Discrete and Algorithmic Geometry

Julian Pfeifle, UPC, 2014

Sheet 2

due on Monday, November 17, 2014

Writing

(1) Let $P \subset \mathbb{R}^d$, $Q \subset \mathbb{R}^e$ be two non-empty polytopes. Prove that the set of faces of the cartesian product polytope $P \times Q = \{(p,q) \in \mathbb{R}^{d+e} : p \in P,\ q \in Q\}$ exactly equals $\{F \times G : F \text{ is face of } P,\ G \text{ is face of } Q\}$. Conclude that

$$f_k(P \times Q) \;=\; \sum_{i+j=k,\ i,j\geq 0} f_i(P) f_j(Q) \qquad \text{for } k \geq 0,$$

and use this formula to calculate the entire f-vector of the permutahedron

$$P_n \;=\; \operatorname{conv}\left\{ \big(\pi(1), \pi(2), \dots, \pi(n)\big)^\top : \pi \in S_n \right\}.$$

(2) A *lattice polytope* is the convex hull of finitely many vertices with integer coordinates. Two lattice polytopes $P, Q \subset \mathbb{R}^d$ are *lattice equivalent* or *lattice isomorphic* if there exist $A \in \mathbb{Z}^{d\times d}$ with $\det(A) = \pm 1$ and $b \in \mathbb{Z}^d$ such that $Q = f(P)$ for the affine map $f(x) = Ax + b$. (How is this different from demanding that P and Q be $\mathrm{Sl}_d(\mathbb{Z})$-equivalent?)

  (a) Prove that the *modular group* $\mathrm{Sl}_2(\mathbb{Z})$ is generated by the elements $S = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ and $T = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, i.e., any $A \in \mathrm{Sl}_2(\mathbb{Z})$ is expressible as a product of matrices S and T.

  (b) Interpret the preceding result geometrically, and use it to classify the lattice polygons with (i) no, (ii) exactly one strictly interior lattice point up to lattice equivalence.

Software

(3) Explain the difference between a public/private key pair for `ssh` and a public/private key pair for `gpg`. Gather information about the recommended keysizes, and explain briefly the advantages and disadvantages of the `gpg` software, making special mention of the latest versions. Then place a public/private `gpg`(!) keypair of the recommended size into `2014/public_keys` . Again, don't forget to commit and push your changes, and issue a pull request.

(4) Using your favorite software environment, write a function χ that checks if two 3-dimensional lattice tetrahedra are lattice equivalent. Your function should take two 4×4 matrices P, Q as input, and output a 4×4 matrix A. The columns of the input matrices P and Q are to be interpreted as the homogeneous coordinates of the vertices of the respective lattice tetrahedra. The output A is either 0 (if $\widetilde{P} = \operatorname{conv}\operatorname{cols} P$ is not lattice isomorphic to $\widetilde{Q}$), or encodes a $\mathbb{Z}$-affine map that sends $\widetilde{P}$ to $\widetilde{Q}$.

In particular, your function should correctly reproduce the calculation

$$\chi\left(\begin{bmatrix}1&1&1&1\\0&1&0&2\\0&0&1&3\\0&0&0&5\end{bmatrix},\begin{bmatrix}1&1&1&1\\0&0&1&2\\0&1&0&3\\0&0&0&5\end{bmatrix}\right)=I_4.$$

Provide a rudimentary testing facility and testsuite for your function. You should be able to input the two matrices from two plain text files, where each row encodes a vertex of the tetrahedron, and write the output matrix to a third file. For instance, the matrix P above would be encoded as

```
1 0 0 0
1 1 0 0
1 0 1 0
1 2 3 5
```

Also prepare a function `test_chi` that reads a file containing lines of the form

```
matrix_P1.txt matrix_P2.txt result_P1_P2.text
matrix_P3.txt matrix_P3.txt result_P3_P4.text
```

and executes the tests with the corresponding filenames. When you are done, your instructor will provide you with more examples to test your function on!

Create a directory of the form `2014/exercises/sheet2/your_name_here/code` and put all relevant source code in there. Don't forget a README so that other people (i.e., you in the future) can figure out how your function works! And, as always, don't forget the pull request.