Some useful structures for categorical approach for program behavior

Viliam Slodičák, Martina Ľaľová
Department of computers and informatics, FEI TU Kosice
Letná 9, 042 00 Košice, Slovakia
{viliam.slodicak, martina.lalova}@tuke.sk

Abstract. *Using of category theory in computer science has extremely growth in the last decade. Categories allow us to express mathematical structures in unified way. Algebras are used for constructing basic structures used in computer programs. A program can be considered as an element of the initial algebra arising from the used programming language. In our contribution we formulate two ways of expressing algebras in categories. We also construct the codomain functor from the arrow category of algebras into the base category of sets which objects are also the carrier-sets of the algebras. This functor expresses the relation between algebras and carrier-sets.*

Keywords. Algebra, category theory, monad, Kleisli category

1 Overview

The aim of programming is to construct such correct programs and program systems that during their execution provide expected behavior. A program can be considered as an element of the initial algebra arising from the used programming language [12]. Algebraic structures and number systems are widely used in computer science. They allow to abstract from concrete objects which leads to the mathematical branches of abstract algebra and universal algebra. On the other hand, category theory provides possibilities to model many important features of computer science [1, 6] and it affords suitable structures for the describing program construction using algebras $T(C) \to C$ and for modelling observable behavior using coalgebras $C \to T(C)$, where C is a category object and T is a polynomial endofunctor induced by a signature. Algebra and coalgebra are from category's point of view dual constructions [17]. In this contribution we present two ways of expressing the relation of T-algebras and their carrier-sets. We define arrow category of algebras and Kleisli arrow category of algebras. The relation we will formulate with the codomain functors.

2 Basic Concepts

Algebraic and coalgebraic concepts are based on category theory. Roughly speaking it is related to algebra and coalgebra, but far more general [3, 9]. A category $\mathcal{C}$ is mathematical structure consisting of objects, e.g. $A, B, \ldots$ and morphisms of the form $f : A \to B$ between them. Every object has the identity morphism $id_A : A \to A$ and morphisms are composable. Because the objects of category can be arbitrary structures, categories are useful in computer science, where we often use more complex structures not expressible by sets. Morphisms between categories are called functors, e.g. a functor $F : \mathcal{C} \to \mathcal{D}$ from a category $\mathcal{C}$ into a category $\mathcal{D}$ which preserves the structure. In this contribution we use only the category $\mathcal{S}et$ with sets as objects and functions between them as morphisms, but this approach can be extended to categories of arbitrary complex objects.

3 Algebras in category

In our research we are interested in formal description of program construction by algebras and behavior theory of programs. Construction of algebras over the signatures is important approach [9, 14]. We define the endofunctor over the appropriate category for substantiation of the signature operations for a given program. Such a functor we call the polynomial functor.

Algebras over signatures we construct in category. We use basic category $\mathcal{S}et$ of sets and functions. Let T be an endofunctor

$$T : \mathcal{S}et \to \mathcal{S}et \tag{1}$$

Operations in signature determine polynomial endofunctor that can be constructed inductively from T using constants, identities, products, coproducts and powersets. One of the most used categorical forms of algebra is as follows: we define T-algebra as a pair

$$(A, a)$$

T-algebra is a model of signature and its carrier-set is the representation of the type. The algebraic structure (or structuring map) is defined as cotuple function of constructors $cons_1, \ldots, cons_n$:

$$a = [cons_1, \ldots, cons_n] : T(A) \to A$$

The relations between algebras are defined by algebra homomorphisms. Let (A, a) and (B, b) be T-algebras. A homomorphism $f^* : (A, a) \to (B, b)$ of T-algebras is the function $f : A \to B$ between carrier-sets, which commutes with the operations as it is illustrated on the following diagram at the Fig. 1

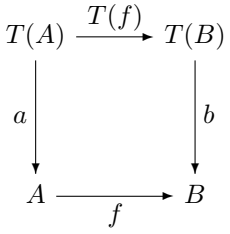


Figure 1: The relation between algebras

so it holds the universal property $f \circ a = b \circ T(f)$.

