THE NSSC (NEC SUBGROUPS SIGNATURE CALCULATION) PACKAGE

ISMAEL CORTÁZAR, ANTONIO F. COSTA (2017)

1. Introduction

The NSSC package provides an implementation of the Hoare's algorithm (ref [2]), with the improvements in the calculation of the direction of the period-cycles described in ref [1].

So given an NEC group Δ and an homomorphism φ, with transitive image, into the symmetrical group of n elements S_n, the algorithm calculates the signature of the subgroup $\mathit{\Gamma} = \varphi^{-1}(Stab(1))$.

The NSSC package contains a most important function named *NSSC_ CalcMain*, this function takes a record with the data of the group Δ and the homomorphism into S_n, and returns a record with the signature of Γ and details on the calculations made.

This function is intended to be used by another functions that generate data of NEC groups and homomorphism and analyze the results.

For the convenience of the user that just want to analyze a particular case the package provides a function called *NSSC_ CalcFile* that uses a text file as input and writes the result in another text file (that could be the screen).

The input file must accomplish the GAP syntax as it is read using the standard GAP function *ReadAsFunction*. But it could be very easily generated starting from any of the multiple examples provided by the package.

The output file is self-explanatory, and besides the signature of the subgroup contains detailed information of the calculations, although there is an option to get a terse output with just the signature.

2. Installation

The installation of the nssc package is rather easy. After uncompress the downloaded file it suffices with copy the nssc directory into the GAP library directory.

3. Functions

The package provides the following functions

NSSC_CalcMain(data). Takes the record *data* as input and return the record *res*. This function is intended to be called by functions inside a bigger GAP project.

NSSC_CalcFile(filein, <fileout>). The parameter *filein* is the name of a text file with a description of the starting NEC group and of the homomorphism into the symmetrical group, *fileout* is the name of the text file in which the results will be written, this is an optional parameter if it is not present results are written in the screen.

It is important to note that filein must follow GAP sintax as it is to be read using *ReadAsFunction*. Also the name of the variable *data*, inside the file, should not be changed.

This function has three options: **dir, terse,** and **debug**: The option *terse* provokes than only the signature is written to the output file. The option *dir* makes the program use the specified directory, instead of the current one, to search *flilein* and to write *fileout*. The directory could be specified either as an string or as a GAP directory object.

The option debug is not intended for the users of the package.

Example:

NSSC_CalcFile("descr.g":dir:=NSSC_TestDir(),terse);

Writes just the signature of the subgroup into the screen using the description file *descr.g* in the package test directory.

NSSC_CheckOrient(data). This function returns a record *ret* with information on the orientability of the cover surface (described by the NEC subgroup $\Gamma = \varphi^{-1}(Stab(1))$). The record data is the same that the used in *NSSC_CalcMain*, actually it calls this function.

NSSC_TestDir(). Returns the package test directory either as a GAP object directory or as an string if the option *str* is used.

4. Records

First of all a word on notation: Every reflection generator of Γ is produced by a reflection generator of Δ whose image by φ fixes an element (coset) of Sn. Since every reflection of Δ lives inside a period-cycle, given a reflection of Γ we can speak of the generating period-cycle, the originating reflection (position inside the period-cycle), and the originating fixed coset.

Similarly every elliptic generator of Γ is produced either by an elliptic generator of Δ or by an dihedral group of reflections of Δ, so we can speak of the generating elliptic generator or the generating period-cycle and reflections,

There are several records that are used as inputs/outputs of the functions:

Data input record. This record contains the data of the NEC group Δ and the homomorphism φ into S_n. The fields starting by *S_* define the group whereas those starting by *H_* define the homomorphism, this record is initialized in the GAP file that is used as input to *NSSC_CalcFile()*, or must be filled up prior to call *NSSC_CalcMain*

The fields are:

- **data.S_ORIENT** true or false depending on the orientability of Δ.
- **data.S_GENUS** genus of Δ.
- **data.S_NELLIP** number of elliptic periods of Δ.
- **data.S_EPERIODS** array with the elliptic periods of Δ.
- **data.S_NPERCYC** number of period-cycles of Δ.
- **data.S_NREFPERCYC** array with the number of reflections of each period-cycle of Δ.
- **data.S_CPERIODS** array of arrays of the period-cycles of Δ.
- **data.H_NC** order of the cover of Δ, so the homomorphism φ will be into the symmetrical group of data.H_NC elements, it is also the index Δ/Γ being $\Gamma = \varphi^{-1}(Stab(1))$ the subgroup whose signature is to be calculated.
- **data.H_AGEN** array of permutations image by φ of the generators of the fundamental group (number equal genus) of the surface defined by Δ neither considering elliptic points nor reflections. These generators will be conformal if the original surface is orientable and anticonformal otherwise.
- **data.H_BGEN** array of permutations image by φ of the generators of the fundamental group (number equal genus) of the surface defined by Δ neither considering elliptic points nor reflections, they exist only in case of orientable surfaces, being so conformal.
- **data.H_XGEN** array of permutations image by φ of the elliptic generators of Δ.
- **data.H_REFGEN** array of arrays of permutations image by φ of the reflections generators of Δ. Indexed by period-cycle and number of reflection inside it.
- **data.H_CONNGEN** array of permutations image by φ of the connection generators of Δ. Indexed by period-cycle.

Results record. This record returned by *NSSC_CalcMain* contains the signature of the NEC subgroup $\Gamma = \varphi^{-1}(Stab(1))$ and aditional data on the calculations made to obtain it.

The fields are:

- **res.OK** false if something has gone wrong in the calculation.
- **res.Diag** text with an intended explanation of the error if it has happened.
- **res.CoverORIENT** true if the Klein surface defined by Γ is orientable.

- **res.BIPARTITE_ONE** if the surface defined by Γ is orientable this is the set of the bipartition that contains the coset numbered 1. The colour associated to this set is 0.
- **res.BIPARTITE_OTHER** if the surface defined by Γ is orientable this is the set of the bipartition that does not contain the coset numbered 1. The colour associated to this set is 1.
- **res.CoverEPERIODS** array with the elliptic periods of Γ.
- **res.CoverEPERIODSVAL1** array with information on the origin of the the elliptic period of Γ, 0 if it comes from an elliptic period of Δ, or the number of the period-cycle of Δ in which the pair of originating reflections are located.
- **res.CoverEPERIODSVAL2** array with information on the origin of the the elliptic period, case of being originated by a pair of reflections of Δ, it stores the number of the lowest reflection of the pair.
- **res.CoverPERCYC** array of records, one by period-cycle of Γ (explained later).
- **res.AUX_FIXED** array of array of cosets fixed by reflections, indexed by period-cycle of Δ and number of reflection inside it.
- **res.AUX_ORBITS** array of array of orbits in S_n of the dihedral groups generated by reflections $j, j+1$ of the period-cycle i (so indexed by i, j) of Δ (there is an array of orbits corresponding to a pair of indexes i, j being each group of elements in one orbit further indexed by k).
- **res.AUX_FCONVI**, **res.AUX_FCONVF** the first and last reflections of a period-cycle have the same number of fixed cosets related by the permutation associated to the corresponding connection generator. This fields contain the fixed cosets of the last reflection and the corresponding ones of the first reflection. These ararys are indexed by period-cycle.
- **res.AUX_CREF** array of array of records with information of the reflections of Γ, indexed by the originating period-cycle of Δ and another index that is not easily related with anything, so searches inside the array are frequently performed. The record will be detailed later.

Cover reflections record (res.AUX_CREF[i][j]). This record stores data of a reflection generator of Γ.

The fields are:

- **auxcref.orgpcycle** originating period-cycle of Δ.
- **auxcref.pos** originating reflection.
- **auxcref.fix** originating fixed coset.
- **auxcref.name** name of the reflection of Γ. Names are formed as irj_k being i the originating period-cycle, r just this letter, j the originating reflection, and k the originating fixed coset.

