We want to compute A_4^*. In other words, we are looking for the coefficients of a convex polynomial $p(x,y) = \sum_{i=0}^{8} a_i x^i y^{8-i}$ that maximizes $\frac{a_4}{70}$ and satisfies $a_0 = a_8 = 1$.

If $p(x,y)$ is an optimal solution, so is $p(y,x)$. We can therefore take $a_i = a_{8-i}$ for $i = 1, \ldots, 7$.

In [1]:
```
import numpy as np
import sage_sdp
from scipy.sparse import csc_matrix

# helper functions for symmetry reduction
import isotypical_decomposition
```

In [2]:
```
d = 4
a = tuple([polygen(QQ, 'a'+str(i)) for i in range(1, d+1)])
R.<x,y,u,v> = QQ[a]['x,y,u,v']
a = tuple(map(R, a))
a_repeated = (1,) + a[:-1] + a[::-1] + (1,)
show(a)
vars = [x,y,u,v]
q = sum([ a_repeated[i] * x^i * y^(2*d-i) * binomial(2*d, i)
         for i in range(2*d+1)])
obj = q.coefficient(x^d*y^d)
print "We want to maximize the variable `{obj}` and keep the form below convex."\
            .format(obj=obj)
show(q)
show(a)
```

(a_1, a_2, a_3, a_4)

```
We want to maximize the variable `70*a4` and keep the form below convex.
```

$x^8 + (8a_1)x^7y + (28a_2)x^6y^2 + (56a_3)x^5y^3 + (70a_4)x^4y^4 + (56a_3)x^3y^5 + (28a_2)x^2y^6 + (8a_1)xy^7 + y^8$

(a_1, a_2, a_3, a_4)

In [3]:
```
Hq = jacobian(jacobian(q, (x, y)), (x, y))
H_quadratic_form = vector([u, v]) * Hq * vector([u, v])
print("q sos is equivalent to the form below being sos.")
show("H_quadratic_form = ", H_quadratic_form)
```

```
q sos is equivalent to the form below being sos.
```

$$\texttt{H\_quadratic\_form} = 56x^6u^2 + (336a_1)x^5yu^2 + (840a_2)x^4y^2u^2 + (1120a_3)x^3y^3u^2 + (840a_4)x^2y^4u^2 + (336a_3)xy^5u^2 + (56a_2)y^6u^2 + (112a_1)x^6uv + (672a_2)x^5yuv + (1680a_3)x^4y^2uv + (2240a_4)x^3y^3uv + (1680a_3)x^2y^4uv + (672a_2)xy^5uv + (112a_1)y^6uv + (56a_2)x^6v^2 + (336a_3)x^5yv^2 + (840a_4)x^4y^2v^2 + (1120a_3)x^3y^3v^2 + (840a_2)x^2y^4v^2 + (336a_1)xy^5v^2 + 56y^6v^2$$

SDP formulation

`H-quadratic-form` is sos if and only if it can be written as $z^T Qz$, where Q is an $(2d-2) \times (2d-2)$ psd matrix and z is the vector of monomials. (See below.)

In [4]:
```
#  H_quadratic_form sos if H_quadratic_form = z *
mons_xy = vector([x^i * y^(d-i-1) for i in range(d)])
mons = vector(list(u * mons_xy) + list(v * mons_xy))
show("Consider the monomials in (x,y,u,v): ", mons)
show("and the {i} x {i} matrix".format(i=len(mons)))

Q = sage_sdp.symb_matrix(len(mons), "Q")
show(Q[:5, :5], '...')

opt_variables = Q.variables() + a
print("New decision variables:")
show(opt_variables)
```

Consider the monomials in (x,y,u,v): $\left(y^3 u,\ xy^2 u,\ x^2 yu,\ x^3 u,\ y^3 v,\ xy^2 v,\ x^2 yv,\ x^3 v\right)$

and the 8 x 8 matrix

$$\begin{pmatrix} Q_{11} & Q_{21} & Q_{31} & Q_{41} & Q_{51} \\ Q_{21} & Q_{22} & Q_{32} & Q_{42} & Q_{52} \\ Q_{31} & Q_{32} & Q_{33} & Q_{43} & Q_{53} \\ Q_{41} & Q_{42} & Q_{43} & Q_{44} & Q_{54} \\ Q_{51} & Q_{52} & Q_{53} & Q_{54} & Q_{55} \end{pmatrix} \ldots$$

New decision variables:

$$(Q_{11}, Q_{21}, Q_{22}, Q_{31}, Q_{32}, Q_{33}, Q_{41}, Q_{42}, Q_{43}, Q_{44}, Q_{51}, Q_{52}, Q_{53}, Q_{54}, Q_{55}, Q_{61}, Q_{62}, Q_{63}, Q_{64}, Q_{65}, Q_{66}, Q_{71}, Q_{72}, Q_{73}, Q_{74}, Q_{75}, Q_{76}, Q_{77}, Q_{81}, Q_{82}, Q_{83}, Q_{84}, Q_{85}, Q_{86}, Q_{87}, Q_{88}, a_1, a_2, a_3, a_4)$$

The identity $u^T \nabla^2 qu - z^T Qz = 0$ is equivalent to a list of equalities involving the coefficients of Q and the vector a linearly (one for each monomial). We report these linear equalities in the table below

In [5]:
```
residual = H_quadratic_form - mons * Q * mons
RR = QQ[opt_variables][x, y, u, v]
residual = RR(residual)
linear_eq = residual.coefficients()


table([residual.monomials(), [LatexExpr("{}=0".format(mi)) for mi in linear_eq]],
      header_column=["monomial", "corresponding linear equality"])\
    .transpose()
```

