OnAlgClosure

OnAlgClosure/Frobenius and projective linear action on objects of positive characteristic

0.1

07/07/2017

Gábor P. Nagy

Gábor P. Nagy
Email: nagyg@math.u-szeged.hu
Homepage: http://www.math.u-szeged.hu/~nagyg
Address: Bolyai Institute of the University of Szeged
Aradi vértanúk tere 1
H-6720 Szeged (Hungary)

Contents

Chapter 1

Usage

1.1 Installation

Download und unpack the file

http://www.math.u-szeged.hu/~nagyg/OnAlgClosure-0.1.tar.gz

into your pkg directory. You can load the package with the command

`LoadPackage("OnAlgClosure");`

1.2 Functions for AC-Frobenius automorphism actions

1.2.1 AC_FrobeniusAutomorphism

▷ `AC_FrobeniusAutomorphism(q)` (function)

Returns: an AC-Frobenius automorphism

Creates the Frobenius map $x- > x^q$ for the prime power q which operates on objects of characteristic p: vectors, matrices (also in CVEC representation), polynomials and rational functions. The argument may be the finite field `GF(q)` as well.

Although algebraic closure is not defined in GAP one can say that AC-Frobenius automorphisms act on the algebraic closure of the prime field `GF(p)`.

An AC-Frobenius automorphism has infinite order. By default, no inverse of an AC-Frobenius automorphism is defined. AC-Frobenius automorphisms are not mapping as GAP objects.

These are the main differences to the GAP command `FrobeniusAutomorphism`.

It must be easy to install methods for the action of an AC-Frobenius automorphism on new classes of objects.

Example

```
gap> fr:=AC_FrobeniusAutomorphism(9);
AC_FrobeniusAutomorphism(3^2)
gap> Z(3)^fr;
Z(3)
gap> Z(27)^fr;
Z(3^3)^9
gap> v:=Random(GF(27)^5);
[ Z(3)^0, Z(3^3)^23, Z(3^3)^23, Z(3^3)^17, Z(3)^0 ]
gap> v^fr;
[ Z(3)^0, Z(3^3)^25, Z(3^3)^25, Z(3^3)^23, Z(3)^0 ]
gap> cv:=CVec(v);
```

```
<cvec over GF(3,3) of length 5>
gap> cv^fr;
<cvec over GF(3,3) of length 5>
gap> cv^fr=v^fr;
true
```

Example

```
gap> fr:=AC_FrobeniusAutomorphism(7^2);
AC_FrobeniusAutomorphism(7^2)
gap> x:=Indeterminate(GF(7),"x");;
gap> pol:=(x^3-Z(7))/(x^2-Z(7^3));
(x^3+Z(7)^4)/(x^2+Z(7^3)^172)
gap> pol^fr;
(x^3+Z(7)^4)/(x^2+Z(7^3)^220)
gap> pol=pol^(fr^3);
true
```

AC-Frobenius automorphisms of the same characteristic can be multiplied and share a unique multiplicative identity.

Example

```
gap> fr:=AC_FrobeniusAutomorphism(9);
AC_FrobeniusAutomorphism(3^2)
gap> fr^2;
AC_FrobeniusAutomorphism(3^4)
gap> One(fr);
AC_FrobeniusAutomorphism(3^0)
gap> AC_FrobeniusAutomorphism(8)*AC_FrobeniusAutomorphism(16);
AC_FrobeniusAutomorphism(2^7)
gap> One(last)=One(fr);
false
```

1.2.2 AC_FrobeniusAutomorphismOrbit

▷ `AC_FrobeniusAutomorphismOrbit(fr, obj)` (function)

Returns: the Frobenius orbit of the given object as list

The (i+1)th element of the Frobenius orbit is $obj^(fr^i)$.

Example

```
gap> fr:=AC_FrobeniusAutomorphism(7^2);
AC_FrobeniusAutomorphism(7^2)
gap> m:=[[0*Z(7),Z(7^3)],[Z(7^4)^-1,Z(7)^0]];
[ [ 0*Z(7), Z(7^3) ], [ Z(7^4)^2399, Z(7)^0 ] ]
gap> AC_FrobeniusAutomorphismOrbit(fr,m);
[ [ [ 0*Z(7), Z(7^3) ], [ Z(7^4)^2399, Z(7)^0 ] ], [ [ 0*Z(7), Z(7^3)^49 ], [ Z(7^4)^2351, Z(7)^
  [ [ 0*Z(7), Z(7^3)^7 ], [ Z(7^4)^2399, Z(7)^0 ] ], [ [ 0*Z(7), Z(7^3) ], [ Z(7^4)^2351, Z(7)^0
  [ [ 0*Z(7), Z(7^3)^49 ], [ Z(7^4)^2399, Z(7)^0 ] ], [ [ 0*Z(7), Z(7^3)^7 ], [ Z(7^4)^2351, Z(7
```

An AC-Frobenius automorphism has infinite order. By default, no inverse of an AC-Frobenius automorphism is defined. AC-Frobenius automorphisms are not mapping as GAP objects. If you want, they act on the algebraic closure of the prime field `GF(p)`. In fact, these are the main differences to the GAP command `FrobeniusAutomorphism`.