Homomorphisms of T-algebras can be composed, and every T-algebra (A, a) has the identity homomorphism $id_{(A,a)} : (A, a) \to (A, a)$, therefore we can construct the category **ALG** of T-algebras consisting of T-algebras as objects and homomorphisms between them as category morphisms. The most important concept in algebraic approach is the initial T-algebra [9, 10]. An T-algebra is initial if for arbitrary T-algebra there is unique homomorphism from initial to arbitrary T-algebra. Initial T-algebras, if they exists have some important properties:

- they are unique up to isomorphism, therefore we write initial T-algebra as $u : T(U) \cong U$, and
- the initial algebra has an inverse $u^{-1} : U \to T(U)$

In the other words, from the first property it follows that there exists at most one initial T-algebra. Because from the initial T-algebra exists unique homomorphism to every T-algebra, the initial T-algebra is the initial object in the category **ALG**. The second property was proved in [11] and says that the initial T-algebra is the least fixed point of the functor T. Initial algebras are generalizations of the least fixed points of monotone functions, since they have unique maps into arbitrary T-algebra.

Such formulated category of T-algebras allows us to work with algebras as with unique objects. If we want to formulate the relations between algebras and carrier-sets, we need to define the couple of two adjoint functors $\mathcal{F} \dashv \mathcal{U}$ [5, 15]. The functor $\mathcal{U}$ is forgetful functor which assigns to any algebra from category of sets an appropriate carrier-set from the category of sets. Vice versa, the functor $\mathcal{F}$ has to be generating functor. But there is also another form of representation of algebras in category. By availing of properties of the algebras and some special categories, we enclose algebras in the arrow category.

4 Arrow category for algebras

For simpler handling of algebras in category we define algebras in another form: we will interpret al-

gebraic structure given by the (A, a) as a map

$$TA \xrightarrow{a} A$$

For such expressed algebras we define category of morphisms - the arrow category. For formulation of the relation between algebras and carrier-sets we introduce the codomain functor from arrow category into category of carrier-sets.
Now we define the category of algebras as arrow category. This category of algebras we denote **TALG**. It consists of algebras of the form $TA \xrightarrow{a} A$, $TB \xrightarrow{b} B$, ... as objects and morphisms between objects. Morphisms are algebra homomorphisms of the form (f, a, b), where map f is the function between codomains of the appropriate algebras - the carrier-sets A and B

$$f : cod(a) \to cod(b)$$

as it is depicted at Fig. 2.

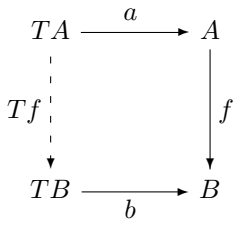


Figure 2: Morphism of algebras

In the category we also define for each algebra $TA \xrightarrow{a} A$ the identity morphism of the form (id_A, a, a) (Fig. 3).

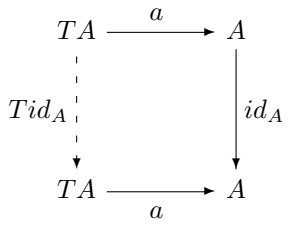


Figure 3: Identity morphism of algebras

It also holds, that morphisms are composable: for (f, a, b) and (g, b, c) we have $(g \circ f, a, c)$ as it is depicted at Fig. 4.

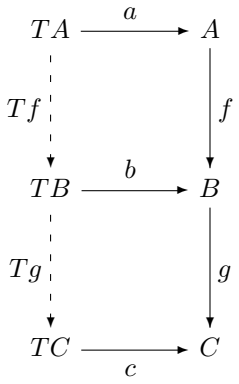


Figure 4: Composition of algebra homomorphism

An initial algebra in the category **TALG** is the initial object of that category. It is the least fixed point of the functor T. The least fixed point of the functor T we denote also as μT. Seeing that it is the T-algebra, there exists operation in defined as

$$in : T(\mu T) \to \mu T$$

The T-algebra $(\mu T, in)$ is the initial T-algebra, if for any T-algebra (A, a) there exists an unique arrow $cata\ a : \mu T \to A$ making the diagram at Fig. 5 commute.