Out[5]:

monomial	corresponding linear equality
x^6u^2	-Q_44 + 56=0
x^5yu^2	-2*Q_43 + 336*a1=0
$x^4y^2u^2$	-Q_33 - 2*Q_42 + 840*a2=0
$x^3y^3u^2$	-2*Q_32 - 2*Q_41 + 1120*a3=0
$x^2y^4u^2$	-Q_22 - 2*Q_31 + 840*a4=0
xy^5u^2	-2*Q_21 + 336*a3=0
y^6u^2	-Q_11 + 56*a2=0
x^6uv	-2*Q_84 + 112*a1=0
x^5yuv	-2*Q_74 - 2*Q_83 + 672*a2=0
x^4y^2uv	-2*Q_64 - 2*Q_73 - 2*Q_82 + 1680*a3=0
x^3y^3uv	-2*Q_54 - 2*Q_63 - 2*Q_72 - 2*Q_81 + 2240*a4=0
x^2y^4uv	-2*Q_53 - 2*Q_62 - 2*Q_71 + 1680*a3=0
xy^5uv	-2*Q_52 - 2*Q_61 + 672*a2=0
y^6uv	-2*Q_51 + 112*a1=0
x^6v^2	-Q_88 + 56*a2=0
x^5yv^2	-2*Q_87 + 336*a3=0
$x^4y^2v^2$	-Q_77 - 2*Q_86 + 840*a4=0
$x^3y^3v^2$	-2*Q_76 - 2*Q_85 + 1120*a3=0
$x^2y^4v^2$	-Q_66 - 2*Q_75 + 840*a2=0
xy^5v^2	-2*Q_65 + 336*a1=0
y^6v^2	-Q_55 + 56=0

Construct a symmetry-adapted basis of monomials

In [6]:
```
# permutation group of symmetries
vars = [x, y, u, v]
symmetries =[[y,x,v,u],]
if d % 2 == 0:
    symmetries += [[-x,y,-u,v], [x,-y,-u,v]]
else:
    symmetries += [[-x,-y,-u,-v]]
symmetries_rho = map(lambda sym_i: matrix(QQ, jacobian(sym_i, vars)), symmetries)
G_sage = MatrixGroup(symmetries_rho)


print("The symmetries of the problem")
show(symmetries)
print "are represented by the group of", len(G_sage), "elements:"
show(G_sage)
```

```
The symmetries of the problem
```

$$[[y, x, v, u], [-x, y, -u, v], [x, -y, -u, v]]$$

```
are represented by the group of 16 elements:
```

$$\left\langle \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right\rangle$$

In [7]:
```
def get_irreducible_repr(G_sage):
    """Utility function to compute irreducible representations of a group using gap."""
    G_gap = gap.Group(map(lambda M: matrix(QQ, M), G_sage.gens()))
    irr_repr = []
    gap_irr_repr = gap.IrreducibleRepresentations(G_gap)
    characters = gap.Irr( G_gap )
    for irr in gap_irr_repr:
        sage_irr = gap.UnderlyingRelation(irr)
        sage_irr = sage_irr.AsList()
        sage_irr = map(lambda u: list(u.AsList()),
                       sage_irr)
        sage_irr = map(lambda elem_img: ((elem_img[0]),
                                         elem_img[1].sage()), sage_irr)
        gens, vals = zip(*sage_irr)
        vals = map(lambda v: matrix(QQ, v), vals)
        gens_vals_dict = {G_sage(g): v for g,v in zip(gens, vals)}
        hom_irr = lambda g, gens_vals_dict=gens_vals_dict: gens_vals_dict[g]
        irr_repr.append(hom_irr)
    return irr_repr

irr_repr = get_irreducible_repr(G_sage)
```

In [8]:
```
# induced action on the vector of monomials z
def action_on_monomials(g):
    mons_permuted = mons.subs({vi: subi
                               for vi, subi in zip(vars, g*vector(vars))})
    # construct permutation matrix
    col = np.array(range(len(mons)))
    basis_idx = {b: i for i,b in enumerate(mons)}
    row = [ basis_idx[m] if m in basis_idx else basis_idx[-m]
           for m in mons_permuted]
    data = [1 if m in basis_idx else -1 for m in mons_permuted]
    return csc_matrix((data, (row, col))).todense()

print("Example of the action of G on the vector of monomials")
g = G_sage.random_element()
show(g, LatexExpr(r"\cdot"), vector(mons),
     " = ", matrix(QQ,action_on_monomials(g)) * mons)
```

Example of the action of G on the vector of monomials

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \cdot \left(y^3 u, xy^2 u, x^2 yu, x^3 u, y^3 v, xy^2 v, x^2 yv, x^3 v\right) = \left(y^3 u, -xy^2 u, x^2 yu, -x^3 u, -y^3 v, xy^2 v, -x^2 yv, x^3 v\right)$$

In [9]:
```
# compute a basis for the isotypical compononents of span(mons)
basis_iso = [isotypical_decomposition.\
             compute_basis_isotypical_comp(chi, action_on_monomials, G_sage)
             for chi in irr_repr]

# compute a symmetry adapted basis
adapted_basis = []
for j, Vj in enumerate(basis_iso):
    if Vj is not None:
        adapted_basis.append([ Vij * mons for Vij in Vj])

print("Symmetry adapted basis:")
show(adapted_basis)
```

```
Symmetry adapted basis:
```

$$\left[\left[\left(xy^2u + x^2yv,\ x^3u + y^3v\right)\right], \left[\left(y^3u + x^3v,\ x^2yu + xy^2v\right)\right], \left[\left(xy^2u - x^2yv,\ x^3u - y^3v\right)\right], \left[\left(y^3u - x^3v,\ x^2yu - xy^2v\right)\right]\right]$$