- **auxcref.scan** used internally when the algorithm scans through reflections, don't care.
- **auxcref.link1** index, inside the array *res.AUX_ CREF[i]*, of one of the reflections of Γ that forms one of the two dihedral groups of contiguous reflections of Γ in which the current reflection is involved (the link is found using the orbits of dihedral groups of Δ or through the image by φ of connection generators).
- **auxcref.dih1** order of the dihedral group formed by the current reflection of Γ and that indexed by link1.
- **auxcref.typ1** type of link1, 1 if each of the two fixed cosets if fixed by a different reflection of Δ or the link1 is through a connection generator (both fixed cosets of the same colour), 2 otherwise.
- **auxcref.orgdh1** order inside the period-cycle of Δ of the reflection corresponding to the fixed coset that generates the reflection of Γ pointed by link1. In case the current reflection is the first one and the pointed by link1 the last one this field is filled with -1.
- **auxcref.name1** name of the reflection indexed by link1.

  The same for the other related reflection of Γ.
- **auxcref.link2**
- **auxcref.dih2**
- **auxcref.typ2**
- **auxcref.orgdh2**
- **auxcref.name2**

Cover period-cycle record. This record stores data of a period-cycle of Γ, they are the elements of *res.CoverPERCYC.*

The fields are indexed in *res.AUX_ CREF[i]* (being i the previous field) of the reflections of the period-cycle.

- **percyc.ORGPC** originating period-cycle in Δ.
- **percyc.ELEMS** array of indexed in *res.AUX_ CREF[i]* (being i the previous field) of the reflections of the period-cycle.
- **percyc.LINKS** array of links ($(1, 2)$ of auxcref) used to obtain the next reflection of the period-cycle .
- **percyc.DIHS** the period-cycle itself, ones should not be considered.
- **percyc.COLORS** array of colours of the cosets fixed by the originating reflections of this period-cycle.

5. Test Directory

There is a test directory, returned by *NSSC_ TestDir* that contains many examples of description and result files. The description files have an extension *.g*. Of course this is not necessary but it has be done to emphasize that these files must accomplish GAP sintax.

6. Example

An example of the use of the package follows. The input data file is:

```
local data; data:=rec();
##########################################
#ORIENTABLE
data.S_ORIENT:=false;
#Genus
data.S_GENUS:=2;
#Elliptic generators
data.S_NELLIP:=2;
#Periods
data.S_EPERIODS:=[3,4];
#Number of period cycles
data.S_NPERCYC:=2;
#Number or reflections en each period cycle data
S_NREFPERCYC:=[4,1];
# Periods of period cycles data
S_CPERIODS:=[[24,6,12], [] ];
#################
#Cover order
data.H_NC:=8;
#Surface generators
data.H_AGEN:=[ (1,2,3,4,5,6,7,8), (3,4,5,6,7,8)(1,2) ];
#Elliptic Generators
data.H_XGEN:=[(1,5,7)(2,4,8),(1,3,5,7)(6,8)];
#Reflections
data.H_REFGEN:=[ #First Cycle [(2,3)(5,6), (1,2)(4,5)(6,7), (1,2)(3,4)(7,8),
(3,4)(6,1)], #Second Cycle [(1,4)(5,8) ] ];
# Connecting generators
data.H_CONNGEN:=[ (1,3,5)(2,4,6), () ];
#################
return data;
```

And the file with results is:

Signature (13;+;[3,3,4,4,2,6,12];{(8,2,12),(4,2,24,6,6),(),(),(),()})
Orientable Cover Genus: 13
Elliptic periods of the cover: [3, 3, 4, 4, 2, 6, 12]
Cover has 6 Period Cycles
Cover Period Cycle #1–> (8,2,12) Cover Period Cycle #2–> (4,2,24,6,6)
Cover Period Cycle #3–> () Cover Period Cycle #4–> ()
Cover Period Cycle #5–> () Cover Period Cycle #6–> ()
*********** CALCULATION DETAILS ***************
****
Bipartite sets [1, 3, 5, 7] [2, 4, 6, 8]
****** Elliptics
Period 3 generated from elliptic 1
Period 3 generated from elliptic 1
Period 4 generated from elliptic 2
Period 4 generated from elliptic 2
Period 2 generated from elliptic 2
Period 6 generated from period cycle 1 Reflections 2 3
Period 12 generated from period cycle 1 Reflections 3 4
****** COSETS FIXED BY REFLECTIONS
PERIOD CYCLE 1
REFLECT 1 [1, 4, 7, 8] REFLECT 2 [3, 8] REFLECT 3 [5, 6] REFLECT 4 [2, 5, 7, 8]
PERIOD CYCLE 2
REFLECT 1 [2, 3, 6, 7]
****** ORBITS OF DIHEDRAL GROUPS OF REFLECTIONS
Period Cycle 1
Orbit R1_R2 [1, 2, 3][4, 5, 6, 7][8] Orbit R2_R3 [1, 2][3, 4, 5][6, 7, 8]
Orbit R3_R4 [1, 2, 6][3, 4][5][7, 8]
Fixed coset conversion R4 R1: [2, 5, 7, 8]–>[4, 1, 7, 8]
Period Cycle 2
Fixed coset conversion R1 R1: [2, 3, 6, 7]–>[2, 3, 6, 7]
****** PERIOD CYCLES OF THE COVER
Cover period cycle #1
Originating period cycle in the original surface 1
Colors [0, 0, 0, 0]
Dihedral orders [8, 2, 12, 1]
Reflections 1r1_1 1r2_3 1r3_5 1r4_5 1r1_1
Cover period cycle #2
Originating period cycle in the original surface 1
Colors [1, 1, 1, 1, 1, 1, 0, 0]
Dihedral orders [1, 4, 2, 24, 1, 6, 1, 6]
Reflections 1r1_4 1r4_2 1r3_6 1r2_8 1r1_8 1r4_8 1r4_7 1r1_7 1r1_4
Cover period cycle #3
Originating period cycle in the original surface 2
Colors [1]
Dihedral orders [1]
Reflections 2r1_2 2r1_2

```
Cover period cycle #4
Originating period cycle in the original surface 2
Colors [ 0 ]
Dihedral orders [ 1 ]
Reflections 2r1_3 2r1_3
Cover period cycle #5
Originating period cycle in the original surface 2
Colors [ 1 ]
Dihedral orders [ 1 ]
Reflections 2r1_6 2r1_6
Cover period cycle #6
Originating period cycle in the original surface 2
Colors [ 0 ]
Dihedral orders [ 1 ]
Reflections 2r1_7 2r1_7
```

7. .Sample session

Let us suppose that there is a directory called *pru* inside the starting directory of GAP, and a description file in it named *descr.* We want to have the results written in file *res* in the same directory

(The starting directory is specified in GAP batch starting file,. A typical example would be:

set HOME=%HOMEDRIVE%%HOMEPATH%\Documents\gap)

After that we run another example in the test directory.

```
gap> LoadPackage("nssc");
———————————————————————————-
************************************************************
Loading nssc 1.0
by Ismael Cortazar
Antonio Costa
************************************************************
———————————————————————————-
true
gap> NSSC_CalcFile("descr","res":dir:="pru");
true
gap> d:=NSSC_TestDir();
dir("/proc/cygdrive/C/gap4r8p7/pkg/nssc/tst/")
gap> NSSC_CalcFile("descr3.g":dir:=d,terse);
Signature
(1;-;{(3),(),()})
true
gap>
```

References

[1] Cortázar, Ismael; Costa Antonio F. Computing the signatures of subgroups of non-Euclidean crystallographic groups, J. of Algebra 477 (2017), 483-493

[2] Hoare, A.H.M. (1990). Subgroups of NEC groups and finite permutation groups, Quart. J. Math 41 (1990), 45-59.