URL: http://www.elsevier.nl/locate/entcs/volume61.html 17 pages

# Computing over K-modules

Neil Ghani [1] and Anne Heyworth [2,3]

*Department of Mathematics and Computer Science*
*University of Leicester*
*Leicester LE1 7RH, UK*

**Abstract**

Kan extensions over the category of Sets provide a unifying framework for computation of group, monoid and category actions allowing a number of diverse problems to be solved with a generalised form of string rewriting. This paper extends these techniques to $K$-algebras and $K$-categories by using Gröbner basis techniques to compute Kan extensions over the category of $K$-modules.

## 1 Introduction

**Computer Algebra:** Computer algebra packages are widely used in mathematics and computer science to solve combinatorial problems whose essence is the computation of the quotient of an algebraic structure. Such problems occur widely throughout mathematics in the theory of groups, rings, modules etc, and computer science, for example in equational reasoning [14] and the study of Petri nets [28]. Central to any computer algebra package is the representation of the algebraic structure to be quotiented as a data structure and the algorithms used to compute the quotient. Current packages suffer from two main drawbacks: i) computation is limited to those algebraic structures and quotients for which data structures and algorithms have been built into the package; and ii) many packages are limited to finite structures as they enumerate the elements of the quotient. Thus, although there are a number of successful computer algebra packages for computation over groups, for example GAP [12], and *KBMAG* [17], there is little support for computation structures such as rings, modules and algebras beyond 'Vector Enumerator' which implements a Todd-Coxeter type algorithm for modules [23].

[1] Email: ng13@mcs.le.ac.uk
[2] Email: ah83@mcs.le.ac.uk
[3] Supported by EPSRC grant number GR/R29604/01 Kan: A Categorical Approach to Computer Algebra

**Categorical Computer Algebra:** Our long-term aim is to develop a generic computer algebra package *Kan* supporting the computation of a variety of quotients in different algebraic structures. Our central idea is to use generalised forms of rewriting to compute *Kan extensions*[4] and then compute other quotients by expressing them as Kan extensions. The use of category theory as a meta-language for algebra can be traced back to Linton [22] and Lawvere [21] and the study of algebra has since remained central to the development of category theory [2]. Our main contribution here is to observe that this meta-language extends naturally, through the use of Kan extensions, to cover computational algebra. The use of rewriting to perform computation ensures that, unlike enumerative methods, our algorithms are not limited to finite structures.

**Computation over Sets:** In our previous work we unified a number of different computational problems as Kan extensions whose codomain was the category Sets. These included a number of fundamental problems in computational group theory, for example the calculation of presentations, cosets, orbits and induced actions as such Kan extensions, as well as analogues in the theory of monoids and categories. The construction of groups and monoids from finite presentations is based upon the free monoid functor and hence leads to a notion of computation based upon a generalised form of string rewriting. The advantage of this approach is that instead of implementing a number of different algorithms, and providing each with a correctness proof, we need only implement a single algorithm which significantly reduces the work required. The correctness is also simplified because the structures being quotiented and their representation as Kan extensions are both algebraic in nature and because the quotient is simply the symmetric closure of the rewrite relation used to compute it. This work is summarised in Section 2 so as to allow the reader to familiarise themselves with our general approach.

**Computation over $K$-modules** This paper extends our previous work [4] to computing $K$-modules, $K$-algebras and $K$-categories by using Kan extensions whose codomain is KMods, the category of $K$-modules. $K$-modules are of particular interest in representation theory, while Lie, Hecke, Serre and String algebras are widely studied examples of $K$-algebras. Rather than invent new computational paradigms for these structures, the categorical approach notes that $K$-algebras are internal monoids in the category KMods of $K$-modules in the same way that monoids are internal monoids in Sets. Hence one can compute with $K$-algebras by using the same Kan extensions as before but now the codomain of the Kan extension will be KMods. Overall, the change in algebraic structure from monoid-like structures to $K$-algebras is elegantly and succinctly modelled by computing the same Kan extensions but over a

[4] Throughout this paper "Kan extension" is taken to mean left Kan extension

more complex base category. Having reduced computation with $K$-algebras to the computation of Kan extensions with codomain KMods, we then turn to the computation of these Kan extensions. The construction of $K$-algebras from finite presentations is based upon the free ring functor and this leads to computation based upon Gröbner basis techniques.

**Background:** This synthesis of category theory and rewriting is part of a strand of research dating back to the late 1980's when it was observed that the traditional denotation models of computation based upon categories could be extended to cover operational aspects. The seminal research in this field has focused on the development of categorical models of rewriting [30,29,18,25,13]. We hope that, by leading to the actual production of software, this research will be seen as part of the maturing of the field of *Categorical Rewriting*.

Our work is related to that of Carmody and Walters [6,7] who provided Todd-Coxeter algorithms for computing Kan Extensions but only over Sets. This was implemented by Rosebrugh in [10]. The rewriting techniques of [4] provided an alternative to their enumerative methods in the same way that string rewriting provides an alternative to traditional Todd-Coxeter for groups [9].

To summarise, this paper provides a number of theoretical and practical insights.

- We show that in the context of $K$-algebras and $K$-modules, different quotients can be modelled uniformly as Kan extensions
- We provide evidence that computing quotients in different algebraic structures can be achieved by changing the codomain of the associated Kan extension.
- We show that Kan extensions into the category KMods can be computed by Gröbner basis techniques.
- We consequently provide algorithms for computing canonical quotients of $K$-algebras.

One further question needs to be addressed, namely the suitability of this paper for a theoretical computer science audience. Although modules and algebras may not be standard fare for such an audience we believe that our methodology is. Certainly, category theory has become widespread throughout the theoretical computer science community in providing a meta-language for computation. Indeed it is hard to find a modern paper on the denotational semantics of programming languages which is not written in the categorical dialect. Furthermore, rewriting falls clearly within the realms of theoretical computer science. We believe that the synthesis of category theory and rewriting will therefore be of interest to the participants of CATS2002.

The rest of the paper is structured as follows: Section 2 uses the examples of groups to explain our general approach, Section 3 introduces $K$-modules, $K$-algebras and $K$-categories and states their basic properties. Section 4 derives algorithms to compute Kan extensions over KMods and we briefly outline some examples in Section 5. We conclude in Section 6 by outlining plans for future research.

