Numerical Semigroups

(Version 0.94)

Manuel Delgado
Pedro A. García-Sánchez
José João Morais

Manuel Delgado — Email: mdelgado@fc.up.pt
— Homepage: http://www.fc.up.pt/cmup/mdelgado

Pedro A. García-Sánchez — Email: pedro@ugr.es
— Homepage: http://www.ugr.es/~pedro

José João Morais — Email: jjoao@netcabo.pt

Copyright

Acknowledgements

Colophon

This work started when the first author visited the University of Granada in part of a sabbatical year. Bug reports, suggestions and comments are, of course, welcome. Please use our email addresses to this effect.

Contents

Chapter 1

Introduction

A *numerical semigroup* is a subset of the set $\mathbb{N}$ of nonnegative integers that is closed under addition, contains 0 and whose complement in $\mathbb{N}$ is finite. The smallest positive integer belonging to a numerical semigroup is its *multiplicity*.

Let S be a numerical semigroup and A be a subset of S. We say that A is a *system of generators* of S if $S = \{k_1 a_1 + \cdots + k_n a_n \mid n, k_1, \ldots, k_n \in \mathbb{N}, a_1, \ldots, a_n \in A\}$. The set A is a *minimal system of generators* of S if no proper subset of A is a system of generators of S.

Every numerical semigroup has a unique minimal system of generators. This is a data that can be used in order to uniquely define a numerical semigroup. Observe that since the complement of a numerical semigroup in the set of nonnegative integers is finite, this implies that the greatest common divisor of the elements of a numerical semigroup is 1, and the same condition must be fulfilled by its minimal system of generators (or by any of its systems of generators).

Given a numerical semigroup S and a nonzero element s in it, one can consider for every integer i ranging from 0 to $s-1$, the smallest element in S congruent with i modulo s, say $w(i)$ (this element exists since the complement of S in $\mathbb{N}$ is finite). Clearly $w(0) = 0$. The set $\text{Ap}(S,s) = \{w(0), w(1), \ldots, w(s-1)\}$ is called the *Apéry set* of S with respect to s. Note that a nonnegative integer x congruent with i modulo s belongs to S if and only if $w(i) \le x$. Thus the pair $(s, \text{Ap}(S,s))$ fully determines the numerical semigroup S (and can be used to easily solve the membership problem to S). This set is in fact one of the most powerfull tools known for numerical semigroups, and it is used almost everywhere in the computation of components and invariants associated to a numerical semigroup. Usually the element s is taken to be the multiplicity, since in this way the resulting Apéry set is the smallest possible.

A *gap* of a numerical semigroup S is a nonnegative integer not belonging to S. The set of gaps of S is usually denoted by $\text{H}(S)$, and clearly determines uniquely S. Note that if x is a gap of S, then so are all the nonnegative integers dividing it. Thus in order to describe S we do not need to know all its gaps, but only those that are maximal with respect to the partial order induced by division in $\mathbb{N}$. These gaps are called *fundamental gaps*.

The largest nonnegative integer not belonging to a numerical semigroup S is the *Frobenius number* of S. If S is the set of nonnegative integers, then clearly its Frobenius number is -1, otherwise its Frobenius number coincides with the maximum of the gaps (or fundamental gaps) of S. In this package we refer to the elements in the semigroup that are less than or equal to the Frobenius number plus 1 as *small elements* of the semigroup. Observe that from the definition, if S is a numerical semigroup with Frobenius number f, then $f + \mathbb{N} \setminus \{0\} \subseteq S$. An integer z is a *pseudo-Frobenius number* of S if $z + S \setminus \{0\} \subseteq S$. Thus the Frobenius number of S is one of its pseudo-Frobenius numbers. The *type* of

a numerical semigroup is the cardinality of the set of its pseudo-Frobenius numbers.

The number of numerical semigroups having a given Frobenius number is finite. The elements in this set of numerical semigroups that are maximal with respect to set inclusion are precisely those numerical semigroups that cannot be expressed as intersection of two other numerical semigroups containing them properly, and thus they are known as *irreducible* numerical semigroups. Clearly, every numerical semigroup is the intersection of (finitely many) irreducible numerical semigroups.

A numerical semigroup S with Frobenius number f is *symmetric* if for every integer x, either $x \in S$ or $f - x \in S$. The set of irreducible numerical semigroups with odd Frobenius number coincides with the set of symmetric numerical semigroups. The numerical semigroup S is *pseudo-symmetric* if f is even and for every integer x not equal to $f/2$ either $x \in S$ or $f - x \in S$. The set of irreducible numerical semigroups with even Frobenius number is precisely the set of pseudo-symmetric numerical semigroups. These two classes of numerical semigroups have been widely studied in the literature due to their nice applications in Algebraic Geometry. This is probably one of the main reasons that made people turn their attention on numerical semigroups again in the last decades. Symmetric numerical semigroups can be also characterized as those with type one, and pseudo-symmetric numerical semigroups are those numerical semigroups with type two and such that its pseudo-Frobenius numbers are its Frobenius number and its Frobenius number divided by two.

Another class of numerical semigroups that catched the attention of researchers working on Algebraic Geometry and Commutative Ring Theory is the class of numerical semigroups with maximal embedding dimension. The *embedding dimension* of a numerical semigroup is the cardinality of its minimal system of generators. It can be shown that the embedding dimension is at most the multiplicity of the numerical semigroup. Thus *maximal embedding dimension* numerical semigroups are those numerical semigroups for which their embedding dimension and multiplicity coincide. These numerical semigroups have nice maximal properties, not only (of course) related to their embedding dimension, but also by means of their presentations. Among maximal embedding dimension there are two classes of numerical semigroups that have been studied due to the connections with the equivalence of algebroid branches. A numerical semigroup S is Arf if for every $x \geq y \geq z \in S$, then $x + y - z \in S$; and it is *saturated* if the following condition holds: if $s, s_1, \ldots, s_r \in S$ are such that $s_i \leq s$ for all $i \in \{1, \ldots, r\}$ and $z_1, \ldots, z_r \in \mathbb{Z}$ are such that $z_1 s_1 + \cdots + z_r s_r \geq 0$, then $s + z_1 s_1 + \cdots + z_r s_r \in S$.

If we look carefully inside the set of fundamental gaps of a numerical semigroup, we see that there are some fulfilling the condition that if they are added to the given numerical semigroup, then the resulting set is again a numerical semigroup. These elements are called *special gaps* of the numerical semigroup. A numerical semigroup other than the set of nonnegative integers is irreducible if and only if it has only a special gap.

The inverse operation to the one described in the above paragraph is that of removing an element of a numerical semigroup. If we want the resulting set to be a numerical semigroup, then the only thing we can remove is a minimal generator.

Let a, b, c, d be positive integers such that $a/b < c/d$, and let $I = [a/b, c/d]$. Then the set $\mathrm{S}(I) = \mathbb{N} \cap \bigcup_{n \geq 0} nI$ is a numerical semigroup. This class of numerical semigroups coincides with that of sets of solutions to equations of the form $Ax \bmod B \leq Cx$ with A, B, C positive integers. A numerical semigroup in this class is said to be *proportionally modular*.

A sequence of positive rational numbers $a_1/b_1 < \cdots < a_n/b_n$ with a_i, b_i positive integers is a *Bézout sequence* if $a_{i+1} b_i - a_i b_{i+1} = 1$ for all $i \in \{1, \ldots, n-1\}$. If $a/b = a_1/b_1 < \cdots < a_n/b_n = c/d$, then $\mathrm{S}([a/b, c/d]) = \langle a_1, \ldots, a_n \rangle$. Bézout sequences are not only interesting for this fact, they have shown to be a major tool in the study of proportionally modular numerical semigroups.

If S is a numerical semigroup and k is a positive integer, then the set $S/k = \{x \in \mathbb{N} \mid kx \in S\}$ is a numerical semigroup, known as the *quotient S by k*.

Let m be a positive integer. A *subadditive* function with period m is a map $f : \mathbb{N} \to \mathbb{N}$ such that $f(0) = 0$, $f(x+y) \leq f(x) + f(y)$ and $f(x+m) = f(x)$. If f is a subadditive function with period m, then the set $M_f = \{x \in \mathbb{N} \mid f(x) \leq x\}$ is a numerical semigroup. Moreover, every numerical semigroup is of this form. Thus a numerical semigroup can be given by a subadditive function with a given period. If S is a numerical semigroup and $s \in S, s \neq 0$, and $\text{Ap}(S,s) = \{w(0), w(1), \ldots, w(s-1)\}$, then $f(x) = w(x \bmod s)$ is a subadditive function with period s such that $M_f = S$.

Let S be a numerical semigroup generated by $\{n_1, \ldots, n_k\}$. Then we can define the following morphism (called sometimes the factorization morphism) by $\varphi : \mathbb{N}^k \to S$, $\varphi(a_1, \ldots, a_k) = a_1 n_1 + \cdots + a_k n_k$. If σ is the kernel congruence of φ (that is, $a \sigma b$ if $\varphi(a) = \varphi(b)$), then S is isomorphic to $\mathbb{N}^k / \sigma$. A *presentation* for S is a system of generators (as a congruence) of σ. If $\{n_1, \ldots, n_p\}$ is a minimal system of generators, then a *minimal presentation* is a presentation such that none of its proper subsets is a presentation. Minimal presentations of numerical semigroups coincide with presentations with minimal cardinality, though in general these two concepts are not the same for an arbitrary commutative semigroup.

A set I of integers is an *ideal relative to a numerical semigroup S* provided that $I + S \subseteq I$ and that there exists $d \in S$ such that $d + I \subseteq S$. If $I \subseteq S$, we simply say that I is an *ideal* of S. If I and J are relative ideals of S, then so is $I - J = \{z \in \mathbb{Z} \mid z + J \subseteq I\}$, and it is tightly related to the operation ":" of ideals in a commutative ring.

In this package we have implemented the functions needed to deal with the elements exposed in this introduction.

Chapter 2

Numerical Semigroups

This chapter describes how to create numerical semigroups in GAP and perform some basic tests.

2.1 Generating Numerical Semigroups

Recalling some definitions from Chapter 1.

A numerical semigroup is a subset of the set $\mathbb{N}$ of nonnegative integers that is closed under addition, contains 0 and whose complement in $\mathbb{N}$ is finite.

We refer to the elements in a numerical semigroup that are less than or equal to the Frobenius number plus 1 as *small elements* of the semigroup.

