The LocalSNF Package

Version 0.0.1

Alexander Neshitov

Alexander Neshitov Email: alexander.neshitov@gmail.com

Copyright

Contents

Chapter 1

LocalSNF package

1.1 Motivation

Calculations of Smith Normal form with transformations are hard for large matrices since the coefficients of the matix become extremely large. In some situations it is sufficient to find Smoth Normal Form and transformation matrices modulo power of a prime number. This package is developed to compute Smith Normal Form with transformations for matrices over the ring Z/2^N. The reduction algorithm is implemented in C for speed-up purposes and ability to use multithreading and binary arithmetic optimizations.

1.2 Functions

In this section we describe two functions implemented in this package. Both functions take an optional parameter num_threads for how many threads to use in reduction algorithm. By default 4 threads are used.

1.2.1 SNFTransform

▷ `SNFTransform(`*mat, N*`)` (function)

This function computes Smith Normal Form modulo 2^N of the integral matrix mat. It returns record with fields: SNF, row_t, row_t_inverse, col_t, rank, power such that

- SNF is diagonal, its entries are powers of 2, and i entry divides i+1 entry for all i
- SNF = row_t * mat * col_t mod 2^N
- row_t * row_t_inverse = I, rank is the rank of SNF, power = N

Example

```
gap> LoadPackage("LocalSNF");
gap> SNFTransform([[1,2],[3,4]], 10: num_threads:=2);
rec( SNF := [ [ 1, 0 ], [ 0, 2 ] ], col_t := [ [ 1, 1022 ], [ 0, 1 ] ],
  power := 10, rank := 2, row_t := [ [ 1, 0 ], [ 515, 511 ] ],
  row_t_inverse := [ [ 1, 0 ], [ 3, 511 ] ] )
```

▷ `SaturationVectors(`*mat, N*`)` (function)

This function a minimal set of vectors such that together with columns of mat they span a direct summand of the module (Z/2^N)^d where d is the number of rows of mat.

Example

```
gap> LoadPackage("LocalSNF");
gap> SaturationVectors([[1,2],[3,4]],10);
[ [ 0 ], [ 511 ] ]
```

Index