## 2 Computation over Sets

We illustrate our previous work on Kan extensions over Sets by describing several problems in computational group theory and our solution to them.

### *2.1 Four Problems in Computational Group Theory*

Let $U$ : Grp $\to$ Sets be the forgetful functor from the category of groups to the category of sets and $F$ be its left adjoint.

**Definition 2.1** A group presentation $grp\langle X|R\rangle$ consists of a finite set $X$ and a finite subset $R \subseteq F(X)$. A group $G$ is presented by a group presentation $grp\langle X|R\rangle$ if and only if $G$ is isomorphic to the quotient of $F(X)$ by the equivalence relation induced by $r \sim 0$ for $r \in R$.

Categorically, $G$ is presented by $grp\langle X|R\rangle$ if and only if $G$ is the coequalizer in Grp of

$$F(R) \underset{0}{\overset{r^*}{\rightrightarrows}} F(X)$$

where $r^*$ is induced by the function sending $r$ as an element of $R$ to $r$ as an element of $F(X)$ and 0 is the constantly 0 function. Either way, $G$ is a quotient of $F(X)$ and for $p \in F(X)$, its equivalence class in $G$ is written $[p]_G$.

**Problem 2.2** *Given a group presentation $grp\langle X|R\rangle$ of $G$ and elements $p, q \in F(X)$, is it the case that $[p]_G = [q]_G$?*

Our second problem concerns *cosets*.

**Definition 2.3 (Cosets)**
Let $H$ be a subgroup of $G$. The set of cosets $G/H$ is the quotient of the carrier set of $G$ by the equivalence relation $g \sim g + h$ where $g \in G$ and $h \in H$. The equivalence class of $g \in G$ is written $gH$

In general, the cosets $G/H$ form a set and not a group — the obvious notion of multiplication is not well-defined on the equivalence classes of $G/H$. This example shows that one cannot base a general model of algebraic computation on *coequalisers* as all objects would have to reside in the same category.

**Problem 2.4** *Let $H$ be a subgroup of $G$ and $g_1, g_2 \in G$. Does $g_1H = g_2H$?*

Our third problem concerns the notion of an action, or $G$-set, and seeks to calculate the quotient of the carrier set by the action.

**Definition 2.5** If $G$ is a group, then a $G$-set is a set $X$ with a function $G \times X \to X$ (here written by juxtaposition) such that $1x = x$ and $(g_1 + g_2)x = g_1(g_2 x)$.

The *orbits* of the action are the collections of elements of the set which can be obtained by applying the action repeatedly to an element.

**Problem 2.6** *Let $\phi : G \times X \to X$ be a $G$-set. The orbits of $\phi$ is the quotient of $X$ under the equivalence relation $x \sim gx$ for $g \in G$. The orbit of $x \in X$ is denoted $[x]_\phi$. Given elements $x_0, x_1 \in X$, is it the case that $[x_0]_\phi = [x_1]_\phi$.*

Finally, we introduce the problem of induced actions.

**Problem 2.7** *Let $\phi : G \times X \to X$ be a $G$-set and $f : G \to G'$ be a group homomorphism. The induced action $\phi/f$ is the $G'$-set whose carrier is the quotient of $X \times G'$ by the equivalence relation*

$$\langle gx, g' \rangle \sim \langle x, f(g)g' \rangle \tag{1}$$

### *2.2 Unification of Quotients as Kan-extensions*

One could write an algorithm, together with an associated correctness proof, for each quotient one wants to compute. Although this is possible, the volume of work makes this a lengthy process and increases the possibility of errors arising in the algorithms or their implementation.

Our alternative approach begins by translating these problems into category theory. First note that we can regard a group $G$ as a category with one object and whose hom-set is the set $G$. Composition in the category is given by addition in the group and the identity is the zero of the group. Of course, this construction works for any monoid — the presence of inverses in the group means that every arrow in the associated category is an isomorphism. We use $G$ to denote both a group and its associated category. In the same way, group homomorphisms form functors. Actions are also functors. In particular, given an action $\phi : G \times X \to X$ we define a functor $\phi : G \to \mathsf{Sets}$ whose action sends the object of $G$ to $X$ and every $g \in G$ to the function $\phi(g) : X \to X$ defined by $\phi(g)(x) = gx$. The $G$-set axioms correspond precisely to the functoriality axioms. If $G$ is a group, we also use $G$ to denote its categorical representation and similarly for homomorphisms and actions.

Let us turn to our four computational problems. We start with, because it turns out to be most general, the problem of induced actions. Recall that we have a $G$-set $\phi : G \times X \to X$ and a homomorphism $f : G \to G'$. The induced action is the $G'$-action $\phi/f$ whose carrier is the equivalence classes of $X \times G'$ under the equivalence relation $\langle gx, h \rangle \sim \langle x, f(g)h \rangle$ and whose action is $h'\langle x, h \rangle = \langle x, h'h \rangle$. A priori, this particular quotient does not appear to have any simple categorical explanation in the way that groups, homomorphisms

and actions did. However, note there is a natural transformation

$$\epsilon : \phi \Rightarrow \phi/f \circ f : G \to \mathsf{Sets}$$

whose only component $\epsilon_* : X \to X \times G'/\sim$ is defined by $\epsilon(x) = [\langle x, 1\rangle]$, i.e. the map sending $x$ to the equivalence class of $\langle x, 1\rangle$. The naturality of $\epsilon$ is precisely equation 1. The fact that the induced action is the quotient of equation 1 means that the induced action is the smallest $G'$-set equipped with such a natural transformation. That is, there is a natural bijection between $G'$-sets $\psi$ equipped with a natural transformation $\alpha : \phi \to \psi \circ f$ and natural transformations $\phi/f \Rightarrow \psi$. The induced action is simply the left Kan extension of $\phi$ along $f$ written $\mathsf{Lan}_f \phi$.

**Definition 2.8 (Left Kan Extensions)** Let $F : A \to C$ and $G : A \to B$ be functors. The left Kan extension of $F$ along $G$ is a functor $\mathsf{Lan}_G F : B \to C$ such that there is a natural bijection $Nat(\mathsf{Lan}_G F, H) \cong Nat(F, HG)$.

