A BRIEF DESCRIPTION OF THE FUNCTIONALITY OF THE ACCOMPANYING CODE

IGNAT SOROKO

The file `hindex.g` accompanies the paper

> "Divergence, thickness and hypergraph index for general Coxeter groups" by Pallavi Dani, Yusra Naqvi, Ignat Soroko and Anne Thomas

and contains some code for the computer algebra system `GAP4` for computing:

- the hypergraph index
- the duplex construction
- the number of ends of the Coxeter group
- some Coxeter matrices used in the text.

1. A typical `GAP` session

```
gap> Read("hindex.g");
gap> m3:=DaniThomasGamma(3);
[ [ 1, 0, 2, 2, 2, 2, 0, 0 ], [ 0, 1, 2, 2, 2, 2, 0, 0 ], [ 2, 2, 1, 0, 0, 0, 2, 0 ], [ 2, 2, 0, 1, 0, 0, 2, 0 ],
  [ 2, 2, 0, 0, 1, 0, 0, 2 ], [ 2, 2, 0, 0, 0, 1, 0, 0 ], [ 0, 0, 2, 2, 0, 0, 1, 2 ], [ 0, 0, 0, 0, 2, 0, 2, 1 ] ]
gap> h3:=HypergraphIndex(m3);
[ 2, [ [ [ 1, 2, 3, 4, 5, 6 ], [ 1, 2, 3, 4, 7 ], [ 1, 5, 8 ], [ 2, 5, 8 ], [ 3, 7, 8 ], [ 4, 7, 8 ], [ 5, 7, 8 ] ],
      [ [ 1 .. 7 ], [ 1, 5, 8 ], [ 2, 5, 8 ], [ 3, 7, 8 ], [ 4, 7, 8 ], [ 5, 7, 8 ] ],
      [ [ 1 .. 8 ], [ 1, 5, 8 ], [ 2, 5, 8 ], [ 3, 7, 8 ], [ 4, 7, 8 ] ] ] ]
```

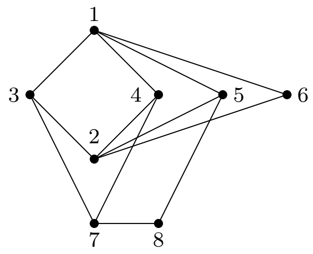
Here the command

```
gap> Read("hindex.g");
```

loads the file `hindex.g`, and it needs to be done just once.

```
gap> m3:=DaniThomasGamma(3);
```

loads the Coxeter matrix of the right-angled Coxeter group Γ_3, whose *defining graph* is depicted below (see also Figure 5 of the article). Note that, by our convention, in a Coxeter matrix `0` represents infinity. The numeration of vertices in the Coxeter matrix for Γ_3 is as follows:



Recall that in the defining graph an edge corresponds to a pair of commuting generators s, t, i.e. such that $m_{st} = 2$, and no edge between s, t corresponds to $m_{st} = \infty$.

```
gap> h3:=HypergraphIndex(m3);
```

returns a list of length two. Its first component, `2`, is the hypergraph index h of the corresponding Coxeter system. The second component is the list $[\Lambda_0, \Lambda_1, \ldots, \Lambda_h]$, where each sublist Λ_i is a list of lists of integers, which represent subsets of the generating set. I.e. Λ_0 is

```
[ [ 1, 2, 3, 4, 5, 6 ], [ 1, 2, 3, 4, 7 ], [ 1, 5, 8 ], [ 2, 5, 8 ], [ 3, 7, 8 ], [ 4, 7, 8 ], [ 5, 7, 8 ] ]
```

which means that Λ_0 contains wide subsets $\{1,2\} \times \{3,4,5,6\}$ and $\{3,4\} \times \{1,2,7\}$ and slab subsets $\{1,8\} \times \{5\}$, $\{2,8\} \times \{5\}$, $\{3,8\} \times \{7\}$, $\{4,8\} \times \{7\}$, and $\{5,7\} \times \{8\}$. Moreover, we see that the two wide subsets intersect in a nonspherical system $\{1,2\} \times \{3,4\}$, so they should be united in Λ_1, which is equal to the second entry in the list discussed above:

```
[ [ 1 .. 7 ], [ 1, 5, 8 ], [ 2, 5, 8 ], [ 3, 7, 8 ], [ 4, 7, 8 ], [ 5, 7, 8 ] ]
```

And finally we see that the elements $\{1,\dots,7\}$ and $\{5,7\} \times \{8\}$ in Λ_1 intersect in a nonspherical subsystem $\{5,7\}$, and hence should be united in Λ_2. Thus Λ_2 contains the whole generator set $\{1,\dots,8\}$, which is represented by the third entry in the above list: Λ_2 equals

```
[ [ 1 .. 8 ], [ 1, 5, 8 ], [ 2, 5, 8 ], [ 3, 7, 8 ], [ 4, 7, 8 ] ]
```

Since the whole generator set `[ 1 .. 8 ]` belongs to Λ_2, the hypergraph index h is equal to 2.

2. Main functions

▶ `HypergraphIndex(mat)`

- Input: A Coxeter matrix `mat`.
- Output: A list `[hi,L]` of length two:
  - `hi` is the hypergraph index, which is an integer `0,1,2,...` or `infinity`.
  - `L` is the list of lists which represents sets $[\Lambda_0, \Lambda_1, \dots, \Lambda_h]$.

Each Λ_i is a list of lists of integers corresponding to the subsets of the generating set S, which is identified with `[1..N]` for `N` $=$ Card(S), see the example above.

