_2. Then:*

$$\begin{array}{rcl} \mathrm{Pred}([\![\mathcal{S}]\!])(P_1) & \subseteq & \mathrm{Pred}([\![\mathcal{S}]\!])(P_2) \\ \mathrm{Rel}([\![\mathcal{S}]\!])(R_1) & \subseteq & \mathrm{Rel}([\![\mathcal{S}]\!])(R_2) \end{array}$$

Proof. Predicate and relation lifting for iterated polynomial functors is monotone [5]. Further, the coproduct $\coprod_f$ along an arbitrary function is a functor between fibre categories, thus it also monotone. $\square$

The result about monotonicity ensures that $\mathrm{Pred}([\![\mathcal{S}]\!])$ and $\mathrm{Rel}([\![\mathcal{S}]\!])$ are functors on $\mathbf{Pred}^T$ and $\mathbf{Rel}^T$, respectively.

Remark 4.2 Monotonicity also ensures that one can apply the definition of Hensel and Jacobs to get liftings for the (constant) functors $\mu X\,.\,[\![\mathcal{S}]\!](X)$ and

$\nu X \,.\, [\![\mathcal{S}]\!](X)$. These are not iterated polynomial functors, so next to nothing is known about these functors and their liftings.

Proposition 4.3 (Truth and Equality) *Predicate and relation lifting of $\mathcal{S}$ commute with truth and equality in the following sense:*

$$\begin{array}{rcl} \mathrm{Pred}([\![\mathcal{S}]\!])(\top_{|T|}) & = & \top_{[\![\mathcal{S}]\!](T)} \\ \mathrm{Rel}([\![\mathcal{S}]\!])(\mathrm{Eq}(|T|)) & = & \mathrm{Eq}([\![\mathcal{S}]\!](T)) \end{array}$$

Proof. The liftings of iterated polynomial functors commute with truth and equality [5]. Further, the Definition 3.1.i ensures that C_T contains terms from every equivalence class, so $\coprod_{q_Q} C_T = \top_{[\![\mathcal{S}]\!](X)}$. □

The preceding result about truth and equality ensures that the truth predicate is an invariant for $[\![\mathcal{S}]\!]$–coalgebras and that the equality relation is a bisimulation. The next result can be used to prove that bisimulations are closed under union.

Proposition 4.4 (Unions) *Consider the models T and V of $\mathcal{T}$ and let $(P_i \subseteq |T|)_{i\in I}$ and $(R_i \subseteq |T| \times |V|)_{i\in I}$ be collections of predicates and relations over an arbitrary index set I. Assume that the liftings for $\mu X \,.\, \mathit{Sig}(-, X)$ preserve the unions of these collections in the following sense:*

$$\begin{array}{rcl} \bigcup_i \mathrm{Pred}(\mu X \,.\, \mathit{Sig}(-,X))(P_i) & \subseteq & \mathrm{Pred}(\mu X \,.\, \mathit{Sig}(-,X))(\bigcup_i P_i) \\ \bigcup_i \mathrm{Rel}(\mu X \,.\, \mathit{Sig}(-,X))(R_i) & \subseteq & \mathrm{Rel}(\mu X \,.\, \mathit{Sig}(-,X))(\bigcup_i R_i) \end{array}$$

Then, the liftings of $[\![\mathcal{S}]\!]$ do also preserve unions of arbitrary collections:

$$\begin{array}{rcl} \bigcup_i \mathrm{Pred}([\![\mathcal{S}]\!])(P_i) & \subseteq & \mathrm{Pred}([\![\mathcal{S}]\!])(\bigcup_i P_i) \\ \bigcup_i \mathrm{Rel}([\![\mathcal{S}]\!])(R_i) & \subseteq & \mathrm{Rel}([\![\mathcal{S}]\!])(\bigcup_i R_i) \end{array}$$

The assumption about the liftings of $\mu X \,.\, \mathit{Sig}(-, X)$ is necessary, because there is (up to my knowledge) no general result available. Further, it only makes sense to assume an inclusion, because equality does not hold in general.

