Calculations in convex geometry

Robert Vollmert

July 30, 2006

`convex` by Matthias Franz

`http://www-fourier.ujf-grenoble.fr/~franz/convex/`

`TorDiv` by Florian Berchtold, Jürgen Hausen and Marcel Widmann

`http://www.uni-konstanz.de/FuF/mathe/homepages/berchtof/Software/TorDiv.html`

1 toric geometry

1.1 affine toric varieties

```
> sigma := poshull([2, -1], [0, 1]);
                  sigma := CONE(2, 2, 0, 2, 2)
> rays(sigma);
                       [[2, -1], [0, 1]]
> sigmadual := dual(sigma);
                sigmadual := CONE(2, 2, 0, 2, 2)
> rays(sigmadual);
                        [[1, 0], [1, 2]]
> hilbertbasis(sigmadual);
                    [[1, 2], [1, 1], [1, 0]]
```

1.2 global toric varieties

```
> s := stdsimplex(2);
                   s := POLYTOPE(2, 2, 3, 3)
> vertices(s);
                    [[0, 0], [1, 0], [0, 1]]
> fanP2 := normalfan(s);
                  fanP2 := FAN(2, 0, [0, 3])
> map(rays, maximal(fanP2));
    [[[1, 0], [0, 1]], [[-1, -1], [0, 1]], [[-1, -1], [1, 0]]]
```

2 computations

2.1 projectivity, line bundles

```
> ispolytopal(fanP2);
                                true

> picardgroup(fanP2);
                                [[1]]

> amplecone(fanP2);
                        CONE(1, 1, 0, 1, 1)
```

2.2 resolution of singularities

```
> isregular(sigma);
                                false

> Sigma := regularsubdiv(fan(sigma));
                     Sigma := FAN(2, 0, [0, 2])

> map(rays, maximal(Sigma));
               [[[0, 1], [1, 0]], [[2, -1], [1, 0]]]
```

2.3 refine fans

```
> neg := [[-1,0],[0,-1]]:
   > fanP2b := image(fanP2, neg);
                    fanP2b := FAN(2, 0, [0, 3])

> ispolytopal(fanP2b, t);
                                true

> u := minkowskisum(s, t);
                     u := POLYTOPE(2, 2, 6, 6)

> fanref := normalfan(u);
                    fanref := FAN(2, 0, [0, 6])
```

2.4 GIT quotients

> Q := $\begin{pmatrix} 1 & 0 & 0 & 0 & 2 & 1 & 2 & 1 & 2 \\ 0 & 1 & 0 & 0 & 1 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & -2 & 0 & -2 \end{pmatrix}$;

```
> P := ...; R := ...; S := ...;
> delta := posorthant(9); deltadual := dual(delta);

> omega := image(deltadual, Q);

                    omega := CONE(4, 4, 0, 8, 10)

> u := relint(omega);

                          u := [4, 3, -1, -2]

posfiber := proc(u, deltadual, Q, R)
    image(intersection(convert(deltadual, affine),
                       preimage(convhull(u), Q)),
          R)
end proc

> Pi_u := posfiber(u, deltadual, Q, R);

                    Pi_u := POLYTOPE(5, 5, 18, 9)

> Sigma_u := normalfan(Pi_u);

                Sigma_u := FAN(5, 0, [0, 0, 0, 0, 18])

> regularsubdiv(Sigma_u);

                     FAN(5, 0, [0, 0, 0, 0, 66])
```

2.5 polyhedral divisors

```
> ppdiv := ppdivisor(Q):
   > ppdiv[2];

                     FAN(5, 0, [0, 0, 0, 0, 98])

   > ppdiv[1];

[[POLYHEDRON(4, 4, 0, [4, 10], [8]), [-1, 0, -1, -1, -1]],
 [POLYHEDRON(4, 4, 0, [4, 10], [8]), [0, -1, 1, -1, 0]],
 [POLYHEDRON(4, 4, 0, [2, 10], [8]), [0, 0, 1, 0, 0]],
 [POLYHEDRON(4, 4, 0, [2, 10], [8]), [0, 0, 0, 1, 0]],
 [POLYHEDRON(4, 4, 0, [2, 10], [8]), [0, 0, 0, 0, 1]],
 [POLYHEDRON(4, 4, 0, [4, 10], [9]), [-1, 0, 0, 0, 0]],
 [POLYHEDRON(4, 4, 0, [4, 10], [8]), [1, 0, 1, 1, 1]],
 [POLYHEDRON(4, 4, 0, [1, 10], [8]), [1, -1, 0, 0, -1]],
 [POLYHEDRON(4, 4, 0, [2, 10], [8]), [-1, -1, 0, -1, -1]],
 [POLYHEDRON(4, 4, 0, [2, 10], [8]), [-1, 0, -1, -1, 0]],
 [POLYHEDRON(4, 4, 0, [2, 10], [8]), [0, 1, 0, 1, 0]],
 [POLYHEDRON(4, 4, 0, [2, 10], [8]), [1, 0, 1, 0, 1]],
 [POLYHEDRON(4, 4, 0, [2, 10], [8]), [0, 1, -1, 0, 0]],
 [POLYHEDRON(4, 4, 0, [4, 10], [8]), [0, 1, -1, 1, 0]],
 [POLYHEDRON(4, 4, 0, [4, 10], [8]), [-1, -1, 0, -2, -1]],
 [POLYHEDRON(4, 4, 0, [4, 10], [9]), [0, 1, 0, 0, 1]],
 [POLYHEDRON(4, 4, 0, [4, 10], [8]), [1, 1, 0, 2, 1]],
 [POLYHEDRON(4, 4, 0, [4, 10], [8]), [-1, 1, -2, 0, -1]],
 [POLYHEDRON(4, 4, 0, [4, 10], [8]), [1, -1, 2, 0, 1]]]
```