Kan extensions were originally developed 40 years ago [19] and have since become a fundamental construction in category theory [8,24]. There are a number of alternative formulations of left Kan extensions, e.g. when it exists, $\mathsf{Lan}_{G\_} \dashv \_\circ G : C^A \to C^B$. The above discussion shows that induced actions are Kan extensions — the natural transformation $\epsilon$ is the unit of the adjunction and the bijection property of $\epsilon$ is the universal property of the unit. By itself, representing induced actions as Kan extensions is not so interesting. However, what is interesting is that all of the other problems encountered so far are also induced actions and hence left Kan extensions.

**Lemma 2.9** *Let* $1$ *be the group with one element and* $!_G : G \to 1$ *the unique group homomorphism from* $G$ *to* $1$*. The orbit of an action* $\phi : G \to \mathsf{Sets}$ *is the induced action of* $\phi$ *along* $!_G$*, or equivalently,* $\mathsf{Lan}_{!_G}\phi$*.*

**Proof.** The induced action is a quotient of $\phi(*) \times 1$ which is isomorphic to $\phi(*)$. The equivalence relations are also isomorphic: $\langle gx, * \rangle \sim \langle x, !_G(g)* \rangle = \langle x, * \rangle$. □

**Lemma 2.10** *Let* $H$ *be a subgroup of* $G$*. If* $\phi$ *is the trivial action of* $H$ *on the one point set* $*$*, then the right (and left) cosets* $G/H$ *are isomorphic to the induced action of* $\phi$ *along the inclusion* $i : H \to G$ *or, equivalently, the Kan extension* $\mathsf{Lan}_i \phi$*.*

**Proof.** The induced action is a quotient of $1 \times G$ which is isomorphic to $G$. The defining equivalence relations are also isomorphic $\langle *, g \rangle = \langle *h, g \rangle \sim \langle *, hg \rangle$. □

So we have an elegant and abstract way of encoding computational problems as Kan extensions. The reader may argue that since all of our examples are induced actions, induced actions could be taken as the primary concept. In fact induced actions are precisely Kan extensions of functors whose domain

and codomain are 1-object categories. The extra generality afforded by Kan extensions is crucial in modelling several computational problems , for example colimits and path algebras as we mention in Section 5.
subsectionComputation via String Rewriting
The key to the computation of Kan extensions is their expression as coends

$$(\mathsf{Lan}_F M)B = \int^{A \in \mathrm{Ob}\mathsf{A}} MA \otimes \mathsf{B}(FA, B)$$

where $\otimes$ is the tensor operation. In the examples presented above, the codomain of $M$ is $\mathsf{Sets}$ and hence $MA$ is just a set while the tensor operation is cartesian product. In addition, the categories $\mathsf{A}$ and $\mathsf{B}$ are generated from graphs and hence $\mathsf{B}(FA, B)$ a set of equivalence classes of paths over a generating set $\Lambda_B$. Thus $(\mathsf{Lan}_F M)B$ is contained in the quotient of the free monoid $(MA \times \Lambda_B)^*$ by the equivalence relation induced by the relations of $\mathsf{B}$ and the Kan relations $\langle F(f)(a), g\rangle \ \langle M(f)g\rangle$. Since these relations are strings, we have expressed the Kan extension as a quotient of a free monoid by a set of equations between words, ie a problem expressible within the string rewriting formalism. In particular, one can use Knuth-Bendix completion for string rewriting to generate (if possible) a complete string rewrite system and hence decide when two elements of $(MA \times \Lambda_B)^*$ represent the same element in the Kan extension.

## 3 Computation over $K$-modules

Rather than developing new data structures and algorithms for computing with $K$-modules, algebras and categories, we regard them as internal constructions in $\mathsf{KMods}$ and then compute with them as before. In order to do so, we define the notion of internal monoid — see [24] for details.

**Definition 3.1** A monoid in a monoidal category $(C, \otimes, I)$ consists of an object $X$ of $C$ together with maps $e : 1 \to X$ and $m : X \otimes X \to X$ such that the obvious monoid laws hold. Given a monoid $(X, e, m)$, the map $Z \mapsto X \otimes Z$ defines the action of a monad on $C$. An $X$-action is an $X \otimes \_$-algebra.

For example, a monoid $M$ is a monoid in $\mathsf{Sets}$, while a $M$-action is precisely a $M$-set. In the category $\mathsf{Ab}$ of abelian groups, a monoid in $\mathsf{Ab}$ is a ring $R$, while an $R$-action is a $R$-module. In the category $\mathsf{KMods}$ of $K$-modules, a monoid is a $K$-algebra $A$ while an $A$-action is an $A$-module. We now give more traditional definitions of $K$-modules etc.

**Definition 3.2** ($K$**-modules)**
Let $K$ be a field. A $K$-module is an abelian group $\Sigma$ together with a scalar

multiplication $\alpha : K \times \Sigma \to \Sigma$ (written by juxtaposition) such that

$$\begin{aligned} k(g+h) &= kg +_k h \\ (k_1+k_2)g &= k_1 g + k_2 g \\ (k_1 k_2)g &= k_1(k_2 g) \\ 1m &= m. \end{aligned}$$

The first three equations guarantee that $\alpha$ is a bilinear map and, by the universal property of the tensor of abelian groups, therefore it can be identified with a group homomorphism $\alpha : K \otimes A \to A$. The third and fourth equations then make $\alpha$ into a $K \otimes \_$-algebra. Since $K$-modules are $K \otimes \_$-algebras, we could define KMods to be the Eilenberg-Moore category of the $K \otimes \_$-monad. More concretely,

**Definition 3.3 (The category KMods)**
Given $K$-modules $\alpha : K \otimes A \to A$ and $\beta : K \otimes B \to B$, a $K$-module homomorphism is a group homomorphism $f : A \to B$ such that $f(kg) = kf(g)$. The category KMods has as objects $K$-modules and as morphisms $K$-module homomorphisms.

The condition on $K$-module homomorphisms can be written as a commuting diagram in Ab

$$\begin{array}{ccc} K \otimes A & \longrightarrow & A \\ {\scriptstyle K\otimes f}\downarrow & & \downarrow{\scriptstyle f} \\ K \otimes B & \longrightarrow & B \end{array}$$

which illustrates $K$-module homomorphisms as $K \otimes \_$-algebra homomorphisms. The categorical approach to modules gives immediate results on the structure of KMods. Firstly, one can construct free $K$-modules over sets.