Proof. The coproduct along every function f commutes with arbitrary unions of collections: $\coprod_f \bigcup_i P_i = \bigcup_i \coprod_f P_i$. □

As next property I consider fibredness. The predicate lifting for a functor F is fibred if the equation $F(f)^* \,\mathrm{Pred}(F)(P) = \mathrm{Pred}(F)(f^* \,P)$ holds for all suitable functions f and predicates P. Similarly the relation lifting for F is fibred if $(F(f) \times F(g))^* \,\mathrm{Rel}(F)(R) = \mathrm{Rel}(F)((f \times g)^* \,R)$. At first fibredness is the technical property that ensure that $\mathrm{Pred}(F)$ and $\mathrm{Rel}(F)$ are fibred functors (i.e., functors between fibrations) over F. However, fibredness of relation lifting admits an elegant proof for the proposition that says that a function f is a coalgebra homomorphism precisely if its graph is a bisimulation.

The following example shows that fibredness can fail if the canonical forms are not carefully chosen.

T : **Loose Specification**
  Operation
    k : T
End T

S[T : T] : **Initial Specification**
  Operations
    nil : S
    cons : S $\times T \longrightarrow$ S
    m : S $\times T \longrightarrow$ S
  Variables l : S; $t : T$
  Axioms
    a : $\mathrm{m}(l,t) = \mathrm{cons}(l, \mathsf{k})$
End S

Fig. 2. Specification for Example 4.5

Example 4.5 Consider the specification S from Figure 2. It specifies lists with an additional (nonsensical) Operation m such that $\mathsf{m}(l,t)$ equals $\mathsf{cons}(l,\mathsf{k})$ for an identified constant k of the parameter.

Let T be a model of the parameter T such that $|T| = \{t_1, t_2\}$ is a two element set and $\mathsf{k}_T = t_1$. Then the function f that maps both t_1 and t_2 to t_1 is a morphism $T \longrightarrow T$. The initial semantics of S are the finite lists containing just t_1 or t_2 with an additional operation m that conses t_1 to its argument list. Let the set of canonical forms C_T be the set of all terms.

Consider now the predicate $P = \{t_2\}$. Note that $f^* P = \emptyset$. Let $[-]_Q$ denote the equivalence class with respect to the relation Q that is used for factoring the initial $\mu X \,.\, Sig_{\mathsf{S}}(T, X)$ algebra. We have that $[\mathsf{cons}(\mathsf{nil}, t_1)]_Q \in ([\![\mathsf{S}]\!](f))^* \,\mathrm{Pred}([\![\mathsf{S}]\!])(P)$ (because $\mathsf{m}(\mathsf{nil}, t_2) \in \mathrm{Pred}(\mu X \,.\, Sig_{\mathcal{S}}(-, X))(P)$ and $\mathcal{Q}(\mu X \,.\, Sig_{\mathcal{S}}(f, X))$ maps $[\mathsf{cons}(\mathsf{nil}, t_1)]_Q$ to $[\mathsf{cons}(\mathsf{nil}, t_1)]_Q$). However $\mathrm{Pred}([\![\mathsf{S}]\!])(f^* P) = \{[\mathsf{nil}]_Q\}$.

One obtains a counterexample for fibredness of the relation lifting when considering the relation $\{(t_2, t_2)\} \subseteq |T| \times |T|$. Both, predicate and relation lifting are fibred if one restricts the set of canonical forms to those terms that do not contain m. ■

The last two examples demonstrate that, depending on the canonical forms, the liftings for $[\![\mathcal{S}]\!]$ might not commute with intersection and composition of relations.