In [10]:
```
print("Let's build the semidefinite program")
size_Qs = map(lambda u: len(u[0]), adapted_basis)
Qs = [sage_sdp.symb_matrix(len(bi[0]), "Q"+str(i+1)) for i, bi in enumerate(adapted_basis)]
print("The matrix Q in the symmetry adapted basis has the following block diagonalization (with multiplicity):")
show([(len(bi), LatexExpr(r'\times'), Qi) for bi, Qi in zip(adapted_basis, Qs)])


Qs_vars = tuple(sum([list(Qi.variables()) for Qi in Qs], []))
decision_vars = Qs_vars + a
RR = QQ[decision_vars][x,y,u,v]
reduced_r = RR(H_quadratic_form  - \
       sum( sum( bij*Qi*bij for bij in bi ) for bi, Qi in zip(adapted_basis, Qs) ))
```

```
Let's build the semidefinite program
The matrix Q in the symmetry adapted basis has the following block diagonalization (with multiplicity):
```

$$\left[\left(1, \times, \begin{pmatrix} Q1_{11} & Q1_{21} \\ Q1_{21} & Q1_{22} \end{pmatrix}\right), \left(1, \times, \begin{pmatrix} Q2_{11} & Q2_{21} \\ Q2_{21} & Q2_{22} \end{pmatrix}\right), \left(1, \times, \begin{pmatrix} Q3_{11} & Q3_{21} \\ Q3_{21} & Q3_{22} \end{pmatrix}\right), \left(1, \times, \begin{pmatrix} Q4_{11} & Q4_{21} \\ Q4_{21} & Q4_{22} \end{pmatrix}\right)\right]$$

The identity $u^T \nabla^2 qu - z^T Qz = 0$ is equivalent to a list of equalities involving the coefficients of Q linearly (one for each monomial m)

$$\langle A_m, Q \rangle = 0.$$

We report these linear equalities in the table below

In [11]:
```
linear_eq = reduced_r.coefficients()
table([reduced_r.monomials(), [LatexExpr("{}=0".format(mi)) for mi in linear_eq]],
      header_column=["monomial", "corresponding linear equality"])\
    .transpose()
```

Out[11]:

monomial	corresponding linear equality
x^6u^2	-Q1_22 - Q3_22 + 56=0
x^5yu^2	336*a1=0
$x^4y^2u^2$	-2*Q1_21 - Q2_22 - 2*Q3_21 - Q4_22 + 840*a2=0
$x^3y^3u^2$	1120*a3=0
$x^2y^4u^2$	-Q1_11 - 2*Q2_21 - Q3_11 - 2*Q4_21 + 840*a4=0
xy^5u^2	336*a3=0
y^6u^2	-Q2_11 - Q4_11 + 56*a2=0
x^6uv	112*a1=0
x^5yuv	-2*Q1_21 - 2*Q2_21 + 2*Q3_21 + 2*Q4_21 + 672*a2=0
x^4y^2uv	1680*a3=0
x^3y^3uv	-2*Q1_11 - 2*Q1_22 - 2*Q2_11 - 2*Q2_22 + 2*Q3_11 + 2*Q3_22 + 2*Q4_11 + 2*Q4_22 + 2240*a4=0
x^2y^4uv	1680*a3=0
xy^5uv	-2*Q1_21 - 2*Q2_21 + 2*Q3_21 + 2*Q4_21 + 672*a2=0
y^6uv	112*a1=0
x^6v^2	-Q2_11 - Q4_11 + 56*a2=0
x^5yv^2	336*a3=0
$x^4y^2v^2$	-Q1_11 - 2*Q2_21 - Q3_11 - 2*Q4_21 + 840*a4=0
$x^3y^3v^2$	1120*a3=0
$x^2y^4v^2$	-2*Q1_21 - Q2_22 - 2*Q3_21 - Q4_22 + 840*a2=0
xy^5v^2	336*a1=0
y^6v^2	-Q1_22 - Q3_22 + 56=0

Full dimensional description of the set

$\mathcal{L} := \{v := (a, Q) \quad | \quad Q$ is psd and has the block diagonal structure described above and a and Q satisfy the linear equalities above$\}$
$\mathcal{L} := \{v := (a, Q) \quad | \quad Q$ is psd and has the block diagonal structure described above and $Av = b\}$ $\mathcal{L} := \{v := (a, Q) \quad | \quad v = v_0 + N\alpha, \alpha \in \mathbb{R}^k\}$, where $ker(A) =: \{N\alpha | \alpha \in \mathbb{R}^k\}$, and $k = dim(Ker(A))$

Let us construct the matrix A, the vector b, the vector v_0 , and the matrix N.

In [12]:
```
def get_full_dim_description(linear_eq, vars, name='alpha'):
    A = matrix(QQ, jacobian(vector(SR, linear_eq), vector(vars)))
    b = vector(linear_eq) - A * vector(vars)
    v0 = A.solve_right(b)
    ker_A = matrix(SR, A.transpose().kernel().basis())
    alpha = tuple([var(name+str(i)) for i in range(1, ker_A.dimensions()[0]+1)])

    full_dim_vars = vector(v0) + ker_A.transpose() * vector(alpha)
    assert A * full_dim_vars - b == 0
    return full_dim_vars

full_dim_vars = get_full_dim_description(linear_eq, decision_vars, 'alpha')
```

In [13]:
```
original_to_full_dim = dict(zip(list(map(SR, decision_vars)), list(full_dim_vars)))
table([original_to_full_dim.keys(), original_to_full_dim.values()],
      header_column=["original variables", "full dimensional description"]).transpose()
```

Out[13]:

