From: **Martin Schoenert** martin.schoenert@math.rwth-aachen.de
Subject: Re: Re: Bye, NewObject
Date: 20 September 1996 at 08:03
To: gap-dev@math.rwth-aachen.de

And one more follow-up message.

If constructors take a result family or a result kind as first argument,
then 'ObjByExtRep' is simply such a constructor.

The only thing special about 'ObjByExtRep' is that the interpretation of the
arguments only depends on the family. For most other constructors the
interpretation will mostly depend on the constructor (and the constructor
will be applicable only for certain families).

Remember I wrote that the interpretation of the arguments depends on the
constructor and the result family (but not the categories of the result
kind).   In this sense 'ObjByExtRep' is at one extreme, and most other
constructors are at the other extreme.

BTW. I opt for 'ObjByExtRep' and 'ExtRepOfObj'.
In fact I think 'New' would be a nice alias for 'ObjByExtRep'.

Martin.

-- .- .-. - .. -. .-.. --- ..- . ... .- -. -. .. -.- .-
Martin Sch"onert,   Martin.Schoenert@Math.RWTH-Aachen.DE,   +49 241 804551
Lehrstuhl D f"ur Mathematik, Templergraben 64, RWTH, 52056 Aachen, Germany

From: **Martin Schoenert** martin.schoenert@math.rwth-aachen.de
Subject: Re: Bye, NewObject
Date: 20 September 1996 at 07:45
To: gap-dev@math.rwth-aachen.de

In my previous message I wrote about constructors that take the wanted result family or the wanted result kind as first argument.

The problem that Thomas describes with associative words can then be solved.

If you want to construct a word in a particular representation, simply make certain that the result kind requires that representation.

Now for efficiency reason one could take the following path.

The constructor should *behave* as implied by the general concept.
But it need not be *implemented* in this manner.

For example if one knows that the method selection will depend only on the kind (i.e. not on the other arguments), then one could store the constructor methods at a special position in the kind, and the constructor operation would select the method by taking the value at that position in the kind.

Of course doing so would limit the possible extensions (e.g. with new methods). But since the specially implemented constructor behaves like an ordinary constructor, it should be possible to fall back to an ordinary constructor without too much work.

Martin.

-- .- .-. - .. -. .-.. --- ...- . ... .- -. -. .. -.- .-
Martin Sch"onert,   Martin.Schoenert@Math.RWTH-Aachen.DE,   +49 241 804551
Lehrstuhl D f"ur Mathematik, Templergraben 64, RWTH, 52056 Aachen, Germany

From: frank.celler@math.rwth-aachen.de
Subject: Re: Bye, NewObject
Date: 20 September 1996 at 04:51
To: thomas.breuer@math.rwth-aachen.de
Cc: gap-dev@math.rwth-aachen.de

Bye, bye NewObject! Welcome ObjExtRep?

As we are now getting rid of 'NewObject' and 'ObjExtRep' in its unkind meaning, it would be handy to reuse 'ObjExtRep' as inverse of 'ExtRepObj':

ExtRepObj(<obj>)
----
returns an external representation <ext> for <obj>

ObjExtRep(<fam>, <ext>)
----
returns an objects which has <ext> as external representation. Therefore ObjExtRep(FamilyObj(obj), ExtRepObj(obj)) and obj describe the same element.

Comments? Should it be named 'ExtRepOfObj' and 'ObjByExtRep'?
 Frank

From: **Thomas Breuer** thomas.breuer@math.rwth-aachen.de
Subject: Bye, NewObject
Date: 20 September 1996 at 03:41
To: gap-dev@math.rwth-aachen.de

There are two places in the GAP 4 library where it is not obvious to me how to get rid of 'NewObject'. These are the constructions of mappings and of associative words.

In both cases, the old implementation used the facility to dispatch according to the type of the result, which was specified by the first argument of 'NewObject', i.e., the required kind of the result.

1. The proposed solution to settle the mapping case without 'NewObject' (and of course better than with 'NewObject') is to set up the kind of the mapping with 'NewKindOfGeneralMapping', a common function for all mappings (with three arguments, namely the source and the range and a filter that describes the categories/representation of the desired mapping), and to have really individual constructors for the mapping itself. The latter means that we do not have a default constructor for all mappings. Note that we want the flexibility to choose 'IsPositionalObjectRep' for mappings; one step in this direction is that source and range of a mapping can be stored in its kind.

2. The situation with associative words is more delicate. For efficiency reasons, the arithmetic operations may claim to decide about the representation of their results. For example, if the product of two 8 Bits words is known to fit into the 8 Bits representation, the multiplication may call the internal function that constructs an 8 Bits word. (Note that 'Objectify' is *not* called for this.) 'NewObject' was used to choose such internal functions as methods, depending on its first argument, which had the desired representation that could be used by the method selection. The proposed solution without 'NewObject' is to store the appropriate internal constructor function itself in the kind of the word, and to call this function directly instead of calling 'NewObject' with the appropriate kind. This should be even more efficient since it avoids one way through the method selection.

   But note that all such attempts to speed up the computations make it very difficult to add new representations at a later stage, since the central role of the method selection process is not respected. (Recently Werner reminded me not to do ``premature optimizations''.)

Thomas