

mvpoli - Prolog Tests

avivace - r0.2

coefficients

```
coefficients(y^0, C1),
coefficients(0 * x -3, C2),
coefficients(-y * x + y, C3),
coefficients(0 * x, C4),
coefficients(poly([m(-3, 0, []), m(-4, 1, [v(1, x)])]), C5),
coefficients(m(-1, 1, [v(1, x)]), C6),
coefficients(m(1, 0, []), C7).

C1 = [1]
C2 = [-3]
C3 = [-1, 1]
C4 = []
C5 = [-3, -4]
C6 = [-1]
C7 = [1]
```

variables

```
variables(x * y * z, V1),
variables(0 * x * y, V2),
variables(x^0, V3),
variables(-x + y ^2, V4),
variables(3 * b -a, V5),
variables(m(1, 0, []), V6),
variables(poly([m(-3, 0, []), m(-4, 1, [v(1, x)])]), V7).

V1 = [x, y, z],
V2 = V3, V3 = V6, V6 = [],
V4 = [x, y],
V5 = [a, b],
V7 = [x].
```

monomials

```
monomials(x * y + 3, M1),
monomials(0 * x -y, M2),
monomials(-pippo^123 * pluto, M3),
monomials(x + x + x + x, M4),
monomials(x + x + x + 0 * x, M5).

M1 = [m(3, 0, []), m(1, 2, [v(1, x), v(1, y)])],
M2 = [m(-1, 1, [v(1, y)])],
M3 = [m(-1, 124, [v(123, pippo), v(1, pluto)])],
M4 = [m(4, 1, [v(1, x)])],
M5 = [m(3, 1, [v(1, x)])].
```

maxdegree

```
maxdegree(x ^ 0, MAX1),
maxdegree(pippo ^ 3 * pippo, MAX2),
maxdegree(a ^ 3 -b + ciao^2, MAX3),
maxdegree(-a -b ^ 7 -c^2, MAX4),
maxdegree(-b -b ^ 7 -b^2, MAX5).
```

```
MAX1 = 0
MAX2 = 4
MAX3 = 3
MAX4 = 7
MAX5 = 7
```

mindegree

```
mindegree(x ^ 0, MIN1),
mindegree(pippo ^ 3 * pippo, MIN2),
mindegree(a ^ 3 -b + ciao^2, MIN3),
mindegree(-a -b ^ 7 -c^2, MIN4),
mindegree(-b ^ 0 -b ^ 7 -b^2, MIN5).
```

```
MIN1 = 0
MIN2 = 4
MIN3 = MIN4 = 1
MIN5 = 0
```

polyplus

```
polyplus(a^ 3, 3 * b, PP1),
polyplus(0 * x, b, PP2),
polyplus(a + c ^ 3, a ^ 0, PP3),
polyplus(x ^ 0, 3 *x ^ 0, PP4),
polyplus(3, 5, PP5),
polyplus(m(-1, 1, [v(1, x)]), 3 * x, PP6),
polyplus(poly([m(1, 2, [v(1, a), v(1, b)])]), m(1, 2, [v(1, a), v(1, b)]),
PP7).
```

```
PP1 = poly([m(3, 1, [v(1, b)]), m(1, 3, [v(3, a)])]),
PP2 = poly([m(1, 1, [v(1, b)])]),
PP3 = poly([m(1, 0, []), m(1, 1, [v(1, a)]), m(1, 3, [v(3, c)])]),
PP4 = poly([m(4, 0, [])]),
PP5 = poly([m(8, 0, [])]),
PP6 = poly([m(2, 1, [v(1, x)])]),
PP7 = poly([m(2, 2, [v(1, a), v(1, b)])]).
```

polyminus

```
polyminus(a^ 3, 3 * b, PM1),
polyminus(0 * x, b, PM2),
polyminus(a + c ^ 3, a ^ 0, PM3),
polyminus(x ^ 0, 3 *x ^ 0, PM4),
polyminus(3, 5, PM5),
polyminus(m(-1, 1, [v(1, x)]), 3 * x, PM6),
```

```
polyminus(poly([m(1, 2, [v(1, a), v(1, b)])]), m(1, 2, [v(1, a), v(1, b)]),
PM7).
```

```
PM1 = poly([m(-3, 1, [v(1, b)]), m(1, 3, [v(3, a)])]),
PM2 = poly([m(-1, 1, [v(1, b)])]),
PM3 = poly([m(-1, 0, []), m(1, 1, [v(1, a)]), m(1, 3, [v(3, c)])]),
PM4 = PM5, PM5 = poly([m(-2, 0, [])]),
PM6 = poly([m(-4, 1, [v(1, x)])]),
PM7 = poly([m(0, 0, [])]).//caso molto particolare, coef è zero e quindi
elimino monomio!
```

