Experiments with Numerical Semigroups

Manuel Delgado

www.fc.up.pt/cmup/mdelgado/

International meeting on numerical semigroups - Cortona 2014

Cortona, 8 September, 2014
Outline

1 A crash course on GAP

2 Examples

3 Programming examples
A crash course on GAP

The GAP system: http://www.gap-system.org/

Installation: in the installation page [link], look for the “alternative installation methods”

A look through the packages...

Note: the stable versions of the packages provided by GAP are included in any “standard” GAP installation.

Examples of packages: numericalsgps and intpic

The manuals are included in the packages. They are also available online, from the packages webpages.
Examples with “numericalsgps” and “intpic”

After having a working copy of GAP installed on the computer, one has to load the packages to be used.

A GAP session...

LoadPackage("numericalsgps");
LoadPackage("intpic");
ns := NumericalSemigroup(9,13,15);
arr := [SmallElementsOfNumericalSemigroup(ns),
       GapsOfNumericalSemigroup(ns),
       MinimalGeneratingSystemOfNumericalSemigroup(ns),
       FundamentalGapsOfNumericalSemigroup(ns),
       [ConductorOfNumericalSemigroup(ns)],
       PseudoFrobeniusOfNumericalSemigroup(ns)];;
tkz := IP_TikzArrayOfIntegers(rec(highlights:=arr));;

By executing

Print(tkz);

one obtains tikz code that can be copy/pasted into a latex document...
Alternatively, see the “intpic” manual for details on saving the tikz code into
a file or on automatic displaying...
A single row may not be the most convenient way to visualize

tkz := IP_TikzArrayOfIntegers(8,rec(highlights:=arr));;
A short programming example

Some notation and terminology

- $S$: numerical semigroup
- $I$: relative ideal of $S$ (generated by more than one element)

Consider the relative ideals $S - I$ (the dual of $I$) and $I + (S - I)$.

The pair $(S, I)$ is said to be a $k \times m$ brick if the number of minimal generators of $I$ is $k$, the number of minimal generators of $(S - I)$ is $m$ and the number of minimal generators of $I + (S - I)$ is $km$.

Let us find some bricks...
S := RandomNumericalSemigroup(5,100);;
MinimalGeneratingSystemOfNumericalSemigroup(S);
genI := [0,1];;
ideal := genI+S;

One can use the following to check if the pair (S, ideal) is a brick.

genideal := MinimalGeneratingSystem(ideal);
dual := S-ideal;
gendual := MinimalGeneratingSystem(dual);
##
ideal_plus_dual := ideal + dual;
genipd := MinimalGeneratingSystem(ideal_plus_dual);

One can do something more sophisticated just by putting these commands into a function.
# input : numerical semigroup
#     list of integers (generators of relative ideal)

isBrick := function(S, genI)
    local ideal, genideal, dual, gendual, ideal_plus_dual, genipd;

    ideal := genI+S;
genideal := MinimalGeneratingSystem(ideal);
dual := S-ideal;
gendual := MinimalGeneratingSystem(dual);
##
    ideal_plus_dual := ideal + dual;
genipd := MinimalGeneratingSystem(ideal_plus_dual);
##
    return Length(genideal)+Length(gendual) = Length(genipd);
end;
##

One example:

ss := NumericalSemigroup(14,21,27,36,45);
isBrick(ss,[0,1]);
##
DeclareInfoClass("InfoBrick");
##
testBricks := function(n)
    local  brks, i, ns, genI, brk;
    brks := [];
    for i in [1..n] do
        ns := RandomNumericalSemigroup(5,100);;
        genI := [0,RandomList([1..5])];
        if isBrick(ns,genI) then
            brk := [MinimalGeneratingSystem(ns), genI];
            Info(InfoBrick,1,"I found a brick: ", brk, "\n");
            Append(brks, [brk]);
        fi;
    od;
    return brks;
end;
By executing

SetInfoLevel(InfoBrick,1);
testBricks(100);

one can get some bricks (and get information while the computations are being done). And executing

testBricks(1000000); # do not run this on a laptop...

one can get some more...