OSCAR: The Dream

Mohamed Barakat Janko Boehm Wolfram Decker Claus Fieker Michael Joswig Frank Lübeck

September 27, 2018

TR 195 SYMBOLIC TOOLS

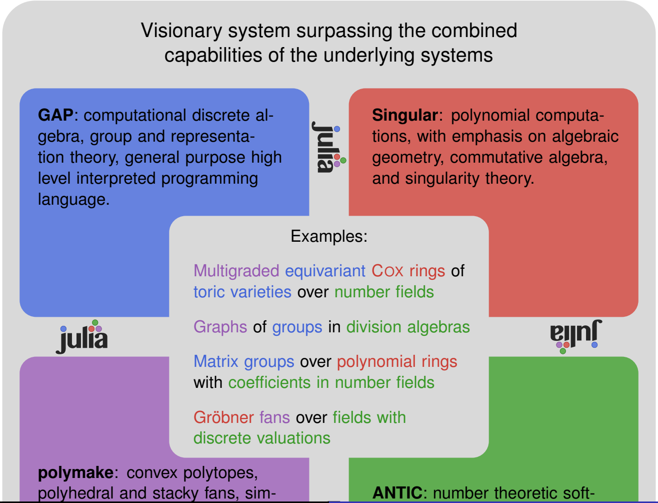
OSCAR SYMBOLIC TOOLS

Develop a visionary, next generation, open source computer algebra system, integrating all systems, libraries and packages developed within the TRR.

OSCAR

SYMBOLIC TOOLS

Overview



Visionary system surpassing the combined capabilities of the underlying systems

GAP: computational discrete algebra, group and representation theory, general purpose high level interpreted programming language.

Singular: polynomial computations, with emphasis on algebraic geometry, commutative algebra, and singularity theory.

Examples:

Multigraded equivariant Cox rings of toric varieties over number fields

Graphs of groups in division algebras

Matrix groups over polynomial rings with coefficients in number fields

Gröbner fans over fields with discrete valuations

polymake: convex polytopes, polyhedral and stacky fans, sim-

ANTIC: number theoretic soft-

- The technical aspects:
  - Integration
  - Data exchange
  - Tools (Gröbner basis, linear algebra, coset enumeration, . . .
- Mathematics
  - Modelling
  - Abstraction
  - Cross-disciplinary language

- The technical aspects:
  - Integration
  - Data exchange
  - Tools (Gröbner basis, linear algebra, coset enumeration, . . .
- Mathematics
  - Modelling
  - Abstraction
  - Cross-disciplinary language - *not programming language*

Aim

This talk aims to look at the 2nd block.

Aim

This talk aims to look at the 2nd block.

Starting with (the few) projects that are/were successfully using OSCAR.

Aim

This talk aims to look at the 2nd block.

Starting with (the few) projects that are/were successfully using OSCAR. and with general progress.

Aim

This talk aims to look at the 2nd block.

Starting with (the few) projects that are/were successfully using OSCAR. and with general progress.

Other aspects will be covered tomorrow.

Binomial Ideals

Binomial ideals are ideals in $K[x_1, \ldots, x_n]$ that are generated by binomials, ie. polynomials with at most 2 terms. They form an important class of ideals, containing

- toric ideals
- ideals coming from algebraic statistics

Clara Petroll implemented in her bachelor thesis special algorithms for the primary decomposition of binomial ideals over $\mathbb{Q}$.

Binomial Ideals

In OSCAR:

- Singular for multivariate ideals
- Hecke for the abelian closure

Shafarevich

Given a soluble finite group G, a famous theorem of Shafarevich shows that there exist number fields (polynomials) having G as Galois group.

Shafarevich

Given a soluble finite group G, a famous theorem of Shafarevich shows that there exist number fields (polynomials) having G as Galois group.

The problem is to find such polynomials/ fields.

Shafarevich

Given a soluble finite group G, a famous theorem of Shafarevich shows that there exist number fields (polynomials) having G as Galois group.

The problem is to find such polynomials/ fields.

As part of his PhD, Carlo Sircana is working on this.