original variables	full dimensional description
$Q3_{21}$	$\frac{1}{14}\,\alpha_1 + \frac{1}{7}\,\alpha_2 - \frac{3}{7}\,\alpha_3 - \frac{3}{7}\,\alpha_4 + \frac{8}{7}\,\alpha_5 - \frac{3}{7}\,\alpha_6 + \frac{1}{2}\,\alpha_7$
$Q2_{21}$	α_5
$Q1_{21}$	α_2
a_1	0
$Q4_{11}$	α_8
$Q3_{11}$	$\alpha_7 - 28$
$Q2_{11}$	α_4
$Q1_{11}$	$\alpha_1 + 28$
a_4	$\frac{1}{980}\,\alpha_1 + \frac{1}{490}\,\alpha_2 + \frac{1}{980}\,\alpha_3 - \frac{13}{980}\,\alpha_4 + \frac{1}{490}\,\alpha_5 + \frac{1}{980}\,\alpha_6 - \frac{1}{70}\,\alpha_8$
$Q4_{22}$	$-\frac{1}{7}\,\alpha_1 - \frac{16}{7}\,\alpha_2 + \frac{6}{7}\,\alpha_3 + \frac{111}{7}\,\alpha_4 - \frac{16}{7}\,\alpha_5 - \frac{1}{7}\,\alpha_6 - \alpha_7 + 15\,\alpha_8$
$Q3_{22}$	$-\alpha_3$
$Q2_{22}$	α_6
$Q1_{22}$	$\alpha_3 - 56$
a_3	0
a_2	$\frac{1}{56}\,\alpha_4 + \frac{1}{56}\,\alpha_8$
$Q4_{21}$	$-\frac{1}{14}\,\alpha_1 + \frac{6}{7}\,\alpha_2 + \frac{3}{7}\,\alpha_3 - \frac{39}{7}\,\alpha_4 - \frac{1}{7}\,\alpha_5 + \frac{3}{7}\,\alpha_6 - \frac{1}{2}\,\alpha_7 - 6\,\alpha_8$

```
In [14]: def sub_list_matrices(list_matrices, subs):
             """Substitue according to the dict `subs`."""
             sub_map = lambda Mij: Mij.subs(subs)
             return list(map(lambda M: M.apply_map(sub_map), list_matrices))

         Q_reduced = block_diagonal_matrix(sub_list_matrices(Qs, original_to_full_dim))
         objective_reduced = original_to_full_dim[SR(a[-1])]
         show("max ", objective_reduced)
         show("s.t. the following matrix is psd")
         show(Q_reduced)
```

$$\texttt{max}\,\frac{1}{980}\,\alpha_1 + \frac{1}{490}\,\alpha_2 + \frac{1}{980}\,\alpha_3 - \frac{13}{980}\,\alpha_4 + \frac{1}{490}\,\alpha_5 + \frac{1}{980}\,\alpha_6 - \frac{1}{70}\,\alpha_8$$

s.t. the following matrix is psd

$$\left(\begin{array}{rr|rr|rr|r}
\alpha_1 + 28 & \alpha_2 & 0 & 0 & 0 & 0 & \\
\alpha_2 & \alpha_3 - 56 & 0 & 0 & 0 & 0 & \\
\hline
0 & 0 & \alpha_4 & \alpha_5 & 0 & 0 & \\
0 & 0 & \alpha_5 & \alpha_6 & 0 & 0 & \\
\hline
0 & 0 & 0 & 0 & \alpha_7 - 28 & \frac{1}{14}\,\alpha_1 + \frac{1}{7}\,\alpha_2 - \frac{3}{7}\,\alpha_3 - \frac{3}{7}\,\alpha_4 + \frac{8}{7}\,\alpha_5 - \frac{3}{7}\,\alpha_6 + \frac{1}{2}\,\alpha_7 & \\
0 & 0 & 0 & 0 & \frac{1}{14}\,\alpha_1 + \frac{1}{7}\,\alpha_2 - \frac{3}{7}\,\alpha_3 - \frac{3}{7}\,\alpha_4 + \frac{8}{7}\,\alpha_5 - \frac{3}{7}\,\alpha_6 + \frac{1}{2}\,\alpha_7 & -\alpha_3 & \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{14}\,\alpha_1 + \frac{6}{7}\,\alpha_2 + \frac{3}{7}\,\alpha
\end{array}\right.$$

Dual problem

```
In [15]: Ds = [sage_sdp.symb_matrix(2, "D{}".format(i)) for i in range(4)]
         D = block_diagonal_matrix(Ds)
         show(D)
```

$$\left(\begin{array}{rr|rr|rr|rr}
D0_{11} & D0_{21} & 0 & 0 & 0 & 0 & 0 & 0 \\
D0_{21} & D0_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & D1_{11} & D1_{21} & 0 & 0 & 0 & 0 \\
0 & 0 & D1_{21} & D1_{22} & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & D2_{11} & D2_{21} & 0 & 0 \\
0 & 0 & 0 & 0 & D2_{21} & D2_{22} & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & D3_{11} & D3_{21} \\
0 & 0 & 0 & 0 & 0 & 0 & D3_{21} & D3_{22}
\end{array}\right)$$

In [16]:
```
hessian = lambda pp: jacobian(jacobian(pp, (x,y)), (x,y))
dot = lambda u,v: vector(u) * vector(v)
mdot = lambda A, B: dot(A.list(), B.list())


lagrangian = mdot(D, Q_reduced) + objective_reduced
var_primal = Q_reduced.variables()
var_dual = D.variables()
lagrangian = QQ[var_dual][var_primal] (lagrangian)
show(LatexExpr(r"\mathcal L(\alpha, D) = "), lagrangian)
```

$$\begin{aligned}\mathcal{L}(\alpha, D) = &\left(D0_{11} + \frac{1}{7}D2_{21} - \frac{1}{7}D3_{21} - \frac{1}{7}D3_{22} + \frac{1}{980}\right)\alpha_1 + \left(2D0_{21} + \frac{2}{7}D2_{21} + \frac{12}{7}D3_{21} - \frac{16}{7}D3_{22} + \frac{1}{490}\right)\alpha_2 \\ &+ \left(D0_{22} - \frac{6}{7}D2_{21} - D2_{22} + \frac{6}{7}D3_{21} + \frac{6}{7}D3_{22} + \frac{1}{980}\right)\alpha_3 + \left(D1_{11} - \frac{6}{7}D2_{21} - \frac{78}{7}D3_{21} + \frac{111}{7}D3_{22} - \frac{13}{980}\right)\alpha_4 \\ &+ \left(2D1_{21} + \frac{16}{7}D2_{21} - \frac{2}{7}D3_{21} - \frac{16}{7}D3_{22} + \frac{1}{490}\right)\alpha_5 + \left(D1_{22} - \frac{6}{7}D2_{21} + \frac{6}{7}D3_{21} - \frac{1}{7}D3_{22} + \frac{1}{980}\right)\alpha_6 + \left(D2_{11} + D2_{21} - D3_{21} - D3_{22}\right)\alpha_7 \\ &+ \left(D3_{11} - 12D3_{21} + 15D3_{22} - \frac{1}{70}\right)\alpha_8 + 28D0_{11} - 56D0_{22} - 28D2_{11}\end{aligned}$$