**Lemma 3.4** *The forgetful functor $U_M$ : KMods $\to$ Sets has a left adjoint $F_M$ whose action maps a set $S$ to the set of all polynomials $k_1 s_1 + \cdots + k_n s_n$ for $s_i \in S$ and $k_i \in K$.*

In fact KMods arises as a finitary algebraic theory, so we have the following lemma.

**Lemma 3.5** KMods *is locally finitely presentable and hence complete and cocomplete.*

A consequence of this lemma is that a $K$-module $\Sigma$ is finitely presented if there is a pair $mod_K\langle \Lambda | R \rangle$ where $\Lambda$ is a finite set and $R \subseteq F_M(\Lambda)$ is a finite set such that $\Sigma$ is the coequaliser in KMods of the following diagram

$$F_M(R) \underset{0}{\overset{r^*}{\rightrightarrows}} F_M(\Lambda)$$

where $r^*$ is the module homomorphism defined by sending $r$ to its canonical interpretation in $F_M(\Lambda)$ and 0 is the map which is constantly 0. More concretely, given a presentation $mod_K\langle\Lambda|R\rangle$, define an equivalence relation on $F_M(\Lambda)$ by $f =_R h$ if and only if $f = h + k_1r_1 + \cdots + k_nr_n$ for $k_i \in K$ and $r_i \in R$. Then, the set theoretic quotient $F_M(\Lambda)/=_R$ is a $K$-module and we say $\Sigma$ is presented by $mod_K\langle\Lambda|R\rangle$ if and only if $\Sigma \sim F_M(\Lambda)/=_R$.

**Lemma 3.6** KMods *is a symmetric monoidal closed category as follows*

- *The unit is the free $K$-module on one generator, namely $K$*
- *The tensor of $K$-modules $R \otimes A \to A$ and $R \otimes B \to B$ is denoted $A \otimes_K B$ and is the quotient of $A \otimes B$ by the relation $r(a \otimes b) = ra \otimes b = a \otimes rb$.*
- *The exponential $[A, B]$ is the set of $K$-module homomorphisms between $A$ and $B$. This set is an abelian group:*

$$(f + g)x = fx + gx, \qquad 0x = 0, \qquad (f^{-1})x = (fx)^{-1}.$$

*The scalar multiplication $K \otimes [A, B] \to [A, B]$ is given by $(rf)(x) = r(fx)$.*

**Proof.** The proof rests upon KMods arising as an example of a commutative algebraic theory. Note that the commutativity of the theory is essential, eg the category of groups is not closed. See Borceux 2 pp172 or MacLane pp180 [3,24]. □

### *3.1 Enrichment over* KMods

We have represented $K$-algebras as monoids in the category of $K$-modules. Recall that computing with groups amounted to turning them into categories and we do the same for $K$-algebras. Indeed, this is a general construction mapping monoids in a monoidal category to categories. In fact the category we get is enriched over the ambient category. Enriched categories are categories whose hom is not a set but an object of some other category. We give a basic description and refer the reader to [3,20] for more details.

If $V$ is a monoidal category, a $V$-category $C$ consists of a class of objects $|C|$ and, for each pair of objects a hom $C(A, B)$ which is an object of $V$. In addition,

- Identities are given by requiring for each object $A \in |C|$, a map $1_A : I \to C(A, A)$ in $V$
- Composition is given by requiring for each triple of objects $A, B, C \in |C|$ a map $m_{A,B,C} : C(A, B) \otimes C(B, C) \to C(A, C)$ in $V$.

The maps $1_A$ are required to be identities for composition and composition is required to be associative. $V$-functors are defined similarly. If $V =$ Sets we get the usual definition of a category. If $V =$ Pre we get ordered categories while if $V =$ Cat we get 2-categories. In this paper, we are interested $K$-algebras which are monoids in KMods which will then turn into one object, KMods-enriched categories, or $K$-categories for short.

**Lemma 3.7** *Let $X$ be a monoid in a monoidal category $V$. Then $X$ defines a one-object $V$-enriched category. If $V$ is closed, then every $X$-action $a : X \otimes Y \to Y$ defines a $V$-functor $a : X \to V$.*

**Proof.** The proof is the obvious generalisation the presentation in Section 2 of groups as categories. Let $X$ be a monoid in $V$. definitionine the $V$-category $X$ to have one object and hom $X(*,*) = X$. The monoid structure of $X$ ensures that $X$ is then a $V$-category.

For the second part of the lemma, note first that the assumption that $V$ is closed is required to ensure that $V$ is a $V$-category. To define a $V$-functor $a : X \to V$ we must map the single object of $X$ to an object of $V$ and the obvious choice is $a(*) = X$. The rest of the proof is then standard with the action axioms translating precisely into the axioms of a functor. □

Although computing with $K$-algebras requires only one object $K$-categories, as we remarked before this is overly limiting and in general we want to compute with finitely presented $K$-categories. These are a synthesis of the usual presentation of categories based upon the path functor and the presentations of $K$-modules given above. The free $K$-category on the graph $\Delta$ is the category $P_K\Delta$ whose objects are the objects of $\Delta$ and whose homs are $P_K\Delta(A,B) = F_M(P\Delta)(A,B)$ where $A, B \in \mathrm{Ob}\Delta$, $P : \mathsf{Gph} \to \mathsf{Cat}$ is the path functor and $F_M$ is the free module functor. More concretely, $P_K\Delta(A,B)$ consists of all polynomials of the form $p = k_1w_1 + \cdots + k_nw_n$ where $k_1, \ldots, k_n \in K$ and $w_1, \ldots, w_n \in P\Delta(A,B)$. As usual, finitely presented $K$-categories are a quotient of a free $K$-category by a set of relations [26].

**Definition 3.8 ($K$-category presentation)**
A $K$-category presentation cosists of a finite directed graph $\Delta$ and a set of elements $R$ of the free $K$-category $P_K\Delta$ on $\Delta$. We may write the presentation as $cat_K\langle\Delta|R\rangle$. The category presented has the same objects as $\Delta$ and its arrows are the equivalence classes of $\mathrm{Arr}P_K\Delta$ under the relation generated by $R$; i.e. $=_R$ which is defined by

$$f =_R h \text{ if and only if } f =_R h + k_1p_1r_1q_1 + \cdots + k_np_nr_nq_n$$

where $r_i \in R, k_i \in K$ and $p_i, q_i$ are arrows of $P_K\Delta$ whose composites are defined.