Shafarevich

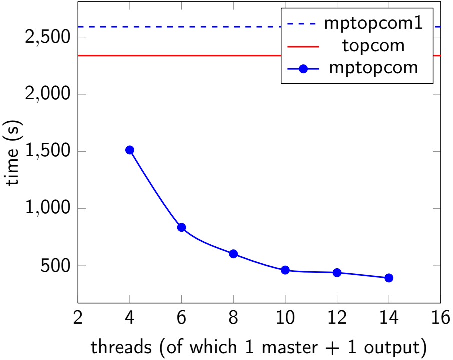
In OSCAR:

- Gap for lower derived series and isomorphism test for groups
- Hecke for class field theory

mptopcom

Jordan, Joswig & Kastner 2018

- enumerate all (regular) triangulations of a point configuration
  - crucial, e.g., for computing tropical moduli spaces



toy example: regular 4-cube
@ AMD Ryzen 7 1700 [32GB RAM]

- embarassingly parallel algorithm, runs in several hundreds of threads
- almost linear scaling until competition for CPU cache/main memory/disk space kicks in
- output: data base

Hecke

- relative extensions
- non-simple extensions
- class field theory
- non-commutative orders

Nemo

Multivariate polynomials over $\mathbb{Q}$ and finite fields

- arithmetic
- division
- gcd

Theory

Class Field Theory: Given a number field k, the class group Cl_K is the Picard group of the ring of integers (similar to the divisor class group of a normal curve). This is a finite abelian group, one of the core invariants of a number field.

Given an ideal $\mathfrak{A}$, there is a similar, but more general group $\mathrm{Cl}_{\mathfrak{A}}$ the ray class group.

Theory

Class field theory shows that for all subgroups $U < \mathrm{Cl}_{\mathfrak{A}}$ there is exactly one abelian elxtension K/k s.th.

$$\mathrm{Aut}(K/k) = \mathrm{Cl}_{\mathfrak{A}} / U$$

canonically. Furthermore this correspondence behaves well under operations of $\mathrm{Aut}(k)$.

E.g. if $k/\mathbb{Q}$ is normal, then K/k is normal over $\mathbb{Q}$ iff

- $\mathfrak{A}$ is invariant under $\mathrm{Aut}(k)$

Theory

Class field theory shows that for all subgroups $U < \mathrm{Cl}_{\mathfrak{A}}$ there is exactly one abelian elxtension K/k s.th.

$$\mathrm{Aut}(K/k) = \mathrm{Cl}_{\mathfrak{A}}/U$$

canonically. Furthermore this correspondence behaves well under operations of $\mathrm{Aut}(k)$.

E.g. if $k/\mathbb{Q}$ is normal, then K/k is normal over $\mathbb{Q}$ iff

- $\mathfrak{A}$ is invariant under $\mathrm{Aut}(k)$ then $\mathrm{Aut}(k)$ acts on $\mathrm{Cl}_{\mathfrak{A}}$
- U is (set) invariant under the action

Code

Summary: Class Field Theory associates “in some natural way” some weired finite abelian groups (related to ideals) to finite extensions on number fields with Abelian Galois group.

```
oscar> k, a = NumberField(x^2-10)
oscar> Z_k = MaximalOrder(k)
oscar> I = Ideal(Z_k, 1271, Z_k(107+a))
oscar> Factorisation(I)
    <31, a+14> => 1
    <41, a-16> => 1
oscar> au = Automorphisms(k)
oscar> all(\phi -> \phi(I) == I, au)
  false
```

Code

```
oscar> J = lcm(\phi(I) for phi = au)
oscar> R, mR = RayClassGroup(I)
  C_10, Map:C_10 -> Ideals
oscar> K = RayClassField(mR)
oscar> isNormal(K, QQ)
  false
oscar> S, mS = RayClassGroup(J)
oscar> \Gamma = RayClassField(mS)
oscar> isNormal(\Gamma)
  true
oscar> isSubfield(K, \Gamma)
  true
```

Code

```
oscar> L = NormalClosure(K)
oscar> L == \Gamma
  false
oscar> h = induced_map(mS, mR, x->x)
oscar> U = kernel(h)
oscar> K == RayClassField(mS, quo(S, U)[2])
  true
oscar> act = induced_action(mS, au)
oscar> V = intersect(phi(U) for phi = act)
oscar> NormalClosure(K) == RayClassField(mS, quo(S, V)[2
  true
```

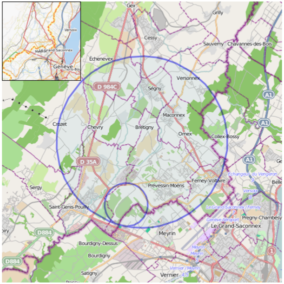
Feynman integrals

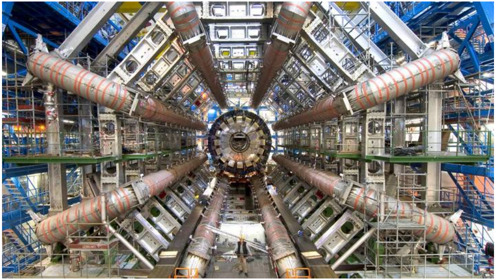
Experimental measurements of scattering processes at the Large Hadron Collider (LHC) require theoretical computation of scattering amplitudes (probabilities of particle interactions) as Feynman integrals.