In [17]:
```
# Find a full rank description of the dual varialbe DQ
lagrang_coeffs = jacobian(lagrangian, var_primal)[0]
lagrang_coeffs = list(lagrang_coeffs)
full_dim_vars_dual = get_full_dim_description(lagrang_coeffs, var_dual, 'beta')
full_dim_vars_dual
```

Out[17]:
```
(beta1 + 1/980, beta2 + 1/980, beta3 + 1/980, beta4 - 13/980, 5*beta1 + 5*beta2 + 1/3*beta4 + 1/980, -4*beta1 - 4*beta2 - 1/3*beta4
+ 1/980, 7*beta1, -2*beta1 - 9*beta2 - 2/3*beta4, 6*beta1 + beta3, 6*beta1 + beta4 - 1/70, 3*beta1 - 5*beta2 - 1/3*beta4, 2*beta1 -
4*beta2 - 1/3*beta4)
```

In [18]:
```
original_to_full_dim_var = dict(zip(list(map(SR, var_dual)), list(full_dim_vars_dual)))
table([original_to_full_dim_var.keys(), original_to_full_dim_var.values()],
      header_column=["original dual variables", "full dimensional description"]).transpose()
```

Out[18]:

original dual variables	full dimensional description
$D3_{22}$	$2\,\beta_1 - 4\,\beta_2 - \frac{1}{3}\,\beta_4$
$D2_{22}$	$6\,\beta_1 + \beta_3$
$D1_{22}$	$-4\,\beta_1 - 4\,\beta_2 - \frac{1}{3}\,\beta_4 + \frac{1}{980}$
$D0_{22}$	$\beta_3 + \frac{1}{980}$
$D3_{21}$	$3\,\beta_1 - 5\,\beta_2 - \frac{1}{3}\,\beta_4$
$D2_{21}$	$-2\,\beta_1 - 9\,\beta_2 - \frac{2}{3}\,\beta_4$
$D1_{21}$	$5\,\beta_1 + 5\,\beta_2 + \frac{1}{3}\,\beta_4 + \frac{1}{980}$
$D0_{21}$	$\beta_2 + \frac{1}{980}$
$D3_{11}$	$6\,\beta_1 + \beta_4 - \frac{1}{70}$
$D2_{11}$	$7\,\beta_1$
$D1_{11}$	$\beta_4 - \frac{13}{980}$
$D0_{11}$	$\beta_1 + \frac{1}{980}$

In [19]:
```
D_reduced = block_diagonal_matrix(sub_list_matrices(Ds, original_to_full_dim_var))
show(D_reduced)
```

$$\left(\begin{array}{cc|cc|cc|cc}
\beta_1 + \frac{1}{980} & \beta_2 + \frac{1}{980} & 0 & 0 & 0 & 0 & 0 \\
\beta_2 + \frac{1}{980} & \beta_3 + \frac{1}{980} & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & \beta_4 - \frac{13}{980} & 5\,\beta_1 + 5\,\beta_2 + \frac{1}{3}\,\beta_4 + \frac{1}{980} & 0 & 0 & 0 \\
0 & 0 & 5\,\beta_1 + 5\,\beta_2 + \frac{1}{3}\,\beta_4 + \frac{1}{980} & -4\,\beta_1 - 4\,\beta_2 - \frac{1}{3}\,\beta_4 + \frac{1}{980} & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 7\,\beta_1 & -2\,\beta_1 - 9\,\beta_2 - \frac{2}{3}\,\beta_4 & 0 \\
0 & 0 & 0 & 0 & -2\,\beta_1 - 9\,\beta_2 - \frac{2}{3}\,\beta_4 & 6\,\beta_1 + \beta_3 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 6\,\beta_1 + \beta_4 - \frac{1}{70} & 3\,\beta_1 - 5\,\beta_2 - \cdot \\
0 & 0 & 0 & 0 & 0 & 0 & 3\,\beta_1 - 5\,\beta_2 - \frac{1}{3}\,\beta_4 & 2\,\beta_1 - 4\,\beta_2 - \cdot
\end{array}\right)$$

use KKT conditions to transform the SDP to a system of polynomial equations¶

In [20]:
```
# w = objective_reduced
w = var('omega')
R = QQ[Q_reduced.variables() + D_reduced.variables() + (w,)]

# KKT equations
KKT_eqn = (D_reduced * Q_reduced).list() + [w - objective_reduced]
KKT_eqn = list(set(KKT_eqn))
print("KKT equations = ")

table([list(map(R, KKT_eqn)), ["= 0" for _ in KKT_eqn]]).transpose()
```

KKT equations =

Out[20]:

$$0 = 0$$

$$\alpha_1\beta_1 + \alpha_2\beta_2 + \tfrac{1}{980}\alpha_1 + \tfrac{1}{980}\alpha_2 + 28\beta_1 + \tfrac{1}{35} = 0$$

$$-\tfrac{3}{14}\alpha_1\beta_1 + \tfrac{18}{7}\alpha_2\beta_1 + \tfrac{9}{7}\alpha_3\beta_1 - \tfrac{117}{7}\alpha_4\beta_1 - \tfrac{3}{7}\alpha_5\beta_1 + \tfrac{9}{7}\alpha_6\beta_1 - \tfrac{3}{2}\alpha_7\beta_1 - 12\alpha_8\beta_1 + \tfrac{5}{14}\alpha_1\beta_2 - \tfrac{30}{7}\alpha_2\beta_2 - \tfrac{15}{7}\alpha_3\beta_2 + \tfrac{195}{7}\alpha_4\beta_2 + \tfrac{5}{7}\alpha_5\beta_2 - \tfrac{15}{7}\alpha_6\beta_2 + \tfrac{5}{2}\alpha_7\beta_2 + 30\alpha_8\beta_2 + \tfrac{1}{42}\alpha_1\beta_4 - \tfrac{2}{7}\alpha_2\beta_4 - \tfrac{1}{7}\alpha_3\beta_4 + \tfrac{13}{7}\alpha_4\beta_4 + \tfrac{1}{21}\alpha_5\beta_4 - \tfrac{1}{7}\alpha_6\beta_4 + \tfrac{1}{6}\alpha_7\beta_4 + 3\alpha_8\beta_4 - \tfrac{1}{70}\alpha_8 = 0$$