A *gap* of a numerical semigroup S is a nonnegative integer not belonging to S. The *fundamental gaps* of S are those gaps that are maximal with respect to the partial order induced by division in $\mathbb{N}$.

Given a numerical semigroup S and a nonzero element s in it, one can consider for every integer i ranging from 0 to $s-1$, the smallest element in S congruent with i modulo s, say $w(i)$ (this element exists since the complement of S in $\mathbb{N}$ is finite). Clearly $w(0) = 0$. The set $\mathrm{Ap}(S,s) = \{w(0), w(1), \ldots, w(s-1)\}$ is called the *Apéry set* of S with respect to s.

Let a, b, c, d be positive integers such that $a/b < c/d$, and let $I = [a/b, c/d]$. Then the set $\mathrm{S}(I) = \mathbb{N} \cap \bigcup_{n \geq 0} nI$ is a numerical semigroup. This class of numerical semigroups coincides with that of sets of solutions to equations of the form *Ax mod B $\leq$ Cx* with A, B, C positive integers. A numerical semigroup in this class is said to be *proportionally modular*. If $C = 1$, then it is said to be *modular*.

There are several different ways to specify a numerical semigroup S, namely, by its generators; by its gaps, its fundamental or special gaps by its Apéry set, just to name some. In this section we describe functions that may be used to specify, in one of these ways, a numerical semigroup in GAP.

To create a numerical semigroup in GAP the function `NumericalSemigroup` is used.

2.1.1 NumericalSemigroup

◊ `NumericalSemigroup( Type, List )` (function)

`Type`

May be `"generators"`, `"minimalgenerators"`, `"modular"`, `"propmodular"`, `"elements"`, `"gaps"`, `"fundamentalgaps"`, `"subadditive"` or `"apery"` according to whether the semigroup is to be given by means of a condition of the form *ax mod m <= x*, a system of generators, a condition

9

of the form *ax mod m <= cx*, a set of all elements up to the Frobenius number +1, the set of gaps, the set of fundamental gaps, a periodic subaditive function or the Apéry set.

When no string is given as first argument it is assumed that the numerical semigroup will be given by means of a set of generators.

```
List
```

When the semigroup is given through a set of generators, this set may be given as a list or through its individual elements.

The set of all elements up to the Frobenius number +1, the set of gaps, the set of fundamental gaps or the Apéry set are given through lists.

A periodic subadditive function with period *m* is given through the list of images of the elements, from 1 to *m*. The image of *m* has to be 0.

```
 ─────────────────────── Example ───────────────────────
 gap> s1 := NumericalSemigroup("generators",3,5,7);
 <Numerical semigroup with 3 generators>
 gap> s2 := NumericalSemigroup("generators",[3,5,7]);
 <Numerical semigroup with 3 generators>
 gap> s1=s2;
 true
 gap> s := NumericalSemigroup("minimalgenerators",3,7);
 <Numerical semigroup with 2 generators>
 gap> s := NumericalSemigroup("modular",3,5);
 <Modular numerical semigroup satisfying 3x mod 5 <= x >
 gap> s1 := NumericalSemigroup("generators",2,5);
 <Numerical semigroup with 2 generators>
 gap> s = s1;
 true


 ....................................
```

2.1.2 ModularNumericalSemigroup

◊ ModularNumericalSemigroup(a, b)   (function)

Given two positive integers a and b, this function returns a modular numerical semigroup satisfying *ax mod b <= x*.

```
 ─────────────────────── Example ───────────────────────
 gap> ModularNumericalSemigroup(3,7);
 <Modular numerical semigroup satisfying 3x mod 7 <= x >
```

2.1.3 ProportionallyModularNumericalSemigroup

◊ ProportionallyModularNumericalSemigroup(a, b, c)   (function)

Given three positive integers a, b and c, this function returns a proportionally modular numerical semigroup satisfying *ax mod b <= cx*.

```
 ─────────────────────── Example ───────────────────────
 gap> ProportionallyModularNumericalSemigroup(3,7,12);
 <Proportionally modular numerical semigroup satisfying 3x mod 7 <= 12x >
```

2.1.4 NumericalSemigroupByGenerators

◇ NumericalSemigroupByGenerators(List)   (function)
◇ NumericalSemigroupByMinimalGenerators(List)   (function)
◇ NumericalSemigroupByMinimalGeneratorsNC(List)   (function)
◇ NumericalSemigroupByInterval(List)   (function)
◇ NumericalSemigroupByOpenInterval(List)   (function)
◇ NumericalSemigroupBySubAdditiveFunction(List)   (function)
◇ NumericalSemigroupByAperyList(List)   (function)
◇ NumericalSemigroupBySmallElements(List)   (function)
◇ NumericalSemigroupByGaps(List)   (function)
◇ NumericalSemigroupByFundamentalGaps(List)   (function)

The function NumericalSemigroup (2.1.1) is a front-end for these functions. The argument of each of these functions is a list representing an entity of the type to which the function's name refers.

```
————— Example —————
gap> s:=NumericalSemigroup(3,11);
<Numerical semigroup with 2 generators>
gap> GapsOfNumericalSemigroup(s);
[ 1, 2, 4, 5, 7, 8, 10, 13, 16, 19 ]
gap> t:=NumericalSemigroupByGaps(last);
<Numerical semigroup>
gap> s=t;
true

gap> AperyListOfNumericalSemigroupWRTElement(s,20);;
gap> t:=NumericalSemigroupByAperyList(last);
<Numerical semigroup>
gap> s=t;
true
...
```

2.2 Some basic tests

This section describes some basic tests on numerical semigroups. The first described tests refer to the way the semigroup was created. Then are presented functions to test if a given list represents the small elements, gaps or the Apéry set (see 1) of a numerical semigroup; to test if an integer belongs to a numerical semigroup and if a numerical semigroup is a subsemigroup of another one.

2.2.1 IsNumericalSemigroup

◇ IsNumericalSemigroup(NS)   (attribute)
◇ IsNumericalSemigroupByGenerators(NS)   (attribute)
◇ IsNumericalSemigroupByMinimalGenerators(NS)   (attribute)
◇ IsNumericalSemigroupByInterval(NS)   (attribute)
◇ IsNumericalSemigroupByOpenInterval(NS)   (attribute)
◇ IsNumericalSemigroupBySubAdditiveFunction(NS)   (attribute)
◇ IsNumericalSemigroupByAperyList(NS)   (attribute)

◇ `IsNumericalSemigroupBySmallElements( NS )` (attribute)

◇ `IsNumericalSemigroupByGaps( NS )` (attribute)

◇ `IsNumericalSemigroupByFundamentalGaps( NS )` (attribute)

◇ `IsProportionallyModularNumericalSemigroup( NS )` (attribute)

◇ `IsModularNumericalSemigroup( NS )` (attribute)

`NS` is a numerical semigroup and these attributes are available (their names should be self explanatory).

```
───────────────────────── Example ─────────────────────────
gap> s:=NumericalSemigroup(3,7);
<Numerical semigroup with 2 generators>
gap> AperyListOfNumericalSemigroupWRTElement(s,30);;
gap> t:=NumericalSemigroupByAperyList(last);
<Numerical semigroup>
gap> IsNumericalSemigroupByGenerators(s);
true
gap> IsNumericalSemigroupByGenerators(t);
false
gap> IsNumericalSemigroupByAperyList(s);
true
gap> IsNumericalSemigroupByAperyList(t);
true
```

2.2.2 RepresentsSmallElementsOfNumericalSemigroup

◇ `RepresentsSmallElementsOfNumericalSemigroup( L )` (attribute)

Tests if the list `L` (which has to be a set) may represent the "small" elements of a numerical semigroup.

```
───────────────────────── Example ─────────────────────────
gap> L:=[ 0, 3, 6, 9, 11, 12, 14, 15, 17, 18, 20 ];
[ 0, 3, 6, 9, 11, 12, 14, 15, 17, 18, 20 ]
gap> RepresentsSmallElementsOfNumericalSemigroup(L);
true
gap> L:=[ 6, 9, 11, 12, 14, 15, 17, 18, 20 ];
[ 6, 9, 11, 12, 14, 15, 17, 18, 20 ]
gap> RepresentsSmallElementsOfNumericalSemigroup(L);
false
```

2.2.3 RepresentsGapsOfNumericalSemigroup

◇ `RepresentsGapsOfNumericalSemigroup( L )` (attribute)

Tests if the list `L` may represent the gaps (see 1) of a numerical semigroup.

```
───────────────────────── Example ─────────────────────────
gap> s:=NumericalSemigroup(3,7);
<Numerical semigroup with 2 generators>
gap> L:=GapsOfNumericalSemigroup(s);
[ 1, 2, 4, 5, 8, 11 ]
gap> RepresentsGapsOfNumericalSemigroup(L);
```

```
true
gap> L:=Set(List([1..21],i->RandomList([1..50])));
[ 2, 6, 7, 8, 10, 12, 14, 19, 24, 28, 31, 35, 42, 50 ]
gap> RepresentsGapsOfNumericalSemigroup(L);
false
```

2.2.4 IsAperyListOfNumericalSemigroup

◊ IsAperyListOfNumericalSemigroup(L)   (function)

Tests whether a list L of integers may represent the Apéry list of a numerical semi-group. It returns true when the periodic function represented by L is subadditive (see RepresentsPeriodicSubAdditiveFunction (A.2.1)) and the remainder of the division of L[i] by the length of L is i and returns false otherwise (the crieterium used is the one explained in [Ros96b]).

```
                              Example
gap> IsAperyListOfNumericalSemigroup([0,21,7,28,14]);
true
```

2.2.5 IsSubsemigroupOfNumericalSemigroup

◊ IsSubsemigroupOfNumericalSemigroup(S, T)   (function)

S and T are numerical semigroups. Tests whether T is contained in S.

```
                              Example
gap> S := NumericalSemigroup("modular", 5,53);
<Modular numerical semigroup satisfying 5x mod 53 <= x >
gap> T := NumericalSemigroup(2,3);
<Numerical semigroup with 2 generators>
gap> IsSubsemigroupOfNumericalSemigroup(T,S);
true
gap> IsSubsemigroupOfNumericalSemigroup(S,T);
false
```