It must be easy to install methods for the action of an AC-Frobenius automorphism on new classes of objects.

1.3 Functions for AC-projective linear transformations

1.3.1 AC_ProjectiveLinearTransformation

▷ `AC_ProjectiveLinearTransformation(M)` (function)

Returns: an AC-projective linear transformation

Creates the projective linear transformation φ corresponding to the $n \times n$ matrix `M`. Here, `M` is a nonsingular matrix over the finite field `GF(q)`. By definition φ acts via `OnLines` on row vectors of length n with entries from the algebraic closure of `GF(q)`.

Example

```
gap> m1:=[[ Z(5^2), 0*Z(5), 0*Z(5) ],[ 0*Z(5), Z(5)^0, 0*Z(5) ],[ 0*Z(5), 0*Z(5), Z(5^2)^19 ]];;
gap> m2:=[[ Z(5^2)^13, Z(5)^2, Z(5)^0 ],[ Z(5)^2, Z(5)^2, 0*Z(5) ],[ Z(5)^0, 0*Z(5), 0*Z(5) ]];;
gap> t1:=AC_ProjectiveLinearTransformation(m1);
AC_ProjectiveLinearTransformation([ [ Z(5)^0, 0*Z(5), 0*Z(5) ], [ 0*Z(5), Z(5^2)^23, 0*Z(5) ], [
gap> t2:=AC_ProjectiveLinearTransformation(m2);
AC_ProjectiveLinearTransformation([ [ Z(5)^0, Z(5^2)^23, Z(5^2)^11 ], [ Z(5^2)^23, Z(5^2)^23, 0*
  [ Z(5^2)^11, 0*Z(5), 0*Z(5) ] ])
gap> Order(t1);
24
gap> t1*t2;
AC_ProjectiveLinearTransformation([ [ Z(5)^0, Z(5^2)^23, Z(5^2)^11 ], [ Z(5^2)^22, Z(5^2)^22, 0*
  [ Z(5^2)^5, 0*Z(5), 0*Z(5) ] ])
gap> t1/t2;
AC_ProjectiveLinearTransformation([ [ 0*Z(5), 0*Z(5), Z(5)^0 ], [ 0*Z(5), Z(5^2)^11, Z(5^2)^11 ]
gap> One(t1);
AC_ProjectiveLinearTransformation([ [ Z(5)^0, 0*Z(5), 0*Z(5) ], [ 0*Z(5), Z(5)^0, 0*Z(5) ], [ 0*
gap> Characteristic(t1);
5
```

AC-projective linear transformations of the same characteristic and dimension can be multiplied, possess an inverse AC-projective linear transformation and share a unique multiplicative identity.

AC-proejctive linear transformations defined over `GF(q)` have a regular permutation action on $PG(n-1,q)$. Via nice monomorphism, φ knows this permutation. This enables efficient arithmetics for groups generated by AC-projective linear transformations.

It must be easy to implement other actions of AC-projective linear transformations on GAP objects.

Example

```
gap> q:=5;
5
gap> mg:=GU(3,q);
GU(3,5)
gap> Size(mg);
2268000
gap> oset:=Orbit(mg,Z(q)^0*[0,0,1],OnLines);;
gap> Size(oset); q^3+1;
126
126
gap> pg:=AC_ProjectiveTransformationGroupWithShortOrbit(mg,oset);
<group with 2 generators>
gap> Size(pg); Size(PGU(3,q));
378000
```

```
378000
gap> vec:=Z(q)*[1,Z(q^3),0];
[ Z(5), Z(5^3)^32, 0*Z(5) ]
gap> OrbitLength(pg,NormedRowVector(vec));
75600
gap> StructureDescription(pg);
"PSU(3,5) : C3"
```

1.3.2 AC_ProjectiveTransformationGroupWithShortOrbit

▷ `AC_ProjectiveTransformationGroupWithShortOrbit(`*matgr, orb*`)` (function)

Returns: a projective linear cycle

Creates the AC-projective linear transformation group G corresponding to the matrix group *matgr*. `matgr` must have a faithful action on the orbit *orb*. This permutation action is stored as nice monomorphism of G.

Index