Introduction to **Magma**

E. Kowalski

22 October 2008

The **Magma** software

This is the best computational *algebra* package currently available, both in terms of *what objects* it knows about, and *how efficiently* it can compute with them.

Magma is non-commercial, but not free or open source. The best existing free software are

– **Pari/GP** for algebraic number theory;

– **GAP** for group theory and representation theory;

– **Sage**, which tries to combine together many different free mathematical software for a better user interface, around the language **Python**.

There are other software such as **Macaulay**, **Singular**, **KANT**, but I don't know so much about them.

What **Magma** knows

Magma has some knowledge of the following types of mathematical objects (non-exhaustive list):
– Sets, sequences, multisets, tuples, strings;
– Rings, algebras, fields, including exact rational numbers, arbitrary (finite) precision real and complex numbers;
– Finite groups, permutation groups, finitely presented groups, Lie groups, Coxeter groups;
– Representation theory of finite and algebraic groups;
– Number fields, function fields, local fields, finite fields;
– Modular forms;
– Schemes for algebraic geometry, commutative algebras (in particular elliptic curves and modular curves);
– Codes, graphs, finite geometries...

Language

Magma is also a complete programming language, which is easy to learn and has very natural constructs to "express" mathematical constructions. For instance

```
F:=FiniteField(3); A<x>:=PolynomialRing(F);
liste:=[x^3+a*x^2+b*x+c : a,b,c in F |
           IsSquarefree(x^3+a*x^2+b*x+c)];
```

gives an ordered list of all monic polynomials of degree 3 in $\mathbf{F}_3[x]$ which are squarefree.
(Note that data in **Magma** is strongly typed, so one must often get used to explicit typing and conversions, such as defining F and A above).

Other features

Magma also contains databases which make experimentation with some objects particularly easy:
– Groups of small order, almost simple groups, transitive permutation groups of small degree;
– Graphs, codes and lattices;
– Elliptic curves...
And its algorithms are usually among the best known, and highly optimized. For instance, it can compute *L*-functions of hyperelliptic curves in families using very recent algorithms.
For many types of objects, **Magma** also provides a way to get a "random" element, which can be very useful for testing and exploring (though there isn't that much support for probability in general).

An example

F. Jouve, D. Zywina and I found[1] the first entirely explicit integral polynomial $P \in \mathbf{Z}[T]$ such that the splitting field $K/\mathbf{Q}$ generated by the roots of P is a Galois extension with

$$\mathrm{Gal}(K/\mathbf{Q}) \simeq W(E_8)$$

where $W(E_8)$ is the Weyl group of the exceptional Lie group of type E_8.

There are three components of the proof:

– Find a candidate;

– Show that the Galois group is a subgroup of $W(E_8)$;

– Show that it is not a proper subgroup.

Magma was used for the first and third step.

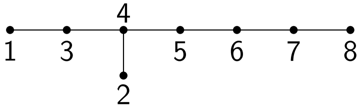
[1] `arXiv:0801.1733`.

Background on $W(E_8)$

$W(E_8)$ is a finite group of order $2^{14} \cdot 3^5 \cdot 5^2 \cdot 7$. It has a faithful permutation representation of degree 240 and a presentation as Coxeter group

$$W(E_8) = \langle w_1, \ldots, w_8 \mid w_i^2 = (w_i w_j)^{m(i,j)} = 1, \; i \neq j \rangle$$

where $m(i,j) = 2$ except if (i,j) are connected in the Dynkin diagram



The composition factors are given by

$$\mathbf{Z}/2\mathbf{Z}, \quad O^+(8, \mathbf{F}_2), \quad \mathbf{Z}/2\mathbf{Z}.$$

Constructing the candidate

The idea is the principle that if $G/\mathbf{Z}$ is a split semisimple algebraic group, and $\rho : G \to GL(N)$ is a faithful representation, then for a "random" element $g \in G(\mathbf{Z})$, the characteristic polynomial

$$P_{\rho,g} = \det(T - \rho(g)) \in \mathbf{Z}[T]$$

should have splitting field with Galois group $W(G)$, the Weyl group of G.
If $G = SL(N)$ and ρ is the inclusion, then $W(G)$ is the symmetric group on N letters, which is the typical Galois group for a random polynomial, so this is not too surprising.

(cont.)

We take $G = E_8/\mathbf{Z}$, the split Chevalley group of type E_8, and $\rho : G \to GL(248)$ the adjoint representation on the Lie algebra. To construct a “random” element (of low complexity), we take the Chevalley generators (as given by **Magma**)

$$x_1, \ldots, x_8, x_9, \ldots, x_{16}$$

and their product

$$g = x_1 \cdots x_{16}.$$

So our candidate is

$$P = \det(T - \rho(x_1 \cdots x_{16})),$$

and we divide by $(T-1)^8$ (because any P obtained this way is divisible by this factor).