2.2.6 BelongsToNumericalSemigroup

◊ BelongsToNumericalSemigroup(n, S)   (operation)

n is an integer and S is a numerical semigroup. Tests whether n belongs to S. n in S is the short for BelongsToNumericalSemigroup(n,S).

```
                              Example
gap> S := NumericalSemigroup("modular", 5,53);
<Modular numerical semigroup satisfying 5x mod 53 <= x >
gap> BelongsToNumericalSemigroup(15,S);
false
gap> 15 in S;
false
gap> SmallElementsOfNumericalSemigroup(S);
[ 0, 11, 12, 13, 22, 23, 24, 25, 26, 32, 33, 34, 35, 36, 37, 38, 39, 43 ]
gap> BelongsToNumericalSemigroup(13,S);
```

```
true
gap> 13 in S;
true
```

Chapter 3

Basic operations with numerical semigroups

3.1 The definitions

3.1.1 MultiplicityOfNumericalSemigroup

◊ MultiplicityOfNumericalSemigroup(NS)   (attribute)

NS is a numerical semigroup. Returns the multiplicity of NS, which is the smallest positive integer belonging to NS.

```
━━━━━━━━━━━━━━━━━━━━━━━ Example ━━━━━━━━━━━━━━━━━━━━━━━
gap> S := NumericalSemigroup("modular", 7,53);
<Modular numerical semigroup satisfying 7x mod 53 <= x >
gap> MultiplicityOfNumericalSemigroup(S);
8
```

3.1.2 GeneratorsOfNumericalSemigroup

◊ GeneratorsOfNumericalSemigroup(S)   (function)
◊ GeneratorsOfNumericalSemigroupNC(S)   (function)
◊ MinimalGeneratingSystemOfNumericalSemigroup(S)   (attribute)

S is a numerical semigroup. GeneratorsOfNumericalSemigroup returns a set of generators of S, which may not be minimal. GeneratorsOfNumericalSemigroupNC returns the set of generators recorded in S!.generators, which may not be minimal. MinimalGeneratingSystemOfNumericalSemigroup returns the minimal set of generators of S.

```
━━━━━━━━━━━━━━━━━━━━━━━ Example ━━━━━━━━━━━━━━━━━━━━━━━
gap> S := NumericalSemigroup("modular", 5,53);
<Modular numerical semigroup satisfying 5x mod 53 <= x >
gap> GeneratorsOfNumericalSemigroup(S);
[ 11, 12, 13, 32, 53 ]
gap> S := NumericalSemigroup(3, 5, 53);
<Numerical semigroup with 3 generators>
gap> GeneratorsOfNumericalSemigroup(S);
[ 3, 5, 53 ]
```

```
gap> MinimalGeneratingSystemOfNumericalSemigroup(S);
[ 3, 5 ]
```

3.1.3 SmallElementsOfNumericalSemigroup

◊ SmallElementsOfNumericalSemigroup(NS)   (attribute)

NS is a numerical semigroup. It returns the list of small elements of NS. Of course, the time consumed to return a result may depend on the way the semigroup is given.

```
————— Example —————
gap> SmallElementsOfNumericalSemigroup(NumericalSemigroup(3,5,7));
[ 0, 3, 5 ]
```

3.1.4 FirstElementsOfNumericalSemigroup

◊ FirstElementsOfNumericalSemigroup(n, NS)   (function)

NS is a numerical semigroup. It returns the list with the first n elements of NS.

```
————— Example —————
gap> FirstElementsOfNumericalSemigroup(2,NumericalSemigroup(3,5,7));
[ 0, 3 ]
gap> FirstElementsOfNumericalSemigroup(10,NumericalSemigroup(3,5,7));
[ 0, 3, 5, 6, 7, 8, 9, 10, 11, 12 ]
```

3.1.5 AperyListOfNumericalSemigroupWRTElement

◊ AperyListOfNumericalSemigroupWRTElement(S, m)   (operation)

S is a numerical semigroup and m is a positive element of S. Computes the Apéry list of S wrt m. It contains for every $i \in \{0,\ldots,m-1\}$, in the $i+1$th position, the smallest element in the semigroup congruent with i modulo m.

```
————— Example —————
gap> S := NumericalSemigroup("modular", 5,53);
<Modular numerical semigroup satisfying 5x mod 53 <= x >
gap> AperyListOfNumericalSemigroupWRTElement(S,12);
[ 0, 13, 26, 39, 52, 53, 54, 43, 32, 33, 22, 11 ]
```

3.1.6 DrawAperyListOfNumericalSemigroup

◊ DrawAperyListOfNumericalSemigroup(ap)   (function)

ap is the Apéry list of a numerical semigroup. This function draws the graph (ap,E) where the edge $u->v$ is in E iff $v-u$ is in *ap*. To use this function, Graphviz (http://www.graphviz.org) should be installed and also Evince (http://www.gnome.org/projects/evince/) or ggv (http://directory.fsf.org/ggv.html).

3.1.7 AperyListOfNumericalSemigroupAsGraph

◊ AperyListOfNumericalSemigroupAsGraph(ap) (function)

ap is the Apéry list of a numerical semigroup. This function returns the adjacency list of the graph (ap, E) where the edge $u-> v$ is in E iff $v - u$ is in ap. The 0 is ignored.

```
———————————— Example ————————————
gap> s:=NumericalSemigroup(3,7);
<Numerical semigroup with 2 generators>
gap> AperyListOfNumericalSemigroupWRTElement(s,10);
[ 0, 21, 12, 3, 14, 15, 6, 7, 18, 9 ]
gap> AperyListOfNumericalSemigroupAsGraph(last);
[ ,, [ 3, 6, 9, 12, 15, 18, 21 ],,, [ 6, 9, 12, 15, 18, 21 ],
[ 7, 14, 21 ],, [ 9, 12, 15, 18, 21 ],,, [ 12, 15, 18, 21 ],,
[ 14, 21 ], [ 15, 18, 21 ],,, [ 18, 21 ],,, [ 21 ] ]
```

3.2 Frobenius Number

The largest nonnegative integer not belonging to a numerical semigroup S is the *Frobenius number* of S. If S is the set of nonnegative integers, then clearly its Frobenius number is -1, otherwise its Frobenius number coincides with the maximum of the gaps (or fundamental gaps) of S. An integer z is a *pseudo-Frobenius number* of S if $z + S \setminus \{0\} \subseteq S$.

3.2.1 FrobeniusNumberOfNumericalSemigroup

◊ FrobeniusNumberOfNumericalSemigroup(NS) (attribute)

NS is a numerical semigroup. It returns the Frobenius number of NS. Of course, the time consumed to return a result may depend on the way the semigroup is given or on the knowledge already produced on the semigroup.

```
———————————— Example ————————————
gap> FrobeniusNumberOfNumericalSemigroup(NumericalSemigroup(3,5,7));
4
```

3.2.2 FrobeniusNumber

◊ FrobeniusNumber(NS) (attribute)

This is just a synonym of FrobeniusNumberOfNumericalSemigroup (3.2.1).

3.2.3 PseudoFrobeniusOfNumericalSemigroup

◊ PseudoFrobeniusOfNumericalSemigroup(S) (attribute)

S is a numerical semigroup. It returns set of pseudo-Frobenius numbers of S.

```
———————————— Example ————————————
gap> S := NumericalSemigroup("modular", 5,53);
<Modular numerical semigroup satisfying 5x mod 53 <= x >
```

```
gap> PseudoFrobeniusOfNumericalSemigroup(S);
[ 21, 40, 41, 42 ]
```

3.3 Gaps

A *gap* of a numerical semigroup *S* is a nonnegative integer not belonging to *S*. The *fundamental gaps* of *S* are those gaps that are maximal with respect to the partial order induced by division in $\mathbb{N}$. The *special gaps* of a numerical semigroup *S*, are those fundamental gaps such that if they are added to the given numerical semigroup, then the resulting set is again a numerical semigroup.

3.3.1 GapsOfNumericalSemigroup

◊ GapsOfNumericalSemigroup(NS)   (attribute)

NS is a numerical semigroup. It returns the set of gaps of NS.

──────────────── Example ────────────────
```
gap> GapsOfNumericalSemigroup(NumericalSemigroup(3,5,7));
[ 1, 2, 4 ]
```

3.3.2 FundamentalGapsOfNumericalSemigroup

◊ FundamentalGapsOfNumericalSemigroup(S)   (attribute)

S is a numerical semigroup. It returns the set of fundamental gaps of S.

──────────────── Example ────────────────
```
gap> S := NumericalSemigroup("modular", 5,53);
<Modular numerical semigroup satisfying 5x mod 53 <= x >
gap> FundamentalGapsOfNumericalSemigroup(S);
[ 16, 17, 18, 19, 27, 28, 29, 30, 31, 40, 41, 42 ]
gap> GapsOfNumericalSemigroup(S);
[ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 18, 19, 20, 21, 27, 28, 29,
  30, 31, 40, 41, 42 ]
```

3.3.3 SpecialGapsOfNumericalSemigroup

◊ SpecialGapsOfNumericalSemigroup(S)   (attribute)

S is a numerical semigroup. It returns the special gaps of S.

──────────────── Example ────────────────
```
gap> S := NumericalSemigroup("modular", 5,53);
<Modular numerical semigroup satisfying 5x mod 53 <= x >
gap> SpecialGapsOfNumericalSemigroup(S);
[ 40, 41, 42 ]
```

Chapter 4

Presentations of Numerical Semigroups

In this chapter we explain how to compute a minimal presentation of a numerical semigroup. There are three functions involved in this process.

4.1 Presentations of Numerical Semigroups

4.1.1 FortenTruncatedNCForNumericalSemigroups

◇ FortenTruncatedNCForNumericalSemigroups(L)   (function)

L contains the list of coefficients of a single linear equation. This function gives a minimal generator of the affine semigroup of nonnegative solutions of this equation with the first coordinate equal to one (see [CD94]). Returns `fail` if no solution exists.

```
———————————————— Example ————————————————
gap> FortenTruncatedNCForNumericalSemigroups([ -57, 3 ]);
[ 1, 19 ]
gap> FortenTruncatedNCForNumericalSemigroups([ -57, 33 ]);
fail
gap> FortenTruncatedNCForNumericalSemigroups([ -57, 19 ]);
[ 1, 3 ]
```

4.1.2 MinimalPresentationOfNumericalSemigroup

◇ MinimalPresentationOfNumericalSemigroup(S)   (function)

S is a numerical semigroup. The output is a list of lists with two elements. Each list of two elements represents a relation between the minimal generators of the numerical semigroup. If $\{\{x_1, y_1\}, \ldots, \{x_k, y_k\}\}$ is the output and $\{m_1, \ldots, m_n\}$ is the minimal system of generators of the numerical semigroup, then $\{x_i, y_i\} = \{\{a_{i_1}, \ldots, a_{i_n}\}, \{b_{i_1}, \ldots, b_{i_n}\}\}$ and $a_{i_1} m_1 + \cdots + a_{i_n} m_n = b_{i_1} m_1 + \cdots + b_{i_n} m_n$.

