Group Operations

Alexander Hulpke

April 1999

G-sets

A G-*set* is a set Ω together with a group action $\mu: \Omega \times G \to \Omega$ of a group G.

Group operations are naturally considered in the category of G-sets:

G-sets (Ω, G, μ) and (Δ, H, ν) are equivalent if there is a bijection $\alpha: \Omega \to \Delta$ and an isomorphism $\varphi: G \to H$ such that

$$\mu(\omega, g)\alpha = \nu(\omega\alpha, g\varphi).$$

G-sets can be transitive, regular, primitive, etc.

If $\Delta \subset \Omega$ is G-invariant, $(\Delta, G, \mu_{|\Delta})$ is a G-(sub)set. The most frequent case is the G-orbit ω^G (for $\omega \in \Omega$).

Every G-set induces a permutation representation $\phi: G \to S_\Omega$.

If $\phi: H \to G$ then $(\Omega, H, \mu(\cdot, \cdot\phi))$ is an H-set.

Note that in GAP all group operations act *from the right*, that is

$$\mu(\omega, gh) = \mu(\mu(\omega, g), h).$$

External sets

In GAP, G-sets are implemented via the category `IsExternalSet`. An external set (the name alludes to the similarity with vector spaces or modules, which are `IsExtLSet`) is created from a collection `Omega`, a group `G`, and an operation function (an ordinary 2-argument GAP function) `opfun(`*`omega`*`,`*`g`*`)` by

```
ExternalSet(G,Omega,opfun);
```

The external set stores the group in the attribute `ActingDomain`, the set in the attribute `HomeEnumerator` (the `Enumerator` of an external set consisting of several orbits enumerates the orbits) and the operation function in `FunctionOperation`.

Standard operation functions are:

`OnPoints` Action via ^ (permutation on points, group on itself, matrices on vectors). This is the default if no operation function is given.

`OnRight` Right multiplication (group on cosets, matrices on vectors).

`OnLeftInverse` Left multiplication by inverse of group element.

`OnSets` Action on sets of elements induced by `OnPoints` on the elements. This is also used for the action on blocks in a block system.

`OnTuples` ditto for tuples of elements.

`Permuted` Action on lists by permuting the indices.

`OnIndeterminants` Permutation of indeterminants for multivariate polynomials.

An external set can have properties like `IsTransitive`, `IsRegular`, `IsPrimitive` and attributes like `RankOperation` or `Transitivity`. These also can be called as operations with the full set of arguments, for example

```
Transitivity(G,[1..5],OnPoints)
```

Operation via an homomorphism

The case of G-sets induced by a representation ϕ merits special treatment: Sometimes we can evaluate ϕ (or take preimages) only on generators in practice. This however is sufficient for the standard orbit/stabilizer algorithm, as only the generators act and we always obtain words in the generators. The syntax here is

```
ExternalSet(G,Omega,gens,genimages,opfun);
```

for a list `gens` of generators of `G` and their images `genimages` under ϕ (which otherwise is not given and remains unevaluated at other elements).

External subsets

Transitive external sets can be created by

```
ExternalOrbit(G,extset,pnt,opfun)
ExternalOrbit(G,pnt,opfun)
```

Here `pnt` is stored in the attribute `Representative` and the attribute `StabilizerOfExternalSet` will compute its stabilizer.

The variant `ExternalSubset(G,extset,start,opfun)` creates the subset consisting of the orbits of all points in `start`.

A list of the separate external orbits within one external set can be obtained by the `Enumerator`.

`ExternalOrbits(`*`extset`*`)` computes a list of `ExternalOrbits` consisting of all the orbits, and `ExternalOrbitsStabilizers` simultaneously computes the stabilizers.

The `Enumerator` of an external orbit gives the elements of the orbit.

If no further usage of the external sets is envisioned their use would be a bit clumsy.