## 4 Computing Kan Extensions over $K$-modules

We now formally define those Kan extensions to which we will compute with Gröbner basis techniques.

**Definition 4.1 (Kan Presentations)**
A Kan presentation for $K$-categories is a quintuple $\mathcal{P} := kan\langle\Gamma, \Delta, R, M, F\rangle$ where

i) $\Gamma$ and $\Delta$ are (directed) finite graphs;

ii) $M : \Gamma \to \mathsf{KMods}$ and $F : \Gamma \to P_K\Delta$ are graph morphisms; For every object $A \in \Gamma$, $M(A)$ is presented by $mod_K\langle \Lambda_A, R_A\rangle$.

iii) $R$ is a finite set of relations on $P_K\Delta$.

$kan\langle \Gamma, \Delta, R, M, F\rangle$ presents the Kan extension of $(M', F')$ where $M' : \mathsf{A} \to \mathsf{KMods}$ and $F' : \mathsf{A} \to \mathsf{B}$ if $A$ is the free $K$-category on $\Gamma$, $cat_K\langle \Delta, R\rangle$ is a $K$-category presentation for $\mathsf{B}$, $M$ induces $M'$ and $F$ induces $F'$.
As we have seen, a Kan extension $\mathsf{Lan}_F M$ can be computed pointwise by the coend formula

$$(\mathsf{Lan}_F M)B = \int^{A \in \mathrm{Ob}\mathsf{A}} MA \otimes \mathsf{B}(FA, B)$$

In our setting, both $MA$ and $\mathsf{B}(FA, B)$ are finitely presented $K$-modules. The tensor product of two finitely presented $K$-modules is finitely presented and so we consider the free module over the generators as a basic data structure over which three forms of equation, and later rewrite rule, will occur: firstly equations pertaining to the presentation of $MA$, secondly the relations defining $\mathsf{B}$ and finally the equations defining the actual Kan extension. Thus, for each $B \in \mathrm{Ob}\mathsf{B}$ and $A \in \mathrm{Ob}\mathsf{B}$ define $T_{A,B}$ to be

$$T_{A,B} = F_M(\Lambda_A \times P\Delta(FA, B)).$$

Further define $T_B := \bigsqcup_{A \in \mathrm{Ob}\Gamma} T_{A,B}$ and $T := \bigsqcup_{B \in \mathrm{Ob}\Delta} T_B$. Alternatively, $T_{A,B}$ is the set of all elements $k_1\lambda_1 p_1 + \cdots + k_n\lambda_n p_n$ where $k_1, \ldots, k_n \in K$, $\lambda_1, \ldots, \lambda_n \in \Lambda_A$ and $p_1, \ldots, p_n \in P\Delta(FA, B)$. We will refer to the elements $\lambda p$ as *the terms of* $T$, whilst noting that not all formal sums of these elements are defined in $T$. In addition, let $\sigma, \tau : T \to \mathrm{Ob}\Delta$ be defined by $\sigma(t) := F(A)$ and $\tau(t) := B$ for $t \in T_{A,B}$. These are, in effect, source and target functions.

As mentioned above, to construct the Kan extension $\mathsf{Lan}_F M$ we need to combine the relations for the category $\mathsf{B}$ with the relations in the $K$-module presentations defining $M$ and relations to force there to be only one natural transformation $\varepsilon$ from $M$ to $\mathsf{Lan}_F M \circ F$. Given three sets of relations; $Q_T \subseteq T$, $Q_M \subseteq \bigsqcup_{A \in \mathrm{Ob}\Gamma} F_M[\Lambda_A]$ and $Q_R \subseteq \mathrm{Arr}P_K\Delta$, define $Q = Q_T + Q_R + Q_M$.
We compute $\leftrightarrow_Q$ in $T$ by embedding $T$ in the free polynomial ring $T_+ = K[(\Lambda + \mathrm{Arr}\Delta)^*]$ where $\Lambda = \sqcup_{A \in \mathrm{Ob}\Gamma}\Lambda_A$. We choose an admissible well-ordering $>$ on the monoid $(\Lambda + \mathrm{Arr}\Delta)^*$, i.e. a well-ordering on the elements of $(\Lambda + \mathrm{Arr}\Delta)^*$ such that if $u_1 > u_2$ then $tu_1v > tu_2v$ for $t, v \in (\Lambda + \mathrm{Arr}\Delta)^*$. Note that this ordering is stronger than we need, but it is computationally practical as well as more easily defined.

The *leading term* of any polynomial $q = k_1u_1 + \cdots + k_nu_n$ of $T_+$ is defined to be the element $\mathtt{LT}(q) = u_i$ in $(\Lambda + \mathrm{Arr}\Delta)^*$ which is largest with respect to the given ordering. The coefficient of $u_i$ in $q$ is $k_i$. We note that for polynomials

generating an ideal in $T_+$ we can divide them all by the coefficient of their leading terms; so we can assume that the coefficient of the leading term is 1.

**Definition 4.2** The **reduction relation** $\rightarrow_Q$ on $T$ is defined by

$$f \rightarrow_Q f - kuqv$$

when $u(\mathtt{LT}(q))v$ occurs in $f$ with coefficient $k \in K$ where either

i) $q$ is in $Q_T$ or $Q_M$, $u = 1$ and $v \in P\Delta$ with $\tau(q) = src(v)$,

ii) $q \in Q_R$, $u \in T$ and $v \in P\Delta$ and $\tau(u) = src(q)$ and $tgt(q) = src(v)$.

The reflexive, symmetric and transitive closure of $\rightarrow_Q$ is denoted $\overset{*}{\leftrightarrow}_Q$. The equivalence classes of $T$ under $\overset{*}{\leftrightarrow}_Q$ are denoted $[t]_Q$. Note that if $t \in T_B$ and $t_1 \rightarrow_Q t_2$, then $t_2 \in T_B$ and also that the relation preserves addition and scalar multiplication. This gives us the following result.

**Lemma 4.3** *For $Q$, $T_B$ as above, the restriction of reduction relation $\rightarrow_Q$ to the module $T_B$, is well defined, i.e. if $t \in T_B$ then $[t]_Q \subseteq T_B$ and $T_B/\overset{*}{\leftrightarrow}_Q$ is a $K$-module.*

We now prove that the reduction relation generated by $Q$ on the set of terms $T$, that we have described, captures the Kan extension.

