

JacktoM_demo

February 1, 2019

0.0.1 This is the package to compute transition matrices from various Jack polynomials to monomials

```
In [1]: load('JacktoM.sage')
```

0.0.2 This is based on the built-in symmetric functions in SageMath.

```
In [2]: def CJacktoM1(k,alpha):
    m = SymmetricFunctions(QQ).m();
    Jack = SymmetricFunctions(QQ).jack(t=alpha);
    Q = Jack.Q();
    D = Q.transition_matrix(m,k);
    Plist = Partitions(k).list();
    c0 = alpha^k*factorial(k)
    for i in range(len(Plist)):
        c = c0/Plist[i].hook_product(alpha);
        D[i,i:] = c*D[i,i:];
    return D
```

```
In [3]: def JJacktoM1(k,alpha):
    m = SymmetricFunctions(QQ).m();
    Jack = SymmetricFunctions(QQ).jack(t=alpha);
    J = Jack.J();
    D = J.transition_matrix(m,k);
    return D
```

```
In [4]: def PJacktoM1(k,alpha):
    m = SymmetricFunctions(QQ).m();
    Jack = SymmetricFunctions(QQ).jack(t=alpha);
    P = Jack.P();
    D = P.transition_matrix(m,k);
    return D
```

```
In [5]: def QJacktoM1(k,alpha):
    m = SymmetricFunctions(QQ).m();
    Jack = SymmetricFunctions(QQ).jack(t=alpha);
    Q = Jack.Q();
    D = Q.transition_matrix(m,k);
    return D
```

In [2]: JacktoM(5,3, 'J')

```
Out[2]: [3640 1400 1120 560 480 240 120]
[ 0 308 132 248 168 162 120]
[ 0 0 160 80 140 120 120]
[ 0 0 0 72 36 96 120]
[ 0 0 0 0 60 72 120]
[ 0 0 0 0 0 42 120]
[ 0 0 0 0 0 0 120]
```

In [7]: JJacktoM1(5,3)

```
Out[7]: [3640 1400 1120 560 480 240 120]
[ 0 308 132 248 168 162 120]
[ 0 0 160 80 140 120 120]
[ 0 0 0 72 36 96 120]
[ 0 0 0 0 60 72 120]
[ 0 0 0 0 0 42 120]
[ 0 0 0 0 0 0 120]
```

In [7]: JacktoM(5,3, 'P')

```
Out[7]: [ 1 5/13 4/13 2/13 12/91 6/91 3/91]
[ 0 1 3/7 62/77 6/11 81/154 30/77]
[ 0 0 1 1/2 7/8 3/4 3/4]
[ 0 0 0 1 1/2 4/3 5/3]
[ 0 0 0 0 1 6/5 2]
[ 0 0 0 0 0 1 20/7]
[ 0 0 0 0 0 0 1]
```

In [9]: PJacktoM1(5,3)

```
Out[9]: [ 1 5/13 4/13 2/13 12/91 6/91 3/91]
[ 0 1 3/7 62/77 6/11 81/154 30/77]
[ 0 0 1 1/2 7/8 3/4 3/4]
[ 0 0 0 1 1/2 4/3 5/3]
[ 0 0 0 0 1 6/5 2]
[ 0 0 0 0 0 1 20/7]
[ 0 0 0 0 0 0 1]
```

In [8]: JacktoM(5,3, 'Q')

```
Out[8]: [ 91/729 35/729 28/729 14/729 4/243 2/243 1/243]
[ 0 154/3159 22/1053 124/3159 28/1053 1/39 20/1053]
[ 0 0 8/189 4/189 1/27 2/63 2/63]
[ 0 0 0 1/33 1/66 4/99 5/99]
[ 0 0 0 0 5/168 1/28 5/84]
[ 0 0 0 0 0 7/270 2/27]
[ 0 0 0 0 0 0 1/21]
```

In [11]: QJacktoM1(5,3)

```
Out[11]: [ 91/729  35/729  28/729  14/729  4/243  2/243  1/243]
          [      0 154/3159 22/1053 124/3159 28/1053  1/39 20/1053]
          [      0      0  8/189  4/189  1/27  2/63  2/63]
          [      0      0      0  1/33  1/66  4/99  5/99]
          [      0      0      0      0  5/168  1/28  5/84]
          [      0      0      0      0      0  7/270  2/27]
          [      0      0      0      0      0      0  1/21]
```

In [3]: JacktoM(5,3,'C')

```
Out[3]: [      1      5/13      4/13      2/13      12/91      6/91      3/91]
          [      0     60/13    180/91 3720/1001 360/143 2430/1001 1800/1001]
          [      0      0     54/7     27/7     27/4     81/14     81/14]
          [      0      0      0    135/11   135/22   180/11   225/11]
          [      0      0      0      0    405/28   243/14   405/14]
          [      0      0      0      0      0      18    360/7]
          [      0      0      0      0      0      0     81/7]
```

In [11]: CJacktoM1(5,3)

```
Out[11]: [      1      5/13      4/13      2/13      12/91      6/91      3/91]
          [      0     60/13    180/91 3720/1001 360/143 2430/1001 1800/1001]
          [      0      0     54/7     27/7     27/4     81/14     81/14]
          [      0      0      0    135/11   135/22   180/11   225/11]
          [      0      0      0      0    405/28   243/14   405/14]
          [      0      0      0      0      0      18    360/7]
          [      0      0      0      0      0      0     81/7]
```

In [10]: A=JacktoP(5,3,'P').inverse();

