Graph Operations and Free Graph Algebras

Uwe Wolter[1(✉)], Zinovy Diskin[2], and Harald König[3]

[1] University of Bergen, Bergen, Norway
Uwe.Wolter@uib.no
[2] McMaster University, Hamilton, Canada
diskinz@mcmaster.ca
[3] FHDW Hannover, Hannover, Germany
Harald.Koenig@fhdw.de

Abstract. We introduce a concept of *graph algebra* that generalizes the traditional concept of algebra in the sense that (1) we use graphs rather than sets as carriers, and (2) we generalize algebraic operations to diagrammatic operations over graphs, which we call *graph operations*.

Our main objective is to extend the construction of term algebras, i.e., free algebras, for the new setting. The key mechanism for the construction of free graph algebras are pushout-based graph transformations for non-deleting injective rules. The application of rules, however, has to be controlled in such a way that "no confusion" arises. For this, we introduce *graph terms* and present a concrete construction of free graph algebras as graph term algebras.

As the main result of the paper, we obtain for any graph signature Γ an adjunction between the category Graph of graphs and the category $\mathsf{GAlg}(\Gamma)$ of graph Γ-algebras. In such a way, we establish an "integrating link" between the two areas Hartmut Ehrig contributed most: algebraic specifications with initial/free semantics and pushout-based graph transformations.

Keywords: Universal Algebra · Term · Term algebra
Free algebra · Graph operation · Graph algebra · Graph term
Graph term algebra · Free graph algebra · Kleisli morphism

1 Introduction

Graph operations have been a key ingredient of the generalized sketch framework, developed in the 90s by a group around the second author and motivated by applications in databases and data modeling [1,3,5]. What was missing, until now, is a proper formal substantiation of the "Kleisli mapping" construct heavily employed in those papers. When we re-launched, ten years later, generalized sketches under the name Diagrammatic Predicate Framework (DPF) [6,12–14], we dropped operations due to the lack of a proper formalization appropriate for our applications in Model Driven Software Engineering. Finally, during his period in Hartmut's group in 1991-00, the first author had always been wondering if one

R. Heckel and G. Taentzer (Eds.): Ehrig Festschrift, LNCS 10800, pp. 313–331, 2018.
https://doi.org/10.1007/978-3-319-75396-6_17

should look for a uniform mechanism to create names for new items produced by injective graph transformation rules via pushouts.

In the paper, we present a concept of graph algebra that generalizes the traditional concept of algebra in the sense that (1) we use graphs as carriers, instead of sets, and (2) we generalize algebraic operations to graph operations. We introduce graph terms and present a concrete construction of free graph algebras as graph term algebras. As a side effect, graph terms provide a uniform mechanism for the new names problem mentioned above.

As the main result of the paper we obtain for any graph signature Γ an adjunction between the category Graph of graphs and the category $\mathsf{GAlg}(\Gamma)$ of graph Γ-algebras. These adjunctions generalize the traditional adjunctions between the category Set and categories $\mathsf{Alg}(\Sigma)$ of Σ-algebras. The Kleisli categories of the new adjunctions provide the necessary substantiation of the idea of "Kleisli morphisms" of the second author, we have been looking for.

As a pleasant surprise, we realized that the new setting of graph algebras establishes an "integrating link" between the two areas Hartmut Ehrig contributed most - algebraic specifications with initial semantics and graph transformations.

To keep technicalities simple, and to meet the space limitations, we only consider unsorted/untyped signatures and algebras, and leave the straightforward generalization for the many-sorted/typed case for future work. As a running example for a graph signature Γ, we have chosen Γ consisting of arrow composition, identity, initial object, and pullback, which hopefully most of the readers are familiar with.

The paper is organized as follows. In Sect. 2 we recapitulate the basic algebraic concepts signature, operation, algebra, variable, term and term algebra, and we discuss the characterization of term algebras as free algebras. In Sect. 3 we analyze algebraic operations and "diagrammatic" operations, like composition and pullbacks, in the light of graphs, and develop the new concepts graph signature, graph operation and graph algebra. We define corresponding categories $\mathsf{GAlg}(\Gamma)$ of graph algebras for given graph signatures Γ. In Sect. 4, we analyse the construction of terms in the light of graph transformations, and develop the new concepts of a graph term and a graph term algebra. We show that graph term algebras are free graph algebras and discuss applications of this main result. Finally, we discuss related work in Sect. 5 and Sect. 6 outlines different dimensions of further research.

2 Background: Algebras and Term Algebras

An *(algebraic) signature* $\Sigma = (F, ar)$ is given by a finite set F of *operation symbols* and an *arity function* $ar : F \to \mathbb{N}$. A *Σ-algebra* $\mathcal{A} = (A, F^{\mathcal{A}})$ is provided by a *(carrier) set* A, also denoted by $|\mathcal{A}|$, and a family $F^{\mathcal{A}} = (\omega^{\mathcal{A}} : A^{ar(\omega)} \to A \mid \omega \in F)$ of operations. For $n \in \mathbb{N}$ we denote by A^n the set of all n-tuples $\bar{a} = (a_1, \ldots, a_n)$ of elements in A. For $n = 0$ we obtain, in such a way, the singleton set $A^0 = \{()\}$ containing the empty tuple. A symbol $c \in F$ with $ar(c) = 0$ is

also called a *constant symbol*. The corresponding operation $c^{\mathcal{A}} : A^0 \to A$ in a Σ-algebra $\mathcal{A}$ is a "pointer" with the only element () in A^0 pointing to the element $c^{\mathcal{A}}()$ in A.

A Σ-algebra $\mathcal{A}$ is a subalgebra of a Σ-algebra $\mathcal{B}$ if, and only if, $A \subseteq B$ and $\omega^{\mathcal{A}}(\bar{a}) = \omega^{\mathcal{B}}(\bar{a})$ for all $\omega \in F$ and all $\bar{a} \in A^{ar(\omega)} \subseteq B^{ar(\omega)}$. This means that the subset A of the carrier of $\mathcal{B}$ has to be closed under the operations in $\mathcal{B}$. Specifically, A has to contain all the constants $c^{\mathcal{B}}()$ from $\mathcal{B}$.

A *Σ-homomorphism* $f : \mathcal{A} \to \mathcal{B}$ between two Σ-algebras $\mathcal{A}$ and $\mathcal{B}$ is a map $f : A \to B$ such that for every $\omega \in F$, $ar(\omega) = n$ we have $f \circ \omega^{\mathcal{A}} = \omega^{\mathcal{B}} \circ f^n$ where the n'th power $f^n : A^n \to B^n$ of the map f is defined by $f^n(\bar{a}) = (f(a_1), \ldots, f(a_n))$ for all $\bar{a} = (a_1, \ldots, a_n) \in A^n$. That is, for each $\omega \in F$, $ar(\omega) = n$ we require

$$f(\omega^{\mathcal{A}}(a_1, \ldots, a_n)) = \omega^{\mathcal{B}}(f(a_1), \ldots, f(a_n)) \text{ for all } (a_1, \ldots, a_n) \in A^n. \quad (1)$$

$f^0 : A^0 \to B^0$ is the identity on $\{()\}$. Note that requirement (1) for constants $c \in F$ means that constants are mapped to constants: $f(c^{\mathcal{A}}()) = c^{\mathcal{B}}(f^0()) = c^{\mathcal{B}}()$.

The composition $g \circ f : \mathcal{A} \to \mathcal{C}$ of two Σ-homomorphisms $f : \mathcal{A} \to \mathcal{B}$ and $g : \mathcal{B} \to \mathcal{C}$ is given by the composition $g \circ f : A \to C$ of the underlying maps $f : A \to B$ and $g : B \to C$. In such a way, Σ-algebras and Σ-homomorphisms constitute a category $\mathsf{Alg}(\Sigma)$, and the assignments $\mathcal{A} \mapsto |\mathcal{A}|$ and $(f : \mathcal{A} \to \mathcal{B}) \mapsto (f : |\mathcal{A}| \to |\mathcal{B}|)$ define a forgetful functor $|\_| : \mathsf{Alg}(\Sigma) \to \mathsf{Set}$.