**Theorem 4.4**
*Let $\mathcal{P} := kan\langle \Gamma, \Delta, R, M, F\rangle$ be a presentation of a left Kan extension over* $\mathsf{KMods}$. *Define*

*i)* $Q_T := \lambda \cdot F(a) - M(a)(\lambda)$ *for all* $\lambda \in \Lambda_{M(src(a))}$, *for all* $a \in \mathrm{Arr}\Gamma$,

*ii)* $Q_M := \bigsqcup_{A \in \mathrm{Ob}\Gamma} R_A$,

*iii)* $Q_R := R$.

*Then the left Kan extension presented by $\mathcal{P}$ is $(\mathsf{Lan}_F M, \varepsilon)$ where*

*i)* $\mathsf{Lan}_F M(B)$ *is the $K$-module $T_B/\overset{*}{\leftrightarrow}_Q$,*

*ii)* $\mathsf{Lan}_F M(b)$ *is defined by* $\mathsf{Lan}_F M(b)[t]_Q := [tb]_Q$ *for $b$ in* $\mathrm{Arr}\Delta$,

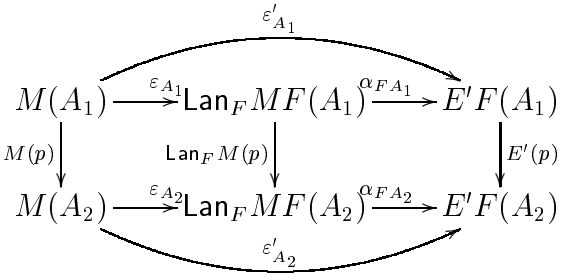
*iii)* $\varepsilon : M \rightarrow \mathsf{Lan}_F M \circ F$ *is given by* $\varepsilon_A(\lambda) := [\lambda 1_{FA}]_Q$.

**Proof.** It is required to verify that $\mathsf{Lan}_F M$, as defined above, is a well-defined $K$-functor. This can be deduced from the fact that the congruence preserves addition, scalar multiplication and right-multiplication.

To verify that $\varepsilon$ is a natural transformation of $K$-functors is straightforward. Let $a : A_1 \rightarrow A_2$ in $\mathsf{A}$. Then let $\mu$ be an element of $M(A_1)$. Now by definition $(\mathsf{Lan}_F M)(Fa)(\varepsilon_{A_1}(\mu)) = (\mathsf{Lan}_F M)(Fa)([\mu 1_{FA_1}]_Q) = [\mu 1_{FA_1} p]_Q = [\mu p]_Q$ where $[p]_R = Fa$, and $\varepsilon_{A_2}(Ma(\mu)) = \varepsilon_{A_2}([\mu p]_Q) = [\mu p 1_{FA_2}]_Q = [\mu p]$ so $\mathsf{Lan}_F MFa \circ \varepsilon_{A_1} = \varepsilon_{A_2} \circ Ma$ for all arrows $a : A_1 \rightarrow A_2$ in $\mathsf{A}$.

The universal property completes the proof. Let $(E', \varepsilon')$ be a pair such that

$E'$ is a $K$-functor from $\mathsf{B} \to \mathsf{KMods}$ and $\varepsilon'$ is a natural transformation of $K$-functors. Any natural transformation of $K$-functors $\alpha : \mathsf{Lan}_F M \to E'$ such that $\varepsilon \circ \alpha = \varepsilon'$ must satisfy the commutative diagram:




which allows the unique definition $\alpha(p) = E'(p)(\varepsilon'(1_A))$. Hence $(\mathsf{Lan}_F M, \varepsilon)$ is universal. □

By making the following observations about $T$ and $M$, we can apply the standard methods of noncommutative Gröbner bases [11,27] to obtain a set of polynomials $Q'$ so that $\overset{*}{\leftrightarrow}_Q$ coincides with $\overset{*}{\leftrightarrow}_{Q'}$ and $\overset{*}{\rightarrow}_{Q'}$ is complete. Recall $T$ is a submodule of the $K$-module $T_+ = K[(\Lambda + \mathrm{Arr}\Delta)^*]$ which is a free polynomial ring. Secondly, we can define $\rightarrow_{Q+}$ on $K[(\Lambda + \mathrm{Arr}\Delta)^*]$ by $f \rightarrow_{Q+} f - kuqv$ for all $k \in K$ and $u, v \in (\Lambda + \mathrm{Arr}\Delta)^*$ such that $uLT(q)v$ occurs in $f$ with coefficient $k$. Observe that the restriction to $T$ of $\rightarrow_{Q+}$ coincides with our original relation $\rightarrow_Q$, and that if $\rightarrow_{Q+}$ is complete on $T_+$, then $\rightarrow_Q$ is complete on $T$. Recall that we can use Buchberger's Algorithm to try to compute a Gröbner basis for $Q$ in $T_+$, and thus find $Q'$ such that $\rightarrow_{Q'+}$ is complete. Furthermore, no computation during the execution of Buchberger's Algorithm for $Q$ will yield a polynomial which is not a member of the submodule $T$ of $T_+$. Therefore, if Buchberger's Algorithm applied to $Q$ in $T_+$ terminates, giving a Gröbner basis $Q'$, then $Q'$ is a subset of $T + \mathrm{Arr}P_K\Delta$ so $\rightarrow_{Q'}$ is well-defined and convergent on $T$. This gives us the following result.