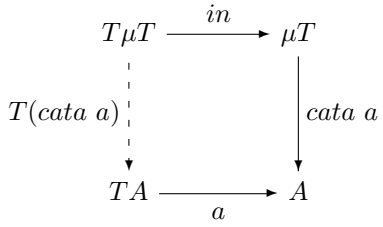


Figure 5: Diagram for initial algebra

satisfying the universal property:

$$cata\ a \circ in = a \circ T(cata\ a)$$

The morphism $cata(-)$ we call the catamorphism. The initial algebra $(\mu T, in)$ is the initial object in the category **TALG** and the catamorphism $cata(-)$ is the mediating arrow out of it. It also holds that the initial algebra exists if T is ω-cocontinuous (i.e. it preserves the colimits of ω-chains) [3]. From the existence of initial algebra

it implies the property called the *reflection*, that it holds

$$id = cata\ in$$

5 Monads

From one point of view, a monad is an abstraction of certain properties of algebraic structures. From another point of view, it is an abstraction of certain properties of adjoint functors. Theory of monads has turned out to be an important tool for studying toposes [5, 15].

5.1 Definition

A *monad*

$$\mathcal{T} = (T, \eta, \mu)$$

on a category $\mathbf{C}$ is an endofunctor

$$T : \mathbf{C} \to \mathbf{C}$$

together with two natural transformations

- $\eta : id_{\mathbf{C}} \to T$ called *unit*
- $\mu : T^2 \to T$ called *multiplication*

subject to the condition that the following diagrams commute.

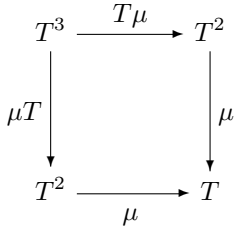


Figure 6: Coherence square for monad

If we consider monad over category $\mathcal{S}et$ of sets, then the unit transformation is a map $\eta_X : X \to TX$ for each set X satisfying a suitable naturality condition; and the multiplication transformation consists of functions $\mu_X : T^2X \to TX$ with X ranging over sets. Next we will handle with the endofunctor T 1 over the category of sets.

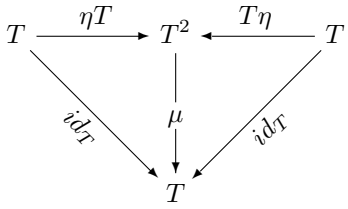


Figure 7: Coherence triangle for monad

Example. The simplest example of monad involves monoids. Let M be a monoid and define $T : \mathcal{S}et \to \mathcal{S}et$ by $TX = M \times X$.

Let $\eta_X : X \to M \times X$ takes x to (id_M, X) and $\mu_X : M \times M \times X \to M \times X$ takes (m, n, x) to (mn, x). Then the associative and unitary identities follow from those on M.

□

The monad structures play a crucial rôle in modeling "branching". Intuitively, the unit η embeds a non-branching behavior as a trivial branching with only one possibility to choose. The multiplication μ "flattens" two successive branching into one branching, abstracting away internal branching [7].

The notion of "algebras for a monad" generalizes classical notions from universal algebra, and in this sense, monads can be thought of as "theories". Every monad is defined by its T-algebras [2]. T-algebras for a monad $\mathcal{T}$ should interact properly with the extra structure on $\mathcal{T}$. A T-algebra is an arrow $a : TA \to A$ as before, such that the diagrams at Fig. 8 and Fig. 9 commute.

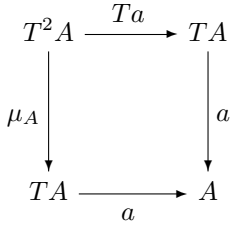


Figure 8: Algebra in monad via multiplication

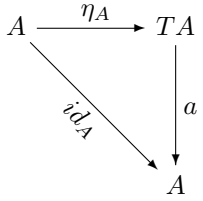


Figure 9: Algebra in monad via unit

5.2 Kleisli categories

Kleisli category is the kind of category which should be investigated for the functional programming paradigm or for the generalizing the structures in category [5, 13]. This category is an extremal solution of the problem of constructing an adjunction that gives rise to a given monad. A monad is a functor from a category to itself consisting of composition of adjoint functors [4]. Especially the dual concept, comonad has useful properties for behavioral theory [15]. Recognizing the categories of coalgebras for a comonad is an important tool of topos theory [16].