$$\alpha_1\beta_2 + \alpha_2\beta_3 + \tfrac{1}{980}\alpha_1 + \tfrac{1}{980}\alpha_2 + 28\beta_2 + \tfrac{1}{35} = 0$$

$$5\alpha_5\beta_1 + 5\alpha_5\beta_2 + \alpha_4\beta_4 + \tfrac{1}{3}\alpha_5\beta_4 - \tfrac{13}{980}\alpha_4 + \tfrac{1}{980}\alpha_5 = 0$$

$$5\alpha_6\beta_1 + 5\alpha_6\beta_2 + \alpha_5\beta_4 + \tfrac{1}{3}\alpha_6\beta_4 - \tfrac{13}{980}\alpha_5 + \tfrac{1}{980}\alpha_6 = 0$$

$$-\tfrac{1}{2}\alpha_1\beta_1 - 2\alpha_2\beta_1 + 3\alpha_3\beta_1 + 15\alpha_4\beta_1 - 5\alpha_5\beta_1 + \alpha_6\beta_1 - \tfrac{7}{2}\alpha_7\beta_1 + 12\alpha_8\beta_1 + \tfrac{13}{14}\alpha_1\beta_2 + \tfrac{34}{7}\alpha_2\beta_2 - \tfrac{39}{7}\alpha_3\beta_2 - \tfrac{249}{7}\alpha_4\beta_2 + \tfrac{69}{7}\alpha_5\beta_2 - \tfrac{11}{7}\alpha_6\beta_2 + \tfrac{13}{2}\alpha_7\beta_2 - 30\alpha_8\beta_2 + \tfrac{1}{14}\alpha_1\beta_4 + \tfrac{10}{21}\alpha_2\beta_4 - \tfrac{3}{7}\alpha_3\beta_4 - \tfrac{24}{7}\alpha_4\beta_4 + \tfrac{17}{21}\alpha_5\beta_4 - \tfrac{2}{21}\alpha_6\beta_4 + \tfrac{1}{2}\alpha_7\beta_4 - 3\alpha_8\beta_4 = 0$$

$$-\tfrac{1}{7}\alpha_1\beta_1 + \tfrac{12}{7}\alpha_2\beta_1 + \tfrac{6}{7}\alpha_3\beta_1 - \tfrac{78}{7}\alpha_4\beta_1 - \tfrac{2}{7}\alpha_5\beta_1 + \tfrac{6}{7}\alpha_6\beta_1 - \alpha_7\beta_1 - 9\alpha_8\beta_1 + \tfrac{2}{7}\alpha_1\beta_2 - \tfrac{24}{7}\alpha_2\beta_2 - \tfrac{12}{7}\alpha_3\beta_2 + \tfrac{156}{7}\alpha_4\beta_2 + \tfrac{4}{7}\alpha_5\beta_2 - \tfrac{12}{7}\alpha_6\beta_2 + 2\alpha_7\beta_2 + 19\alpha_8\beta_2 + \tfrac{1}{42}\alpha_1\beta_4 - \tfrac{2}{7}\alpha_2\beta_4 - \tfrac{1}{7}\alpha_3\beta_4 + \tfrac{13}{7}\alpha_4\beta_4 + \tfrac{1}{21}\alpha_5\beta_4 - \tfrac{1}{7}\alpha_6\beta_4 + \tfrac{1}{6}\alpha_7\beta_4 + \tfrac{5}{3}\alpha_8\beta_4 = 0$$

$$-\tfrac{1}{980}\alpha_1 - \tfrac{1}{490}\alpha_2 - \tfrac{1}{980}\alpha_3 + \tfrac{13}{980}\alpha_4 - \tfrac{1}{490}\alpha_5 - \tfrac{1}{980}\alpha_6 + \tfrac{1}{70}\alpha_8 + \omega = 0$$

$$5\alpha_5\beta_1 - 4\alpha_6\beta_1 + 5\alpha_5\beta_2 - 4\alpha_6\beta_2 + \tfrac{1}{3}\alpha_5\beta_4 - \tfrac{1}{3}\alpha_6\beta_4 + \tfrac{1}{980}\alpha_5 + \tfrac{1}{980}\alpha_6 = 0$$

$$-\tfrac{1}{7}\alpha_1\beta_1 - \tfrac{2}{7}\alpha_2\beta_1 + \tfrac{6}{7}\alpha_3\beta_1 + \tfrac{6}{7}\alpha_4\beta_1 - \tfrac{16}{7}\alpha_5\beta_1 + \tfrac{6}{7}\alpha_6\beta_1 + 6\alpha_7\beta_1 - \tfrac{9}{14}\alpha_1\beta_2 - \tfrac{9}{7}\alpha_2\beta_2 + \tfrac{27}{7}\alpha_3\beta_2 + \tfrac{27}{7}\alpha_4\beta_2 - \tfrac{72}{7}\alpha_5\beta_2 + \tfrac{27}{7}\alpha_6\beta_2 - \tfrac{9}{2}\alpha_7\beta_2 - \tfrac{1}{21}\alpha_1\beta_4 - \tfrac{2}{21}\alpha_2\beta_4 + \tfrac{2}{7}\alpha_3\beta_4 + \tfrac{2}{7}\alpha_4\beta_4 - \tfrac{16}{21}\alpha_5\beta_4 + \tfrac{2}{7}\alpha_6\beta_4 - \tfrac{1}{3}\alpha_7\beta_4 - 196\beta_1 = 0$$