**Corollary 4.5 (Application of Gröbner Basis Theory)**
*Gröbner bases can be used to compute left Kan extensions of the above type.*

**Outline Proof** Given $Q$, we can use the noncommutative version of Buchberger's Algorithm in the usual way [27] to attempt to compute a Gröbner basis in $T_+$. Suppose $Q'$ is a Gröbner basis for $Q$ in $T_+$, then $Q$ generates a convergent reduction relation on $T$ and the Kan extension is given by the following:

i) $\mathsf{Lan}_F M(B) := \mathtt{irr}_{Q'}(T_B)$,

ii) $\mathsf{Lan}_F M(b) : t \mapsto \mathtt{irr}_{Q'}(tp)$, for $t$ in $\mathsf{Lan}_F M(B)$, $p$ in $P_K\Delta$ such that $\theta(p) = b$ and $src(p) = \tau(t)$,

iii) $\varepsilon_A(\lambda) := \lambda$.

where $\mathtt{irr}_{Q'}(t)$ is the irreducible result of repeated reduction of $t$ by $\rightarrow_{Q'}$ and $\mathtt{irr}_{Q'}(T_B)$ is the set of all terms in $T_B$ which are irreducible with respect to $\overset{*}{\rightarrow}_Q$. □

## 5 Examples

We complete the paper by showing how our computational Kan extensions can be used to solve a number of problems of importance to researchers in mathematics. The first concerns presentations of $K$-algebras, which recall we represent as 1-object $K$-categories. Thus, a $K$-algebra presentation is a $K$-category presentation whose graph has one vertex. Given a $K$-algebra presentation $cat_K\langle\Delta, R\rangle$ of $B$ and two elements $\alpha, \beta$ of the free $K$-algebra on $\Delta$, is it the case that $[\alpha]_B = [\beta]_B$?

**Example 5.1 (Algebra presentation)** *Let* $\mathsf{A}$ *be the trivial* $K$*-category and let* $\mathsf{B}$ *be the* $K$*-category* $B$*. Let* $F : \mathsf{A} \to \mathsf{B}$ *be unique functor and let* $M : \mathsf{A} \to \mathsf{KMods}$ *map the object of* $\mathsf{A}$ *to the* $K$*-module* $K$*. Then computing* $\mathsf{Lan}_F M$ *is equivalent to computing the algebra presented by* $K[X^*]/=_R$ *where* $X$ *is the set of edges in* $\Delta$*. In detail, the functor* $\mathsf{Lan}_F M$*, when applied to the object of* $B$*, gives a* $K$*-module isomorphic to the algebra (quotient of a monoid ring)* $K[X^*]/=_R$*. On arrows, the functor gives us automorphisms of the module, which define a right action of the module on itself* $\mathsf{Lan}_F M(b)p = pb$*. This gives the multiplication for the algebra. The natural transformation* $\varepsilon$ *picks out the monomial which is the multiplicative identity of the algebra i.e.* $\varepsilon_A(1) = [1_X]_R$*.*

The construction of the quiver algebra over a graph is a fundamental construction in representation theory. This can be modelled as follows. Note that this example requires Kan extensions of functors whose domain/codomain have more than one object.

**Example 5.2 (Path Algebra)** *Let* $B$ *be a path algebra, i.e. the free* $K$*-category over a graph* $\Delta$*. Let* $\Gamma$ *have the same vertices as* $\Delta$ *but an empty set of edges and* $F$ *the inclusion. Let* $M : \mathsf{A} \to \mathsf{KMods}$ *map each object of* $\mathsf{A}$ *to the* $K$*-module* $K$*. Then the Kan extension* $\mathsf{Lan}_F M$ *defines the quiver algebra over* $B$*.*

The free module over an algebra may be computed as follows.

**Example 5.3 (Free module over an algebra)** *Let* $\mathsf{A}$ *be the trivial* $K$*-category and let* $M$ *map it to a free* $K$*-module on a set of generators* $\Lambda$*. Then let* $\mathsf{B}$ *be* $K$*-algebras regarded as a one object* $K$*-category. Let* $F$ *be the functor from* $\mathsf{A}$ *to* $\mathsf{B}$*. Then the Kan extension of* $M$ *along* $F$ *gives the free* $\mathsf{B}$*-module on* $\Lambda$*.*

The coset construction is fundamental throughout algebra. We have already seen it in the context of group theory and the following example constructs cosets of $K$-algebras. Note how the construction is only changed by the enrichment with the unit of the monoidal structure on $\mathsf{Sets}$ namely 1 being replaced by the unit of the monoidal structure on $\mathsf{KMods}$, namely $K$.

**Example 5.4 (Cosets of a sub-algebra in a $K$-algebra)** *Let* $\mathsf{A}$ *and* $\mathsf{B}$ *be $K$-algebras, regarded as one object $K$-categories enriched over* $\mathsf{KMods}$*. Let $F$ be inclusion of* $\mathsf{A}$ *into* $\mathsf{B}$*. Then let $M$ map the object of* $\mathsf{A}$ *the $K$-module $K$, and all the arrows to the identity $K$-module morphism. Then* $\mathsf{Lan}_F M$ *maps the single object of $B$ to the module of cosets of* $\mathsf{A}$ *in* $\mathsf{B}$*.*

As is well known, colimits in categories are Kan extensions along a functor into the terminal category. Enriching this construction allows us to calculate colimits of $K$-modules. For simplicity, we tackle coproducts i.e. direct sums and note that this example requires Kan extensions of functors whose domain is not a single object $K$-category. See MacLane for details [24]

**Example 5.5 (Coproducts/Direct sums of $K$-modules)** *Let* $\mathsf{A}$ *be a discrete category with $n$ objects, and let* $\mathsf{B}$ *be the trivial $K$-category. Let $F$ be the unique functor from* $\mathsf{A}$ *to* $\mathsf{B}$ *and let $M$ map the objects of* $\mathsf{A}$ *to $K$-modules $M_1, \ldots, M_n$. Then the Kan extension* $\mathsf{Lan}_F M$ *calculates the coproduct/direct sum $M_1 + .. + M_n$ of $K$-modules.*

Our last example is that of induced modules

**Example 5.6 (Induced Modules)** *Let* $\mathsf{A}$ *and* $\mathsf{B}$ *be $K$-algebras represented as one object $K$-categories and let $F : \mathsf{A} \to \mathsf{B}$. Let $M$ map the object of* $\mathsf{A}$ *to a $K$-module $M(A)$ and the arrows to endomorphisms of the $K$-module; then $M$ defines a right $A$-module. The Kan extension of $M$ along $F$ gives the right $B$-module induced by $F$ on $M$. In detail,* $\mathsf{Lan}_F M(B)$ *is a $K$-module and* $\mathsf{Lan}_F M(b) : \mathsf{Lan}_F M(B) \to \mathsf{Lan}_F M(B)$ *gives a right action of the elements of the $K$-algebra* $\mathsf{B}$ *on* $\mathsf{Lan}_F MB$*. The universal property of the natural transformation $\varepsilon_A : M(A) \to \mathsf{Lan}_F M(B)$ confirms that* $\mathsf{Lan}_F M(B)$ *is the induced module.*

## 6 Further work

We have shown that category theory, in particular Kan extensions, provides an expressive meta-language for describing various quotients involving $K$-modules, $K$-algebras and $K$-categories. We also showed how Gröbner bases techniques could be applied to compute these Kan extensions, thereby opening the way to their formal implementation as part of a computer algebra package. We feel that the unification of quotients at the level of Kan extensions, and the unification of computation in different algebraic structures by a change of enrichment, is an elegant theoretical insight which will also significantly improve the quality and reliability of the software.