Example 1 (Natural numbers). We consider the signature $\Sigma = (F, ar)$ with $F = \{\mathsf{z}, \mathsf{s}, \mathsf{p}\}$ consisting of a constant symbol z, $ar(\mathsf{z}) = 0$, a unary operation symbol s, $ar(\mathsf{s}) = 1$, and a binary operation symbol p, $ar(\mathsf{p}) = 2$. As sample Σ-algebra $\mathcal{N} = (\mathbb{N}, F^{\mathcal{N}})$ we consider the natural numbers with a zero, a successor, and a plus operation: $\mathsf{z}^{\mathcal{N}}() = 0$, $\mathsf{s}^{\mathcal{N}} = \_ + 1 : \mathbb{N} \to \mathbb{N}$, $\mathsf{p}^{\mathcal{N}} = \_ + \_ : \mathbb{N}^2 \to \mathbb{N}$.

Let be given an algebraic signature Σ and a set X of variables. Σ-terms on X are strings build of three kinds of symbols: operation symbols from F, variables from X and three auxiliary symbols ",", "$\langle$", "$\rangle$". The inductive definition of terms goes traditionally as follows (compare [8], p. 18):

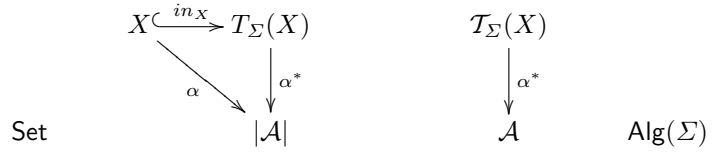
Definition 1 (Terms). *The set $T_\Sigma(X)$ of all Σ-terms on a set X of variables is the smallest set of strings of symbols such that:*
(Variables). $X \subseteq T_\Sigma(X)$,
(Constants). $c\langle\rangle' \in T_\Sigma(X)$ *for all* $c \in F$ *with* $ar(c) = 0$,
(Operations). $\omega\langle t_1, \ldots, t_n\rangle \in T_\Sigma(X)$ *for all operation symbols* $\omega \in F$ *with* $ar(\omega) = n \geq 1$ *and all Σ-terms* $t_1, \ldots, t_n \in T_\Sigma(X)$.

A simple, but crucial, observation is, that the generation of terms can be interpreted as operations in special Σ-algebras (compare [8], p. 67):

Definition 2 (Term algebra). *For a given set X of variables we denote by $\mathcal{T}_\Sigma(X) = (T_\Sigma(X), F^{\mathcal{T}_\Sigma(X)})$ the Σ-algebra of Σ-terms on X with:*
(Constants). $c^{\mathcal{T}_\Sigma(X)}() = c\langle\rangle \in T_\Sigma(X)$ *for all* $c \in F$ *with* $ar(c) = 0$,
(Operations). $\omega^{\mathcal{T}_\Sigma(X)}(t_1, \ldots, t_n) = \omega\langle t_1, \ldots, t_n\rangle \in T_\Sigma(X)$ *for all operation symbols* $\omega \in F$ *with* $ar(\omega) = n \geq 1$ *and all n-tuples* $(t_1, \ldots, t_n) \in T_\Sigma(X)^n$.

That any term is generated in a unique way, is abstractly reflected by the characterization of term algebras as free algebras (compare [8], p. 68):

Proposition 1 (Free algebras). *For each set X of variables the Σ-algebra $\mathcal{T}_\Sigma(X) = (T_\Sigma(X), F^{\mathcal{T}_\Sigma(X)})$ has the following universal property: For any Σ-algebra $\mathcal{A}$ and any variable assignment $\alpha : X \to |\mathcal{A}|$ there exists a unique Σ-homomorphism $\alpha^* : \mathcal{T}_\Sigma(X) \to \mathcal{A}$ such that: $\alpha^* \circ in_X = \alpha$.*



Proposition 1 can be shown by structural induction according to the inductive definition of terms in Definition 1: For the basic case of variables the defining condition forces $\alpha^*(x) = \alpha(x)$ for all $x \in X$. For the basic case of constant symbols the definition of operations in $\mathcal{T}_\Sigma(X)$ and the homomorphism condition entail $\alpha^*(c\langle\rangle) = \alpha^*(c^{\mathcal{T}_\Sigma(X)}()) = c^{\mathcal{A}}()$ for all $c \in F$ with $ar(c) = 0$. And, for the induction step the definition of operations in $\mathcal{T}_\Sigma(X)$ and the homomorphism condition provide the necessary induction/recursion scheme

$$\alpha^*(\omega\langle t_1, \ldots, t_n\rangle) = \alpha^*(\omega^{\mathcal{T}_\Sigma(X)}(t_1, \ldots, t_n)) = \omega^{\mathcal{A}}(\alpha^*(t_1), \ldots, \alpha^*(t_n)) \quad (2)$$

for all operation symbols $\omega \in F$ with $ar(\omega) = n \geq 1$ and all $t_1, \ldots, t_n \in T_\Sigma(X)$.

The universal property determines $\mathcal{T}_\Sigma(X)$ up to isomorphism in $\mathsf{Alg}(\Sigma)$. A Σ-algebra $\mathcal{A}$ is isomorphic to $\mathcal{T}_\Sigma(X)$ iff the following conditions are satisfied:

1. **Generators:** There is an injective mapping $em : X \to |\mathcal{A}|$.
2. **No confusion:** $em(x) \neq \omega^{\mathcal{A}}(\bar{a})$ for any $x \in X$, any operation symbol $\omega \in F$ and any tuple $\bar{a} \in A^{ar(\omega)}$. For any operation symbols $\omega_1, \omega_2 \in F$ and any tuples $\bar{a}_1 \in A^{ar(\omega_1)}$, $\bar{a}_2 \in A^{ar(\omega_2)}$ we have

$$\omega_1^{\mathcal{A}}(\bar{a}_1) \neq \omega_2^{\mathcal{A}}(\bar{a}_2) \quad \text{iff} \quad \omega_1 \neq \omega_2 \text{ or } \bar{a}_1 \neq \bar{a}_2.$$

3. **No junk:** $\mathcal{A}$ has no proper Σ-subalgebra containing $em(X)$.

As any free construction [10], the universal property in Proposition 1 ensures that we can extend the assignments $X \mapsto \mathcal{T}_\Sigma(X)$ to a functor $\mathcal{T}_\Sigma(\_) : \mathsf{Set} \to \mathsf{Alg}(\Sigma)$ that is left-adjoint to the forgetful functor $|\_| : \mathsf{Alg}(\Sigma) \to \mathsf{Set}$. The adjunction

$$\mathsf{Set} \underset{|\_|}{\overset{\mathcal{T}_\Sigma(\_)}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathsf{Alg}(\Sigma)$$

is the fundament for the area of algebraic specifications as the two volumes [8,9] exemplify. Just to mention, that any variant of equational and/or first-order specifications is syntactically based on terms while the semantics relies on the

uniqueness of the evaluation of terms w.r.t. variable assignments. And, not to forget, the Kleisli category of this adjunction provides us a substitution calculus: A substitution of terms for variables is a morphism in the Kleisli category, i.e., a map $\sigma : X \to T_\Sigma(Y)$. The corresponding extended map $\sigma^* : T_\Sigma(X) \to T_\Sigma(Y)$ describes the simultaneous substitution of all variables x in Σ-terms on X by the corresponding terms $\sigma(x) \in T_\Sigma(Y)$. The composition of two substitutions $\sigma : X \to T_\Sigma(Y)$ and $\delta : Y \to T_\Sigma(Z)$ is given by $\delta^* \circ \sigma : X \to T_\Sigma(Z)$.

3 From Algebras to Graph Algebras

As graphs we consider "directed multigraphs" [7]. A graph $G = (G_V, G_E, \mathsf{sr}^G, \mathsf{tg}^G)$ consists of a set G_V of vertices, a set G_E of edges, and two maps $\mathsf{sr}^G, \mathsf{tg}^G : G_E \to G_V$. A homomorphism $\varphi = (\varphi_V, \varphi_E)$ between two graphs $G = (G_V, G_E, \mathsf{sr}^G, \mathsf{tg}^G)$ and $H = (H_V, H_E, \mathsf{sr}^H, \mathsf{tg}^H)$ consists of two maps $\varphi_V : H_V \to G_V$ and $\varphi_E : H_E \to G_E$ such that $\varphi_V \circ \mathsf{sr}^G = \mathsf{sr}^H \circ \varphi_E$ and $\varphi_V \circ \mathsf{tg}^G = \mathsf{tg}^H \circ \varphi_E$.

The identity graph homomorphism on a graph G is the pair $id_G = (id_{G_V}, id_{G_E})$ of identity maps and graph homomorphisms are composed componentwise. By Graph we denote the category with graphs as objects and graph homomorphisms as morphisms. To establish the concept of graph algebras, we need, first, an adequate concept of signature.