polytimes

```
polytimes(1, a * b * c, PT1),
polytimes(0, a * b * c, PT2),
polytimes(a^0, a, PT3),
polytimes(pippo * pluto, 0 + pippo, PT4),
polytimes(x + y, a * b * c, PT5),
polytimes(x + y, a + b, PT6),
polytimes(a + b, a + b, PT7).
```

```
PT1 = poly([m(1, 3, [v(1, a), v(1, b), v(1, c)])]),
PT2 = poly([m(0, 0, [])]),
PT3 = poly([m(1, 1, [v(1, a)])]),
PT4 = poly([m(1, 3, [v(2, pippo), v(1, pluto)])]),
PT5 = poly([m(1, 4, [v(1, a), v(1, b), v(1, c), v(1, x)]), m(1, 4, [v(1, a),
v(1, b), v(1, c), v(1, y)])]),
PT6 = poly([m(1, 2, [v(1, a), v(1, x)]), m(1, 2, [v(1, a), v(1, y)]), m(1, 2,
[v(1, b), v(1, x)]), m(1, 2, [v(1, b), v(1, y)])]),
PT7 = poly([m(1, 2, [v(2, a)]), m(2, 2, [v(1, a), v(1, b)]), m(1, 2, [v(2,
b)])]).
```

as_monomial

```
as_monomial(3 * ciao ^2 * bea ^ 3, AM1),
as_monomial(3 * x ^ 0, AM2),
as_monomial(0 * a * b * c * quellochevoglio, AM3),
as_monomial(-foo^42, AM4),
as_monomial(-42 * foo * bar ^ 42, AM5),
as_monomial(pippo * pippo * pippo * pippo * pippo^5, AM6),
as_monomial(a * b * a * b * a * c * b, AM7).
```

```
AM1 = m(3, 5, [v(3, bea), v(2, ciao)]).
AM2 = m(3, 0, [])
AM3 = m(0, 0, [])
AM4 = m(-1, 42, [v(42, foo)]).
AM5 = m(-42, 43, [v(42, bar), v(1, foo)])
AM6 = m(1, 9, [v(9, pippo)]).
AM7 = m(1, 7, [v(3, a), v(3, b), v(1, c)]).
```

as_polynomial

```

as_polynomial( 0 + 0 + 0, AP1),
as_polynomial( 0 + 0 + x, AP2),
as_polynomial(a^2 + b^2 + a * b + b * a, AP3),
as_polynomial(pippo + gennaro + 3 * pippo, AP4),
as_polynomial(a * b + c * d + a * d + s * d, AP5),
as_polynomial(-3 * pippo -4*gennaro + 1, AP6),
as_polynomial(-3 * a ^ 4 +12 * a ^2 * a ^2, AP7).

AP1 = poly([m(0, 0, [])]),
AP2 = poly([m(1, 1, [v(1, x)])]),
AP3 = poly([m(1, 2, [v(2, a)]), m(2, 2, [v(1, a), v(1, b)]), m(1, 2, [v(2,
b)])]),
AP4 = poly([m(1, 1, [v(1, gennaro)]), m(4, 1, [v(1, pippo)])]),
AP5 = poly([m(1, 2, [v(1, a), v(1, b)]), m(1, 2, [v(1, a), v(1, d)]), m(1, 2,
[v(1, c), v(1, d)]), m(1, 2, [v(1, d), v(1, s)])]),
AP6 = poly([m(1, 0, []), m(-4, 1, [v(1, gennaro)]), m(-3, 1, [v(1, pippo)])]),
AP7 = poly([m(9, 4, [v(4, a)])]).

```

polyval

```

polyval(3, [], PV1),
polyval(-x, [12], PV2),
polyval(-2*x + y, [3, 2], PV3),
polyval(0 * x * pluto, [2, 1], PV4),
polyval(x * pippo, [0, 100000], PV5),
polyval(x ^ 0, [], PV6),
polyval(poly([m(1, 1, [v(1, a)]), m(4, 1, [v(1, x)])]), [1, 1], PV7),
polyval(poly([m(3, 1, [v(1, x)]), m(1, 2, [v(2, a)])]), [-1, -2], PV8).

PV1 = 3.
PV2 = -12.
PV3 = -4.
PV4 = parametri non validi //monomio = m(0, 0, [])
PV5 = 0.
PV6 = 1.
PV7 = 5.
PV8 = -5.