The relevance of Kleisli categories in usual coalgebraic approach is that Kleisli category can be thought of as a category where the branching is implicit [7, 13].

Given any monad $\mathcal{T}$, its *Kleisli category* $\mathcal{K}(\mathcal{T})$ is defined as follows. Its objects are the objects of the base category, hence sets in the current setting. An arrow $A \to B$ in $\mathcal{K}(\mathcal{T})$ is the same thing as an arrow $A \to TB$ in the base category, here $\mathcal{S}et$:

$$\frac{A \to B \ \text{ in } \mathcal{K}(\mathcal{T})}{A \to TB \ \text{ in } \mathcal{S}et}$$

Identities and compositions of arrows are defined using the unit and the multiplication of T. Moreover, there is a canonical adjunction

$$\mathcal{F} \dashv \mathcal{U}$$

where functors are:

$$\mathcal{F} : \mathcal{S}et \to \mathcal{K}(\mathcal{T}) \qquad \mathcal{U} : \mathcal{K}(\mathcal{T}) \to \mathcal{S}et$$

In this adjunction the right adjoint $\mathcal{U}$ carries arrow $f : A \to B$ in $\mathcal{K}(\mathcal{T})$ that is a function $f : A \to TB$ in $\mathcal{S}et$ to map

$$TA \xrightarrow{Tf} T^2B \xrightarrow{\mu_B} TB$$

in $\mathcal{S}et$. Moreover, compositions of arrows in category $\mathcal{K}(\mathcal{T})$ are given by

$$A \xrightarrow{f} B \xrightarrow{g} C$$

as the composition

$$A \xrightarrow{f} TB \xrightarrow{Tg} T^2C \xrightarrow{\mu_C} TC$$

in the category $\mathcal{S}et$. $\mu_C \circ Tg$ is the unique lifting of g to the free T-algebra on its domain. The Kleisli category $\mathcal{K}(\mathcal{T})$ is equivalent to the subcategory of **TALG** consisting of the free algebras of the form

$$\mu_A : T^2A \to TA$$

Objects of $\mathcal{K}(\mathcal{T})$ uniquely determine free algebras TA and actions μ_A and Kleisli arrows lift uniquely to arrows $\mu_B \circ Tf$. These are morphisms of free algebras by virtue of the diagram at Fig. 10

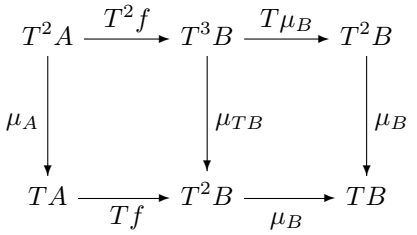


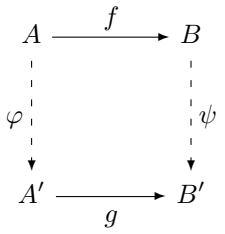
Figure 10: Algebras in Kleisli category

6 Codomain functor and the expression of algebras

Codomain functor is the special functor defined for arrow category. Codomain functor is always defined for the arrow category and the appropriate base category [8]. The arrow category of the base category is the mathematical structure which contents are

- an object of arrow category is an arrow (morphism) of the base category

- given two objects say $A \xrightarrow{f} B$, $A' \xrightarrow{g} B'$ a morphism from T to g consists of an ordered pair (φ, ψ), where $A \xrightarrow{\varphi} A'$, $B \xrightarrow{\psi} B'$ such that the following diagram



  is a commutative diagram. For purpose of this approach we can consider the morphism of an arrow category also in the form (ψ, f, g).