Example 4.6 For polynomial functors one has $\mathrm{Pred}(F)(P) \cap \mathsf{Pred}(F)(Q) = \mathrm{Pred}(F)(P \cap Q)$. To obtain a counterexample for this equation for the lifting of $[\![\mathcal{S}]\!]$ consider a specification similar to that in Figure 2. This time assume two special elements k and l in T with the axiom $\mathsf{cons}(\mathsf{nil}, \mathsf{k}) = \mathsf{cons}(\mathsf{nil}, \mathsf{l})$ in S (and ignore the operation m). Let the canonical forms be the set of all terms and consider two predicates $P = \{\mathsf{k}\}$ and $Q = \{\mathsf{l}\}$. Obviously $t = [\mathsf{cons}(\mathsf{nil}, \mathsf{k})]$ is in both $\mathrm{Pred}(\llbracket\mathcal{S}\rrbracket)(P)$ and $\mathrm{Pred}(\llbracket\mathcal{S}\rrbracket)(Q)$. However, $t \notin \mathrm{Pred}(\llbracket\mathcal{S}\rrbracket)(P \cap Q)$. One obtains a counterexample for the intersection of relations from relations that contain (k, k) and (l, l), respectively. ■

Example 4.7 Again for polynomial functor it holds that $\mathrm{Rel}(F)(R) \circ \mathrm{Rel}(F)(S) = \mathrm{Rel}(F)(R \circ S)$. Here is a counterexample for the relation lifting of $\llbracket\mathcal{S}\rrbracket$: Consider again the specification of Figure 2 without the operation m. Assume an axiom $\mathsf{cons}(\mathsf{nil}, \mathsf{k}) = \mathsf{nil}$ and let all terms be canonical. Let R be a relation that contains the pair (a, k) for $a \neq \mathsf{k}$. Let S be the empty relation. We have that the pair $(\mathsf{cons}(\mathsf{nil}, a), \mathsf{nil})$ is in the lifting of R and of course $(\mathsf{nil}, \mathsf{nil})$ is in the lifting of S. However, $R \circ S$ is the empty relation, so $(\mathsf{cons}(\mathsf{nil}, a), \mathsf{nil}) \notin \mathrm{Rel}(\llbracket\mathcal{S}\rrbracket)(R \circ S)$. ■

5 Conclusion

This paper presents a definition of predicate and relation lifting for functors obtained as initial semantics of parametric algebraic specifications. This makes it possible to defined the notion of bisimulation and invariant for coalgebras of such functors. The paper contains examples that show that alternative approaches to define bisimulations for such functors are not satisfactory.

There is an obvious approach to obtain liftings for algebraic specifications: One takes the image of the liftings of Hensel and Jacobs [5] for the initial algebra of the signature along the quotient map that factors the carrier of the initial algebra into a model of the specification. However, this approach *does not* yield satisfactory notions. Therefore I propose a modified definition that employs *sets of canonical forms*, which must be developed together with the specification.

The paper investigates some interesting properties of the new liftings. Some of them hold trivially (for instance preserving truth, equality, and unions), other essential properties fail (fibredness, preservation of intersection and relational composition) for a particularly chosen set of canonical forms (the properties might hold for a different choice). So the proposed definition is not completely satisfactory, however, it provides the only means to define bisimulations for functors stemming from algebraic specification.

This work has been inspired by possible extensions of the coalgebraic specification language CCSL [16]. The design of CoCASL [10] also permits nested algebraic and coalgebraic specifications. However, Mossakowski and colleagues do not discuss how to define bisimulations. Several incarnations of the OBJ–Family [3,15] do also permit nested algebraic and coalgebraic (behavioural—in OBJ terms) specifications. In OBJ the notion of behavioural equivalence is not derived in a general way from the signature. Instead the user must specify observers in the subsignature of congruent operations. Behavioural equivalence is then defined via the visible contexts [13] of the congruent operations.

Future work includes the investigation of more properties and a better understanding of the role of the canonical forms. In particular, I would like to derive conditions on the set of canonical forms that ensure for instance that the liftings commute with intersection. It would also be interesting to implement algebraic specifications and their liftings in the CCSL compiler to permit larger case studies.