Any other relation among the minimal generators of the semigroup can be deduced from the ones given in the output.

The algorithm implemented is described in [Ros96a] (see also [RGS99]).

```
———————————————— Example ————————————————
gap> s:=NumericalSemigroup(3,5,7);
<Numerical semigroup with 3 generators>
```

```
gap> MinimalPresentationOfNumericalSemigroup(s);
[ [ [ 1, 0, 1 ], [ 0, 2, 0 ] ], [ [ 4, 0, 0 ], [ 0, 1, 1 ] ],
  [ [ 3, 1, 0 ], [ 0, 0, 2 ] ] ]
```

The first element in the list means that $1 \times 3 + 1 \times 7 = 2 \times 5$, and so on.

4.1.3 GraphAssociatedToElementInNumericalSemigroup

◊ GraphAssociatedToElementInNumericalSemigroup(n, S)   (function)

S is a numerical semigroup and n is an element in S.

The output is a pair. If $\{m_1, \ldots, m_n\}$ is the set of minimal generators of S, then the first component is the set of vertices of the graph associated to n in S, that is, the set $\{m_i \mid n - m_i \in S\}$, and the second component is the set of edges of this graph, that is, $\{\{m_i, m_j\} \mid n - (m_i + m_j) \in S\}$.

This function is used to compute a minimal presentation of the numerical semigroup S, as explained in [Ros96a].

```
———————————————— Example ————————————————
gap> s:=NumericalSemigroup(3,5,7);
<Numerical semigroup with 3 generators>
gap> GraphAssociatedToElementInNumericalSemigroup(10,s);
[ [ 3, 5, 7 ], [ [ 3, 7 ], [ 5, 5 ], [ 7, 3 ] ] ]
```

Chapter 5

Constructing numerical semigroups from others

5.1 Adding and removing elements of a numerical semigroup

In this section we show how to construct new numerical semigroups from a given numerical semigroup. Two dual operations are presented. The first one removes a minimal generator from a numerical semigroup. The second adds a special gap to a semigroup (see [JCRJM03]).

5.1.1 RemoveMinimalGeneratorFromNumericalSemigroup

◇ RemoveMinimalGeneratorFromNumericalSemigroup(n, S)   (function)

   S is a numerical semigroup and n is one if its minimal generators.

   The output is the numerical semigroup $S \setminus \{n\}$ (see [JCRJM03]; $S \setminus \{n\}$ is a numerical semigroup if and only if n is a minimal generator of S).

```
——————————— Example ———————————
gap> s:=NumericalSemigroup(3,5,7);
<Numerical semigroup with 3 generators>
gap> RemoveMinimalGeneratorFromNumericalSemigroup(7,s);
<Numerical semigroup with 3 generators>
gap> MinimalGeneratingSystemOfNumericalSemigroup(last);
[ 3, 5 ]

```

5.1.2 AddSpecialGapOfNumericalSemigroup

◇ AddSpecialGapOfNumericalSemigroup(g, S)   (function)

   S is a numerical semigroup and g is a special gap of S

   The output is the numerical semigroup $S \cup \{g\}$ (see [JCRJM03], where it is explained why this set is a numerical semigroup).

```
——————————— Example ———————————
gap> s:=NumericalSemigroup(3,5,7);
<Numerical semigroup with 3 generators>
```

```
gap> s2:=RemoveMinimalGeneratorFromNumericalSemigroup(5,s);
<Numerical semigroup with 3 generators>
gap> s3:=AddSpecialGapOfNumericalSemigroup(5,s2);
<Numerical semigroup>
gap> SmallElementsOfNumericalSemigroup(s) =
> SmallElementsOfNumericalSemigroup(s3);
true
gap> s=s3;
true
```

5.1.3 IntersectionOfNumericalSemigroups

◇ IntersectionOfNumericalSemigroups(S, T)   (function)

S and T are numerical semigroups. Computes the intersection of S and T (which is a numerical semigroup).

```
_____ Example _____
gap> S := NumericalSemigroup("modular", 5,53);
<Modular numerical semigroup satisfying 5x mod 53 <= x >
gap> T := NumericalSemigroup(2,17);
<Numerical semigroup with 2 generators>
gap> SmallElementsOfNumericalSemigroup(S);
[ 0, 11, 12, 13, 22, 23, 24, 25, 26, 32, 33, 34, 35, 36, 37, 38, 39, 43 ]
gap> SmallElementsOfNumericalSemigroup(T);
[ 0, 2, 4, 6, 8, 10, 12, 14, 16 ]
gap> IntersectionOfNumericalSemigroups(S,T);
<Numerical semigroup>
gap> SmallElementsOfNumericalSemigroup(last);
[ 0, 12, 22, 23, 24, 25, 26, 32, 33, 34, 35, 36, 37, 38, 39, 43 ]
```

5.1.4 QuotientOfNumericalSemigroup

◇ QuotientOfNumericalSemigroup(S, n)   (function)

S is a numerical semigroup and n is an integer. Computes the quotient of S by n, that is, the set $\{x \in \mathbb{N} \mid nx \in S\}$, which is again a numerical semigroup. S / n may be used as a short for QuotientOfNumericalSemigroup(S, n).

```
_____ Example _____
gap> s:=NumericalSemigroup(3,29);
<Numerical semigroup with 2 generators>
gap> SmallElementsOfNumericalSemigroup(s);
[ 0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 29, 30, 32, 33, 35, 36, 38,
  39, 41, 42, 44, 45, 47, 48, 50, 51, 53, 54, 56 ]
gap> t:=QuotientOfNumericalSemigroup(s,7);
<Numerical semigroup>
gap> SmallElementsOfNumericalSemigroup(t);
[ 0, 3, 5, 6, 8 ]
gap> u := s / 7;
<Numerical semigroup>
```

```
gap> SmallElementsOfNumericalSemigroup(u);
[ 0, 3, 5, 6, 8 ]
```

5.2 Constructing the set of all numerical semigroups containing a given numerical semigroup

In order to construct the set of numerical semigroups containing a fixed numerical semigroup S, one first constructs its unitary extensions, that is to say, the sets $S \cup \{g\}$ that are numerical semigroups with g a positive integer. This is achieved by constructing the special gaps of the semigroup, and then adding each of them to the numerical semigroup. Then we repeat the process for each of this new numerical semigroups until we reach $\mathbb{N}$.

These procedures are described in [JCRJM03].

5.2.1 OverSemigroupsNumericalSemigroup

◊ OverSemigroupsNumericalSemigroup(s)   (function)

s is a numerical semigroup. The output is the set of numerical semigroups containing it.

```
───────────────────────── Example ─────────────────────────
gap> OverSemigroupsNumericalSemigroup(NumericalSemigroup(3,5,7));
[ <Numerical semigroup with 1 generators>, <Numerical semigroup>,
  <Numerical semigroup>, <Numerical semigroup with 3 generators> ]
]]

gap> List(last,s->MinimalGeneratingSystemOfNumericalSemigroup(s));
[ [ 1 ], [ 3, 4, 5 ], [ 2, 3 ], [ 3, 5, 7 ] ]
```

5.2.2 NumericalSemigroupsWithFrobeniusNumber

◊ NumericalSemigroupsWithFrobeniusNumber(f)   (function)

f is an non zero integer greater than or equal to -1. The output is the set of numerical semigroups with Frobenius number f. The algorithm implemented is given in [JCRM04].

```
───────────────────────── Example ─────────────────────────
gap> Length(NumericalSemigroupsWithFrobeniusNumber(20));
900
```

Chapter 6

Irreducible numerical semigroups

6.1 Irreducible numerical semigroups

An irreducible numerical semigroup is a semigroup that cannot be expressed as the intersection of numerical semigroups properly containing it.

It is not difficult to prove that a semigroup is irreducible if and only if it is maximal (with respect to set inclusion) in the set of all numerical semigroup having its same Frobenius number (see [RB03]). Hence, according to [RFH87] (respectively [VBF97]), symmetric (respectively pseudo-symmetric) numerical semigroups are those irreducible numerical semigroups with odd (respectively even) Frobenius number.

In [JCRJM03] it is shown that a numerical semigroup is irreducible if and only if it has only one special gap. We use this characterization.

In this section we show how to construct the set of all numerical semigroups with a given Frobenius number. First we construct an irreducible numerical semigroup with the given Frobenius number (as explained in [Ros04]), and then we construct the rest from it. That is why we have separated both functions.

Every numerical semigroup can be expressed as an intersection of irreducible numerical semigroups. If S can be expressed as $S = S_1 \cap \cdots \cap S_n$, with S_i irreducible numerical semigroups, and no factor can be removed, then we say that this decomposition is minimal. Minimal decompositions can be computed by using Algorithm 26 in [JCRJM03].