6.1 Codomain functor for the category of algebras TALG

In this approach we define codomain functor between category of algebras **TALG** and category of sets $\mathcal{S}et$ which objects are also the carrier-sets of the algebras. Codomain functor $\mathcal{C}\text{od}$ is defined as

$$\mathcal{C}\text{od} : \mathbf{TALG} \to \mathcal{S}et$$

The functor $\mathcal{C}\text{od}$ sends the object of the category **TALG** - the algebra into the category $\mathcal{S}et$, into the appropriate carrier-set:

$$\mathcal{C}\text{od}(FA \xrightarrow{a} A) = A$$

The functor $\mathcal{C}\text{od}$ according the definition sends the morphism of the category **TALG** - the algebra homomorphism into the appropriate morphism of the category of sets. For objects of **TALG**: $FA \xrightarrow{a} A$, $FB \xrightarrow{b} B$ and $FC \xrightarrow{c} C$ and their morphism (f, a, b), (g, b, c) where f and g are the codomain maps

$$f : cod(a) \to cod(b) \qquad g : cod(b) \to cod(c)$$

it holds

$$\mathcal{C}\text{od}(f, a, b) = f$$

and for the identity morphism

$$\mathcal{C}\text{od}(id, a, a) = id$$

which satisfy the definition of the codomain functor. Functor $\mathcal{C}\text{od}$ also preserves the composition of the morphisms:

$$\mathcal{C}\text{od}(g \circ f, a, c) = g \circ f$$

The codomain functor from arrow category into appropriate base category always exists. It also doesn't need to define extra adjoint functors to formulate the relation between algebras and the carrier-sets.

6.2 Codomain functor for the Kleisli category

We defined the Kleisli category $\mathcal{K}(\mathcal{T})$ of a monad $\mathcal{T} = (T, \mu, \eta)$. Now we construct the arrow category over $\mathcal{K}(\mathcal{T})$ denoted $\mathcal{K}(\mathcal{T})^{\to}$ as follows:

- objects are algebras of the form

$$\mu_A : T^2A \to TA$$

- morphisms are algebra homomorphisms of the form (Tf, μ_A, μ_B) such that the following diagram at Fig. 11 commutes

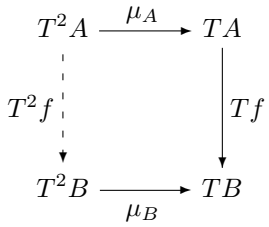


Figure 11: Morphism of algebras in category $\mathcal{K}(\mathcal{T})$

- identity has the form (Tid_A, μ_A, μ_A)
- compostion of two algebras (Tf, μ_A, μ_B) and (Tg, μ_B, μ_C) is $(Tg \circ Tf, \mu_A, \mu_C)$

Next we define the codomain functor $\mathcal{K}\text{od}$ for the Kleisli arrow category. The functor has the form

$$\mathcal{K}\text{od} : \mathcal{K}(\mathcal{T})^{\to} \to \mathcal{S}et$$

Codomain functor $\mathcal{K}\text{od}$ sends the objects of Kleisli arrow category of algebras into the category

of carrier-sets $\mathcal{S}et$ such that it assigns to any algebra $\mu_A : T^2A \to TA$ the appropriate carrier-set:

$$\mathcal{K}\mathrm{od}\left(T^2A \xrightarrow{\mu_A} TA\right) = TA$$

Functor $\mathcal{K}$od according to definition maps the morphisms of category $\mathcal{K}(\mathcal{T})^{\to}$ (the algebra homomorphisms) into the appropriate morphisms of the category of carrier-sets. Let's have the algebras $\mu_A : T^2A \to TA$, $\mu_B : T^2B \to TB$, $\mu_C : T^2C \to TC$ with their morphisms (Tf, μ_A, μ_B) and (Tg, μ_B, μ_C), where Tf and Tg are the codomain maps

$$Tf : cod(\mu_A) \to cod(\mu_B)$$

$$Tg : cod(\mu_B) \to cod(\mu_C)$$

For the algebra homomorphisms according to the definition of Kleisli category it holds that

$$\mathcal{K}\mathrm{od}\,(Tf, \mu_A, \mu_B) = \mu_B \circ Tf$$

For identity homomorphisms it holds that

$$\mathcal{K}\mathrm{od}\,(Tid_A, \mu_A, \mu_A) = \mu_A \circ \eta_A$$

Codomain functor $\mathcal{K}$od also preserves the composition of algebra homomorphisms:

$$\mathcal{K}\mathrm{od}\,(Tg \circ Tf, \mu_A, \mu_C) = (\mu_C \circ Tg) \circ (\mu_B \circ Tf)$$

As in the previous case, the codomain functor from Kleisli arrow category into the appropriate base category always exists. It also doesn't need to define extra adjoint functors to formulate the relation between algebras and the carrier-sets.