Definition 3 (Graph signature). *A* graph signature $\Gamma = (OP, I, R)$ *is given by a finite set OP of operation symbols and two maps I and R assigning to each operation symbol $\omega \in OP$ a finite graph $I(\omega)$, its* input arity*, and a finite graph $R(\omega)$, its* result arity*, respectively. Moreover, we assume that there is an inclusion $\iota_\omega : I(\omega) \hookrightarrow R(\omega)$ between the two arity graphs.*

To substantiate this definition, we discuss, in more detail, the transition from algebraic signatures to graph signatures.

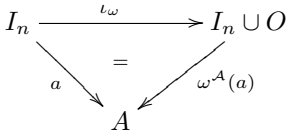
Let $I_n = \{in_1, \ldots, in_n\}$ be a set of indices. We can consider an n-tuple $\bar{a} = (a_1, \ldots, a_n)$ of elements from a set A as a representation of a map $a : I_n \to A$ where $a_i = a(in_i)$ for all $1 \leq i \leq n$. The empty tuple () represents, in this view, the unique map from the empty set $I_0 = \emptyset$ into A.

For any $n \in \mathbb{N}$ there is a bijection between A^n and the set A^{I_n} of all maps from I_n into A, thus we can consider maps $a \in A^{I_n}$ as inputs for operations in a Σ-algebra $\mathcal{A}$. What about the output? Algebraic operations give only a single value as output thus we can consider the codomain of an operation in $\mathcal{A}$ as the set A^O of all maps from a singleton $O = \{out\}$ into A. From this viewpoint, we can consider the declaration of an operation symbol ω with $ar(\omega) = n$ as declaring a span $I_n \hookleftarrow \emptyset \hookrightarrow O$ of set inclusions, where the corresponding operation in a Σ-algebra $\mathcal{A}$ would be a map from A^{I_n} into A^O.

Operations are assumed, however, to have no side effects. This means that the input is neither deleted nor changed. In such a way, we can consider the declaration of the arity of an operation symbol ω as declaring a set inclusion

$\iota_\omega : I_n \hookrightarrow I_n \cup O$ (obtained by pushing out the above span of inclusions). The corresponding operation in $\mathcal{A}$ becomes then a map

$$\omega^{\mathcal{A}} : A^{I_n} \longrightarrow A^{I_n \cup O} \quad \text{such that } a = \omega^{\mathcal{A}}(a) \circ \iota_\omega \text{ for all } a \in A^{I_n}. \tag{3}$$

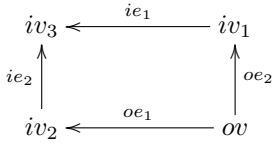


We can recognize the same pattern in "graph operations", like composition of morphisms and limit/colimit constructions in categories, for example. There are, however, three essential differences to the case of algebraic operations:

1. There are two different kinds of input items, namely vertices and edges.
2. Operations can produce arbitrary many output items instead of exactly one.
3. To relate output edges in an appropriate way to the input, we have to work with non-empty "boundary graphs" instead of just the empty set.

As an example we consider the construction of pullbacks. Let a category C with pullbacks be given and let $|C|$ denote the underlying graph of C. To turn the existence of pullbacks into an operation, we have to choose for any cospan $A \xrightarrow{f} C \xleftarrow{g} B$ in C one of the existing pullbacks, i.e., we have to choose an object D and morphisms $g^* : D \to A$, $f^* : D \to B$ such that the resulting square is a pullback in C.

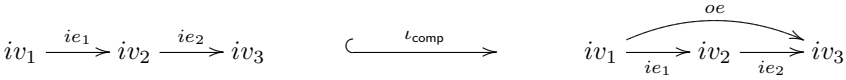
The input arity of a corresponding operation symbol pb can be described by the graph $\mathsf{Cospan} = (iv_1 \xrightarrow{ie_1} iv_3 \xleftarrow{ie_2} iv_2)$, i.e., a cospan in C is a graph homomorphism from Cospan into $|C|$. Here, "iv" stands for *input vertex* while "ie" refers to *input edge*. We will often use the term *binding* for these graph homomorphisms. The output arity of the operation could be described by the graph $\mathsf{Span} = (iv_1 \xleftarrow{oe_2} ov \xrightarrow{oe_1} iv_2)$, where "$ov$" stands for *output vertex* while "oe" refers to *output edge*. The "boundary graph" $\underline{2}$, consisting of the two vertices iv_1 and iv_2, connects the output items with the input items. Instead of a span of graph inclusions $\mathsf{Cospan} \hookleftarrow \underline{2} \hookrightarrow \mathsf{Span}$ we consider, however, the inclusion ι_{pb} of the graph Cospan into the graph Square,



obtained by pushing out the above span of graph inclusions, as the declaration of the arity of the operation symbol pb.

Convention 4 (Graph signature). *For notational convenience we use canonical names for input vertices and edges. For a given operation symbol $\omega \in OP$, we denote the elements of $I(\omega)_V$ by $\{iv_1, \ldots, iv_{nv_\omega}\}$ and the elements of $I(\omega)_E$ by $\{ie_1, \ldots, ie_{ne_\omega}\}$, where nv_ω and ne_ω are the numbers of vertices and edges in $I(\omega)$ resp. Output vertices and edges will be denoted by ov_i and oe_j.*

Example 2 (Graph signature). We consider a graph signature $\Gamma = (OP, I, R)$ with $OP = \{\mathsf{pb}, \mathsf{comp}, \mathsf{id}, \mathsf{ini}\}$. For the operation symbol pb we declare $I(\mathsf{pb}) = \mathsf{Cospan}$ and $R(\mathsf{pb}) = \mathsf{Square}$. The arity of the composition operation symbol comp is given by the following inclusion of graphs

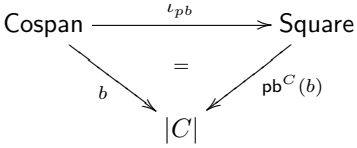


The input arity of id is the graph $\underline{1}$ with exactly one vertex iv and the result arity is the graph Loop with exactly one vertex iv and one edge oe. Finally, the input arity of ini is the *empty graph* $\underline{\emptyset}$, and the result arity is a graph $\underline{1}$ with exactly one vertex ov. That is, ini is a constant symbol with a trivial result arity, but in general the result arity of a constant could be any finite graph!

For graphs G and H we denote by G^H the set of all graph homomorphisms from H into G. A fixed choice of pullbacks in category C gives rise to a map

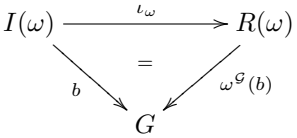
$$\mathsf{pb}^C : |C|^{\mathsf{Cospan}} \longrightarrow |C|^{\mathsf{Square}}$$

such that $b = pb^C(b) \circ \iota_{pb}$ for all bindings $b : \mathsf{Cospan} \to |C|$.



Generalizing the pullback example, we coin now the new concept of graph algebra.

Definition 5 (Graph algebra). *For a graph signature $\Gamma = (OP, I, R)$ a Γ-*algebra $\mathcal{G} = (G, OP^{\mathcal{G}})$ *is given by a* (carrier) graph $G = (G_V, G_E, \mathsf{sr}^G, \mathsf{tg}^G)$, *also denoted by $|\mathcal{G}|$, and a family $OP^{\mathcal{G}} = (\omega^{\mathcal{G}} : G^{I(\omega)} \to G^{R(\omega)} \mid \omega \in OP)$ of maps such that $b = \omega^{\mathcal{G}}(b) \circ \iota_\omega$ for all $\omega \in OP$ and all $b \in G^{I(\omega)}$. These maps will be called* graph operations.



For a vertex $v \in R(\omega)_V$ and edge $e \in R(\omega)_E$, we write $\omega^{\mathcal{G}}_V(b)(v)$ and $\omega^{\mathcal{G}}_E(b)(e)$ rather than $\omega^{\mathcal{G}}(b)_V(v)$ and $\omega^{\mathcal{G}}(b)_E(e)$. This eases reading formulas with b defined by long tuples. We will also omit V, E subindices if it eases reading formulas.

For any graph G there is exactly one graph homomorphism $\underline{\emptyset}_G : \underline{\emptyset} \to G$, i.e., $G^{\underline{\emptyset}} = \{\underline{\emptyset}_G\}$ is a singleton, thus for any constant symbol $c \in OP$, i.e., $I(c) = \underline{\emptyset}$, the corresponding graph operation in a graph Γ-algebra $\mathcal{G}$ just points at a subgraph of G, namely the image of $R(c)$ w.r.t. $c^{\mathcal{G}}(\underline{\emptyset}_G)$.