Feynman integrals

Experimental measurements of scattering processes at the Large Hadron Collider (LHC) require theoretical computation of scattering amplitudes (probabilities of particle interactions) as Feynman integrals.

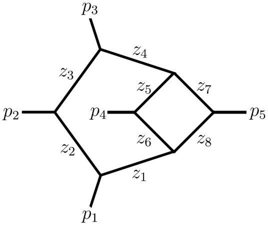
The LHC is the world's largst particle accelerator with a diameter of 9km. It is run by CERN, which a funding of about 1 billion EUR.



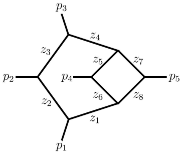


Feynman integrals

A Feynman graph describes an interaction process of particles with external impulses p_i (given constant vectors) and internal impulses z_i (integration variables) which satisfy impulse conservation (the balancing condition):



Feynman integrals



M is the matrix of scalar products of the impulses, $F = \det M$, then the Feynman integral ia a linear combination of integrals

$$\int \frac{dz_1 \cdots dz_m}{z_1 \cdots z_m} F(z)^{\frac{D-L-E-1}{2}}$$

with D a parameter, L the genus of the graph, $E+1$ is the number of external momenta, and $m = LE + \frac{L(L+1)}{2}$.

IBP Relations

Integrals of total differentials vanish, hence yield an integration-by-parts identity

$$0 = \int d\left(\sum_i \frac{a_i(z_1,\dots,z_m)}{z_1\cdots z_m} F(z)^{\frac{D-L-E-1}{2}} dz_1\cdots\widehat{dz_i}\cdots dz_m\right)$$

which translate into a relations

$$\sum_{i=1}^m a_i(z)\frac{\partial F(z)}{\partial z_i} + b(z)F(z) = 0. \qquad (*)$$

Given a full set of relations up to a bound d in z and with $z_i \mid a_i(z)$, any integal reduces to master integrals.

IBP Relations

Integrals of total differentials vanish, hence yield an integration-by-parts identity

$$0 = \int d \left(\sum_i \frac{a_i(z_1, \ldots, z_m)}{z_1 \cdots z_m} F(z)^{\frac{D-L-E-1}{2}} dz_1 \cdots \widehat{dz_i} \cdots dz_m \right)$$

which translate into a relations

$$\sum_{i=1}^{m} a_i(z) \frac{\partial F(z)}{\partial z_i} + b(z) F(z) = 0. \qquad (*)$$

Given a full set of relations up to a bound d in z and with $z_i \mid a_i(z)$, any integal reduces to master integrals. Given

$$M_1 = \langle a(z) \text{ with } (*) \rangle \qquad M_2 = \langle z_i e_i \mid i \leq m \rangle + \langle e_i \mid i > m \rangle$$

we have to calculate $(M_1 \cap M_2)_{\leq d}$.

IBP

Algorithm:

- Find special generators for M_1, then compute $M_1 \cap M_2$ using Gröbner bases over the field of rational functions

IBP

Algorithm:

- Find special generators for M_1, then compute $M_1 \cap M_2$ using Gröbner bases over the field of rational functions
- Generate a large linear system of relations between IBPs in $(M_1 \cap M_2)_{\leq d}$ and compute a RREF over $\mathbb{K}$, trimming the generating system of $(M_1 \cap M_2)$ along our way.

IBP

Algorithm:

- Find special generators for M_1, then compute $M_1 \cap M_2$ using Gröbner bases over the field of rational functions
- Generate a large linear system of relations between IBPs in $(M_1 \cap M_2)_{\leq d}$ and compute a RREF over $\mathbb{K}$, trimming the generating system of $(M_1 \cap M_2)$ along our way.
- Over a function field with a small number of variables, determine good REF via a pivoting aimed at small size and sparseness.

