MatrixAlgebras Package

Peter A. Brooksbank

Bucknell University
pbrooksb@bucknell.edu

Joshua Maglione

Universität Bielefeld
jmaglione@math.uni-bielefeld.de

James B. Wilson

Colorado State University
james.wilson@colostate.edu

Version 0.1
January 27, 2019

Contents

CHAPTER 1

Introduction

This documentation describes Magma functions for computation with various types of algebras of matrices, notably associative algebras, Lie algebras, and $*$-algebras (i.e. algebras with involution). Some of these functions, such as those that aim to access the structure of algebras, supplement existing Magma machinery. Others, such as normalizer and conjugacy functions, are designed with specific applications in mind, but seem likely to be of interest as standalone functions.

Citing MatrixAlgebras. To cite the MatrixAlgebras package, please use the following

Peter A. Brooksbank, Joshua Maglione, James B. Wilson, *MatrixAlgebras*, version 0.1, GitHub, 2018. https://github.com/galois60/MatrixAlgebras.

For AMSRefs:

```
\bib{MatrixAlgebras}{misc}{
   author={Brooksbank, Peter A.},
   author={Maglione, Joshua},
   author={Wilson, James B.},
   title={MatrixAlgebras},
   publisher={GitHub},
   year={2018},
   edition={version 0.1},
   note={\texttt{https://github.com/galois60/MatrixAlgebras}},
}
```

1.1. Overview

1.2. Version

CHAPTER 2

Main chapter 1

Here is sample documentation to demonstrate the latex functions.

2.1. Black-box tensors

A user can specify a tensor by a black-box function that evaluates the required multilinear map.

```
Tensor(S, F) : SeqEnum, UserProgram -> TenSpcElt, List
Tensor(S, F) : List, UserProgram -> TenSpcElt, List
Tensor(S, F, Cat) : SeqEnum, UserProgram, TenCat -> TenSpcElt, List
Tensor(S, F, Cat) : List, UserProgram, TenCat -> TenSpcElt, List
```

Returns a tensor t and a list of maps from the given frame into vector spaces of the returned frame. Note that t is a tensor over vector spaces—essentially forgetting all other structure. The last entry of `S` is assumed to be the codomain of the multilinear map. The user-defined function F should take as input a tuple of elements of the domain and return an element of the codomain. If no tensor category is provided, the Albert's homotopism category is used.

Example 2.1. `BBTensorsFrame`

We demonstrate the black-box constructions by first constructing the dot product $\cdot : \mathbb{Q}^4 \times \mathbb{Q}^4 \rightarrowtail \mathbb{Q}$. The function used to evaluate our black-box tensor, `Dot`, must take exactly one argument. The argument will be a `Tup`, an element of the Cartesian product $U_v \times \cdots \times U_1$. Note that `x[i]` is the ith entry in the tuple and not the ith coordinate.

```
> Q := Rationals();
> U := VectorSpace(Q, 4);
> V := VectorSpace(Q, 4);
> W := VectorSpace(Q, 1);  // Vector space, not the field Q
> Dot := func< x | x[1]*Matrix(4, 1, Eltseq(x[2])) >;
```

Now we will construct the tensor from the data above. The first object returned is the tensor, and the second is a list of maps, mapping the given frame into the vector space frame. In this example, since the given frame consists of vector spaces, these maps are trivial. Note that the list of maps are not needed to work with the given tensor, we will demonstrate this later.

```
> Tensor([U, V, W], Dot);
Tensor of valence 3, U2 x U1 >-> U0
U2 : Full Vector space of degree 4 over Rational Field
U1 : Full Vector space of degree 4 over Rational Field
U0 : Full Vector space of degree 1 over Rational Field
[*
    Mapping from: ModTupFld: U to ModTupFld: U given by a rule,
    Mapping from: ModTupFld: U to ModTupFld: U given by a rule,
    Mapping from: ModTupFld: W to ModTupFld: W given by a rule
*]
```

We will provide a tensor category for the dot product tensor, so that the returned tensor is not in the default homotopism category. We will use instead the $\{2, 1\}$-adjoint category. While the returned tensor prints out the same as above, it does indeed live in a universe. The details of tensor categories are discussed in Chapter **??**.

```
> Cat := AdjointCategory(3, 2, 1);
> Cat;
```

```
Tensor category of valence 3 (<-,->,==) ({ 1 },{ 2 },{ 0 })
>
> t := Tensor([U, V, W], Dot, Cat);
> t;
Tensor of valence 3, U2 x U1 >-> U0
U2 : Full Vector space of degree 4 over Rational Field
U1 : Full Vector space of degree 4 over Rational Field
U0 : Full Vector space of degree 1 over Rational Field
>
> TensorCategory(t);
Tensor category of valence 3 (<-,->,==) ({ 1 },{ 2 },{ 0 })
```

CHAPTER 3

Main chapter 2

Bibliography

[BMW] Peter A. Brooksbank, Joshua Maglione, and James B. Wilson, *Groups acting on densors*, in preparation.