Example 3 (Graph algebra). The composition in a category C can be presented as a graph operation $\mathsf{comp}^C : |C|^{I(\mathsf{comp})} \to |C|^{R(\mathsf{comp})}$ where the only output of the graph operation is given by $\mathsf{comp}^C_E(b)(oe) = b(ie_2) \circ b(ie_1)$ for all bindings $b : I(\mathsf{comp}) \to |C|$. Note, that comp^C is a total operation. The identity in C gives another graph operation $\mathsf{id}^C : |C|^{\underline{1}} \to |C|^{\mathsf{Loop}}$ such that $\mathsf{id}^C_E(b)(oe) = id_{b(iv)}$.

If C has pullbacks, we can define a graph operation $\mathsf{pb}^C : |C|^{\mathsf{Cospan}} \to |C|^{\mathsf{Square}}$, in such a way, that for any binding $b : \mathsf{Cospan} \to |C|$ the result $\mathsf{pb}^C(b) : \mathsf{Square} \to |C|$ is a chosen pullback diagram. And, if C has initial objects, we can define a constant $\mathsf{ini}^C : |C|^{\underline{\emptyset}} \to |C|^{\underline{1}}$, in such a way, that $\mathsf{ini}^C_V(\underline{\emptyset}_G)(ov)$ is a (chosen) initial object in C.

A Γ-algebra $\mathcal{G}$ is a subalgebra of a Γ-algebra $\mathcal{H}$ if G is a subgraph of H and $in \circ \omega^{\mathcal{G}}(b) = \omega^{\mathcal{H}}(in \circ b)$ for all $\omega \in OP$ and all $b \in G^{I(\omega)}$, where $in : G \to H$ is the corresponding inclusion graph homomorphism (compare Definition 6). This means that the subgraph G of the carrier of $\mathcal{H}$ has to be closed under the operations in $\mathcal{H}$ in the sense that for all $\omega \in OP$ and all $b \in G^{I(\omega)}$ the image $\omega^{\mathcal{H}}(in \circ b)(R(\omega))$ is a subgraph of G. Especially, G has to contain the image graph $c^{\mathcal{H}}(\underline{\emptyset}_H)(R(c))$ for any constant symbol c in OP.

A functor $\mathfrak{F} : C \to D$ between two categories C and D is a graph homomorphism $\mathfrak{F} : |C| \to |D|$ compatible with composition, i.e., for all morphisms $f : A \to B$, $g : B \to C$ in C we have $\mathfrak{F}(g \circ f) = \mathfrak{F}(g) \circ \mathfrak{F}(f)$. We can reformulate this condition in terms of the corresponding graph operations $comp^C$ and $comp^D$ by requiring that the following diagram commutes:

$$\begin{array}{ccc} |C|^{I(\mathsf{comp})} & \xrightarrow{\mathsf{comp}^C} & |C|^{R(\mathsf{comp})} \\ {\scriptstyle \mathfrak{F}\circ\_}\downarrow & = & \downarrow{\scriptstyle \mathfrak{F}\circ\_} \\ |D|^{I(\mathsf{comp})} & \xrightarrow{\mathsf{comp}^D} & |D|^{R(\mathsf{comp})} \end{array}$$

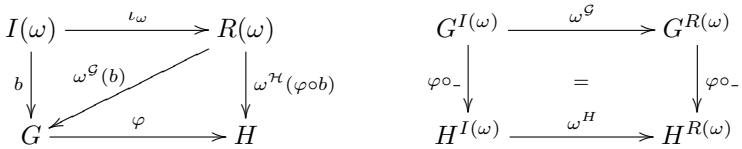
That is, for any binding $b : I(\mathsf{comp}) \to |C|$ we require (compare (1))

$$\mathfrak{F} \circ \mathsf{comp}^C(b) = \mathsf{comp}^D(\mathfrak{F} \circ b).$$

This example motivates our concept of homomorphisms between graph algebras.

Definition 6. *A* Γ-homomorphism $\varphi : \mathcal{G} \to \mathcal{H}$ *between two* Γ*-algebras* $\mathcal{G} = (G, OP^{\mathcal{G}})$ *and* $\mathcal{H} = (H, OP^{\mathcal{H}})$ *is a graph homomorphism* $\varphi : G \to H$ *such that*

$$\varphi \circ \omega^{\mathcal{G}}(b) = \omega^{\mathcal{H}}(\varphi \circ b) \quad \textit{for all } \omega \in OP \textit{ and all } b \in G^{I(\omega)}. \tag{4}$$



Example 4. Given two categories C and D with pullback operations, a functor $\mathfrak{F} : C \to D$, that preserves pullbacks, establishes a graph algebra homomorphism only if it maps chosen pullbacks in C to chosen pullbacks in D.

The composition $\psi \circ \varphi : \mathcal{G} \to \mathcal{K}$ of Γ-homomorphisms $\varphi : \mathcal{G} \to \mathcal{H}$ and $\psi : \mathcal{H} \to \mathcal{K}$ is given by the composition $\psi \circ \varphi : G \to K$ of the underlying graph homomorphisms $\varphi : G \to H$ and $\psi : H \to K$. For any graph Γ-algebra $\mathcal{G}$ the identity Γ-homomorphism $id_{\mathcal{G}} : \mathcal{G} \to \mathcal{G}$ is given by the identity graph homomorphism $id_G : G \to G$. In such a way, Γ-algebras and Γ-homomorphisms constitute a category $\mathsf{GAlg}(\Gamma)$ where the assignments $\mathcal{G} \mapsto |\mathcal{G}|$ and $(\varphi : \mathcal{G} \to \mathcal{H}) \mapsto (\varphi : |\mathcal{G}| \to |\mathcal{H}|)$ define a forgetful functor $|\_| : \mathsf{GAlg}(\Gamma) \to \mathsf{Graph}$.

We conclude this section with a discussion how algebras can be transformed into corresponding graph algebras. Any set A can be transformed into a graph $\mathfrak{V}(A)$ with an empty set of edges, and any map $f : A \to B$ provides trivially a graph homomorphism $\mathfrak{V}(f) : \mathfrak{V}(A) \to \mathfrak{V}(B)$ thus the assignments $A \mapsto \mathfrak{V}(A)$ and $(f : A \to B) \mapsto (\mathfrak{V}(f) : \mathfrak{V}(A) \to \mathfrak{V}(B))$ define a functor $\mathfrak{V} : \mathsf{Set} \to \mathsf{Graph}$.

Turning back to the discussion, at the beginning of this section, it becomes obvious that we can transform any algebraic signature $\Sigma = (F, ar)$ into a graph signature $\Gamma_\Sigma = (F, I_\Sigma, R_\Sigma)$ with $I_\Sigma(\omega) = \mathfrak{V}(I_{ar(\omega)})$ and $R_\Sigma(\omega) = \mathfrak{V}(I_{ar(\omega)} \cup \{ov\})$. It is easy to see that any Σ-algebra can be transformed into a Γ_Σ algebra $\mathfrak{G}(\mathcal{A})$, and any Σ-algebra homomorphism $f : \mathcal{A} \to \mathcal{B}$ gives rise to a Γ_Σ-homomorphism $\mathfrak{V}(f) : \mathfrak{G}(\mathcal{A}) \to \mathfrak{G}(\mathcal{B})$.

Finally, the assignments $\mathcal{A} \mapsto \mathfrak{G}(\mathcal{A})$ and $(f : \mathcal{A} \to \mathcal{B}) \mapsto (\mathfrak{V}(f) : \mathfrak{G}(\mathcal{A}) \to \mathfrak{G}(\mathcal{B}))$ define an embedding $\mathfrak{G} : \mathsf{Alg}(\Sigma) \to \mathsf{GAlg}(\Gamma_\Sigma)$ where we have, by construction, that $|\_| \circ \mathfrak{G} = \mathfrak{V} \circ |\_|$ (see Fig. 1). Note, that $\mathsf{Alg}(\Sigma)$ and $\mathsf{GAlg}(\Gamma_\Sigma)$ are, in general, neither isomorphic nor equivalent since the carrier of a Γ_Σ-algebra can have edges even if the operations only work on vertices.

