# ExceptionAlge Package

**James B. Wilson**

Colorado State University
james.wilson@colostate.edu

**Joshua Maglione**

Universität Bielefeld
jmaglione@math.uni-bielefeld.de

Version 2.1
January 23, 2019

# Contents

CHAPTER 1

# Introduction

**Citing ExceptionAlge.**

## 1.1. Overview

## 1.2. Version

CHAPTER 2

# Exceptional Algebras

Magma provides functionality with common exceptional tensors. Many are used in the construction of nonassociative algebras. A few supporting functions for nonassociative algebras are also provided.

## 2.1. Generics for nonassociative algebras

### 2.1.1. Nonassociative algebras with involutions.

`IsStarAlgebra(A) : AlgGen -> BoolElt`

Decides if algebra has an involution, i.e. a $*$-algebra.

`Star(A) : AlgGen -> Map`

Returns involution of given $*$-algebra.

**Example 2.1.** `StarAlgebra`

We demonstrate the functions dealing with involutions of nonassociative algebras.

```
> A := OctonionAlgebra(Rationals(),-1,-1,-1);
> IsStarAlgebra(A);
true
>
> s := Star(A);
> A.1; // A.1 is the mult. id.
(1 0 0 0 0 0 0 0)
> A.1 @ s;
(1 0 0 0 0 0 0 0)
>
> A.2;
(0 1 0 0 0 0 0 0)
> A.2 @ s;
( 0 -1  0  0  0  0  0  0)
```

### 2.1.2. Operations on power associative algebras.

The following operations are defined for nonassociative algebras for which $x * (x * x) = (x * x) * x$.

`GenericMinimalPolynomial(x) : AlgGenElt -> FldElt`
`GenericMinimumPolynomial(x) : AlgGenElt -> FldElt`

The generic minimum polynomial of an element in a power associative algebra.

`GenericNorm(x) : AlgGenElt -> FldElt`

The generic norm of an element in a power associative algebra.

`GenericTrace(x) : AlgGenElt -> FldElt`

The generic trace of an element in a power associative algebra.

`GenericTracelessSubspaceBasis(A) : AlgGen -> Any`

Given a power associative algebra return a basis for the elements of generic trace 0.

**Example 2.2.** `TenGeneric`

The trace $x+\bar{x}$ of a quaternion doubles the rational component, producing degenerate behavior in characteristic 2. The generic trace avoids this.

```
> Q := QuaternionAlgebra(Rationals(), 1,1);
> Trace(Q!1);
2
> GenericTrace(Q!1);
1
> Q := QuaternionAlgebra(GF(2), 1,1);
> Trace(Q!1);
0
> GenericTrace(Q!1);
1
```

The generic minimum polynomial of an element $x$ in power associative algebra need only be a factor of the minimal polynomial of its right regular matrix $yR_x := x * y$.

```
> J := ExceptionalJordanCSA(GF(5));
> p := GenericMinimumPolynomial(J.3+J.12);
> Rx := AsMatrices(Tensor(J), 2,0);      // yR_x = y*x.
> q := MinimalPolynomial(Rx[3]+Rx[12]);
> Degree(p);
3
> Degree(q);
6
> q mod p;
0
```

## 2.2. Compositions algebras

`CompositionAlgebra(K, a) : Fld, [FldElt] -> AlgGen`
`CompositionAlgebra(K, a) : Fld, [RngIntElt] -> AlgGen`

Constructs the composition algebra with specified parameters. The algebra returned has an involution. The method is modestly intentional choosing Magma's favored representation of the individually classified algebras according to Hurwitz's theorem. In the case of fields the type returned is an algebra with involution, possibly the identity.

`OctonionAlgebra(K, a, b, c) : Fld, FldElt, FldElt, FldElt -> AlgGen`
`OctonionAlgebra(K, a, b, c) : Fld, RngIntElt, RngIntElt, RngIntElt -> AlgGen`

Octonion algebra with involution given by the specified parameters. This builds the Cayley-Dickson algebra over the quaternion algebra $\left(\frac{a,b}{K}\right)$. In particular, Magma's implementation of quaternion algebras is applied.

`SplitOctonionAlgebra(K) : Fld -> AlgGen`

Returns the split octonion algebra over the field $F$.

**Example 2.3.** `TenTriality`

The following example demonstrates some of the mechanics by exploring the concept of triality [S, III.8].

The Cartan-Jacobson theorem asserts that for fields of characteristic other than 2 and 3, the derivation algebra of an octonion algebra is of Lie type $G_2$.

```
> O := OctonionAlgebra(GF(7),-1,-1,-1);
> L := DerivationAlgebra(O);    // Derivations as an algebra.
> SemisimpleType(L);
```

```
G2
```

Cartan's triality obtains $G_2$ from $D_4$ by relaxing to derivations of the octonions as a generic tensor, rather than as an algebra. This is done computationally by changing the category of the octonion product from an algebra to a tensor.

```
> T := Tensor(O);
> T := ChangeTensorCategory(T,HomotopismCategory(2));
> M := DerivationAlgebra(T);  // Derivations as a tensor.
> SemisimpleType(M/SolvableRadical(M));
D4
```

## 2.3. Jordan algebras

`JordanTripleProduct(J) : AlgGen -> TenSpcElt`

Returns the tensor describing the Jordan triple product.

`JordanSpinAlgebra(F) : TenSpcElt -> AlgGen`
`JordanSpinAlgebra(F) : Mtrx -> AlgGen`

Returns the special Jordan algebra of spin type for given symmetric form.

**Example 2.4.** `JordanBasic`

Jordan algebras have suggestive analogues of commutative associative algebras, but experimenting shows serious differences.

```
> F := IdentityMatrix(Rationals(),2);
> J := JordanSpinAlgebra(F);
> T := Tensor(J);
> R := AsMatrices( T, 2,0);
> R[1];   // Is J.1 the identity?
[1 0 0]
[0 1 0]
[0 0 1]
> J.2*J.2 eq J.1;  // J.2^2=1?
true
> J.2*J.3 eq 0;  // Yet J.2 is a zero-divisor.
true
> e := (1/2)*(J.1+J.2);
> e^2 eq e;  // An idempotent of J?
true
```

Pierce decompositions in Jordan algebras have the usual 0 and 1 eigenspaces but an additional 1/2-eigenspace emerges as well.

```
> Re := (1/2)*(R[1]+R[2]);
> Eigenvalues(Re);
{ <1, 1>, <1/2, 1>, <0, 1> }
```

`ExceptionalJordanCSA(O) : AlgGen -> AlgGen`
`ExceptionalJordanCSA(K) : Fld -> AlgGen`

The exception central simple Jordan algebra over the given octonions. If a field is supplied instead then the split octonion algebra over the field is used.

**Example 2.5.** `ChevalleyShaferF4`

In characteristic not 2 or 3, the exceptional central simple Jordan algebra can be used to construct the exceptional Lie algebra of type $F_4$.

```
> J := ExceptionalJordanCSA(Rationals());
> T := Tensor(J);
> T := ChangeTensorCategory(T, HomotopismCategory(3));
> D := DerivationAlgebra(T);
> D2 := Codomain(Induce(D, 2));         // Represent D on U2.
> F4 := D2*D2;                                // Commutator.
> SemisimpleType(F4);
F4
> F4;                             // F4 represented on a 27-dim module.
Matrix Lie Algebra of degree 27 over Rational Field
```

# Bibliography

[S] Richard D. Schafer, *An introduction to nonassociative algebras*, Pure and Applied Mathematics, Vol. 22, Academic Press, New York-London, 1966. MR0210757

# Intrinsics