Introduction to GAP

Part II

Thomas Breuer

July 1, 2008

Groups

Generators of Groups:
permutations, matrices, abstract words, ...

Subgroups:
`SylowSubgroup`, `DerivedSubgroup`, `Centre`

Factor Groups:
`CommutatorFactorGroup`, `G / N`

Implementation:
group objects storing attributes such as order, finiteness

Group Actions

$M \times G \to M$, $(m, g) \mapsto m * g$
such that $m * 1 = m$
and $m * (g_1 g_2) = (m * g_1) * g_2$

In GAP, group elements act from the right:
matrices on row vectors, row spaces $(v * A)$
permutations on positive integers $(i\hat{}p)$
elements on right cosets of a subgroup $(Hg_1 * g_2)$
...

Group Actions

`OnPoints`, `OnTuples`, `OnSets`, `OnSetsSets`, ...

`OnRight`

`Permutation( g, set, actfun )`

`Action( G, set, actfun )`

`Orbit( G, obj, actfun )`

`Stabilizer( G, obj, actfun )`

`Centralizer`, `Normalizer`

Group Homomorphisms

$\varphi: G \to H$
$\varphi(g_1 g_2) = \varphi(g_1)\varphi(g_2)$

`ActionHomomorphism( G, set, actfun )`
`GroupHomomorphismByImages( G, H, elmsG, elmsH )`
`NaturalHomomorphismByNormalSubgroup( G, N )`

Group Libraries

Finite Perfect Groups:
all perfect groups of order up to 10^6, except for eleven orders

Primitive Permutation Groups:
all primitive permutation groups of degree up to 2500

Small Groups:
all of order up to 2000, except 1024; p-groups of order up to p^6;
all of squarefree order; ...

Transitive Permutation Groups:
all of degree up to 30

... (an increasing number of group libraries in GAP packages)

Domains

Domain is GAP's name for a structured set.

- operational structure
- generation/closure concept
- homomorphism concept
- attributes, operations

Domains

Groups:
`*, Inverse, One`; `GeneratorsOfGroup`;
`DerivedSubgroup`, `SylowSubgroup`;
`SymmetricGroup(n)`, `AlternatingGroup(n)`, `GL(d,q)`, `Group( ... )`

Fields:
`*, Inverse, One, /, +, AdditiveInverse, Zero`;
`GeneratorsOfField`;
`GaloisGroup`;
`Rationals`, `GF(q)`, `CF(n)`

Vector Spaces:
`+, AdditiveInverse, Zero, *`; `GeneratorsOfVectorSpace`;
`Dimension`, `Basis`;
`Rationals^d`, `GF(q)^d`, `CF(n)^d`

Rings:
`+, AdditiveInverse, Zero, *`;
`GeneratorsOfRing`, `GeneratorsOfRingWithOne`;
...;
`Integers`

Algebras:
`+, AdditiveInverse, Zero, *, *`;
`GeneratorsOfAlgebra`, `GeneratorsOfAlgebraWithOne`;
`ProductSpace`;
`Rationals^[d,d]`, `QuaternionAlgebra( F )`,
`AlgebraByStructureConstants( F, ... )`, `GroupRing( F, G )`

Semigroups:
...

Operations and Methods

How does GAP actually compute all this?
(For example: `Size` or `IsAbelian`)
Operations know which function shall be used.
Methods do the work.
This is some kind of object orientation.
Attributes are special operations.