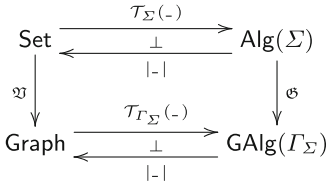


Fig. 1. Two compatible adjunctions

In the next section we will discuss the construction of free Γ-algebras for arbitrary graph signatures Γ providing a functor $\mathcal{T}_\Gamma(\_) : \mathsf{Graph} \to \mathsf{GAlg}(\Gamma)$ to be shown to be left adjoint to the forgetful functor $|\_| : \mathsf{GAlg}(\Gamma) \to \mathsf{Graph}$. This construction should generalize the construction of term algebras, in the sense, that for any algebraic signature Σ there is a natural isomorphism between the two functors $\mathfrak{G} \circ \mathcal{T}_\Sigma(\_)$ and $\mathcal{T}_{\Gamma_\Sigma}(\_) \circ \mathfrak{V}$ from Set into $\mathsf{GAlg}(\Gamma_\Sigma)$.

4 From Terms to Graph Terms

To have a guideline how to define terms in the setting of graph algebras, we analyze the construction of terms in Definition 1 in the light a graph signatures. As example we consider the algebraic signature Σ in Example 1.

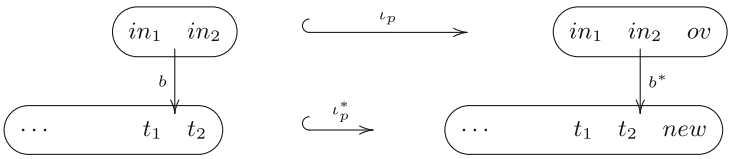


Fig. 2. Term construction as pushout

For the corresponding graph signature Γ_Σ the graph inclusion $\iota_\mathsf{p} : I_{\Gamma_\Sigma}(\mathsf{p}) \to R_{\Gamma_\Sigma}(\mathsf{p})$ is depicted in the upper part of the diagram in Fig. 2. In the left lower corner we depict the set of terms that have been constructed until now. Applying rule 3 in Definition 1 for $\omega = \mathsf{p}$ and two already constructed terms t_1, t_2 means to apply ι_p, considered as a graph transformation rule, for the binding $b = (in_1 \mapsto t_1, in_2 \mapsto t_2)$ and to construct a pushout, i.e., to produce exactly one new vertex new, as depicted in the lower right corner in Fig. 2. Denoting this new item by the term $\mathsf{p}\langle t_1, t_2\rangle$, solves two problems:

1. The term notation provides a uniform mechanism to create identifiers for new graph items introduced by applying non-deleting injective graph transformation rules (at least for graphs without edges). This problem is seldom addressed in the graph transformation literature.
2. The term $\mathsf{p}\langle t_1, t_2\rangle$ codes all the information about the pushout that has been creating the new item:
   (a) The symbol "p" identifies the rule that has been applied and, by consulting the signature, we find the necessary information about the input and result arity, respectively.
   (b) The string "t_1, t_2" codes the actual binding (match) $b = (in_1 \mapsto t_1, in_2 \mapsto t_2)$ for the input arity.
   (c) Since there is exactly one new item, we do have all information to identify uniquely the new item, and thus to define the resulting binding b^*.

In such a way, the term notation offers two possibilities to deal with the problem of applying the same rule twice for the same binding:

1. *A priori:* Before we apply a rule for a certain binding, we check the term denotations of all the items that have already been constructed. In such a way, we can find out, if the rule had already been applied for this binding. If this is the case, we do not apply the rule. Note, that the idea to use a rule as its own negative application condition [7], would be too rigid here. In case of the operation p, e.g., we couldn't apply ι_{p} to any graph with more than two vertices.
2. *A posteriori:* After applying a rule another time for a certain binding we repair the mistake silently by identifying the newly generated items with the "same" already existing items by the assumption that sets are extensional.

We like to adapt the silent a posteriori reparation mechanism and extend the term notation correspondingly. To deal with the rules arising from declaring arities of graph operations (see Definition 3) we have to address two problems: (1) An item can be of two different kinds - vertex or edge, and (2) there can be any finite number of output items instead of exactly one. To tackle problem (1), we will use for each kind a separate string of given terms, instead of just one string. And, by using output items as additional parts of terms, we solve problem (2).

In the context of graph algebras, we consider a collection of variables to be a graph rather than a set. Given a graph signature $\Gamma = (OP, I, R)$ and a graph X of variables, we will define (graph) Γ-terms over X using the following symbols/names: operation symbols from OP, names of output vertices and edges in $R(\omega) \setminus I(\omega)$ for all $\omega \in OP$, and auxiliary symbols like commas and brackets.

Convention 7. *For a graph G, operation symbol $\omega \in OP$, and binding $b : I(\omega) \to G$, we write the strings "$b(iv_1) \dots b(iv_{nv_\omega})$" and "$b(ie_1) \dots b(ie_{ne_\omega})$" of, resp., vertices and edges in G without commas and brackets, denote them by $\overline{b}_V$ and $\overline{b}_E$ resp., and write $\overline{b}$ for $\overline{b}_V; \overline{b}_E$. Extensionality ensures that $b1 = b2$ iff $\overline{b1} = \overline{b2}$ so that we can omit the overline bar.*

Now we are prepared to define graph terms in parallel to the traditional definition of terms in Definition 1.

Definition 8 (Graph terms). *Let be given a graph signature $\Gamma = (OP, I, R)$ and a graph X of variables. The graph $T_\Gamma(X)$ of all* graph Γ-terms *on X is the smallest graph, which satisfies the following three conditions:*
(Variables). *$T_\Gamma(X)$ contains the graph of variables, $X \sqsubseteq T_\Gamma(X)$;*
(Constants). *For all $c \in OP$ with $I(c) = \underline{\emptyset}$, graph $T_\Gamma(X)$ contains*

- *for each $ov \in R(c)_V$, tuple $\langle ov, c, \langle\rangle\rangle$ as a vertex,*
- *for each $oe \in R(c)_E$, tuple $\langle oe, c, \langle\rangle\rangle$ as an edge, where*

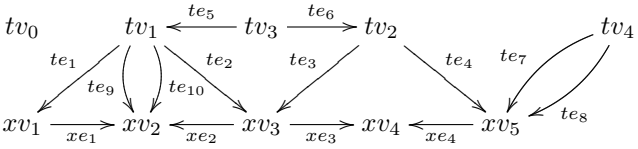
$sc^{T_\Gamma(X)}(\langle oe, c, \langle\rangle\rangle) = \langle sc^{R(c)}(oe), c, \langle\rangle\rangle$ and $tg^{T_\Gamma(X)}(\langle oe, c, \langle\rangle\rangle) = \langle tg^{R(c)}(oe) c \langle\rangle\rangle$;
(Operations) *For all $\omega \in OP$ with $I(\omega) \neq \underline{\emptyset}$ and any $b : I(\omega) \to T_\Gamma(X)$, graph $T_\Gamma(X)$ contains*

– *for each* $ov \in R(\omega)_V \setminus I(\omega)_V$, *tuple* $\langle ov, \omega, b\rangle$ *as a vertex*[1],
– *for each* $oe \in R(\omega)_E \setminus I(\omega)_E$, *tuple* $\langle oe, \omega, b\rangle$ *as an edge*[2], *whose source and target vertices are defined as follows:*

$$\mathsf{sr}^{T_\Gamma(X)}(\langle oe, \omega, b\rangle) = \begin{cases} b(\mathsf{sr}^{R(\omega)}(oe)) & \text{if } \mathsf{sr}^{R(\omega)}(oe) \in I(\omega)_V \\ \langle \mathsf{sr}^{R(\omega)}(oe), \omega, b\rangle & \text{if } \mathsf{sr}^{R(\omega)}(oe) \notin I(\omega)_V \end{cases}$$

$$\mathsf{tg}^{T_\Gamma(X)}(\langle oe, \omega, b\rangle) = \begin{cases} b(\mathsf{tg}^{R(\omega)}(oe)) & \text{if } \mathsf{tg}^{R(\omega)}(oe) \in I(\omega)_V \\ \langle \mathsf{tg}^{R(\omega)}(oe), \omega, b\rangle & \text{if } \mathsf{tg}^{R(\omega)}(oe) \notin I(\omega)_V \end{cases}$$

Example 5. As an example we consider the graph signature Γ from Example 2 and the graph X depicted in the last line in the following diagram.