IBP

Algorithm:

- Find special generators for M_1, then compute $M_1 \cap M_2$ using Gröbner bases over the field of rational functions
- Generate a large linear system of relations between IBPs in $(M_1 \cap M_2)_{\leq d}$ and compute a RREF over $\mathbb{K}$, trimming the generating system of $(M_1 \cap M_2)$ along our way.
- Over a function field with a small number of variables, determine good REF via a pivoting aimed at small size and sparseness.
- Use this REFs over univariate function fields to find the degree of the rational function coefficients for the result.

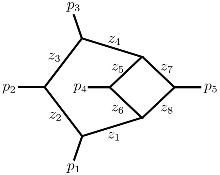
IBP

Algorithm:

- Find special generators for M_1, then compute $M_1 \cap M_2$ using Gröbner bases over the field of rational functions
- Generate a large linear system of relations between IBPs in $(M_1 \cap M_2)_{\leq d}$ and compute a RREF over $\mathbb{K}$, trimming the generating system of $(M_1 \cap M_2)$ along our way.
- Over a function field with a small number of variables, determine good REF via a pivoting aimed at small size and sparseness.
- Use this REFs over univariate function fields to find the degree of the rational function coefficients for the result.
- Compute the coefficients via interpolation, and reduce to the RREF.

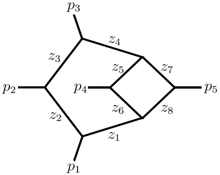
Feynman integrals

- In this way we solved the long-standing open problem of determining a full set of IBPs for the non-planar hexagon box diagram



Feynman integrals

- In this way we solved the long-standing open problem of determining a full set of IBPs for the non-planar hexagon box diagram

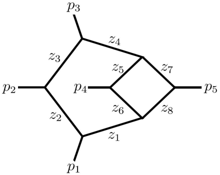


Key algorithmic requirements:

- Fast multivariate function field arithmetic and differentiation.

Feynman integrals

- In this way we solved the long-standing open problem of determining a full set of IBPs for the non-planar hexagon box diagram

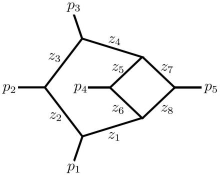


Key algorithmic requirements:

- Fast multivariate function field arithmetic and differentiation.
- Massively parallel computations to obtain the RREF for huge numbers of interpolation points.

Feynman integrals

- In this way we solved the long-standing open problem of determining a full set of IBPs for the non-planar hexagon box diagram



Key algorithmic requirements:

- Fast multivariate function field arithmetic and differentiation.
- Massively parallel computations to obtain the RREF for huge numbers of interpolation points.
- Detection of singular supporting points.
- Exploitation of symmetries

Smoothness of algebraic varieties

For $I = \langle f_1, \dots, f_r \rangle \subset S = K[x_1, \dots, x_n]$, $X = \operatorname{Spec}(S/I) \subset \mathbb{A}^n$

- Jacobian criterion aims at computing the singular locus of X via codimension-sized minors of the Jacobian matrix

$$\mathcal{J}_I = (\partial f_i / \partial x_j)$$

Smoothness of algebraic varieties

For $I = \langle f_1, \ldots, f_r \rangle \subset S = K[x_1, \ldots, x_n]$, $X = \mathrm{Spec}(S/I) \subset \mathbb{A}^n$

- Jacobian criterion aims at computing the singular locus of X via codimension-sized minors of the Jacobian matrix

$$\mathcal{J}_I = (\partial f_i / \partial x_j)$$

  is expensive for large codimension.

Smoothness of algebraic varieties

For $I = \langle f_1, \dots, f_r \rangle \subset S = K[x_1, \dots, x_n]$, $X = \mathrm{Spec}(S/I) \subset \mathbb{A}^n$

- Jacobian criterion aims at computing the singular locus of X via codimension-sized minors of the Jacobian matrix

  $$\mathcal{J}_I = (\partial f_i / \partial x_j)$$

  is expensive for large codimension.
- Hironaka:
  - If X is smooth at $p \in X$, there is smooth hypersurface W

    $$X \cap U \subset W \cap U$$

    in a Zariski open subset $p \in U \subset \mathbb{A}^n$.

