Computing morphisms between modules over nonunital rings using patterns in graphs

Leandro Marin

Department of Applied Mathematics. Faculty of Computer Science
University of Murcia. 30071 Murcia (Spain)
leandro@um.es

Abstract

When we study the structure of nonunital rings, they appear some categories of modules over them, one of those is the category of firm modules, i.e. modules M such that the multiplication $R \otimes_R M \to M$ is an isomorphism. The ring itself use not to be firm, but there is always a special family of firm modules that generate all the others. Those special modules are built from infinite graphs that describe the module structure.
Similar patterns in different graphs generate module homomorphisms. Those are the only possible morphisms in some important cases. In order to represent those infinite graphs on a computer, we need also that some basic patterns are repeated. This family is dense and very useful in order to compute morphisms. We represent them and give an algorithm that says when two such a graph represent isomorphic modules.

1. Algebraic Problem

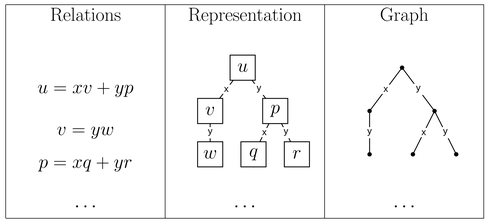
Let X be a set, $\mathbb{Z}\langle X\rangle$ the free noncommutative algebra over the elements of X,

$$R = \mathbb{Z}\langle X\rangle_0 = \sum_{x\in X} x\mathbb{Z}\langle X\rangle \subseteq \mathbb{Z}\langle X\rangle$$

basic family of nonunital rings[1]. In this case, the firm module are exactly the R-modules M such that

$$M^{(X)} \to M \quad (m_x) \mapsto \sum_{x\in X} x m_x$$

is an isomorphism. An element in M has a representation (in this case it is unique) as a sum of elements in M multiplied by elements in X. Those representations can be described with an infinite graph in the following way:



With the basic graph K we can define a module $\langle\langle K\rangle\rangle$ such that all these relations are in fact a module homomorphism $f : \langle\langle K\rangle\rangle \to M$. The modules $\langle\langle K\rangle\rangle$ with K any of those graphs, are generators of the category of firm modules.
There are three basic relations between graphs

- K exactly a subset of L. This induces a canonical epimorphism

$$\langle\langle L\rangle\rangle \to \langle\langle K\rangle\rangle$$

- Let $\Delta_x K$ the (possibly empty) graph that starts moving from the starting vertex of K in the direction given by the edge indexed by x. In this case there is a canonical monomorphism

$$\langle\langle \Delta_x K\rangle\rangle \to \langle\langle K\rangle\rangle$$

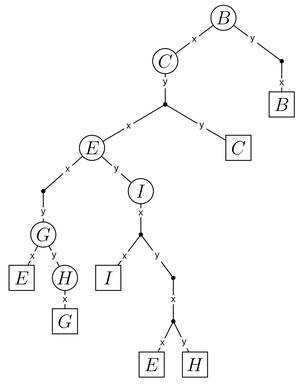
- Let $\nabla_x K$ be the graph generated by adding an edge indexed by x at the starting vertex of K. Here we have a canonical isomorphism

$$\langle\langle \nabla_x K\rangle\rangle \to \langle\langle K\rangle\rangle$$

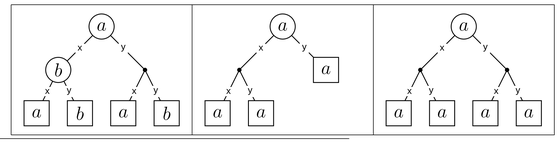
We can compose these basic morphisms and make linear combinations of them. In the case $R = \mathbb{Z}\langle X\rangle_0$ these combinations are the only possible morphisms between the modules $\langle\langle K\rangle\rangle$.

2. Making it Computable

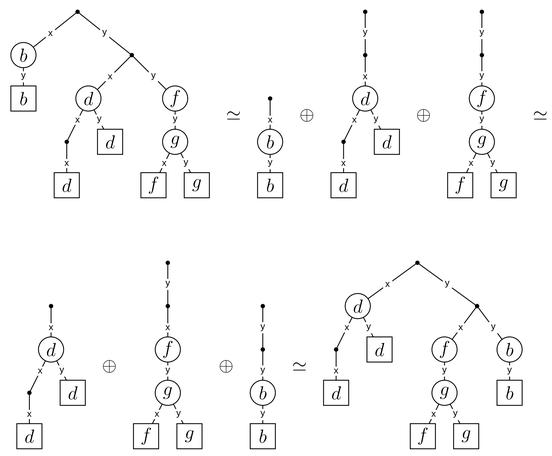
We need to consider only finite sets X. In fact we will consider in the following $X = \{x, y\}$. An infinite graph should be represented by a finite amount of information. Therefore we need that following any branch of the tree, there should be a vertex that have exactly the same structure that a previously considered vertex. We represent this kind of relations naming the vertex that is going to be repeted with a name enclose it with a circle. When the vertex is referenced we use an square and stop drawing. A vertex that is neither referenced nor a reference will be denoted with a dot. A nontrivial example could have the following structure:



The representation given in this way is not unique. The following are representations of the same graph (the complete graph in two variables):



In this case, there seems to be a minimal representation of the graph, but in general it is not trivial to say if there is a minimal representation. In fact it is possible to give several ones depending on the properties we are looking for. The most important is that **Algorithm 1** says when one graph is exacly contained in another one, and it does not depend on the representation. Using the algorithm in both directions, it is possible to say when two representations are exactly from the same graph.
The isomorphism problem is a bit more complex. If we split a graph in a finite number of subgraphs, the module generated by the complete graph is the direct sum of the modules generated by the subgraphs. Using the isomorphisms between $\langle\langle \nabla_x K\rangle\rangle$ and $\langle\langle K\rangle\rangle$ we have for example that



It is possible to prove that in our case, $\langle\langle K\rangle\rangle$ and $\langle\langle L\rangle\rangle$ are isomorhic in the category of firm modules if and only if this kind of operations can be done. In case they are isomorphic there are many isomorphisms. You only have to decide which branches are going to be possitive and which are going to be negative.

3. Algorithms

Algorithm 1

Suppose you have a representation of the graphs such that every vertex k has a memory position and an array indexed by the elements of X with the memory positions of the child vertex in the direction given by $x \in X$. If this child does not exist, the memory position is **NULL**.
Let K and L be representations of graphs. Let k and l be the memory positions of the starting vertex of K and L, and P a set of pairs of memory positions, empty at the beginning. The graph K is contained in L if and only if $\mathbf{LEQ}(k, l, P)$ returns **true** and not if it returns **false**.

1. if $k = \mathbf{NULL}$ or (k, l) is in P return **true**.
2. push (k, l) in P and return **true** if for every $x \in X$, $\mathbf{LEQ}(k[x], l[x], P)$ returns **true** and **false** if any of those callings return **false**. ($k[x]$ is the memory position of the child x branch).

Algorithm 2

Let K be a graph and S be a set of types of graphs. Then $\mathbf{WR}(K, S)$ returns **true** if K can be written using only copies of graphs in S and **false** if this cannot be done. Let k be the memory position of the starting vertex of K and T an empty stack. $\mathbf{WR}(K, S) := \mathbf{WR}(k, S, T)$.

1. if $k = \mathbf{NULL}$ or the type of the vertex over k is in S return **true**.
2. if k is in T return **false**.
3. push k in T and return **true** if $\mathbf{WR}(k[x], S, T)$ is **true** for all $x \in X$ and **false** if it fails for any x.

Algorithm 3

Let K and L be representations of two graphs; $\mathbf{ISO}(K, L)$ says if K and L are isomorphic. First we have to expand the branches until every reference points to a parent vertex in the same branch (not in other branches). Consider the set S of vertices that are referenced in K and L and have under it the same graph.
If $\mathbf{WR}(K, S)$ or $\mathbf{WR}(L, S)$ are not $\mathbf{true}$ return **false**. If not, take a subset $S' \subseteq S$ such that it is minimal with this property. Write all the types in S' with references to types in S' and compute how many copies of each type do you need to write K and L.
Write linear diophantine equations with every type $a \in S'$ saying how many new copies of $a, b, c, \dots \in S'$ appear when you expand a. With all this information you get a system of linear diophantine equations with the number of expansions of every type in S' do you need in K and L for having the same number of copies in K and L. If the system can be solved, return **true** and if not return **false**.

4. Further Research

We are following two lines. To make a MAGMA implementation of these structures and also to consider relations between elements in X, for example $xy = yx$. This kind of relations introduce a lot of complexity with many new possible morphisms.

5. Bibliography

Although there are some references about nonunital rings, the most closely related with this work are the ones in which the firm modules arising from graphs are defined and extensively used:

1. Marin,L.: *The construction of a generator for* R-DMod. Lecture notes in Pure and Applied Mathematics (210) ; Marcel-Dekker, pp. 287–296 (2000)
2. Garcia, J.L.; Marin, L.: *Watts Theorems for associative rings*. Communications in Algebra 29(12), pp. 5799–5834 (2001)

[1] $\mathbb{Z}$ can be replaced by any unital commutative ring or field

MAP 2006. Castro Urdiales (Spain) 09-13th January, 2006. Partially supported by BFM-2003-07569-C02-01.