Vertex tv_0 is generated by the rule ι_{ini}, i.e., $tv_0 = \langle ov, ini, \langle\rangle\rangle$. Vertices $tv_1, \ldots, tv_4$ and edges $te_1, \ldots, te_8$ are generated by the following four applications bi, $i = 1..4$ of the rule ι_{pb} (as there are no isolated vertices in the arity of pb, it's sufficient to specify the values of bindings on edges):

	$b1$	$b2$	$b3$	$b4$
ie_1	xe_1	xe_3	te_2	xe_4
ie_2	xe_2	xe_4	te_3	xe_4

which produce

$tv_1 = \langle ov, \mathsf{pb}, b1\rangle$ $\quad te_1 = \langle oe_2, \mathsf{pb}, b1\rangle$ $\quad te_2 = \langle oe_1, \mathsf{pb}, b1\rangle$
$tv_2 = \langle ov, \mathsf{pb}, b2\rangle$ $\quad te_3 = \langle oe_2, \mathsf{pb}, b2\rangle$ $\quad te_4 = \langle oe_1, \mathsf{pb}, b2\rangle$
$tv_3 = \langle ov, \mathsf{pb}, b3\rangle$ $\quad te_5 = \langle oe_2, \mathsf{pb}, b3\rangle$ $\quad te_6 = \langle oe_1, \mathsf{pb}, b3\rangle$
$tv_4 = \langle ov, \mathsf{pb}, b4\rangle$ $\quad te_7 = \langle oe_2, \mathsf{pb}, b4\rangle$ $\quad te_8 = \langle oe_1, \mathsf{pb}, b4\rangle$

Note, that the edge pair te_7 and te_8 could be declared as kernel of edge xe_4. Finally, edges te_9 and te_{10} are obtained by two applications of rule ι_{comp}: $b5(ie_1) = te_1$, $b5(ie_2) = xe_1$, and $b6(ie_1) = te_2$, $b5(ie_2) = xe_2$, which produce $te_9 = \langle oe, \mathsf{comp}, b5\rangle$, $te_{10} = \langle oe, \mathsf{comp}, b6\rangle$.

Analogously, to the case of terms, we can interpret the construction of graph terms as operations in special Γ-algebras:

[1] To show analogy with Definition 1 clearer, we could denote such tuples as $\omega_{ov}\langle b(iv_1), \ldots, b(iv_{nv_\omega});\ b(ie_1), \ldots, b(ie_{ne_\omega})\rangle$.
[2] Dito for $\omega_{oe}\langle b(iv_1), \ldots, b(iv_{nv_\omega});\ b(ie_1), \ldots, b(ie_{ne_\omega})\rangle$.

Definition 9 (Graph term algebra). *For a graph X of variables we denote by $\mathcal{T}_\Gamma(X) = (T_\Gamma(X), OP^{\mathcal{T}_\Gamma(X)})$ the Γ-algebra of graph Γ-terms on X with:*
(Constants). *For all $c \in OP$ with $I(c) = \underline{\emptyset}$*

$$c_V^{\mathcal{T}_\Gamma(X)}(\underline{\emptyset}_G)(ov) = \langle ov, c, \langle\rangle\rangle \quad \text{for all } ov \in R(c)_V,$$

$$c_E^{\mathcal{T}_\Gamma(X)}(\underline{\emptyset}_G)(oe) = \langle oe, c, \langle\rangle\rangle \quad \text{for all } oe \in R(c)_E;$$

(Operations). *For all $\omega \in OP$ with $I(\omega) \neq \underline{\emptyset}$ and all $b \in T_\Gamma(X)^{I(\omega)}$*

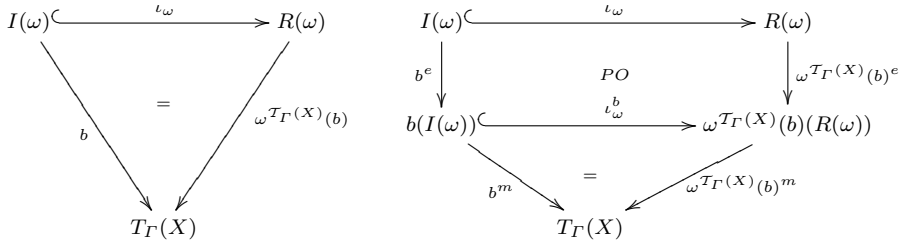
$$\omega_V^{\mathcal{T}_\Gamma(X)}(b)(v) = \begin{cases} b_V(v) & , \text{if } v \in I(\omega)_V \\ \langle v, \omega, b\rangle & , \text{if } v \in R(\omega)_V \setminus I(\omega)_V \end{cases}$$

$$\omega_E^{\mathcal{T}_\Gamma(X)}(b)(e) = \begin{cases} b_E(e) & , \text{if } e \in I(\omega)_E \\ \langle e, \omega, b\rangle & , \text{if } e \in R(\omega)_E \setminus I(\omega)_E \end{cases}$$

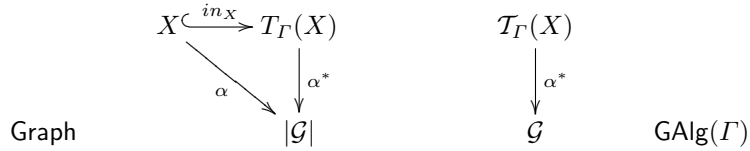
The definitions ensure that all resulting bindings $\omega^{\mathcal{T}_\Gamma(X)}(b) \in T_\Gamma(X)^{R(\omega)}$ are indeed graph homomorphisms and that $b = \omega^{\mathcal{T}_\Gamma(X)}(b) \circ \iota_\omega$, as required.

The condition "the smallest graph" in Definition 8 ensures that $\mathcal{T}_\Gamma(X)$ has "no junk", i.e., no proper subalgebra containing X, and the graph term notation ensures that there is "no confusion", i.e., variables are not identified with items introduced by operation applications. Moreover, items, introduced by different operation applications, are not identified either.

More structurally, "no confusion" means, especially, that for any $\omega \in OP$ and any $b \in T_\Gamma(X)^{I(\omega)}$, the commutative triangle below (on the left) factorizes, by epi-mono-factorization $b = b^m \circ b^e$ and $\omega^{\mathcal{T}_\Gamma(X)}(b) = \omega^{\mathcal{T}_\Gamma(X)}(b)^m \circ \omega^{\mathcal{T}_\Gamma(X)}(b)^e$, into a pushout square and a commutative triangle (below on the right).



Proposition 2 (Free graph algebras). *For each graph X the graph term Γ-algebra $\mathcal{T}_\Gamma(X) = (T_\Gamma(X), OP^{\mathcal{T}_\Gamma(X)})$ has the following universal property: For any Γ-algebra $\mathcal{G}$ and any variable assignment $\alpha : X \to |\mathcal{G}|$ there exists a unique Γ-homomorphism $\alpha^* : \mathcal{T}_\Gamma(X) \to \mathcal{G}$ such that: $\alpha^* \circ in_X = \alpha$.*



Proof. We prove by structural induction according to Definition 8:
(Variables). In this basic case the defining condition forces $\alpha^*_V(xv) = \alpha_V(xv)$ for all $xv \in X_V$ and $\alpha^*_E(xe) = \alpha_E(xe)$ for all $xe \in X_E$.
(Constants). In this basic case the definition of operations in $\mathcal{T}_\Gamma(X)$ and the desired homomorphism condition for α^* forces for all $ov \in R(c)$

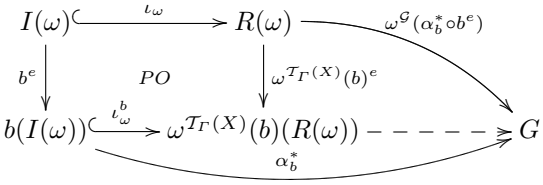
$$\alpha^*_V(\langle ov, c, \langle\rangle\rangle) = \alpha^*_V(c_V^{\mathcal{T}_\Gamma(X)}(\underline{\emptyset}_{T_\Gamma(X)})(ov)) = c_V^{\mathcal{G}}(\alpha^* \circ \underline{\emptyset}_{T_\Gamma(X)})(ov) = c_V^{\mathcal{G}}(\underline{\emptyset}_{\mathcal{G}})(ov)$$

and for all $oe \in R(c)_E$ we get, analogously, $\alpha^*_E(\langle oe, c, \langle\rangle\rangle) = c_E^{\mathcal{G}}(\underline{\emptyset}_{\mathcal{G}})(oe)$.
(Operations). The definition of operations in $\mathcal{T}_\Gamma(X)$ and the desired homomorphism condition forces α^* to be defined according to a corresponding recursion scheme for all $\omega \in OP$ with $I(\omega) \neq \underline{\emptyset}$ and all $b \in T_\Gamma(X)^{I(\omega)}$: The induction hypothesis is that α^* is already defined on the subgraph $b(I(\omega)) \sqsubseteq T_\Gamma(X)$. We denote the restriction of α^* to $b(I(\omega))$ by α^*_b. In the induction step we extend α^* to the subgraph $\omega^{\mathcal{T}_\Gamma(X)}(b)(R(\omega)) \sqsubseteq T_\Gamma(X)$ (that contains $b(I(\omega))$), i.e., to all graph terms that have been constructed exactly by applying rule ι_ω for the binding b: For all $ov \in R(\omega)_V$ we get

$$\alpha^*_V(\langle ov, \omega, b\rangle) = \alpha^*_V(\omega_V^{\mathcal{T}_\Gamma(X)}(b)(ov)) = \omega_V^{\mathcal{G}}(\alpha^*_b \circ b^e)(ov)$$

and for all $oe \in R(\omega)_E$ we get $\alpha^*_E(\langle oe, \omega, b\rangle) = \omega_E^{\mathcal{G}}(\alpha^*_b \circ b^e)(oe)$.