Remark. To speed up computations, we form the slab subsets $\Psi(S)$ in a modified form. Instead of subsets $A \times K$ where $A \subset S$ is minimal nonspherical and $K \subset S$ is a spherical subset commuting with A, such that $A \times K$ have maximality property as per Definition 4.5 of the article, we store subsets of the form $A \times \{s\}$ where A is as before, and s commutes with A. This makes Λ_0 different from the one defined in the article, however this has no bearing on the sets $\Lambda_1, \Lambda_2, \dots$ and they are exactly as they should be.

▶ `DuplexRACG(mat,m,n)`

- Input:
  - A Coxeter matrix `mat` of a right-angled Coxeter system;
  - `m,n` are two integers ≥ 2 or 0 (which represents ∞).
- Output: the Coxeter matrix corresponding to the duplex construction as described in Section 5 of the article.

3. Functions to form Coxeter matrices

▶ `CoxeterMatrixDynkinPath(lst)`

- Input: a list `lst` of edge labels corresponding to the Dynkin diagram which is a simple path.
- Output: the Coxeter matrix of the corresponding Coxeter system.
- Example:
  ```
  gap> mEx9:=CoxeterMatrixDynkinPath([4,4,4,4,4,4,4]);
  ```
  creates the Coxeter matrix of the Coxeter system Δ_8 depicted in Figure 9 of the article.

▶ `CoxeterMatrixDynkinCycle(lst)`

- Input: a list `lst` of edge labels corresponding to the Dynkin diagram which is a simple cycle.
- Output: the Coxeter matrix of the corresponding Coxeter system.
- Examples:
  ```
  gap> mEx10a:=CoxeterMatrixDynkinCycle([3,3,3,3,3,3,3,3,3]);
  gap> mEx10b:=CoxeterMatrixDynkinCycle([4,3,3,4,4,4,3,3,4]);
  gap> mEx10c:=CoxeterMatrixDynkinCycle([3,4,4,3,3,3,4,4,3]);
  gap> mEx10d:=CoxeterMatrixDynkinCycle([4,3,4,4,3,4,4,3,4]);
  ```
  create the Coxeter matrices of the Coxeter systems depicted in Figure 10 of the article.

▶ `CoxeterMatrixRAListOfEdges(edges)`

- Input: a list of pairs of integers representing edges of the defining graph of a right-angled Coxeter system;
- Output: the Coxeter matrix of the corresponding right-angled Coxeter system.

Besides, the constructors of Coxeter matrices for irreducible spherical and affine and Coxeter systems are provided, as well as for the two Lannér hyperbolic systems which are not paths or cycles.

`CoxeterMatrixAn(n)`	`CoxeterMatrixAffineAn_(n)`
`CoxeterMatrixBn(n)`	`CoxeterMatrixAffineBn_(n)`
`CoxeterMatrixCn(n)` (same as the above)	`CoxeterMatrixAffineCn_(n)`
`CoxeterMatrixDn(n)`	`CoxeterMatrixAffineDn_(n)`
`CoxeterMatrixEn(n)` ($\mathtt{n} = 6, 7, 8$, vertex 4 is branch)	`CoxeterMatrixAffineEn_(n)`
`CoxeterMatrixFn(n)` ($\mathtt{n} = 4$)	`CoxeterMatrixAffineFn_(n)`
`CoxeterMatrixGn(n)` ($\mathtt{n} = 2$)	`CoxeterMatrixAffineGn_(n)`
`CoxeterMatrixHn(n)` ($\mathtt{n} = 2, 3, 4$; $m_{12} = 5$)	
`CoxeterMatrixI2m(m)` ($\mathtt{m} \neq 1$)	`CoxeterMatrixLannerZn(n)` ($\mathtt{n} = 4, 5$)

Here for all systems (including affine ones) the argument `n` is the number of vertices. For the function `CoxeterMatrixI2m(m)` the argument `m` is the edge label between the two vertices.

Other Coxeter matrices provided in the file `hindex.g`:

▶ `DaniThomasGamma(d)`

- Input: An integer `d`
- Output: The Coxeter matrix of the right-angled Coxeter system Γ_d from the article of Dani and Thomas, which are also depicted in Figure 6 of the main paper.

▶ `mEx4, mEx9, mEx10a, mEx10b, mEx10c, mEx10d, mEx11`
provide Coxeter matrices for the systems presented in the Figures 4, 9, 10a–d, and 11 of the article.

▶ `CoxeterSubmatrix(mat,T)`

- Input: a Coxeter matrix `mat` and a subset `T` of `[1..Size(mat)]`
- Output: the Coxeter matrix corresponding to the subsystem on the set of vertices `T`.

4. Other useful functions

▶ `IsOneEnded(mat)`

- Input: A Coxeter matrix `mat`;
- Output: `true` or `false` depending on whether the Coxeter group with the Coxeter matrix `mat` has one end. Uses the criterion from Th. 8.7.2 of the Davis's book.

▶ `NrEnds(mat)`

- Input: A Coxeter matrix `mat`;
- Output: an integer `0,1,2` or `infinity` representing the number of ends of the corresponding Coxeter group. Uses Theorems 8.7.2, 8.7.3 and 8.7.4 of the Davis's book.

▶ `CanonicalRepOrbit(mat)`

- Input: A Coxeter matrix `mat`;
- Output: A permuted Coxeter matrix, which is a canonical representative under the action of all permutations of the Coxeter generating set.

It returns the Coxeter matrix which is maximal in the lexicographic order for all simultaneous permutations of rows and columns applied to `mat`. It first computes the vector of sums of edge labels at each vertex, sorts the resulting vector in the descending order, and then applies only those permutations that preserve this vector. This function is useful when comparing Coxeter systems up to isomorphism of edge-labeled graphs.

5. Support

For anything related to the code it is best to contact Ignat Soroko by email `ignat.soroko@gmail.com`.