Future work lies in two directions concerning the implementation of these algorithms and their further theoretical development. The current implementation is a collection of functions written in GAP, which need further development. Interfacing the functions for the Kan extensions with a faster Gröbner basis

program is one possibility for increasing efficiency. We are discussing our algorithms with algebraists in order to get more examples to test, in particular we would like to investigate further the possibility for computing tensor products as Kan extensions. On the theoretical side, there are a number of enhancements we have in mind: using modularity results in rewriting to integrate the different notions of rewriting used; using automata theory to give language-theoretic descriptions of the normal forms of computation; and optimising the Knuth Bendix process for obtaining complete rewrite systems. Overall there is certainly much more to do.

## References


[1] F. Baader and T. Nipkow: *Term Rewriting and All That* Cambridge University Press (1998)

[2] M. Barr and C. Wells : Toposes, Triples and Theories, *Springer, Grundlehren der Mathematischen Wissenschaften Series no.278* (1985)

[3] F. Borceux: Handbook of categorical algebra , Encyclopedia of mathematics and its applications, Vol. 50, p. xv,345p, Basic category theory/ Francis Borceux Cambridge: Cambridge University Press, 1994.

[4] R. Brown and A. Heyworth: *Using Rewrite Systems to Compute Kan Extensions and Induced Actions of Categories* Journal of Symbolic Computation, vol.29 p5-31 (2000)

[5] M. R. Bush, M. Leeming and R. F. C. Walters: *Computing Left Kan Extensions* Journal of Symbolic Computation, vol.11 p11-20 (1997)

[6] S. Carmody, M. Leeming and R. F. C. Walters: *The Todd-Coxeter Procedure and Left Kan Extensions* Journal of Symbolic Computation, 19 p459-488 (1995)

[7] S. Carmody and R. F. C. Walters: *Computing Quotients of Actions on a Free Category* in A. Carboni, M. C. Pedicchio, G. Rosolini (eds), Category Theory, Proceedings of the Int. Conf. Como, Italy 22-28 July 1990 vol.1144 of Lectures Notes in Mathematics, p131-56, Springer-Verlag (2000)

[8] E. J. Dubuc : Kan Extensions in Enriched Category Theory *Springer Lecture Notes in Mathematics, vol.245* (1970)

[9] D. B. A. Epstein with J. W. Cannon et al: *Word processing in groups* Jones and Bartlett, Boston (1992)

[10] M. Fleming, R. Gunther and R. Rosebrugh: *User Guide for the Categories Database and Manual* anonymous `ftp://sun1.mta.ca/pub/papers/rosebrugh/catdsalg.dvi,tex` and `/catuser.dvi,tex` (1996)

[11] R. Fröberg: *An Introduction to Gröbner Bases* John Wiley and Sons (1997)

[12] The GAP Group, 'GAP – Groups, Algorithms, and Programming, Version 4', Aachen, St Andrews, (1998) (`http://www-gap.dcs.st-and.ac.uk/~gap`).

[13] N. Ghani and C. Lüth: *Monads and Modular Rewriting* Proc. Category Theory and Computer Science 1997, Lecture Notes in Computer Science, Springer-Verlag 1290 p69-86 (1997)

[14] J. Goguen and T. Winkler : Introducing obj3, *Technical Report SRI-CSL-88-8, SRI* (1993)

[15] A. Heyworth: *Groebner Basis Techniques for Computing Actions of K-Categories*, Proceedings of Category Theory 2000 16-22 July, Como, Italy, 105-113 (2000)

[16] A. Heyworth and J. Snellman: *Gröbner Bases for Modules University of Leicester, preprint* (2001)

[17] D. F. Holt : Knuth-Bendix in Monoids, and Automatic Groups, *Univ. of Warwick* (1996).

[18] C. B. Jay : Modelling Reduction in Confluent Categories *Applications of Categories in Computer Science p143-62 C.U.P.* (1992)

[19] D. M. Kan: Adjoint Functors, *Trans. Am. Math. Soc. 87 p294-329* (1958)

[20] G. M. Kelly : Basic Concepts of Enriched Category Theory, *London Math Soc. C.U.P. no.64* (1982)

[21] F. W. Lawvere : Functorial Semantics of Algebraic Theories *Proc. Nat. Academ. of Sciences vol.50 p869-872* (1963)

[22] F. E. J. Linton : Some Aspects of Equational Categories *Proc. of Conference on Categorical Algebra p84-94 Springer-Verlag* (1965)

[23] S. A. Linton : On Vector Enumeration, *Linear Algebra and its Applications 192 p235-48* (1993)

[24] S. Mac Lane : Categories for the Working Mathematician, *Springer-Verlag*, (1971)

[25] P. -A. Mellies : A Stability Theorem in Rewriting Theory *LICS: IEEE Symposium on Logic in Computer Science*, (1998)

[26] B. Mitchell: *Rings with Several Objects* Advances in Mathematics vol.8 no.1 (1972)

[27] T. Mora, Seven Variations on Standard Bases, *Preprint, University of Genova,* `http://www.disi.unige.it/ftp/person/MoraF/7Varietions.tar.gz`, 1988.

[28] T. Murata : Petri nets: Properties, Analysis and Applications, *Proceedings of the IEEE, vol.77 no.4* (1989)

[29] D. E. Rydeheard and J. G. Stell : Foundations of Equational Deduction – A Categorical Treatment of Equational Proofs and Unification Algorithms, *Proc. CTCS'87 LNCS 283 114-139 Springer* (1987)

[30] R. A. G. Seely : Modelling Computations: A 2-categorical Framework, Symposium on Logic in Computer Science p65-71 IEEE Computer Society Press (1987)