6.1.1 IsIrreducibleNumericalSemigroup

◇ IsIrreducibleNumericalSemigroup(s)   (function)

s is a numerical semigroup. The output is true if s is irreducible, false otherwise.

```
_____ Example _____
gap> IsIrreducibleNumericalSemigroup(NumericalSemigroup(4,6,9));
true
gap> IsIrreducibleNumericalSemigroup(NumericalSemigroup(4,6,7,9));
false
```

6.1.2   IsSymmetricNumericalSemigroup

◊ IsSymmetricNumericalSemigroup(s)   (function)

s is a numerical semigroup. The output is true if s is symmetric, false otherwise.

```
———————————————— Example ————————————————
gap> IsSymmetricNumericalSemigroup(NumericalSemigroup(10,23));
true
gap> IsSymmetricNumericalSemigroup(NumericalSemigroup(10,11,23));
false
```

6.1.3   IsPseudoSymmetricNumericalSemigroup

◊ IsPseudoSymmetricNumericalSemigroup(s)   (function)

s is a numerical semigroup. The output is true if s is pseudo-symmetric, false otherwise.

```
———————————————— Example ————————————————
gap> IsPseudoSymmetricNumericalSemigroup(NumericalSemigroup(6,7,8,9,11));
true
gap> IsPseudoSymmetricNumericalSemigroup(NumericalSemigroup(4,6,9));
false
```

6.1.4   AnIrreducibleNumericalSemigroupWithFrobeniusNumber

◊ AnIrreducibleNumericalSemigroupWithFrobeniusNumber(f)   (function)

f is an integer greater than or equal to -1. The output is an irreducible numerical semigroup with frobenius number f. From the way the procedure is implemented, the resulting semigroup has at most four generators (see [Ros04]).

```
———————————————— Example ————————————————
gap> FrobeniusNumber(AnIrreducibleNumericalSemigroupWithFrobeniusNumber(28));
28
```

6.1.5   IrreducibleNumericalSemigroupsWithFrobeniusNumber

◊ IrreducibleNumericalSemigroupsWithFrobeniusNumber(f)   (function)

f is an integer greater than or equal to -1. The output is the set of all irreducible numerical semigroups with frobenius number f.

```
———————————————— Example ————————————————
gap> Length(IrreducibleNumericalSemigroupsWithFrobeniusNumber(39));
227
```

6.1.6 DecomposeIntoIrreducibles

◊ DecomposeIntoIrreducibles(s)   (function)

s is a numerical semigroup. The output is a set of irreducible numerical semigroups containing it. These elements appear in a minimal decomposition of s as intersection into irreducibles.

```
————————————————————— Example —————————————————————
gap> DecomposeIntoIrreducibles(NumericalSemigroup(5,6,8));
[ <Numerical semigroup>, <Numerical semigroup> ]
```

Chapter 7

Ideals of numerical semigroups

7.1 Ideals of numerical semigroups

Let S be a numerical semigroup. A set I of integers is an *ideal relative* to a numerical semigroup S provided that $I + S \subseteq I$ and that there exists $d \in S$ such that $d + I \subseteq S$.

If $\{i_1, \ldots, i_k\}$ is a subset of $\mathbb{Z}$, then the set $I = \{i_1, \ldots, i_k\} + S = \bigcup_{n=1}^{k} i_n + S$ is an ideal relative to S, and $\{i_1, \ldots, i_k\}$ is a system of generators of I. A system of generators M is minimal if no proper subset of M generates the same ideal. Usually, ideals are specified by means of its generators and the ambient numerical semigroup to which they are ideals (for more information see for instance [VBF97]).

7.1.1 IdealOfNumericalSemigroup

◇ IdealOfNumericalSemigroup(l, S)   (function)

   S is a numerical semigroup and l a list of integers.

   The output is the ideal of S generated by l

   There are several shortcuts for this function, as shown in the example.

```
———————————————— Example ————————————————
gap> IdealOfNumericalSemigroup([3,5],NumericalSemigroup(9,11));
<Ideal of numerical semigroup>
gap> [3,5]+NumericalSemigroup(9,11);
<Ideal of numerical semigroup>
gap> last=last2;
true
gap> 3+NumericalSemigroup(5,9);
<Ideal of numerical semigroup>
```

7.1.2 IsIdealOfNumericalSemigroup

◇ IsIdealOfNumericalSemigroup(Obj)   (function)

   Tests if the object Obj is an ideal of a numerical semigroup.

```
———————————————— Example ————————————————
gap> I:=[1..7]+NumericalSemigroup(7,19);
<Ideal of numerical semigroup>
```

```
gap> IsIdealOfNumericalSemigroup(I);
true
gap> IsIdealOfNumericalSemigroup(2);
false
```

7.1.3   MinimalGeneratingSystemOfIdealOfNumericalSemigroup

◇ MinimalGeneratingSystemOfIdealOfNumericalSemigroup(I)   (function)

I is an ideal of a numerical semigroup.

The output is the minimal system of generators of I.

```
────────────────────── Example ──────────────────────
gap> I:=[3,5,9]+NumericalSemigroup(2,11);
<Ideal of numerical semigroup>
gap> MinimalGeneratingSystemOfIdealOfNumericalSemigroup(I);
[ 3 ]
```

7.1.4   GeneratorsOfIdealOfNumericalSemigroup

◇ GeneratorsOfIdealOfNumericalSemigroup(I)   (function)
◇ GeneratorsOfIdealOfNumericalSemigroupNC(I)   (function)

I is an ideal of a numerical semigroup.

The output of GeneratorsOfIdealOfNumericalSemigroup is a system of generators of the ideal. If the minimal system of generators is known, then it is used as output. GeneratorsOfIdealOfNumericalSemigroupNC always returns the set of generators stored in I!.generators.

```
────────────────────── Example ──────────────────────
gap> I:=[3,5,9]+NumericalSemigroup(2,11);
<Ideal of numerical semigroup>
gap> GeneratorsOfIdealOfNumericalSemigroup(I);
[ 3, 5, 9 ]
gap> MinimalGeneratingSystemOfIdealOfNumericalSemigroup(I);
[ 3 ]
gap> GeneratorsOfIdealOfNumericalSemigroup(I);
[ 3 ]
gap> GeneratorsOfIdealOfNumericalSemigroupNC(I);
[ 3, 5, 9 ]
```

7.1.5   AmbientNumericalSemigroupOfIdeal

◇ AmbientNumericalSemigroupOfIdeal(I)   (function)

I is an ideal of a numerical semigroup, say *S*.

The output is *S*.

```
 _____ Example _____
gap> I:=[3,5,9]+NumericalSemigroup(2,11);
<Ideal of numerical semigroup>
gap> AmbientNumericalSemigroupOfIdeal(I);
<Numerical semigroup with 2 generators>
```

7.1.6 SmallElementsOfIdealOfNumericalSemigroup

◇ SmallElementsOfIdealOfNumericalSemigroup(I)   (function)

I is an ideal of a numerical semigroup.

The output is a list with the elements in I that are less than or equal to the greatest integer not belonging to the ideal plus one.

```
 _____ Example _____
gap> I:=[3,5,9]+NumericalSemigroup(2,11);
<Ideal of numerical semigroup>
gap> SmallElementsOfIdealOfNumericalSemigroup(I);
[ 3, 5, 7, 9, 11, 13 ]
gap> J:=[2,11]+NumericalSemigroup(2,11);
<Ideal of numerical semigroup>
gap> SmallElementsOfIdealOfNumericalSemigroup(J);
[ 2, 4, 6, 8, 10 ]
```

7.1.7 BelongsToIdealOfNumericalSemigroup

◇ BelongsToIdealOfNumericalSemigroup(n, I)   (function)

I is an ideal of a numerical semigroup, n is an integer.

The output is true if n belongs to I.

n in I can be used for short.

```
 _____ Example _____
gap> J:=[2,11]+NumericalSemigroup(2,11);
<Ideal of numerical semigroup>
gap> BelongsToIdealOfNumericalSemigroup(9,J);
false
gap> 9 in J;
false
gap> BelongsToIdealOfNumericalSemigroup(10,J);
true
gap> 10 in J;
true
```

7.1.8 SumIdealsOfNumericalSemigroup

◇ SumIdealsOfNumericalSemigroup(I, J)   (function)

    I, J are ideals of a numerical semigroup.
    The output is the sum of both ideals $\{i+j \mid i \in \texttt{I}, j \in \texttt{J}\}$.
    I + J is a synonym of this function.

```
 _____ Example _____
gap> I:=[3,5,9]+NumericalSemigroup(2,11);
<Ideal of numerical semigroup>
gap> J:=[2,11]+NumericalSemigroup(2,11);
<Ideal of numerical semigroup>
gap> I+J;
<Ideal of numerical semigroup>
gap> MinimalGeneratingSystemOfIdealOfNumericalSemigroup(last);
[ 5, 14 ]
gap> SumIdealsOfNumericalSemigroup(I,J);
<Ideal of numerical semigroup>
gap> MinimalGeneratingSystemOfIdealOfNumericalSemigroup(last);
[ 5, 14 ]
```

7.1.9 MultipleOfIdealOfNumericalSemigroup

◇ MultipleOfIdealOfNumericalSemigroup(n, I)   (function)

    I is an ideal of a numerical semigroup, n is a non negative integer.
    The output is the ideal I + ⋯ + I (n times).
    n * I can be used for short.

```
 _____ Example _____
gap> I:=[0,1]+NumericalSemigroup(3,5,7);
<Ideal of numerical semigroup>
gap> MinimalGeneratingSystemOfIdealOfNumericalSemigroup(2*I);
[ 0, 1, 2 ]
```

7.1.10 SubtractIdealsOfNumericalSemigroup

◇ SubtractIdealsOfNumericalSemigroup(I, J)   (function)

    I, J are ideals of a numerical semigroup.
    The output is the ideal $\{z \in \mathbb{Z} \mid z + \texttt{J} \subseteq \texttt{I}\}$.
    I - J is a synonym of this function. The following example appears in [HS04].

```
 _____ Example _____
gap> S:=NumericalSemigroup(14, 15, 20, 21, 25);
<Numerical semigroup with 5 generators>
gap> I:=[0,1]+S;
<Ideal of numerical semigroup>
```

```
gap> II:=(0+S)-I;
<Ideal of numerical semigroup>
gap> MinimalGeneratingSystemOfIdealOfNumericalSemigroup(I);
[ 0, 1 ]
gap> MinimalGeneratingSystemOfIdealOfNumericalSemigroup(II);
[ 14, 20 ]
gap> MinimalGeneratingSystemOfIdealOfNumericalSemigroup(I+II);
[ 14, 15, 20, 21 ]
```

7.1.11 DifferenceOfIdealsOfNumericalSemigroup

◊ DifferenceOfIdealsOfNumericalSemigroup(I, J)   (function)

I, J are ideals of a numerical semigroup. J must be contained in I.
The output is the set I \ J.

――――――――――――― Example ―――――――――――――
```
gap> S:=NumericalSemigroup(14, 15, 20, 21, 25);
<Numerical semigroup with 5 generators>
gap> I:=[0,1]+S;
<Ideal of numerical semigroup>
gap> 2*I-2*I;
<Ideal of numerical semigroup>
gap> I-I;
<Ideal of numerical semigroup>
gap> DifferenceOfIdealsOfNumericalSemigroup(last2,last);
[ 26, 27, 37, 38 ]
```

7.1.12 TranslationOfIdealOfNumericalSemigroup

◊ TranslationOfIdealOfNumericalSemigroup(k, I)   (function)

Given an ideal I of a numerical semigroup S and an integer k returns an ideal of the numerical semigroup S generated by $\{i_1 + k, \ldots, i_n + k\}$ where $\{i_1, \ldots, i_n\}$ is the system of generators of I.
As a synonym to TranslationOfIdealOfNumericalSemigroup(k, I) the expression k + I may be used.