Therefore GAP also supports the operations

```
Orbit(G,pnt,opfun)
Orbits(G,Omega,opfun)
Stabilizer(G,pnt,opfun)
```

which simply return lists of elements, respectively the stabilizing subgroup.

Mapping elements

In general mapping elements are computed by

```
RepresentativeOperation(G,omega,delta,opfun)
```

In general, stabilizers or representatives must be computed by an orbit-stabilizer algorithm. There are however efficient methods for solvable groups (solvable orbit algorithm) and permutation groups (backtrack) for many popular operations.

Some prominent external orbits

Many subsets of groups are external orbits for the action of the group on itself:

`ConjugacyClass(G,g)`

`RightCoset(U,g)` (operation `OnLeftInverse`)

`ConjugacyClassSubgroups(G,U)`

As external sets are domains, these objects have methods for `Size` and `in` besides the usual `Representative`. They usually do not evaluate the `HomeEnumerator` unless explicitly asked for.

Their `StabilizerOfExternalSet` is also returned by the operations `Centralizer`, respectively `Normalizer`.

The operations `ConjugacyClasses`, `RightCosets`, `ConjugacyClassesSubgroups` return a list of all orbits that exhaust the full domain.

Canonical representatives

Comparison of external sets becomes easy if there is a normal form. In GAP this can be obtained (if installed) by the attribute `CanonicalRepresentativeOfExternalSet`

For right cosets this canonical representative also is the smallest element in the coset. Therefore it can even be used for $<$ comparisons.

(Because not every external set has a canonical representative defined there is the attribute `CanonicalRepresentativeDeterminatorOfExternalSet` which returns – if available – a function to compute the canonical representative.)

The operation `OperatorOfExternalSet` returns an element that maps the `Representative` to the `CanonicalRepresentativeOfExternalSet`.

Operation homomorphisms

The homomorphism $\phi\colon G \to S_\Omega$ is obtained by

```
OperationHomomorphism(G,Omega,opfun)
```

It returns a GAP homomorphism. (Methods for this homomorphism however do not necessarily use the operation, but the `AsGroupGeneralMappingByImages` if this is quicker, for example for pre-images.)

The command `Operation` (same arguments) is still supported for compatibility and returns the `Image` of the operation homomorphism.

The computation of this image however can be expensive (and may be never asked for). Therefore the `Range` of an `OperationHomomorphism` is usually the full symmetric group. If it is desired (for example for a `NiceMonomorphism`), the string `"surjective"` should be added as a further argument.

The variant

```
SparseOperationHomomorphism(G,pnt,opfun)
```

computes the orbit of `pnt` under `G` and *simultaneously* computes the permutation action. This can save runtime if the operation (or point identification) is expensive.

`SortedSparseOperationHomomorphism` essentially performs the same task, but will sort the domain (thus relying on the points being easily comparable). The actual permutations then are constructed via the operation `Permutation`.

When looking for the position of an element in the domain Ω, GAP actually uses the operation `PositionCanonical`. For ordinary lists this is simply the same as `Position`, but may be different for other objects: For `RightTransversal(G,U)` it returns the position of the *representative for the same coset*, so one can write:

```
Operation(G,RightTransversal(G,U),OnRight);
```

to obtain the action on the cosets.

Declaration and installation

All GAP operations for operations allow the two additional arguments `gens` and `genimages` (as well as replacing `opfun` by a default value `OnPoints`). Alternatively an external set may be given to supply all arguments. In this case the result is stored as attribute of the external set.

The following is true for $\beta 5$ but subject to change:

Technically, the variety of arguments for operation operations is handled by special functions, for example `OrbitsishFOA("Orbits", ...);` defines:

1. The function `Orbits` that takes a variable number of arguments. It sorts out the meaning of these and calls:

2. The operation `OrbitsOp(G,Omega,gens,genimages,opfun)` (which takes all arguments) that does the work. Methods need to be installed only for this full range of arguments.

3. An attribute `OrbitsAttr` that stores the result for external sets.

(In the break loop backtrace, the operation is usually called `orbish`.)