Smoothness of algebraic varieties

For $I = \langle f_1, \dots, f_r \rangle \subset S = K[x_1, \dots, x_n]$, $X = \operatorname{Spec}(S/I) \subset \mathbb{A}^n$

- Jacobian criterion aims at computing the singular locus of X via codimension-sized minors of the Jacobian matrix

  $$\mathcal{J}_I = (\partial f_i / \partial x_j)$$

  is expensive for large codimension.
- Hironaka:
  - If X is smooth at $p \in X$, there is smooth hypersurface W

    $$X \cap U \subset W \cap U$$

    in a Zariski open subset $p \in U \subset \mathbb{A}^n$.
  - Iteration yields tree of charts:

Smoothness of algebraic varieties

For $I = \langle f_1, \ldots, f_r \rangle \subset S = K[x_1, \ldots, x_n]$, $X = \mathrm{Spec}(S/I) \subset \mathbb{A}^n$

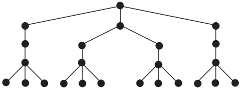
- Jacobian criterion aims at computing the singular locus of X via codimension-sized minors of the Jacobian matrix

  $$\mathcal{J}_I = (\partial f_i / \partial x_j)$$

  is expensive for large codimension.
- Hironaka:
  - If X is smooth at $p \in X$, there is smooth hypersurface W

    $$X \cap U \subset W \cap U$$

    in a Zariski open subset $p \in U \subset \mathbb{A}^n$.
  - Iteration yields tree of charts:



Symmetric GIT-Algorithm

Algorithm to compute GIT-fans with symmetries (B., Keicher, Ren, 2016) via a fan traversal, combining Gröbner bases with computations in polyhedral geometry and group theory.

Symmetric GIT-Algorithm

Algorithm to compute GIT-fans with symmetries (B., Keicher, Ren, 2016) via a fan traversal, combining Gröbner bases with computations in polyhedral geometry and group theory.

- Each GIT-cone is an intersection of orbit cones.

Symmetric GIT-Algorithm

Algorithm to compute GIT-fans with symmetries (B., Keicher, Ren, 2016) via a fan traversal, combining Gröbner bases with computations in polyhedral geometry and group theory.

- Each GIT-cone is an intersection of orbit cones.
- Determine all orbit cones via monomial containment tests.

Symmetric GIT-Algorithm

Algorithm to compute GIT-fans with symmetries (B., Keicher, Ren, 2016) via a fan traversal, combining Gröbner bases with computations in polyhedral geometry and group theory.

- Each GIT-cone is an intersection of orbit cones.
- Determine all orbit cones via monomial containment tests.
- Traverse fan by passing through codim 1 faces to neighbours.

Symmetric GIT-Algorithm

Algorithm to compute GIT-fans with symmetries (B., Keicher, Ren, 2016) via a fan traversal, combining Gröbner bases with computations in polyhedral geometry and group theory.

- Each GIT-cone is an intersection of orbit cones.
- Determine all orbit cones via monomial containment tests.
- Traverse fan by passing through codim 1 faces to neighbours.
- Hash GIT-cones via the binary vector encoding which orbit cones occur in the corresponding intersection. Hash interacts well with symmetry group action.
- Compute in each orbit only a single representative.

Mori Chamber Decomposition of Mov($\overline{M}_{0,6}$)

Cox ring of the moduli space of stable genus zero curves with 6 marked points $\overline{M}_{0,6}$ is $\mathbb{Z}^{16}$-graded, has 40 generators,

Mori Chamber Decomposition of Mov($\overline{M}_{0,6}$)

Cox ring of the moduli space of stable genus zero curves with 6 marked points $\overline{M}_{0,6}$ is $\mathbb{Z}^{16}$-graded, has 40 generators, 225 relations,

Mori Chamber Decomposition of $\mathrm{Mov}(\overline{M}_{0,6})$

Cox ring of the moduli space of stable genus zero curves with 6 marked points $\overline{M}_{0,6}$ is $\mathbb{Z}^{16}$-graded, has 40 generators, 225 relations, and a natural S_6-action.

Mori Chamber Decomposition of Mov($\overline{M}_{0,6}$)

Cox ring of the moduli space of stable genus zero curves with 6 marked points $\overline{M}_{0,6}$ is $\mathbb{Z}^{16}$-graded, has 40 generators, 225 relations, and a natural S_6-action.