――――――――――――― Example ―――――――――――――
```
gap> s:=NumericalSemigroup(13,23);
<Numerical semigroup with 2 generators>
gap> l:=List([1..6], _ -> Random([8..34]));
[ 22, 29, 34, 25, 10, 12 ]
gap> I:=IdealOfNumericalSemigroup(l, s);
<Ideal of numerical semigroup>
gap> It:=TranslationOfIdealOfNumericalSemigroup(7,I);
<Ideal of numerical semigroup>
gap> It2:=7+I;
<Ideal of numerical semigroup>
```

```
gap> It2=It;
true
```

7.1.13 HilbertFunctionOfIdealOfNumericalSemigroup

◇ HilbertFunctionOfIdealOfNumericalSemigroup(n, I)   (function)

I is an ideal of a numerical semigroup, n is a non negative integer. I must be contained in its ambient semigroup.

The output is the cardinality of the set $n\text{I} \setminus (n+1)\text{I}$.

```
————— Example —————
gap> I:=[6,9,11]+NumericalSemigroup(6,9,11);;
gap> List([1..7],n->HilbertFunctionOfIdealOfNumericalSemigroup(n,I));
[ 3, 11, 24, 48, 96, 192, 384 ]
```

7.1.14 BlowUpIdealOfNumericalSemigroup

◇ BlowUpIdealOfNumericalSemigroup(I)   (function)

I is an ideal of a numerical semigroup.

The output is the ideal $\bigcup_{n\geq 0} n\text{I} - n\text{I}$.

```
————— Example —————
gap> I:=[0,2]+NumericalSemigroup(6,9,11);;
gap> BlowUpIdealOfNumericalSemigroup(I);;
gap> SmallElementsOfIdealOfNumericalSemigroup(last);
[ 0, 2, 4, 6, 8 ]
```

7.1.15 ReductionNumberIdealNumericalSemigroup

◇ ReductionNumberIdealNumericalSemigroup(I)   (function)

I is an ideal of a numerical semigroup.

The output is the least integer such that $n\text{I} + i = (n+1)\text{I}$, where $i = min(\text{I})$.

```
————— Example —————
gap> I:=[0,2]+NumericalSemigroup(6,9,11);;
gap> ReductionNumberIdealNumericalSemigroup(I);
2
```

7.1.16 MaximalIdealOfNumericalSemigroup

◇ MaximalIdealOfNumericalSemigroup(S)   (function)

Returns the maximal ideal of the numerical semigroup S.

```
───────────────────── Example ─────────────────────
gap> MaximalIdealOfNumericalSemigroup(NumericalSemigroup(3,7));
<Ideal of numerical semigroup>
```

7.1.17 BlowUpOfNumericalSemigroup

◇ BlowUpOfNumericalSemigroup(S)   (function)

If M is the maximal ideal of the numerical semigroup, then the output is the numerical semigroup $\bigcup_{n\geq 0} n\mathtt{M} - n\mathtt{M}$.

```
───────────────────── Example ─────────────────────
gap> s:=NumericalSemigroup(30, 35, 42, 47, 148, 153, 157, 169, 181, 193);
<Numerical semigroup with 10 generators>
gap> BlowUpOfNumericalSemigroup(s);
<Numerical semigroup with 10 generators>
gap> SmallElementsOfNumericalSemigroup(last);
[ 0, 5, 10, 12, 15, 17, 20, 22, 24, 25, 27, 29, 30, 32, 34, 35, 36, 37, 39,
  40, 41, 42, 44 ]
gap> m:=MaximalIdealOfNumericalSemigroup(s);
<Ideal of numerical semigroup>
gap> BlowUpIdealOfNumericalSemigroup(m);
<Ideal of numerical semigroup>
gap> SmallElementsOfIdealOfNumericalSemigroup(last);
[ 0, 5, 10, 12, 15, 17, 20, 22, 24, 25, 27, 29, 30, 32, 34, 35, 36, 37, 39,
  40, 41, 42, 44 ]
```

7.1.18 MicroInvariantsOfNumericalSemigroup

◇ MicroInvariantsOfNumericalSemigroup(S)   (function)

Returns the microinvariants of the numerical semigroup S defined in [Eli01]. For their computation we have used the formula given in [BF06]. The Apéry set of S and its blow up are involved in this computation.

```
───────────────────── Example ─────────────────────
gap> s:=NumericalSemigroup(30, 35, 42, 47, 148, 153, 157, 169, 181, 193);
<Numerical semigroup with 10 generators>
gap> bu:=BlowUpOfNumericalSemigroup(s);
<Numerical semigroup with 10 generators>
gap> ap:=AperyListOfNumericalSemigroupWRTElement(s,30);;
gap> apbu:=AperyListOfNumericalSemigroupWRTElement(bu,30);;
gap> (ap-apbu)/30;
```

```
[ 0, 4, 4, 3, 2, 1, 3, 4, 4, 3, 2, 3, 1, 4, 4, 3, 3, 1, 4, 4, 4, 3, 2, 4, 2,
  5, 4, 3, 3, 2 ]
gap> MicroInvariantsOfNumericalSemigroup(s)=last;
true
```

7.1.19 IsGradedAssociatedRingNumericalSemigroupCM

◊ IsGradedAssociatedRingNumericalSemigroupCM(S)   (function)

Returns true if the graded ring associated to $K[[S]]$ is Cohen-Macaulay, and false otherwise. This test is the implementation of the algorithm given in [BF06].

```
──────────────── Example ────────────────
gap> s:=NumericalSemigroup(30, 35, 42, 47, 148, 153, 157, 169, 181, 193);
<Numerical semigroup with 10 generators>
gap> IsGradedAssociatedRingNumericalSemigroupCM(s);
false
gap> MicroInvariantsOfNumericalSemigroup(s);
[ 0, 4, 4, 3, 2, 1, 3, 4, 4, 3, 2, 3, 1, 4, 4, 3, 3, 1, 4, 4, 4, 3, 2, 4, 2,
  5, 4, 3, 3, 2 ]
gap> List(AperyListOfNumericalSemigroupWRTElement(s,30),
> w->MaximumDegreeOfElementWRTNumericalSemigroup (w,s));
[ 0, 1, 4, 1, 2, 1, 3, 1, 4, 3, 2, 3, 1, 1, 4, 3, 3, 1, 4, 1, 4, 3, 2, 4, 2,
5, 4, 3, 1, 2 ]
gap> last=last2;
false
gap> s:=NumericalSemigroup(4,6,11);
<Numerical semigroup with 3 generators>
gap> IsGradedAssociatedRingNumericalSemigroupCM(s);
true
gap> MicroInvariantsOfNumericalSemigroup(s);
[ 0, 2, 1, 1 ]
gap> List(AperyListOfNumericalSemigroupWRTElement(s,4),
> w->MaximumDegreeOfElementWRTNumericalSemigroup(w,s));
[ 0, 2, 1, 1 ]
```

7.1.20 CanonicalIdealOfNumericalSemigroup

◊ CanonicalIdealOfNumericalSemigroup(S)   (function)

Computes a canonical ideal of S ([BF06]): $x in Z | g - x not in S$.

```
──────────────── Example ────────────────
gap> s:=NumericalSemigroup(4,6,11);
<Numerical semigroup with 3 generators>
gap> m:=MaximalIdealOfNumericalSemigroup(s);
<Ideal of numerical semigroup>
gap> c:=CanonicalIdealOfNumericalSemigroup(s);
<Ideal of numerical semigroup>
```

```
gap> (m-c)-c=m;
true
gap> id:=3+s;
<Ideal of numerical semigroup>
gap> (id-c)-c=id;
true
```

Chapter 8

Numerical semigroups with maximal embedding dimension

8.1 Numerical semigroups with maximal embedding dimension

If *S* is a numerical semigroup and *m* is its multiplicity (the least positive integer belonging to it), then the embedding dimension *e* of *S* (the cardinality of the minimal system of generators of *S*) is less than or equal to *m*. We say that *S* has maximal embedding dimension (MED for short) when $e = m$. The intersection of two numerical semigroups with the same multiplicity and maximal embedding dimension is again of maximal embedding dimension. Thus we define the MED closure of a non-empty subset of positive integers $M = \{m < m_1 < \cdots < m_n < \cdots\}$ with $\gcd(M) = 1$ as the intersection of all MED numerical semigroups with multiplicity *m*.

Given a MED numerical semigroup *S*, we say that $M = \{m_1 < \cdots < m_k\}$ is a MED system of generators if the MED closure of *M* is *S*. Moreover, *M* is a minimal MED generating system for *S* provided that every proper subset of *M* is not a MED system of generators of *S*. Minimal MED generating systems are unique, and in general are smaller that the classical minimal generating systems (see [JCRB03]).

8.1.1 IsMEDNumericalSemigroup

◊ IsMEDNumericalSemigroup(S)   (function)

    `S` is a numerical semigroup.
    Returns true if `S` is a MED numerical semigroup and false otherwise.

```
─────────────────────────── Example ───────────────────────────
gap> IsMEDNumericalSemigroup(NumericalSemigroup(3,5,7));
true
gap> IsMEDNumericalSemigroup(NumericalSemigroup(3,5));
false
```

8.1.2 MEDNumericalSemigroupClosure

◊ MEDNumericalSemigroupClosure(S)   (function)

S is a numerical semigroup.
Returns the MED closure of S.

```
──────────────────────── Example ────────────────────────
gap> MEDNumericalSemigroupClosure(NumericalSemigroup(3,5));
<Numerical semigroup>
gap> MinimalGeneratingSystemOfNumericalSemigroup(last);
[ 3, 5, 7 ]
```

8.1.3 MinimalMEDGeneratingSystemOfMEDNumericalSemigroup

◊ MinimalMEDGeneratingSystemOfMEDNumericalSemigroup(S)   (function)

S is a MED numerical semigroup.
Returns the minimal MED generating system of S.

```
──────────────────────── Example ────────────────────────
gap> MinimalMEDGeneratingSystemOfMEDNumericalSemigroup(
> NumericalSemigroup(3,5,7));
[ 3, 5 ]
```

8.2 Numerical semigroups with the Arf property and Arf closures

Numerical semigroups with the Arf property are a special kind of numerical semigroups with maximal embedding dimension. A numerical semigroup S is Arf if for every x, y, z in S with $x \geq y \geq z$, one has that $x + y - z \in S$.