A

```
Out[10]: [      1  -5/13  -1/7  5/22  5/56  -1/6  1/7]
          [      1   8/13  -4/7  -4/11  3/14  4/15  -3/7]
          [      1  -5/13   6/7  -3/11 -15/28  1/2  -3/7]
          [      1 21/13      0   6/11  -9/7  -1/10  9/7]
          [      1   8/13  10/7 -15/11 27/28  -4/5  9/7]
          [      1 34/13  18/7 27/11 27/28  -9/5 -27/7]
          [      1 60/13  54/7 135/11 405/28  18  81/7]
```

In [11]: A=JacktoP(5,3,'Q').inverse();

A

```
Out[11]: [ 729/91 -1215/154  -27/8  15/2      3  -45/7      3]
          [ 729/91  972/77  -27/2  -12  36/5  72/7  -9]
          [ 729/91 -1215/154  81/4  -9  -18  135/7  -9]
          [ 729/91  729/22      0  18  -216/5  -27/7  27]
          [ 729/91  972/77  135/4  -45  162/5  -216/7  27]
          [ 729/91  4131/77  243/4  81  162/5  -486/7  -81]
          [ 729/91  7290/77  729/4  405  486  4860/7  243]
```

```
In [12]: A=JacktoP(5,3,'J').inverse()
A
```

```
Out[12]: [ 1/3640 -5/4004 -1/1120  5/1584  1/672 -1/252  1/840]
[ 1/3640  2/1001 -1/280 -1/198  1/280  2/315 -1/280]
[ 1/3640 -5/4004  3/560 -1/264 -1/112  1/84 -1/280]
[ 1/3640  3/572  0  1/132 -3/140 -1/420  3/280]
[ 1/3640  2/1001  1/112 -5/264  9/560 -2/105  3/280]
[ 1/3640 17/2002  9/560  3/88  9/560 -3/70 -9/280]
[ 1/3640 15/1001 27/560 15/88 27/112  3/7 27/280]
```

```
In [4]: A=JacktoP(5,3,'C').inverse()
A
```

```
Out[4]: [ 1 -1/12 -1/54  1/54  1/162 -1/108  1/81]
[ 1  2/15 -2/27 -4/135  2/135  2/135 -1/27]
[ 1 -1/12  1/9 -1/45 -1/27  1/36 -1/27]
[ 1  7/20  0  2/45 -4/45 -1/180  1/9]
[ 1  2/15  5/27 -1/9  1/15 -2/45  1/9]
[ 1 17/30  1/3  1/5  1/15 -1/10 -1/3]
[ 1  1  1  1  1  1  1]
```

```
In [5]: C=PtoJack(5,3,'Q')
C
```

```
Out[5]: [ 729/91 -1215/154 -27/8  15/2  3 -45/7  3]
[ 729/91  972/77 -27/2 -12  36/5  72/7 -9]
[ 729/91 -1215/154  81/4 -9 -18  135/7 -9]
[ 729/91  729/22  0  18 -216/5 -27/7 27]
[ 729/91  972/77 135/4 -45 162/5 -216/7 27]
[ 729/91 4131/77 243/4  81 162/5 -486/7 -81]
[ 729/91 7290/77 729/4  405  486 4860/7 243]
```

```
In [15]: PtoJack(5,2,'C')
```

```
Out[15]: [ 1 -1/8 -1/48  1/24  1/96 -1/32  1/16]
[ 1  1/10 -13/120 -1/30  1/24  1/40 -1/8]
[ 1 -1/8  1/8 -1/40 -1/16  1/16 -1/8]
[ 1 13/40 -1/20  1/40 -1/8  1/40  1/4]
[ 1  1/10 11/60 -1/6  1/12 -1/20  1/4]
[ 1 11/20  3/10  3/20  0 -1/5 -1/2]
[ 1  1  1  1  1  1  1]
```

```
In [6]: %%time
C=PtoJack(20,2,'C');
```

```
CPU times: user 20 s, sys: 985 ms, total: 21 s
Wall time: 21 s
```

```
In [7]: %%time
        C=JacktoP(20,2,'C');
```

```
CPU times: user 21.7 s, sys: 938 ms, total: 22.7 s
Wall time: 22.7 s
```

```
In [8]: %%time
        C=MtoJack(20,3,'J');
```

```
CPU times: user 26.1 s, sys: 1.25 s, total: 27.4 s
Wall time: 27.4 s
```

```
In [19]: JacktoM(5,3,'Q').inverse()
```

```
Out[19]: [ 729/91 -1215/154 -27/8 15/2 3 -45/7 3]
          [ 0 3159/154 -81/8 -39/2 21/5 117/7 -12]
          [ 0 0 189/8 -33/2 -21 180/7 -12]
          [ 0 0 0 33 -84/5 -198/7 30]
          [ 0 0 0 0 168/5 -324/7 30]
          [ 0 0 0 0 0 270/7 -60]
          [ 0 0 0 0 0 0 21]
```

```
In [20]: MtoJack(5,3,'Q')
```

```
Out[20]: [ 729/91 -1215/154 -27/8 15/2 3 -45/7 3]
          [ 0 3159/154 -81/8 -39/2 21/5 117/7 -12]
          [ 0 0 189/8 -33/2 -21 180/7 -12]
          [ 0 0 0 33 -84/5 -198/7 30]
          [ 0 0 0 0 168/5 -324/7 30]
          [ 0 0 0 0 0 270/7 -60]
          [ 0 0 0 0 0 0 21]
```