Example

The GIT-fan decomposition of the moving cone $\mathrm{Mov}(\overline{M}_{0,6}) \subset \mathbb{R}^{16}$ classifies all small modifications (rational maps which are isomorphisms on open subsets which have a complement of codimension ≥ 2).
The moving cone $\mathrm{Mov}(\overline{M}_{0,6})$ has

$$176\,512\,225$$

GIT-cones of maximal dimension 16, which decompose into

$$249\,605$$

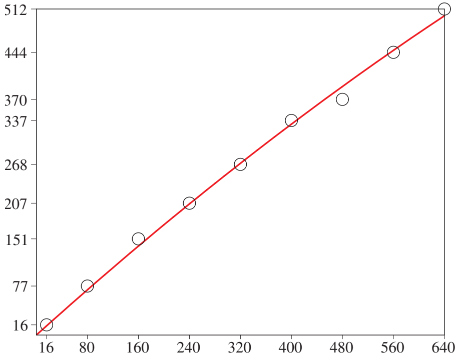
orbits under the S_6-action.

Timings

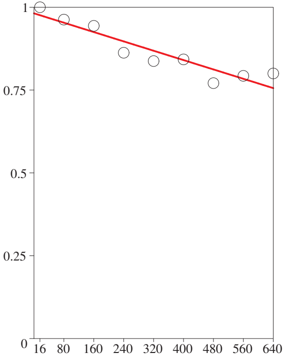
- Singular on 1 core takes 16 days for fan traversal.

Timings

- Singular on 1 core takes 16 days for fan traversal.
- Symmetric GIT-fan algorithm implemented by Christian Reinbold using GPI-Space on 640 cores takes 12.5 minutes.



speedup



parallel efficiency

Tropical varieties

- Algorithm to compute tropical links via Puiseux expansions (Tommy Hofmann, Yue Ren, 2016).

Tropical varieties

- Algorithm to compute tropical links via Puiseux expansions (Tommy Hofmann, Yue Ren, 2016).
- Using fan traversal by Christian Reinbold.

Tropical varieties

- Algorithm to compute tropical links via Puiseux expansions (Tommy Hofmann, Yue Ren, 2016).
- Using fan traversal by Christian Reinbold.
- Puiseux expansions by Santiago Laplagne.

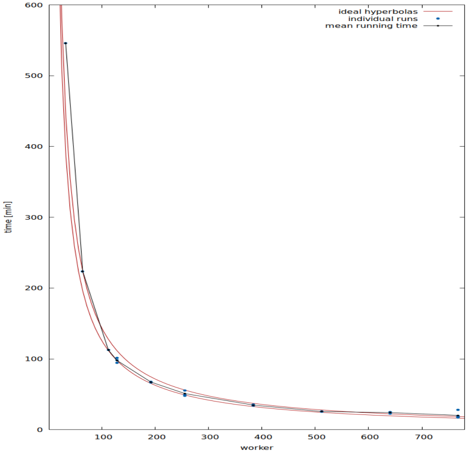
Tropical varieties

- Algorithm to compute tropical links via Puiseux expansions (Tommy Hofmann, Yue Ren, 2016).
- Using fan traversal by Christian Reinbold.
- Puiseux expansions by Santiago Laplagne.
- A parallel and symmetric algorithm for computing tropical varieties by Dominik

Tropical varieties

- Algorithm to compute tropical links via Puiseux expansions (Tommy Hofmann, Yue Ren, 2016).
- Using fan traversal by Christian Reinbold.
- Puiseux expansions by Santiago Laplagne.
- A parallel and symmetric algorithm for computing tropical varieties by Dominik

Example (Tropicalization $\mathbb{G}(3,8)$)



Norm Equations: Theory

Given a maximal order $\mathbb{Z}_k$ and some integer a, try to find all (up to units) $\alpha \in \mathbb{Z}_k$ s.th.

$$N(\alpha) = a$$

This is an important building block in Diophantine Equations.

Algorithm: find all (integral) ideals of the correct norm (which is easy as there is unique factorisation), find the principal ones and take generators.

Theory

Let $\mathfrak{a} = \prod \mathfrak{P}_i^{n_i}$ be an integral ideal of norm $N(\mathfrak{a}) = a$, then

- $n_i \geq 0$ (integrality)
- $N(\mathfrak{a}) = \prod N(\mathfrak{P}_i)^{n_i}$
- $N(\mathfrak{P}_i) = p_i^{f_i}$ for a prime number $p|a$

Hence:

- the possible $\mathfrak{P}_i$ are primes above prime numbers dividing a (hence are known)
- each $p_i|a$ gives rise to a linear equation for the possible exponents n_i
- … and a sign condition: we need all non-negative solutions of a linear equation!