The intersection of two Arf numerical semigroups is again Arf, and thus we can consider the Arf closure of a set of nonnegative integers with greatest common divisor equal to one. Analogously as with MED numerical semigroups, we define Arf systems of generators and minimal Arf generating system for an Arf numerical semigroup. These are also unique(see [JCRB04]).

8.2.1 IsArfNumericalSemigroup

◊ IsArfNumericalSemigroup(S)   (function)

S is a numerical semigroup.
Returns true if S is an Arf numerical semigroup and false otherwise.

```
──────────────────────── Example ────────────────────────
gap>  IsArfNumericalSemigroup(NumericalSemigroup(3,5,7));
true
gap>  IsArfNumericalSemigroup(NumericalSemigroup(3,7,11));
false
gap> IsMEDNumericalSemigroup(NumericalSemigroup(3,7,11));
```

```
   true
```

8.2.2 ArfNumericalSemigroupClosure

◊ ArfNumericalSemigroupClosure(S)   (function)

S is a numerical semigroup.
Returns the Arf closure of S.

```
_____ Example _____
gap> ArfNumericalSemigroupClosure(NumericalSemigroup(3,7,11));
<Numerical semigroup>
gap> MinimalGeneratingSystemOfNumericalSemigroup(last);
[ 3, 7, 8 ]
```

8.2.3 MinimalArfGeneratingSystemOfArfNumericalSemigroup

◊ MinimalArfGeneratingSystemOfArfNumericalSemigroup(S)   (function)

S is an Arf numerical semigroup.
Returns the minimal MED generating system of S.

```
_____ Example _____
gap> MinimalArfGeneratingSystemOfArfNumericalSemigroup(
> NumericalSemigroup(3,7,8));
[ 3, 7 ]
```

Chapter 9

Catenary and Tame degrees of numerical semigroups

9.1 Factorizations in Numerical Semigroups

Let S be a numerical semigroup minimally generated by $\{m_1,\ldots,m_n\}$. A factorization of an element $s \in S$ is an n-tuple $a = (a_1,\ldots,a_n)$ of nonnegative integers such that $n = a_1 n_1 + \cdots + a_n m_n$. The lenght of a is $|a| = a_1 + \cdots + a_n$. Given two factorizations a and b of n, the distance between a and b is $d(a,b) = \max\{|a - \gcd(a,b)|, |b - \gcd(a,b)|\}$, where $\gcd((a_1,\ldots,a_n),(b_1,\ldots,b_n)) = (\min(a_1,b_1),\ldots,\min(a_n,b_n))$.

If $l_1 > \cdots > l_k$ are the lenghts of all the factorizations of $s \in S$, the Delta set associated to s is $\Delta(s) = \{l_1 - l_2,\ldots,l_k - l_{k-1}\}$.

The catenary degree of S is the least positive integer c such that for any two factorizations a and b of an element in S, there exists a chain of factorizations staring in a and ending in b and so that the distance between two consecutive links is at most c.

The tame degree of S is the least positive integer t for any factorization a of an element s in S, and any i such that $s - m_i \in S$, there exists another factorization b of s so that the distance to a is at most t and $b_i \neq 0$.

The basic properties of these constants can be found in [GHK06]. The algorithm used to compute the catenary and tame degree is an adaptation of the algorithms appearing in [STCR06] for numerical semigroup (see [STCL]). The computation of the elascitiy of a numerical semigroup reduces to m/n with m the multiplicity of the semigroup and n its largest minimal generator (see [STCM] or [GHK06]).

9.1.1 FactorizationsElementWRTNumericalSemigroup

◊ FactorizationsElementWRTNumericalSemigroup(n, S)　　　　　　　　　　(function)

S is a numerical semigroup and n a nonnegative integer. The output is the set of factorizations of n in terms of the minimal generating set of S.

```
────────────────────── Example ──────────────────────
gap> s:=NumericalSemigroup(101,113,196,272,278,286);
<Numerical semigroup with 6 generators>
gap> FactorizationsElementWRTNumericalSemigroup(1100,s);
```

```
[ [ 0, 0, 0, 2, 2, 0 ], [ 0, 2, 3, 0, 0, 1 ], [ 0, 8, 1, 0, 0, 0 ],
  [ 5, 1, 1, 0, 0, 1 ] ]
```

9.1.2 LengthsOfFactorizationsElementWRTNumericalSemigroup

◊ LengthsOfFactorizationsElementWRTNumericalSemigroup(n, S)   (function)

S is a numerical semigroup and n a nonnegative integer. The output is the set of lengths of the factorizations of n in terms of the minimal generating set of S.
```
 ───────── Example ─────────
 gap> s:=NumericalSemigroup(101,113,196,272,278,286);
 <Numerical semigroup with 6 generators>
 gap> LengthsOfFactorizationsElementWRTNumericalSemigroup(1100,s);
 [ 4, 6, 8, 9 ]
```

9.1.3 ElasticityOfFactorizationsElementWRTNumericalSemigroup

◊ ElasticityOfFactorizationsElementWRTNumericalSemigroup(n, S)   (function)

S is a numerical semigroup and n a positive integer. The output is the maximum length divided by the minimum length of the factorizations of n in terms of the minimal generating set of S.
```
 ───────── Example ─────────
 gap> s:=NumericalSemigroup(101,113,196,272,278,286);
 <Numerical semigroup with 6 generators>
 gap> ElasticityOfFactorizationsElementWRTNumericalSemigroup(1100,s);
 9/4
```

9.1.4 ElasticityOfNumericalSemigroup

◊ ElasticityOfNumericalSemigroup(S)   (function)

S is a numerical semigroup. The output is the elasticity of S.
```
 ───────── Example ─────────
 gap> s:=NumericalSemigroup(101,113,196,272,278,286);
 <Numerical semigroup with 6 generators>
 gap> ElasticityOfNumericalSemigroup(s);
 286/101
```

9.1.5 DeltaSetOfFactorizationsElementWRTNumericalSemigroup

◊ DeltaSetOfFactorizationsElementWRTNumericalSemigroup(n, S)   (function)

S is a numerical semigroup and n a nonnegative integer. The output is the Delta set of the factorizations of n in terms of the minimal generating set of S.
```
 ───────── Example ─────────
 gap> s:=NumericalSemigroup(101,113,196,272,278,286);
 <Numerical semigroup with 6 generators>
 gap> DeltaSetOfFactorizationsElementWRTNumericalSemigroup(1100,s);
 [ 1, 2 ]
```

9.1.6 MaximumDegreeOfElementWRTNumericalSemigroup

◇ MaximumDegreeOfElementWRTNumericalSemigroup(n, S)   (function)

S is a numerical semigroup and n a nonnegative integer. The output is the maximum length of the factorizations of n in terms of the minimal generating set of S.

```
———————————————————— Example ————————————————————
gap> s:=NumericalSemigroup(101,113,196,272,278,286);
<Numerical semigroup with 6 generators>
gap> MaximumDegreeOfElementWRTNumericalSemigroup(1100,s);
9
```

9.1.7 CatenaryDegreeNumericalSemigroup

◇ CatenaryDegreeNumericalSemigroup(S)   (function)

S is a numerical semigroup. The output is the catenary degree of S.

```
———————————————————— Example ————————————————————
gap> s:=NumericalSemigroup(101,113,196,272,278,286);
<Numerical semigroup with 6 generators>
gap> CatenaryDegreeNumericalSemigroup(s);
8
```

9.1.8 TameDegreeNumericalSemigroup

◇ TameDegreeNumericalSemigroup(S)   (function)

S is a numerical semigroup. The output is the tame degree of S.

```
———————————————————— Example ————————————————————
gap> s:=NumericalSemigroup(101,113,196,272,278,286);
<Numerical semigroup with 6 generators>
gap> TameDegreeNumericalSemigroup(s);
14
```

Appendix A

Generalities

Here we describe some functions which are not specific for numerical semigroups but are used to do computations with them. As they may have interest by themselves, we decribe them here.

A.1 Bézout sequences

A sequence of positive rational numbers $a_1/b_1 < \cdots < a_n/b_n$ with a_i, b_i positive integers is a *Bézout sequence* if $a_{i+1}b_i - a_ib_{i+1} = 1$ for all $i \in \{1, \ldots, n-1\}$.

The following function uses an algorithm presented in [Ros].

A.1.1 BezoutSequence

$\lozenge$ BezoutSequence(arg)   (function)

arg consits of two rational numbers or a list of two rational numbers. The output is a Bézout sequence with ends the two rational numbers given. (Warning: rational numbers are silently transformed into irreducible fractions.)

```
───────────────────────── Example ─────────────────────────
 gap> BezoutSequence(4/5,53/27);
 [ 4/5, 1, 3/2, 5/3, 7/4, 9/5, 11/6, 13/7, 15/8, 17/9, 19/10, 21/11, 23/12,
   25/13, 27/14, 29/15, 31/16, 33/17, 35/18, 37/19, 39/20, 41/21, 43/22,
   45/23, 47/24, 49/25, 51/26, 53/27 ]
```

A.1.2 IsBezoutSequence

$\lozenge$ IsBezoutSequence(L)   (function)

L is a list of rational numbers. IsBezoutSequence returns true or false according to whether L is a Bézout sequence or not.

```
───────────────────────── Example ─────────────────────────
 gap> IsBezoutSequence([ 4/5, 1, 3/2, 5/3, 7/4, 9/5, 11/6]);
 true
 gap> IsBezoutSequence([ 4/5, 1, 3/2, 5/3, 7/4, 9/5, 11/3]);
 Take the 6 and the 7 elements of the sequence
 false
```

A.1.3 **CeilingOfRational**

◊ CeilingOfRational(r)   (function)

Returns the smallest integer greater than or equal to the rational r.

─────────── Example ───────────
```
gap> CeilingOfRational(3/5);
1
```

A.2 **Periodic subadditive functions**

A periodic function f of period m from the set $\mathbb{N}$ of natural numbers into itself may be specified through a list of m natural numbers. The function f is said to be *subadditive* if $f(i+j) \leq f(i) + f(j)$ and $f(0) = 0$.

