New GP features
and how to use them

B. Allombert

IMB
CNRS/Université Bordeaux 1

15/01/2013
New GP features
Simultaneous assignments

The syntax \([a, b, c] = V\) set a to \(V[1]\), b to \(V[2]\) and c to \(V[3]\).

Some examples of use:

\([a, b] = [b, a] \quad \text{\textbackslash{} Swap a and b;}\)
\([q, r] = \text{divrem}(17, 5) \quad \text{\textbackslash{} set } q = 3, \ r = 2\)
\([u, v, d] = \text{bezout}(17, 5) \quad \text{\textbackslash{} set } u = -2, \ v = 7, \ d = 1\)
while(b, [a, b] = [b, a\%b]) \quad \text{\textbackslash{} Euclid algorithm}
Multi-if

GP allows if() statement with an arbitrary number of clauses. This can serve as a replacement for 'else if' of for switches/cases, with less parenthesis.

```plaintext
mycmpold(x, y) = if(x < y, -1, if(x > y, 1, 0));
mycmp(x, y) = if(x < y, -1, x > y, 1, 0)
```

```plaintext
mytype(x) =
{
    t = type(x);
    if (t == "t_INT", "integer",
        t == "t_REAL", "real",
        t == "t_COMPLEX", "complex",
        "unknown")
}
Component extraction

Extracting a subvector:

\[ V[2..4] = [V[2], V[3], V[4]] \]
\[ V[^2] = [V[1], V[3], \ldots, V[\#V]] \]

Extracting a submatrix:

\[ M[, 2..4] = \text{matrix with columns } M[,2] \ldots M[,4] \]
\[ M[2..4,] = \text{matrix with rows } M[2, \ldots M[4,] \]

idem with \(^2\) instead of 2..4 to skip 2.
Example:

```gp
comatrix(M) = matrix(#M, #M, i, j, 
                  (-1)^(i+j) * matdet(M[^[i,^j]]));
M = mathilbert(3)
%7 = [1, 1/2, 1/3; 1/2, 1/3, 1/4; 1/3, 1/4, 1/5]
C = comatrix(M) ~/ matdet(M)
%8 = [9, -36, 30; -36, 192, -180; 30, -180, 180]
C*M
%9 = [1, 0, 0; 0, 1, 0; 0, 0, 1]
```
New GP features

Concatenation

This is the reverse operation: matconcat() allow to build matrices by block:

\[
\text{concat}([1,2],[3,4]) = [1,2,3,4]
\]
\[
M1 = [1,2;3,4]; \quad M2 = [5,6;7,8];
\]
\[
\text{matconcat}([M1,M2;0,M1]) = \\
[1 \ 2 \ 5 \ 6]
\]
\[
[3 \ 4 \ 7 \ 8]
\]
\[
[0 \ 0 \ 1 \ 2]
\]
\[
[0 \ 0 \ 3 \ 4]
\]
New GP features

Vector operations

Ranges:
\[ n..m \] gives the vector \[ n, n+1, \ldots, m \].

Apply:
\[ \{ f(x) \mid x \in V \} \] gives the vector
\[ \{ f(V[1]), \ldots, f[V[\#V]] \} \].

Select:
\[ \{ x \mid x \in V, P(x) \} \] only keep the components such that \( P \) is true.

Both:
\[ \{ f(x) \mid x \in V, P(x) \} \]
Examples:

? [1..5]
%1 = [1, 2, 3, 4, 5]
? [x^2 | x <- [1..5]]
%2 = [1, 4, 9, 16, 25]
? [x | x <- [1..5], isprime(x)]
%3 = [2, 3, 5]
? [x^2 | x <- [1..5], isprime(x)]
%4 = [4, 9, 25]
Iterators

Unbounded forprime: `forprime(p=2, ...)`

```plaintext
? forprime(p=2, if(Mod(2, p^2)^(p-1) == 1, return(p)))
%1 = 1093
```

Loops over lattices vectors of small norms:

```plaintext
? forqfvec(v, matid(6), 1, print(v))
[0, 0, 0, 0, 0, 1]~
[0, 0, 0, 0, 1, 0]~
[0, 0, 0, 1, 0, 0]~
[0, 0, 1, 0, 0, 0]~
[0, 1, 0, 0, 0, 0]~
[1, 0, 0, 0, 0, 0]~
```
New GP features

**Miscellaneous**

? digits(83521) \ \ digits in base 10
%17 = [8, 3, 5, 2, 1]

? digits(83521,16) \ \ digits in base 16
%18 = [1, 4, 6, 4, 1]

? randomprime([100,200]) \ \ between 100 and 200
%19 = 191

? printsep("<",1,2,3,4)
1<2<3<4

? vecmax([2,1,4,3],&m)
%20 = 4

? m
%21 = 3

? ellmul == ellpow