References

[1] Aczel, P. and P. F. Mendler, *A final coalgebra theorem*, in: D. H. Pitt, D. E. Rydeheard, P. Dybjer, A. M. Pitts and A. Poigné, editors, *Proceedings of the Conference on Category Theory and Computer Science*, LNCS **389** (1989), pp. 357–365.

[2] Carboni, A., G. M. Kelly and R. J. Wood, *A 2–categorical approach to change of base and geometric morphisms i.*, Technical Report 90–1, Department of Pure Mathematics, University of Sydney (1990), iSSN 1033–2359.

[3] Goguen, J. and G. Malcolm, *A hidden agenda*, Theoretical Computer Science **245** (2000), pp. 55–101.

[4] Hensel, U., "Definition and Proof Principles for Data and Processes," Ph.D. thesis, Univ. of Dresden, Germany (1999).

[5] Hensel, U. and B. Jacobs, *Proof principles for datatypes with iterated recursion*, in: E. Moggi and G. Rosolini, editors, *Category Theory and Computer Science*, LNCS **1290** (1997), pp. 220–241.

[6] Hermida, C. and B. Jacobs, *Structural induction and coinduction in a fibrational setting*, Information and Computation **145** (1998), pp. 107–152.

[7] Jacobs, B., *Mongruences and cofree coalgebras*, in: V. Alagar and M. Nivat, editors, *Algebraic Methodology and Software Technology*, LNCS **936** (1995), pp. 245–260.

[8] Jacobs, B., "Categorical Logic and Type Theory," Studies in Logic and the Foundations of Mathematics **141**, North Holland, Elsevier, 1999.

[9] Meinke, K. and J. V. Tucker, *Universal algebra*, in: S. Abramsky, D. M. Gabbay and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science. Volume 1. Background: Mathematical Structures*, Oxford University Press, 1992 pp. 189–409.

[10] Mossakowski, T., H. Reichel, M. Roggenbach and L. Schröder, *Algebraic–coalgebraic specification in cocasl*, to appear in the proceedings of the 16th WADT.

[11] Owre, S., S. Rajan, J. Rushby, N. Shankar and M. Srivas, *PVS: Combining specification, proof checking, and model checking*, in: R. Alur and T. Henzinger, editors, *Computer Aided Verification*, LNCS **1102** (1996), pp. 411–414.

[12] Phoa, W., *An introduction to fibrations, topos theory, the effective topos and modest sets*, Research report ECS-LFCS-92-208, Lab. for Foundations of Computer Science, University of Edinburgh (1992).

[13] Reichel, H., *Behavioural validity of conditional equations in abstract data types*, in: *Contributions to General Algebra 3*, Teubner, 1985 Proceedings of the Vienna Conference, June 21-24, 1984.

[14] Rößiger, M., "Coalgebras, Clone Theory and Modal Logic," Ph.D. thesis, Univ. of Dresden, Germany (2000).

[15] Roşu, G., "Hidden Logic," Ph.D. thesis, University of California at San Diego (2000).

[16] Rothe, J., H. Tews and B. Jacobs, *The coalgebraic class specification language CCSL*, Journal of Universal Computer Science **7** (2001), pp. 175–193.

[17] Rutten, J. J. M. M., *Relators and metric bisimulations (extended abstract)*, in: B. Jacobs, L. Moss, H. Reichel and J. Rutten, editors, *Workshop on Coalgebraic Methods in Computer Science '98*, ENTCS **11** (1998), pp. 257–263, available at URL http://www.elsevier.nl/locate/entcs.

[18] Rutten, J. J. M. M., *Universal coalgebra: A theory of systems*, Theoretical Computer Science **249** (2000), pp. 3–80.

[19] Tews, H., *Coalgebras for binary methods: Properties of bisimulations and invariants*, Theoretical informatics and applications **35** (2001), pp. 83–111.

[20] Tews, H., "Coalgebraic Methods for Object-Oriented Specification," Ph.D. thesis, University of Dresden (2002), available via wwwtcs.inf.tu–dresden/∼tews.