$$-\tfrac{6}{7}\alpha_1\beta_1 - \tfrac{12}{7}\alpha_2\beta_1 + \tfrac{36}{7}\alpha_3\beta_1 + \tfrac{99}{7}\alpha_4\beta_1 - \tfrac{54}{7}\alpha_5\beta_1 + \tfrac{15}{7}\alpha_6\beta_1 - 6\alpha_7\beta_1 + 9\alpha_8\beta_1 + \tfrac{5}{7}\alpha_1\beta_2 + \tfrac{80}{7}\alpha_2\beta_2 - \tfrac{30}{7}\alpha_3\beta_2 - \tfrac{555}{7}\alpha_4\beta_2 + \tfrac{80}{7}\alpha_5\beta_2 + \tfrac{5}{7}\alpha_6\beta_2 + 5\alpha_7\beta_2 - 75\alpha_8\beta_2 - \tfrac{1}{42}\alpha_1\beta_4 + \tfrac{34}{21}\alpha_2\beta_4 + \tfrac{1}{7}\alpha_3\beta_4 - \tfrac{76}{7}\alpha_4\beta_4 + \tfrac{13}{21}\alpha_5\beta_4 + \tfrac{10}{21}\alpha_6\beta_4 - \tfrac{1}{6}\alpha_7\beta_4 - 11\alpha_8\beta_4 + \tfrac{1}{980}\alpha_1 - \tfrac{3}{245}\alpha_2 - \tfrac{3}{490}\alpha_3 + \tfrac{39}{490}\alpha_4 + \tfrac{1}{490}\alpha_5 - \tfrac{3}{490}\alpha_6 + \tfrac{1}{140}\alpha_7 + \tfrac{3}{35}\alpha_8 = 0$$

$$\alpha_2\beta_2 + \alpha_3\beta_3 + \tfrac{1}{980}\alpha_2 + \tfrac{1}{980}\alpha_3 - 56\beta_3 - \tfrac{2}{35} = 0$$

$$\alpha_2\beta_1 + \alpha_3\beta_2 + \tfrac{1}{980}\alpha_2 + \tfrac{1}{980}\alpha_3 - 56\beta_2 - \tfrac{2}{35} = 0$$

$$\tfrac{3}{7}\alpha_1\beta_1 + \tfrac{6}{7}\alpha_2\beta_1 - \tfrac{18}{7}\alpha_3\beta_1 - \tfrac{18}{7}\alpha_4\beta_1 + \tfrac{48}{7}\alpha_5\beta_1 - \tfrac{18}{7}\alpha_6\beta_1 + \alpha_7\beta_1 - 9\alpha_7\beta_2 + \tfrac{1}{14}\alpha_1\beta_3 + \tfrac{1}{7}\alpha_2\beta_3 - \tfrac{3}{7}\alpha_3\beta_3 - \tfrac{3}{7}\alpha_4\beta_3 + \tfrac{8}{7}\alpha_5\beta_3 - \tfrac{3}{7}\alpha_6\beta_3 + \tfrac{1}{2}\alpha_7\beta_3 - \tfrac{2}{3}\alpha_7\beta_4 + 56\beta_1 + 252\beta_2 + \tfrac{56}{3}\beta_4 = 0$$

$$5\alpha_4\beta_1 - 4\alpha_5\beta_1 + 5\alpha_4\beta_2 - 4\alpha_5\beta_2 + \tfrac{1}{3}\alpha_4\beta_4 - \tfrac{1}{3}\alpha_5\beta_4 + \tfrac{1}{980}\alpha_4 + \tfrac{1}{980}\alpha_5 = 0$$

$$\tfrac{1}{2}\alpha_1\beta_1 + \alpha_2\beta_1 - \alpha_3\beta_1 - 3\alpha_4\beta_1 + 8\alpha_5\beta_1 - 3\alpha_6\beta_1 + \tfrac{7}{2}\alpha_7\beta_1 + 9\alpha_3\beta_2 + \tfrac{2}{3}\alpha_3\beta_4 = 0$$

$$-\tfrac{1}{7}\alpha_1\beta_1 - \tfrac{2}{7}\alpha_2\beta_1 - \tfrac{36}{7}\alpha_3\beta_1 + \tfrac{6}{7}\alpha_4\beta_1 - \tfrac{16}{7}\alpha_5\beta_1 + \tfrac{6}{7}\alpha_6\beta_1 - \alpha_7\beta_1 - \tfrac{9}{14}\alpha_1\beta_2 - \tfrac{9}{7}\alpha_2\beta_2 + \tfrac{27}{7}\alpha_3\beta_2 + \tfrac{27}{7}\alpha_4\beta_2 - \tfrac{72}{7}\alpha_5\beta_2 + \tfrac{27}{7}\alpha_6\beta_2 - \tfrac{9}{2}\alpha_7\beta_2 - \alpha_3\beta_3 - \tfrac{1}{21}\alpha_1\beta_4 - \tfrac{2}{21}\alpha_2\beta_4 + \tfrac{2}{7}\alpha_3\beta_4 + \tfrac{2}{7}\alpha_4\beta_4 - \tfrac{16}{21}\alpha_5\beta_4 + \tfrac{2}{7}\alpha_6\beta_4 - \tfrac{1}{3}\alpha_7\beta_4 = 0$$

Ideal generated by the KKT equations

In [21]:
```
I = map(lambda p: R(p), KKT_eqn)
I = R*I
print("Ideal I is generated by %d equations in %d variables" % (len(I.gens()), len(R.gens())))
```

```
Ideal I is generated by 18 equations in 13 variables
```