The code

Here is the **Magma** code to do this:

```
A<T>:=PolynomialRing(RationalField());
E8:=GroupOfLieType("E8",RationalField());
gen:=AlgebraicGenerators(E8);
rho:=AdjointRepresentation(E8);
g:=Identity(E8);
for i in gen do g:=g*i ; end for;
m:=rho(g);
pol:=CharacteristicPolynomial(m) div (T-1)^8;
```

Note that it is highly readable for a mathematician.

Upper bound on the Galois group

We prove a fairly simple lemma that states that for any polynomial obtained in this manner for a regular semisimple element g, the Galois group is in a natural way a subgroup of $W(E_8)$.
To explain this, recall we can also write $W(E_8) \simeq N(T)/T$ where $T \subset G$ is a fixed (split) maximal torus $T \simeq \mathbf{G}_m^8$.
The idea is to consider

$$X = \{t \in T \mid t \text{ and } g \text{ are conjugate}\},$$

show that $N(T)/T$ acts simply transitively on X, observe that the Galois group of K acts on X, and then use the map

$$\mathrm{Gal}(K/\mathbf{Q}) \to W(E_8)$$

that sends σ to the unique $n \in W(E_8)$ such that $\sigma(t_0) = n^{-1} \cdot t_0$, where $t_0 \in X$ is fixed.

Lower bound on the Galois group

The basic principle is this: if $P \in \mathbf{Z}[T]$ of degree d factors modulo a prime p as

$$P = S_1 \cdots S_d \pmod{p}$$

where S_i is the product of $n_i \geq 0$ distinct irreducible polynomials of degree i in $\mathbf{F}_p[T]$, then in the faithful permutation representation

$$\mathrm{Gal}(K/\mathbf{Q}) \to \mathfrak{S}_d$$

obtained by the action on the roots of P, the Galois group contains elements with cycle structure given by n_i disjoint cycles of length i for $1 \leq i \leq d$.
For instance if P is irreducible modulo p, then G contains a d-cycle.

(cont.)

Magma can construct the permutation representation for $W(E_8)$ on 240 objects and compute the cycle structure of P modulo primes. Moreover, **Magma** knows all the cycle structures of conjugacy classes of $W(E_8)$ and all maximal subgroups of $W(E_8)$. So one can try to find, by looking at small primes, enough conjugacy classes in $G \subset W(E_8)$ so that the only possibility is that $G = W(E_8)$.

The code

This lists all the cycle structures of all conjugacy classes of maximal subgroups:

```
W:=WeylGroup(E8);
max:=MaximalSubgroups(W);
for m in max do print("----");
  for c in ConjugacyClasses(m`subgroup) do
    print(CycleStructure(c[3]));
  end for;
end for;
```

(cont.)

We find by reducing modulo 11 that G contains an element with cycle structure

$$(16, 15), \text{ i.e. a product of 16 disjoint 15-cycles}$$

and modulo 7 that G contains an element with cycle structure

$$(2, 4),\ (29, 8),\ \text{i.e., a product of 2 disjoint 4-cycles, and 29 disjoint 8-cycles}$$

Inspection of the data using **Magma** shows no proper subgroup of $W(E_8)$ has these properties.

Question. *Is there a conceptual proof of this?*

Another example

We only needed two reductions to prove that our Galois group was the full $W(E_8)$. Is it extraordinarily good luck, or normal?
More generally, let $K/\mathbf{Q}$ be a finite Galois extension with Galois group G. For p prime in K (not dividing the discriminant) we have a conjugacy class $F_p \in G^\sharp$, uniquely determined by the fact that

$$x^{F_p} \equiv x^p \pmod{\mathfrak{p}}$$

for all x in the ring of integers $\mathbf{Z}_K$ of K and a fixed prime ideal $\mathfrak{p} \subset \mathbf{Z}_K$ such that $\mathfrak{p} \cap \mathbf{Z} = p\mathbf{Z}$.
For how many primes do we need to compute F_p before we are sure to generate G?

Probabilistic model

Here is a probabilistic model for this. Let $G \neq 1$ be a finite group.
Definition. A family $(C_1, \ldots, C_m)$ of conjugacy classes in G *generates* G if $(g_1, \ldots, g_m)$ generate G for any choice of $g_i \in C_i$.

Example. The family of *all* conjugacy classes of G generates G.

Now assume given an infinite sequence (X_n) of G-valued random variables, independent, and uniformly distributed:

$$\mathbf{P}(X_n = g) = \frac{1}{|G|} \text{ for all } n \text{ and } g \in G.$$

We want to understand the *waiting time*

$$\tau_G = \min\{n \geq 1 \mid (X_1^\sharp, \ldots, X_n^\sharp) \text{ generate } G\},$$

which is another random variable.

Chebotarev invariants

We define in particular

$$c(G) = \mathbf{E}(\tau_G) = \sum_{n \geq 1} \mathbf{P}(\tau_G \geq n) = 1 + \sum_{n \geq 1} \mathbf{P}((X_1^\sharp, \ldots, X_n^\sharp) \text{ generate } G)$$

the expectation (average) of τ_G, and

$$c_2(G) = \mathbf{E}(\tau_G^2)$$

the mean-square average.