More structurally considered, the induction step constructs the unique mediating morphism from $\omega^{\mathcal{T}_\Gamma(X)}(b)(R(\omega))$ into G in the following diagram (Keep in mind that $\alpha^* \circ b^e = \omega^{\mathcal{G}}(\alpha^*_b \circ b^e) \circ \iota_\omega$ since $\omega^{\mathcal{G}}$ is a graph operation.):



A more traditional presentation of the induction step, analogously to (2), can be given if we use for the binding $b \in T_\Gamma(X)^{I(\omega)}$ the abbreviations $tv_j = b(iv_j)$, $1 \leq j \leq nv_\omega$ and $te_k = b(ie_k)$, $1 \leq k \leq ne_\omega$ (compare Convention 7), consider tuples as presentations of finite maps, as discussed at the beginning of Sect. 3, and represent the two maps constituting a binding for $I(\omega)$ in G as a sequence of vertices and edges of length $nv_\omega + ne_\omega$ in G: For all $ov \in R(\omega)_V$, we get

$$\begin{aligned}&\alpha^*_V(\langle ov, \omega, tv_1 \ldots tv_{nv_\omega} te_1 \ldots te_{ne_\omega}\rangle)\\ &= \omega_V^{\mathcal{G}}(\alpha^*_V(tv_1) \ldots \alpha^*_V(tv_{nv_\omega})\alpha^*_E(te_1) \ldots \alpha^*_E(te_{ne_\omega}))(ov).\end{aligned}$$

The universal property in Proposition 2 ensures that we can extend the assignments $X \mapsto \mathcal{T}_\Gamma(X)$ to a functor $\mathcal{T}_\Gamma(\_) : \mathsf{Graph} \to \mathsf{GAlg}(\Gamma)$ that is left-adjoint to the forgetful functor $|\_| : \mathsf{GAlg}(\Gamma) \to \mathsf{Graph}$ (see Fig. 1). That the adjunction $\mathcal{T}_\Gamma(\_) \dashv |\_|$ generalizes the construction of term algebras, in the sense, that for any algebraic signature Σ there is a natural isomorphism between the

two functors $\mathfrak{G} \circ \mathcal{T}_\Sigma(\_)$ and $\mathcal{T}_{\Gamma_\Sigma}(\_) \circ \mathfrak{V}$ from Set into $\mathsf{GAlg}(\Gamma_\Sigma)$(see Fig. 1) can be shown straightforwardly.

Establishing the adjunctions $\mathcal{T}_\Gamma(\_) \dashv |\_|$ is the main result of the paper. Since the new adjunctions generalize the adjunctions $\mathcal{T}_\Sigma(\_) \dashv |\_|$, we will be able to transfer smoothly many concepts, constructions, and results from the area of algebraic specifications to the new setting of graph algebras (see Sect. 6).

Equations, for example, can be defined as pairs of graph terms and can be used to formulate properties of graph operations. Associativity of composition, e.g., can be expressed by the equation (we recall Convention 7 about denotations of binding mappings)

$$\langle oe, \mathsf{comp}, \langle oe, \mathsf{comp}, xe_1 xe_2 \rangle xe_3 \rangle = \langle oe, \mathsf{comp}, xe_1 \langle oe, \mathsf{comp}, xe_2 xe_3 \rangle \rangle$$

where X is the graph $(xv_1 \overset{xe_1}{\rightarrow} xv_2 \overset{xe_2}{\rightarrow} xv_3 \overset{xe_3}{\rightarrow} xv_4)$. Since there are no isolated vertices in the arities of our sample operations we list only edge variables in the sample equations. We may also require that our choice of pullbacks is symmetric in the sense that the following equations are satisfied:

$$\begin{aligned} \langle ov, \mathsf{pb}, xe_1 xe_2 \rangle &= \langle ov, \mathsf{pb}, xe_2 xe_1 \rangle \\ \langle oe_1, \mathsf{pb}, xe_1 xe_2 \rangle &= \langle oe_2, \mathsf{pb}, xe_2 xe_1 \rangle \\ \langle oe_2, \mathsf{pb}, xe_1 xe_2 \rangle &= \langle oe_1, \mathsf{pb}, xe_2 xe_1 \rangle, \end{aligned}$$

where X is the graph $(xv_1 \overset{xe_1}{\rightarrow} xv_3 \overset{xe_2}{\leftarrow} xv_2)$. Note, that we can not summarize the three equations by a single (hypothetical) equation between bindings

$$\mathsf{pb}\langle xe_1 xe_2 \rangle = \mathsf{pb}\langle xe_2 xe_1 \rangle$$

since oe_1 and oe_2 are interchanged in the last two equations above.

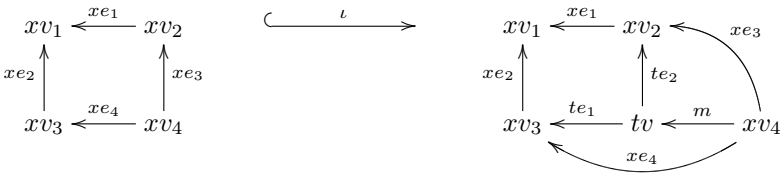
The Kleisli category of the new adjunction provides an appropriate substitution calculus. A substitution is an arrow in the Kleisli category, i.e., a graph homomorphism $\sigma : X \rightarrow T_\Gamma(Y)$. The corresponding extended graph homomorphism $\sigma^* : T_\Gamma(X) \rightarrow T_\Gamma(Y)$ describes the simultaneous substitution of all variable vertices xv and variable edges xe in graph Γ-terms on X by the corresponding graph terms $\sigma_V(xv) \in T_\Gamma(Y)_V$ or $\sigma_E(xe) \in T_\Gamma(Y)_E$, respectively. The composition of two substitutions $\sigma : X \rightarrow T_\Gamma(Y)$ and $\delta : Y \rightarrow T_\Gamma(Z)$ is given by $\delta^* \circ \sigma : X \rightarrow T_\Gamma(Z)$. Substitutions $\sigma : X \rightarrow T_\Gamma(Y)$ allow us, for example, to formalize the concepts of queries and views in databases [3].

Remark 1 (Universal properties). For a categorically minded reader, considering such operations as pullback and pushout without their universal properties does not make too much sense. Below we will show how to include universal properties into our framework of diagram operations. We will consider universality of pullbacks, but the method is quite general and applicable for any limit/colimit operation over graphs.