Solution

Assume, for simplicity, $a = p^k$

```
lP = Factorisation(p*Z_k)
fi = [Valuation(p, Norm(P)) for P = lP]
sol = SolveNonNegative(fi, [k])
for s = sol
  A = prod(P[i]^s[i] for i=1:length(lP))
  fl, g = isPrincipal(A)
  if fl
    print("Found: ", g)
  end
end
```

Variety

```
R, [x,y] = PolynomialRing(Q, 2)
A = AffineVariety(y^2-x^3+3*x+1)
P = ProjectiveClosure(A)
K = FunctionField(P)
L = CanonicalRing(P)
T = TropicalVariety(P)
Genus(P)
Genus(T)
P2 = ChangeRing(P, GF(13))
Genus(P2)
UnramifiedCover(P2)
```

Algebraic Geometry

```
K = QQ
R, [x,y,z] = PolynomialRing(K, 3)
X = Spec(R)
Xf = PrincipalOpenSet(X, x^3+3y+z)
I = ideal(R, ...)
F = Sheaf(X, I)
Y = sub(X, F)
```

Algebraic Geometry

```
Z, mp = BlowUp(Y)
G = pullback(Z, mp)
isSmooth(G)
GenericPoints(Z)
AssociatedPoints(Z)
U = CoordinateSystems(Z)
```

Matroids

```
G = some graph
M = Matroid(G)
ConnectedComponents(M)
Dual(M)

V, mp = sub(VectorSpace(K, n), ...)
N = Matroid(V)
ConfigurationPolynomial(N)
Q = ConfigurationBilinearForm(N)
W = DegeneracyScheme(Q, 2)
AssociatedPoints(W)
isReduced(W)
```

Representation Theory

```
G = QuaterionGroup(8)
C = CharacterTable(G)
\chi = IrreducibleCharacters(C)[5]
SchurIndices(\chi)
[<2, 2>, <2, InfPlc(Q)>]
\rho = Representation(\chi)
ChangeRing(\rho, NumberField(x^2+2))
```

Combinatorial types of finite metric spaces

```
H = Hypersimplex(2,6)
save_data(vertices(H), generators(group.symmetric(6)),
          "h26.dat")
run(‘mptopcom --regular < h26.dat > h26.db‘)
T = Database.Open("h26.db")
print (size(T))
339
```

- Sturmfels & Yu 2004: the 339 combinatorial types of regular triangulations of $\Delta(2,6)$ classify the combinatorial types of (tight spans of) finite metric spaces on six taxa

Combinatorial types of finite metric spaces

```
H = Hypersimplex(2,6)
save_data(vertices(H), generators(group.symmetric(6)),
          "h26.dat")
run(‘mptopcom --regular < h26.dat > h26.db‘)
T = Database.Open("h26.db")
print (size(T))
339
```

- Sturmfels & Yu 2004: the 339 combinatorial types of regular triangulations of $\Delta(2,6)$ classify the combinatorial types of (tight spans of) finite metric spaces on six taxa
- mptopcom supposed to run on a cluster, not interactively

What makes a good computer algebra system?

What makes a good computer algebra system?

> *The best system is the one I know how to use!*

Making people use something new is hard:

- it is new: thus incomplete
- it is new: thus buggy
- it is new: I don't know how to use it

Solving any and all of them for OSCAR is easy and hard: it requires people to use OSCAR and help implement it.

More challenges:

> *Finding the "right" abstraction.*

Which is not always the same abstraction in math and computer algebra.

More challenges:

> *Finding the "right" abstraction.*

Which is not always the same abstraction in math and computer algebra.

Worse: it depends on the user: expert vs. non-expert.

More challenges:

Mathematics is inexact, a lot of crucial information is from context!
(I know what I am doing)

Mathematics is inconsistent: a specific adjective has different meaning depending on the context *even when applied to the identical object*.

Thus choices have to be made.

Goals

Different, conflicting goals:

- Expert: big, bigger, huge examples

Goals

Different, conflicting goals:

- Expert: big, bigger, huge examples can be complicated and strange to use
- non-Expert: small (or impossible) examples from a wide area of mathematics,

Goals

Different, conflicting goals:

- Expert: big, bigger, huge examples can be complicated and strange to use
- non-Expert: small (or impossible) examples from a wide area of mathematics, to combine to an interesting result.