A.2.1 **RepresentsPeriodicSubAdditiveFunction**

◊ RepresentsPeriodicSubAdditiveFunction(L)   (function)

L is a list of integers. RepresentsPeriodicSubAdditiveFunction returns true or false according to whether L represents a periodic subAdditive function f periodic of period m or not. To avoid defining $f(0)$ (which we assume to be 0) we define $f(m) = 0$ and so the last element of the list must be 0. This technical need is due to the fact that positions in a list must be positive (not a 0).

─────────── Example ───────────
```
gap> RepresentsPeriodicSubAdditiveFunction([1,2,3,4,0]);
true
```

Appendix B

Random functions

Here we describe some functions which allow to create several "random" objects.

B.1 Random functions

B.1.1 RandomNumericalSemigroup

◇ RandomNumericalSemigroup(n, m)　　　　　　　　　　　　　　　　　　　　(function)

Returns a "random" numerical semigroup with no more than n generators in [1..m].

```
———————————————————— Example ————————————————————
gap> RandomNumericalSemigroup(3,9);
<Numerical semigroup with 3 generators>
```

B.1.2 RandomListForNS

◇ RandomListForNS(n, m)　　　　　　　　　　　　　　　　　　　　　　　(function)

Returns a set of length not greater than n of random integers in [1..m] whose GCD is 1. It is used to create "random" numerical semigroups.

```
———————————————————— Example ————————————————————
gap> RandomListForNS(13,79);
[ 22, 26, 29, 31, 34, 46, 53, 61, 62, 73, 76 ]
```

B.1.3 RandomModularNumericalSemigroup

◇ RandomModularNumericalSemigroup(k)　　　　　　　　　　　　　　　　(function)

Returns a "random" modular numerical semigroup.

```
———————————————————— Example ————————————————————
gap> RandomModularNumericalSemigroup(9);
<Modular numerical semigroup satisfying 5x mod 6 <= x >
```

B.1.4 RandomProportionallyModularNumericalSemigroup

◇ RandomProportionallyModularNumericalSemigroup(k)   (function)

Returns a "random" proportionally modular numerical semigroup (see 1).

```
────────────────────── Example ──────────────────────
gap> RandomProportionallyModularNumericalSemigroup(9);
<Proportionally modular numerical semigroup satisfying 2x mod 3 <= 2x >
```

B.1.5 RandomListRepresentingSubAdditiveFunction

◇ RandomListRepresentingSubAdditiveFunction(m, a)   (function)

Produces a "random" list representing a subadditive function (see 1) which is periodic with period m (or less). When possible, the images are in [a..20*a]. (Otherwise, the list of possible images is enlarged.)

```
────────────────────── Example ──────────────────────
gap> RandomListRepresentingSubAdditiveFunction(7,9);
[ 173, 114, 67, 0 ]
gap> RepresentsPeriodicSubAdditiveFunction(last);
true
```

Appendix C

A graphical interface

Here we describe a Tcl/Tk interface to specify numerical semigroups.

C.1   Graphical interface

C.1.1   XNumericalSemigroup

◊ XNumericalSemigroup([s])   (function)

   This function pops up a Tcl/Tk window which allows the specification of numerical semigroups. The optional argument `s` is a numerical semigroup whose definition becomes displayed in the graphical interface.
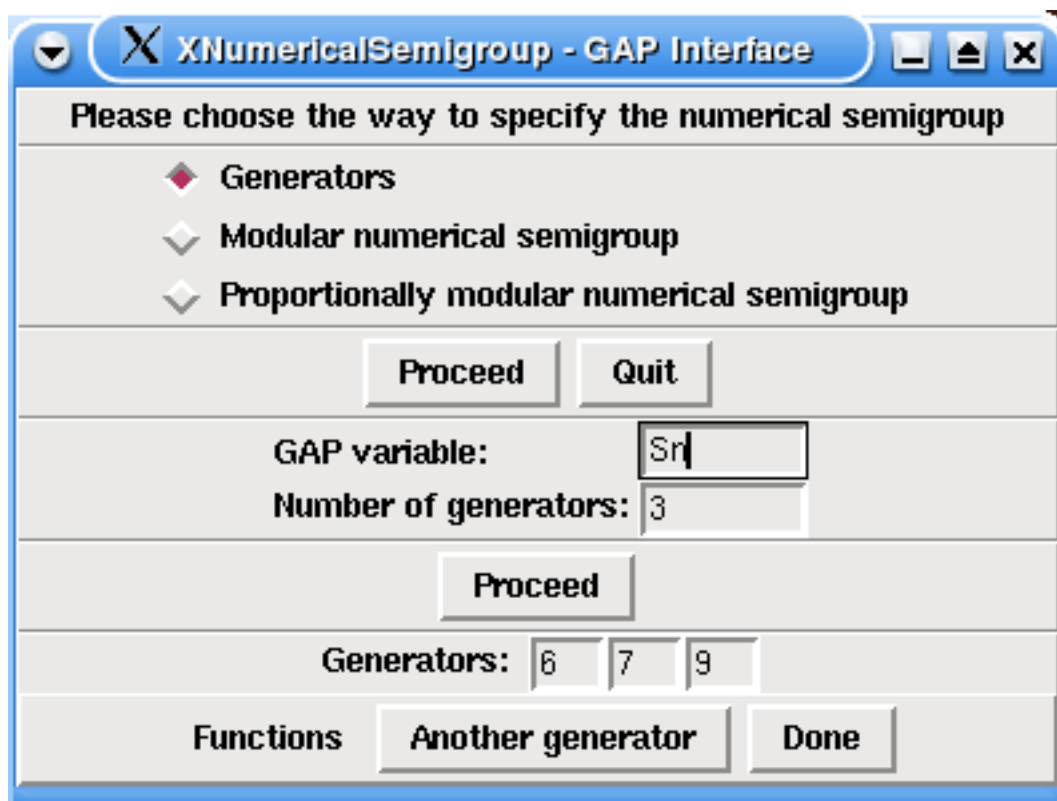
   The interface is very simple, it allows to specify a numerical semigroup by its generators or a (proportionally) modular numerical semigroup by its (proportionally) modular condition. The GAP variable by which the semigroup will be known must also be specified.

   The interface also allows to apply some functions to the semigroup, namely `AperyListOfNumericalSemigroupWRTElement` (3.1.5), `DrawAperyListOfNumericalSemigroup` (3.1.6), `SmallElementsOfNumericalSemigroup` (3.1.3), `GapsOfNumericalSemigroup` (3.3.1), `FrobeniusNumberOfNumericalSemigroup` (3.2.1) and `MinimalPresentationOfNumericalSemigroup` (4.1.2). Finally, it is also possible to add functions to this list and remove them.

─────────────────────────── Example ───────────────────────────
```
gap> s:=RandomNumericalSemigroup(3,9);
<Numerical semigroup with 3 generators>
gap> XNumericalSemigroup(s);
```

This would pop up the following window:

References

[BF06]   V. Barucci and R. Fröberg. Associated graded rings of one-dimensional analytically irreducible rings. *J. Aalgebra*, to appear, 2006. 33, 34

[CD94]   Evelyne Contejean and Hervé Devie. An efficient incremental algorithm for solving systems of linear Diophantine equations. *Inform. and Comput.*, 113(1):143–172, 1994. 19

[Eli01]   J. Elias. On the deep structure of the blowing-up of curve singularities. *Math. Proc. Camb. Phil. Soc.*, 131:227–240, 2001. 33

[GHK06]   A. Geroldinger and F. Halter-Koch. *Non-unique Factorizations: Algebraic, Combinatorial and Analytic Theory*. Chapman Hall/CRC, 2006. 39

[HS04]   Kurt Herzinger and Ryan Sanford. Minimal generating sets for relative ideals in numerical semigroups of multiplicity eight. *Communications in Algebra*, 32(12):4713–4731, 2004. 30

[JCRB03] J. I. García-García J. C. Rosales, P. A. García-Sánchez and M. B. Branco. Numerical semigroups with maximal embedding dimension. *J. Algebra*, 2:47–53, 2003. 36

[JCRB04] J. I. García-García J. C. Rosales, P. A. García-Sánchez and M. B. Branco. Arf numerical semigroups. *J. Algebra*, 276:3–12, 2004. 37

[JCRJM03] J. I. García-García J. C. Rosales, P. A. García-Sánchez and J. A. Jiménez-Madrid. The oversemigroups of a numerical semigroup. *Semigroup Forum*, 67:145–158, 2003. 21, 23, 24

[JCRM04] J. I. García-García J. C. Rosales, P. A. García-Sánchez and J. A. Jiménez Madrid. Fundamental gaps in numerical semigroups. *J. Pure Appl. Algebra*, 189(1-3):301–313, 2004. 23

[RB03]   J. C. Rosales and M. B. Branco. Irreducible numerical semigroups. *Pacific J. Math.*, 209(1):131–143, 2003. 24

[RFH87]   C. Gottlieb R. Fröberg and R. Häggkvist. On numerical semigroups. *Semigroup Forum*, 35(1):63–83, 1987. 24

[RGS99]   J. C. Rosales and P. A. García-Sánchez. *Finitely generated commutative monoids*. Nova Science Publishers, New York, 1999. 19

[Ros]   J. C. Rosales. An algorithm to solve proportionally modular diophantine inequations. *preprint*. 42

[Ros96a] J. C. Rosales. An algorithmic method to compute a minimal relation for any numerical semigroup. *Internat. J. Algebra Comput.*, 6:441–455, 1996. 19, 20

[Ros96b] J. C. Rosales. On numerical semigroups. *Semigroup Forum*, 52:307–318, 1996. 13

[Ros04] J. C. Rosales. Every positive integer is the frobenius number of an irreducible numerical semigroup with at most four generators. *Ark. Mat.*, 42:301–306, 2004. 24, 25

[STCL] P. A. García-Sánchez S. T. Chapman and D. Llena. The catenary and tame degree of numerical semigroups. *submitted*. 39

[STCM] M. T. Holden S. T. Chapman and T. A. Moore. Full elasticity in atomic monoids and integral domains. *submitted*. 39

[STCR06] D. Llena V. Ponomarenko S. T. Chapman, P. A. García-Sánchez and J. C. Rosales. The catenary and tame degree in finitely generated cancellative monoids. *Manuscripta Math.*, to appear, 2006. 39

[VBF97] D. E. Dobbs V. Barucci and M. Fontana. *Maximality properties in numerical semigroups and applications to one-dimensional analytically irreducible local domains*, volume 125. 1997. 24, 27

Index