Commutativity of a pullback square can be expressed by the following equation

$$\langle oe, \mathsf{comp}, \langle oe_1, \mathsf{pb}, ie_1 ie_2 \rangle ie_2 \rangle = \langle oe, \mathsf{comp}, \langle oe_2, \mathsf{pb}, ie_1 ie_2 \rangle ie_1 \rangle$$

where operations pb and comp are defined in Example 3. Universal properties, however, are conditional statements thus we need a kind of implication to express them. The implications, we are looking for, are a further development of the *sketch axioms* in [11] (see also Sect. 5). Those implications are based on graph homomorphisms. To express the existence of mediating morphisms we consider, in case of pullbacks, the following inclusion of graphs:

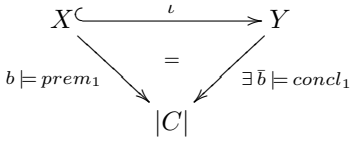


where $tv = \langle ov, \mathsf{pb}, xe_1xe_2\rangle$, $te_1 = \langle oe_2, \mathsf{pb}, xe_1xe_2\rangle$ and $te_2 = \langle oe_1, \mathsf{pb}, xe_1xe_2\rangle$. We denote the graph on the left-hand side by X and the graph on the right-hand side by Y. Since, m is the only item in Y, that is not in X or generated by X, respectively, the inclusion homomorphism allows us to formulate a conditional existence statement of the form

$$\forall\, X.(prem_1 \overset{\iota}{\Rightarrow} \exists\, m.\ concl_1) \quad \text{where}$$

$$prem_1 := \langle oe, \mathsf{comp}, xe_4xe_2\rangle = \langle oe, \mathsf{comp}, xe_3xe_1\rangle$$
$$concl_1 := \langle oe, \mathsf{comp}, m\, te_1\rangle = xe_4 \wedge \langle oe, \mathsf{comp}, m\, te_2\rangle = xe_3.$$

A graph operation $\mathsf{pb}^C : |C|^{\mathsf{Cospan}} \to |C|^{\mathsf{Square}}$, as in Example 3, satisfies this implication iff every binding $b : X \to |C|$, that makes the premise $prem_1$ true, can be extended to a binding $\bar{b} : Y \to |C|$ with $\bar{b} \circ \iota = b$ such that the following two conditions hold: (a) $\bar{b}(te_1) = \langle oe_1, \mathsf{pb}, xe_1xe_2\rangle$ and $\bar{b}(te_2) = \langle oe_2, \mathsf{pb}, xe_1xe_2\rangle$ and (b) the conclusion $concl_1$ becomes true. Note, that condition (a) ensures that only an appropriate match for the edge m needs to be found.



To express the uniqueness of mediating morphisms we exploit a non-injective but surjective graph homomorphism $\varphi : Y' \to Y$ where Y' is Y plus an additional edge m' from xv_4 to tv. φ is the identity except that it maps m and m' in Y' to m in Y. The premise $prem_2$ is given by two corresponding copies of $concl_1$ above and the conclusion $concl_2$ is just *true*. A graph operation pb^C satisfies the implication $\forall\, Y'.(prem_2 \overset{\varphi}{\Rightarrow} true)$ iff every binding $b : Y' \to |C|$, that makes the premise $prem_2$ true, can be extended to a binding $\bar{b} : Y \to |C|$ with $\bar{b} \circ \varphi = b$. In other words, there is no binding $b : Y' \to |C|$ with $\bar{b}(m) \neq \bar{b}(m)$ that makes the premise $prem_2$ true.

5 Related Work

An abstract diagrammatic approach to logic, including a general notion of diagram predicates and their models (generalized sketches), and implications between diagram predicates (sketch axioms), was pioneered by Makkai in [11] (see also historical remarks in our paper [6]). However, Makkai did not work with diagram operations and algebras. Formal definitions of a (diagrammatic) graph operation and a graph algebra were introduced by the second author in [4], and many examples and discussions in the database context can be found in [3]. The latter paper also describes the construction of what they call *sketch parsing*. The idea is that any operation signature Γ gives rise to a predicate signature Γ^* by forgetting the input arity parts in the entire operation arities. Then any Γ-term becomes a Γ^*-sketch [6]. Parsing does the inverse: given an Γ^*-sketch, it tries to convert it into an Γ-term. In these papers, graph terms are defined as trees labeled by diagrams respecting operation arities. In the present paper, we are more interested in the entire object of graph term algebra and its universal properties rather than in the notion of a single graph term. Neither of the papers above formally defined the graph term algebra and proved its universal properties.

Injective graph transformation rules have been studied extensively by Hartmut Ehrig and his co-authors (see [7]). The special feature of injective rules, elucidated in the paper, may shed new light on the "nature" of those rules.

6 Conclusion and Future Work

In the paper, we extended the classical construction of term algebra for operations over sets to the case of diagrammatic operations over graphs. We showed that any graph term algebra freely generated by applying graph operations to a given graph of variables is indeed free: it possesses the respective universal property in the category of graph algebras. This basic result shows that our definitions of graph operations and graph algebras work as we wanted, i.e., in parallel with the ordinary algebra case. Moreover, this result hopefully paves a way to a wider generalization of the core Universal Algebra framework for graph operations and graph algebras. In more detail, we aim at defining congruence relations, quotients and epi-mono factorizations for graph algebras, thus building what we could call *Graph-based Universal Algebra*. More abstractly, it would also be interesting to extend our result in [6] concerning institutions of generalized sketches to any of the envisaged logical extensions.

We see other interesting and useful extensions of the framework.

Typing. The step from unsorted to many-sorted algebras is relatively straightforward. In the same way, we see no principle problems to extend the framework, presented in this paper, to typed graphs [7]. This extension will be necessary to meet the situations in applications (compare [3,12–14]).

Term Language. In the paper we considered two roles of ordinary terms and their extension for graph algebras. These two roles are (a) to denote elements in

free algebras and (b) to provide induction/recursion schemes for evaluating the elements of free algebras in arbitrary algebras. However, terms are also used (c) to denote composed/derived operations in algebras. This role provides the foundation for functorial semantics and thus for a categorical approach to Universal Algebra. By extending the approach in [2], we plan to specify this role in the setting of graph algebras too.

From Graphs to Presheaf Toposes. To meet the spirit of the Festschrift, in the paper we focused on graph-based structures, which have been the basis for research on graph transformations in Hartmut's group for decades [7]. The category of graphs, however, is a very simple instance of a quite general concept of a presheaf topos that encompasses 2-graphs, Petri nets, attributed graphs [7], and many other structures employed in computer science. There should be no principle problems to extend the definitions and results of the paper to the broader class of presheaf topoi.

References

1. Cadish, B., Diskin, Z.: Heterogeneous view integration via sketches and equations. In: Raś, Z.W., Michalewicz, M. (eds.) ISMIS 1996. LNCS, vol. 1079, pp. 603–612. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61286-6_184
2. Claßen, I., Große-Rhode, M., Wolter, U.: Categorical concepts for parameterized partial specifications. Math. Struct. Comput. Sci. **5**(2), 153–188 (1995). https://doi.org/10.1017/S0960129500000700
3. Diskin, Z.: Databases as diagram algebras: specifying queries and views via the graph-based logic of sketches. Technical report 9602, Frame Inform Systems, Riga, Latvia (1996).http://www.cs.toronto.edu/zdiskin/Pubs/TR-9602.pdf
4. Diskin, Z.: Towards algebraic graph-based model theory for computer science. Bull. Symb. Log. **3**, 144–145 (1997)
5. Diskin, Z., Cadish, B.: A graphical yet formalized framework for specifying view systems. In: First East-European Symposium on Advances in Databases and Information Systems, pp. 123–132. Nevsky Dialect (1997)
6. Diskin, Z., Wolter, U.: A diagrammatic logic for object-oriented visual modeling. ENTCS **203**(6), 19–41 (2008). https://doi.org/10.1016/j.entcs.2008.10.041
7. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformations. EATCS Monographs on Theoretical Computer Science. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-31188-2
8. Ehrig, H., Mahr, B.: Fundamentals of Algebraic Specification 1: Equations and Initial Semantics. EATCS Monographs on Theoretical Computer Science, vol. 6. Springer, Heidelberg (1985). https://doi.org/10.1007/978-3-642-69962-7
9. Ehrig, H., Mahr, B.: Fundamentals of Algebraic Specification 2: Module Specifications and Constraints. EATCS Monographs on Theoretical Computer Science, vol. 21. Springer, Heidelberg (1990). https://doi.org/10.1007/978-3-642-61284-8
10. Mac Lane, S.: Categories for the Working Mathematician, 2nd edn. Springer, New York (1978). https://doi.org/10.1007/978-1-4757-4721-8
11. Makkai, M.: Generalized sketches as a framework for completeness theorems. J. Pure Appl. Algebra **115**, 49–79, 179–212, 214–274 (1997)

12. Mantz, F., Taentzer, G., Lamo, Y., Wolter, U.: Co-evolving meta-models and their instance models: a formal approach based on graph transformation. Sci. Comput. Program. **104**, 2–43 (2015). https://doi.org/10.1016/j.scico.2015.01.002
13. Rossini, A., Rutle, A., Lamo, Y., Wolter, U.: A formalisation of the copy-modify-merge approach to version control in MDE. J. Log. Algebraic Programm. **79**(7), 636–658 (2010). https://doi.org/10.1016/j.jlap.2009.10.003
14. Rutle, A., Rossini, A., Lamo, Y., Wolter, U.: A formal approach to the specification and transformation of constraints in MDE. J. Log. Algebraic Programm. **81**(4), 422–457 (2012). https://doi.org/10.1016/j.jlap.2012.03.006