```

1

```

QUERY
poly = 5 * x + y ^ 3 - k*x*x*x
polyval in [90, 80, 40]

RESULTS
5 * X + Y^3 + -K * X^3
polyval = -46015600.

```

2

```

QUERY

```

```
poly1 = x * x + x * y + a * y
poly2 = 3 * y + x * x
polyminus(poly1,poly2)
```

RESULTS

```
polyminus = -3 * Y + A * Y + X * Y
```

3

QUERY

```
poly1 = x * x + x * y + a * y
poly2 = 3 * y + x * x
polyplus(poly1, poly2)
```

RESULTS

```
Result = 3 * Y + A * Y + 2 * X^2 + X * Y
```

4

QUERY

```
poly1 = 5 + 3 + 0 * y - 6*z*a^3
poly2 = u^10 + i + 14
polytimes(poly1, poly2)
```

RESULT

```
112 + 8 * I + -84 * A^3 * Z + -6 * A^3 * I * Z + 8 * U^10 + -6 * A^3 * U^10 *
Z
```

5

QUERY

```
poly1 = x + y
polytimes(poly1, poly1)
```

RESULT

```
X^2 + 2 * X * Y + Y^2
```

6

QUERY

```
as_polynomial(n^2 - l^54 + 2 * x * y + y^2 - x^3 + x^2 + x*y^3 - 2*x^2 - x^3
+ 5*x*y^3 + o + 1, P),
polytimes(P, P, R),
pprint_polynomial(R).
```

RESULT

```
L^2 + 2 * L * O + O^2 + 2 * L * N^2 + -2 * L * X^2 + 4 * L * X * Y + 2 * L *
Y^2 + 2 * N^2 * O + -2 * O * X^2 + 4 * O * X * Y + 2 * O * Y^2 + -4 * L *
X^3 + N^4 + -2 * N^2 * X^2 + 4 * N^2 * X * Y + 2 * N^2 * Y^2 + -4 * O *
X^3 + X^4 + 2 * X * Y^3 + -X^2 * Y^2 + 2 * X * Y^3 + 4 * X^2 * Y^2 + -2 *
X^3 * Y + -X^2 * Y^2 + -2 * X^3 * Y + Y^4 + 12 * L * X * Y^3 + -4 * N^2 *
X^3 + 12 * O * X * Y^3 + 4 * X^5 + -2 * X^3 * Y^2 + -4 * X^4 * Y + -2 *
```

```

X^3 * Y^2 + -4 * X^4 * Y + 12 * N^2 * X * Y^3 + 4 * X^6 + 6 * X * Y^5 +12
* X^2 * Y^4 + -6 * X^3 * Y^3 + 6 * X * Y^5 + 12 * X^2 * Y^4 + -6 * X^3 *
Y^3 + -24 * X^4 * Y^3 + 36 * X^2 * Y^6 + -2 * L^55 + -2 * L^54 * 0 + -2 *
L^54 * N^2 + 2 * L^54 * X^2 + -4 * L^54 * X * Y + -2 * L^54 * Y^2 + 4 *
L^54 * X^3 + -12 * L^54 * X * Y^3 + L^108

```

7

QUERY

```

poly1 = x^3 + x^2 + x*y^3 - 2*x^2 - x^3 + 5*x*y^3 + o + 1
polytimes(poly1, poly1)

```

RESULT

```

L^2 + 2 * L * 0 + 0^2 + -2 * L * X^2 + -2 * 0 * X^2 + X^4 + 12 * L * X * Y^3
+ 12 * 0 * X * Y^3 + -12 * X^3 * Y^3 + 36 * X^2 * Y^6

```

8

QUERY

```

(n+1)^3

```

RESULT

```

1 + 3 * N + 3 * N^2 + N^3

```

9

QUERY

```

as_polynomial(x-4, P),
polytimes(P,P, R1),
polytimes(R1,P, R),
pprint_polynomial(R).

```

RESULT

```

-64 + 48 * X + -12 * X^2 + X^3

```

10, pdf example on sorting

QUERY

```

as_polynomial(y^42*x^4*s*z^2*t^2,P),
pprint_polynomial(P).

```

RESULT

```

S * T^2 * X^4 * Y^42 * Z^2

```