Solve the KKT equations by eliminating all variables except ω

In [22]:
```
I_a = I.elimination_ideal([v for v in R.gens() if SR(v) != w])
show(I_a)
```

$$\begin{gathered}\left(886675935058593750\omega^{14} + 291336378662109375\omega^{13} + 34358793013671875 0\omega^{12} + 160132697292968750\omega^{11} + 9965283909765625\omega^{10}\right.\\ - 1054715538023437 5\omega^{9} - 1139425098437500\omega^{8} + 167729012062500\omega^{7} + 7424588040625\omega^{6} - 843407885375\omega^{5} - 242843650\omega^{4}\\ \left. + 615975630\omega^{3} - 4134137\omega^{2} - 27417\omega\right) \mathbf{Q}[\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \beta_1, \beta_2, \beta_3, \beta_4, \omega]\end{gathered}$$

In [23]:
```
print("The ideal I_a has %d generator" % len(I_a.gens()))
```

```
The ideal I_a has 1 generator
```

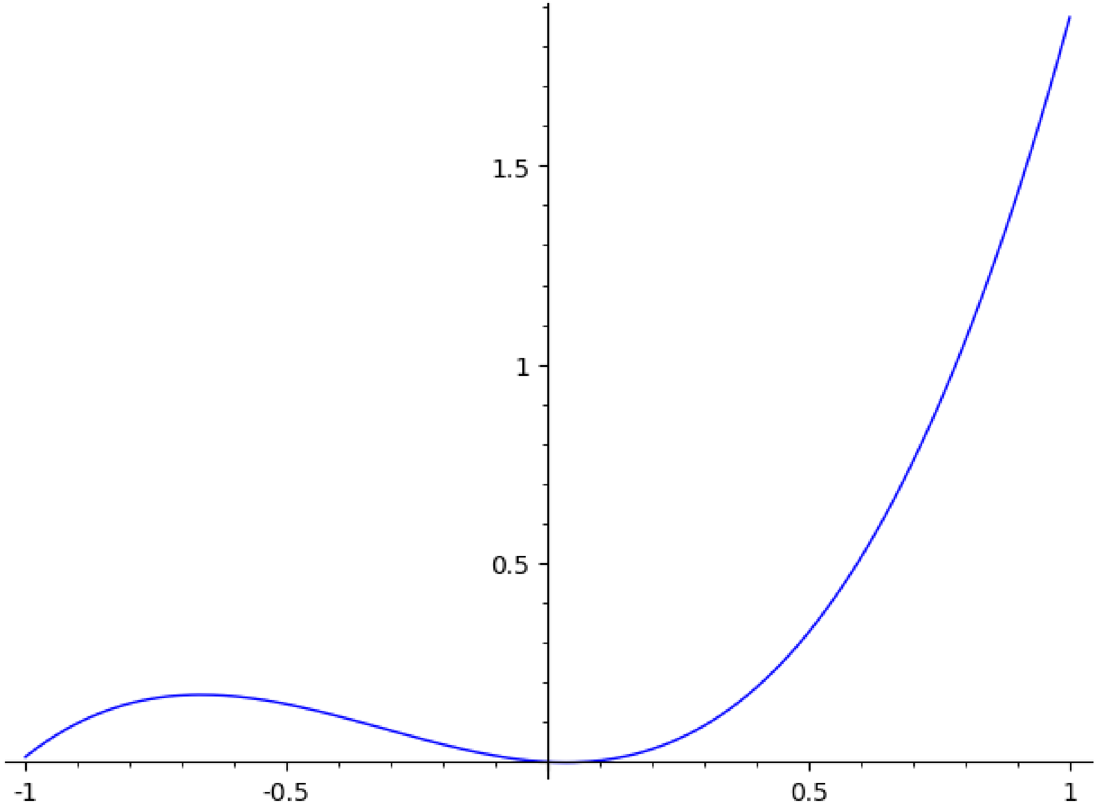
In [24]:
```
p = SR(I_a.gens()[0])
show(factor(p))
```

$$\left(42875\,\omega^{3} + 40425\,\omega^{2} - 2975\,\omega - 13\right)\left(3675\,\omega^{2} + 2870\,\omega - 37\right)(35\,\omega + 19)(35\,\omega + 3)(35\,\omega + 1)(35\,\omega - 1)(15\,\omega - 1)(5\,\omega + 1)(5\,\omega - 1)(\omega + 1)\omega$$

In [25]:
```
minimal_poly = p.factor_list()[0][0]
minimal_poly /= minimal_poly.coefficient(w^3)
show(minimal_poly)
plot(minimal_poly)
```

$$\omega^3 + \frac{33}{35}\,\omega^2 - \frac{17}{245}\,\omega - \frac{13}{42875}$$

Out[25]:



In [26]:
```
show(minimal_poly.roots())
```

$$\left[\left(-\frac{1}{2}\left(i\sqrt{3}+1\right)\left(\frac{256}{55125}i\sqrt{3}-\frac{256}{6125}\right)^{\frac{1}{3}}-\frac{32\left(-i\sqrt{3}+1\right)}{525\left(\frac{256}{55125}i\sqrt{3}-\frac{256}{6125}\right)^{\frac{1}{3}}}-\frac{11}{35},1\right),\right.$$
$$\left(-\frac{1}{2}\left(\frac{256}{55125}i\sqrt{3}-\frac{256}{6125}\right)^{\frac{1}{3}}\left(-i\sqrt{3}+1\right)-\frac{32\left(i\sqrt{3}+1\right)}{525\left(\frac{256}{55125}i\sqrt{3}-\frac{256}{6125}\right)^{\frac{1}{3}}}-\frac{11}{35},1\right),$$
$$\left.\left(\left(\frac{256}{55125}i\sqrt{3}-\frac{256}{6125}\right)^{\frac{1}{3}}+\frac{64}{525\left(\frac{256}{55125}i\sqrt{3}-\frac{256}{6125}\right)^{\frac{1}{3}}}-\frac{11}{35},1\right)\right]$$

In [27]:
```
minimal_poly.roots(ring=RDF)
```

Out[27]:
```
[(-1.0111811061765266, 1),
 (-0.0041380873389235545, 1),
 (0.07246205065830763, 1)]
```