Some new GP features

A tutorial

B. Allombert

IMB
CNRS/Université Bordeaux 1

07/01/2013

Simultaneous assignments

The syntax `[a, b, c] = V` set `a` to `V[1]`, `b` to `V[2]` and `c` to `V[3]`. Now it is also possible to use it inside `my()` and `local()`.

Some examples of use :

```
mygcdext(a,b)=
{
  if (b==0,[1,0],
    my([q,r] = divrem(a,b));
    my([u,v] = mygcdext(b,r));
     [v, u - q*v]);
}
mygcdext(17,5)
```

Multi-vector operations

`[f(x,y) | x<-V; y<-W]` gives the vector `[f(V[1],W[1]), f(V[1],W[2]),...,f(V[#V],W[#W])`.

`[f(x,y) | x<-V, P(x); y<-W, Q(x,y)]` only keep the components such that the predicates $P(x)$ and $Q(x, y)$ is true.

Beware of the semicolon !

Examples :

```
? [a^2+b^2|a<-[1..5];b<-[1..5],gcd(a,b)==1]
%1 = [2,5,10,17,26,5,13,29,10,13,25,34,17,
      25,41,26,29,34,41]
? [a^2+b^2|a<-[1..10],isprime(a); \
              b<-[1..10],a!=b && isprime(b)]
%2 = [13,29,53,13,34,58,29,34,74,53,58,74]
? [[a,b,c]|a<-[1..5];b<-[1..a];c<-[1..b]]
%3 = [[1,1,1],[2,1,1],[2,2,1],[2,2,2],[3,1,1],
[3,2,1],[3,2,2],[3,3,1],[3,3,2],[3,3,3],[4,1,1],
[4,2,1],[4,2,2],[4,3,1],[4,3,2],[4,3,3],[4,4,1],
[4,4,2],[4,4,3],[4,4,4],[5,1,1],[5,2,1],[5,2,2],
[5,3,1],[5,3,2],[5,3,3],[5,4,1],[5,4,2],[5,4,3],
[5,4,4],[5,5,1],[5,5,2],[5,5,3],[5,5,4],[5,5,5]]
```

strictargs

Normally, the arguments of user-defined GP function are all optionnals. Using `default(strictargs,1)`, the arguments are mandatory unless an explicit default value is provided.

strictargs

```
default(strictargs,0);
fun(a,b=1)=[a,b];
fun(2)
fun()
default(strictargs,1);
fun(a,b=1)=[a,b];
fun(2)
fun()
  ***   missing mandatory argument 'a' in user
  ***   function.
```

default

The option `--default` allows to set defaults in the command line

```
gp --default prompt="GP>"
parisize = 8000000, primelimit = 500000
GP>
```

forpart

`forpart` allows to loop over partitions :

```
forpart(X=5,print(X))
Vecsmall([5])
Vecsmall([1,4])
Vecsmall([2,3])
Vecsmall([1,1,3])
Vecsmall([1,2,2])
Vecsmall([1,1,1,2])
Vecsmall([1,1,1,1,1])
```

forpart

It is possible to restrict the lengths and the summand and to fill with 0.

```
\\ at most 3 non-zero parts, all <= 4
forpart(v=5,print(Vec(v)),4,3)
[1, 4]
[2, 3]
[1, 1, 3]
[1, 2, 2]
\\ between 2 and 4 parts less than 5, fill with zer
forpart(v=5,print(Vec(v)),[0,5],[2,4])
[0, 0, 1, 4]
[0, 0, 2, 3]
[0, 1, 1, 3]
[0, 1, 2, 2]
[1, 1, 1, 2]
```

qfauto

GP includes a port of the program ISOM by Bernt Souvignier for computation of automorphisms and isomorphisms of lattices.

- `qfauto` : compute the automorphism group of a lattice.
- `qfisom` : compute an isomorphism between two lattices.
- `qfautoexport` : export the group to GAP or MAGMA format.
- `qfisominit` : precompute invariants for qfisom.

```
qfauto(matid(3))
%1 = [48,[[-1,0,0;0,-1,0;0,0,-1],
     [0,0,1;0,1,0;1,0,0] ,[0,0,1;-1,0,0;0,1,0]]]
K=nfinit(x^3-3*x+1); L=round(K.t2)
%2 = [3,0,0;0,6,-3;0,-3,6]
qfauto(L)
%3 = [24,[[-1,0,0;0,-1,0;0,0,-1],
     [1,0,0;0,0,1;0,1,0] ,[1,0,0;0,1,0;0,1,-1]]]
T=qflllgram(L); M = T~*L*T; qfisom(L,M)
%4 = [1,0,0;0,0,1;0,1,0]
Q=qfisominit(L); qfisom(Q,M)
%5 = [1,0,0;0,0,1;0,1,0]
```

genus2red

GP includes a port of the program genus2reduction by Cohen and Liu to compute the reduction at odd primes of a genus 2 curve $C/\mathbb{Q}$, defined by the hyperelliptic equation $y^2 + Q(x)y = P(x)$.

genus2red

```
genus2red(0,x^6 + 3*x^3 + 63, 3)
%1 = [59049, Mat([3, 10]), x^6 + 3*x^3 + 63,
 [3, [1, []], ["[III{9}] page 184", [3, 3]]]]
[N, FaN, T, V] = genus2red(x^3-x^2-1, x^2-x);
               \\ X_1(13), global reduction
[N, FaN]
%3 = [169,Mat([13,2])]
\\ in particular, good reduction at 2 !
T
%4 = x^6 + 58*x^5 + 1401*x^4 + 18038*x^3
   + 130546*x^2 + 503516*x + 808561
V
%5 = [[13, [5, [Mod(0, 13), Mod(0, 13)]],
["[I{0}-II-0] page 159", []]]]
```

factorization matrices

Arithmetic functions now accept factorization matrices, any of $f(N)$, $f(factor(N))$ or $f([N, factor(N)])$.

```
M=[randomprime(10^100),2;randomprime(10^100),1];
N=factorback(M);
core(M)
divisors(M)
core([N,M])
moebius([N,M])
```

sumdivmult

`sumdivmult` allows to sum arithmetic multiplicative function

```
sumdiv(100,d,moebius(d)*d)
%1 = 4
sumdivmult(100,d,moebius(d)*d)
%2 = 4
sumdivmult(100!,d,moebius(d)*d)
%3 = -2773996904277378399530788061184 00000
? numdiv(100!)
%4 = 39001250856960000
```

matqr

`matqr` compute the QR decomposition of a real square matrix.

```
M=mathilbert(4);
[Q,R]=matqr(M);
norml2(Q*R-M)
[H,R]=matqr(M,1);
norml2(mathouseholder(H, M)-R)
```

nfinit

It is now possible to compute orders of number fields that are maximal at a set of primes.

```
P=polcompositum(x^4+437*x+19,x^5-571*x+27)[1];
B=nfbasis(P,[2,7]);
D=nfdisc(P,[2,8;7,5]]);
K=nfinit([P,10^6]);
nfcertify(K);
```

Miscellaneous

```
sqrtnint(65,3)
%1 = 4
lambertw(3)
%2 = 1.0499088949640399599886970705528979046
%*exp(%)
%3 = 3
readstr("CHANGES")[1]
%4 = "# $Id$"
seralgdep(sum(i=0,5,y^2^i)*Mod(1,2)+O(y^64),2,2)
%5 = x^2+x+y
gcdext == bezout
```