What can be said of these invariants?

Let $\max(G)$ be the set of conjugacy classes of (proper) maximal subgroups of G. For $I \subset \max(G)$, let

$$H_I = \bigcap_{H \in I} H^\sharp \subset G$$

be the union of all conjugacy classes which intersect *all* the H in I. Let

$$\nu(H_I^\sharp) = \frac{|H_I^\sharp|}{|G|}$$

be the density of this set.

(cont.)

An easy inclusion-exclusion argument leads to formulas

$$c(G) = -\sum_{\emptyset \neq I \subset G} \frac{(-1)^{|I|}}{1 - \nu(H_I^\sharp)}$$

$$c_2(G) = -\sum_{\emptyset \neq I \subset G} (-1)^{|I|} \frac{1 + \nu(H_I^\sharp)}{(1 - \nu(H_I^\sharp))^2}.$$

These formulas are useful for certain theoretical computations when the maximal subgroups are well known, e.g., $\mathbf{Z}/n\mathbf{Z}$, $\mathbf{F}_p^k$,

$$H_q = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} \mid a \in \mathbf{F}_q^\times,\ b \in \mathbf{F}_q \right\}.$$

Experiments

It can also be programmed and used for experiments.

```
Chebotarev:= function (G)
  C := ConjugacyClasses(G);
  f:=ClassMap(G);
  M := MaximalSubgroups(G);

  // Construct an array indicating which maximal subgroups
  // intersect which conjugacy classes
  J := [ [false : i in [1..#C]] : k in [1..#M] ];
  for k in [1..#M] do
    H := M[k]`subgroup;
    CH := ConjugacyClasses(H);
    for j in [1..#CH] do
      J[k][f(CH[j][3])] := true;
    end for;
  end for;
```

(cont.)

```
  // Then loop to compute the invariants
  c:=0.0; s:=0.0;
  for I in Subsets({1..#M}) do
    if #I ne 0 then
      v:=0;
      for i in [1..#C] do
        if forall(t) {k: k in I | J[k][i]} then
          v:= v + C[i][2]/#G;
        end if;
      end for;
      c  := c + (-1)^(#I+1)/(1-v);
      s  := s+ (-1)^(#I)/(1-v)*(1-2/(1-v));
    end if;
  end for;
  return([c,s]);
end function;
```

Some results

Name	Order	$c(G)$	$c_2(G)$
$W(G_2)$	12	$4.315\underline{15}$	$23.45407\ldots$
H_{17}	272	$17.21053\ldots$	$562.3851\ldots$
$W(C_4)$	384	$4.864890\ldots$	$29.10488\ldots$
$W(F_4)$	1152	$5.417656\ldots$	$35.12470\ldots$
M_{11}	7920	$4.850698\ldots$	$29.72918\ldots$
$G_2(\mathbf{F}_2)$	12096	$5.246204\ldots$	$34.24515\ldots$
$Sz(8)$	29120	$3.101639\ldots$	$11.92233\ldots$
$W(E_6)$	51840	$4.470824\ldots$	$23.93050\ldots$
M_{12}	95040	$4.953188\ldots$	$29.53947\ldots$
J_1	175560	$3.423739\ldots$	$14.76364\ldots$
M_{22}	443520	$4.164445\ldots$	$22.70981\ldots$
J_2	604800	$3.891094\ldots$	$18.06798\ldots$
$W(C_7)$	645120	$4.632612\ldots$	$25.54504\ldots$
$W(E_7)$	2903040	$5.398250\ldots$	$36.04850\ldots$
$G_2(\mathbf{F}_3)$	4245696	$4.511630\ldots$	$24.06106\ldots$

(cont.)

Name	Order	$c(G)$	$c_2(G)$
M_{23}	10200960	4.030011...	20.98580...
$W(C_8)$	10321920	4.928996...	28.53067...
$Sz(32)$	32537600	2.755449...	9.107751...
HS	44352000	4.002027...	18.66327...
J_3	50232960	3.972161...	19.09843...
$W(C_9)$	185794560	4.716359...	26.41344...
M_{24}	244823040	4.967107...	29.84845...
$W(E_8)$	696729600	4.194248...	20.79438...
McL	898128000	4.531381...	25.52575...
$G_2(\mathbf{F}_5)$	5859000000	3.855868...	18.68766...
$\mathfrak{S}_{16}$	20922789888000	4.461633...	24.12713...
$\mathfrak{S}_{17}$	355687428096000	4.282141...	22.79488...
$\mathfrak{S}_{18}$	6402373705728000	4.531784...	24.67680...
$\mathfrak{S}_{19}$	121645100408832000	4.308469...	23.01145...
$\mathfrak{S}_{20}$	2432902008176640000	4.497047...	24.37207...
Rub	43252003274489856000	5.668645...	36.78701...