7 Conclusion

In this contribution we formulated the expression of algebras in the arrow category. The relation between algebras and their carrier-sets we constructed as codomain functor from the arrow category **TALG** into the base category $\mathcal{S}et$ of sets. We also formulated another approach of expressing algebras in Kleisli arrow category. We defined that relation between algebras and their carrier-sets as codomain functor from $\mathcal{K}(\mathcal{T})^{\to}$ into the base category of sets. The codomain functor for the arrow category is always defined, that's why our approach does not need to construct the couple of adjoint functors and to prove the construction. In our next research we will focus on suitable categorical structures as a base for algebraic description of construction and coalgebraic behavior of program systems. We would like to apply achieved theoretical results to real non trivial program systems from the area of computer networks, database systems and distributed systems.

8 Acknowledgments

This work was supported by VEGA Grant No.1/0175/08: Behavioral categorical models for complex program systems.

References

[1] Awodey, S. *Category Theory.* Carnegie Mellon University, 2005.

[2] Banda, A. Algebras over monads, 2007. African Institute for Mathematical Sciences, University of Cape Town.

[3] Barr, M. Terminal coalgebras for endofunctors on sets, 1999.

[4] Barr, M., and Wells, C. *Category Theory for Computing Science.* Prentice Hall International, 1990. ISBN 0-13-120486-6.

[5] Barr, M., and Wells, C. *Toposes, Triples and Theories.* Springer-Verlag, 2002.

[6] Ehrig, H. Applied and computational category theory. In *Bulletin of the EATCS no 89* (June 2006), European Association for Theoretical Computer Science, pp. 134–135.

[7] Hasuo, I. *Tracing Anonymity with Coalgebras.* PhD thesis, Radboud University Nijmegen, 2008.

[8] Jacobs, B. *Categorical Logic and Type Theory.* No. 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.

[9] Jacobs, B., and Rutten, J. A tutorial on (co)algebras and (co)induction. *Bulletin of the European Association for Theoretical Computer Science*, No. 62 (1997), 222–259.

[10] J.Adámek, Rosický, J., and E.Vitale. Algebraic theories: a categorical introduction to general algebra. With a Preface by F.W. Lawvere, April 2008.

[11] Lambek, J., and Scott, P.-J. Introduction to higher-order categorical logic. *Studies in Adv. Math, Cambridge University Press*, No. 7 (1986).

[12] Nivela, M. P., and Orejas, F. Initial behavior semantics for algebraic specifications. In *Lecture notes in Computer Science on Recent trends in data type specification* (New York, NY, USA, 1987), Springer-Verlag New York, Inc., pp. 184–207.

[13] Novitzká, V., Hužvár, R., Mihályi, D., Slodičák, V., Szabó, C., and Verbová, A. *Categorical models for behavioural description of program systems.* elfa s.r.o., 2008, pp. 7–12.

[14] Novitzká, V., Mihályi, D., and Verbová, A. Coalgebras as models of systems behaviour. In *International Conference on Applied Electrical Engineering and Informatics, Greece, Athens* (2008), pp. 31–36.

[15] Slodičák, V. *The Rôle of Toposes in the Informatics.* PhD thesis, Technical University of Košice, Slovakia, 2008. (in slovak).

[16] Slodičák, V., and Verbová, A. Some aspects about extending toposes. In *Proceedings from 8th Scientific Conference of Young Researchers - SCYR 2008* (2008), elfa, s.r.o., pp. 149–151. ISBN 978-80-553-0036-8.

[17] Wisbauer, R. *Algebras versus coalgebras.* University of Düsseldorf, Germany, 2007.