In[1]:= `<< GeneratingFunctions.m`

GeneratingFunctions Package by Christian Mallinger — © RISC Linz — V 0.69 (28-Sep-2009)

In[2]:= `(* Example 3.7: closure properties addition *)`

In[3]:=
```
REPlus[{q[n + 2] - 2 x q[n + 1] + q[n] == 0, q[0] == 1, q[1] == x},
  {q[n + 1] - (1 + x) q[n] == 0, q[0] == 1}, q[n]]
```

Out[3]= $\{(-1 - x)\, q[n] + (1 + 2x + 2x^2)\, q[1 + n] + (-1 - 3x)\, q[2 + n] + q[3 + n] == 0,\ q[0] == 2,\ q[1] == 1 + 2x,\ q[2] == 2x + 3x^2\}$

In[4]:= `(* help: *)`

In[5]:= `? REPlus`

```
RecurrenceEquationPlus[re1,re2,a[n]] gives a recurrence equation that
is satisfied by the sum of solutions of the recurrences re1 and re2.
All recurrences are given in a[n].

Alias: REPlus

See also: REInfo, DEPlus
```

In[6]:= `? RE*`

▼ GeneratingFunctions`

RE	RECauchy	REInterlace	REShadow
RE2DE	REHadamard	REOut	RESubsequence
RE2L	REInfo	REPlus	

In[7]:=
```
(*
Generating function for Legendre polynomials:
  Input: three term recurrence for Legendre polynomials
    Output: differential equation for the generating function
*)
```

In[8]:=
```
ode = RE2DE[{(n + 2) p[n + 2] - (2 n + 3) x p[n + 1] + (n + 1) p[n] == 0, p[0] == 1, p[1] == x}, p[n], F[z]]
```

Out[8]= $\{-(x - z)\, F[z] - (-1 + 2xz - z^2)\, F'[z] == 0,\ F[0] == 1\}$

In[9]:= `DSolve[ode, F[z], z]`

Out[9]= $\left\{\left\{F[z] \to \frac{1}{\sqrt{1 - 2xz + z^2}}\right\}\right\}$

In[10]:= `(* without initial conditions *)`

In[11]:=
```
RE2DE[{(n + 2) p[n + 2] - (2 n + 3) x p[n + 1] + (n + 1) p[n] == 0}, p[n], F[z]]
```

Out[11]= $F[z] - 3\,(x - z)\, F'[z] - (-1 + 2xz - z^2)\, F''[z] == 0$

In[12]:=
```
(* Legendre kernel polynomials for y=1: first using closure properties,
then guessing and proving *)
```

In[13]:=
```
prod = REHadamard[{(n + 2) s[n + 2] - (2 n + 3) x s[n + 1] + (n + 1) s[n] == 0, s[0] == 1, s[1] == x},
  {(2 n + 1) s[n + 1] - (2 n + 3) s[n] == 0, s[0] == 1 / 2}, s[n]]
```

Out[13]= $\left\{(1+n)(5+2n)\,s[n]-(1+2n)(5+2n)\,x\,s[1+n]+(2+n)(1+2n)\,s[2+n]==0,\ s[0]==\frac{1}{2},\ s[1]==\frac{3x}{2}\right\}$

In[14]:=
```
sum = RECauchy[prod, {s[n + 1] - s[n] == 0, s[0] == 1}, s[n]]
```

Solve::svars : Equations may not give solutions for all "solve" variables. >>

Out[14]= $\left\{(2+n)(7+2n)\,s[n]-(7+2n)(2+n+3x+2nx)\,s[1+n]+(3+2n)(3+n+7x+2nx)\,s[2+n]-(3+n)(3+2n)\,s[3+n]==0,\ s[0]==\frac{1}{2},\ s[1]==\frac{1}{2}+\frac{3x}{2},\ s[2]==-\frac{3}{4}+\frac{3x}{2}+\frac{15x^2}{4}\right\}$

In[15]:=
```
data = Table[Factor[Sum[(2 k + 1) / 2 LegendreP[k, x], {k, 0, n}]], {n, 0, 30}];
```

In[16]:=
```
Take[data, 3]
```

Out[16]= $\left\{\frac{1}{2},\ \frac{1}{2}(1+3x),\ \frac{3}{4}\left(-1+2x+5x^2\right)\right\}$

In[17]:=
```
guess = GuessRE[data, s[n]]
```

Out[17]= $\left\{\left\{(2+n)(5+2n)\,s[n]+\left(-1-15x-16nx-4n^2x\right)s[1+n]+(2+n)(3+2n)\,s[2+n]==0,\ s[0]==\frac{1}{2},\ s[1]==\frac{1}{2}(1+3x)\right\},\ \text{ogf}\right\}$

In[18]:=
```
(* now proving - need to adapt initial values *)
```

In[19]:=
```
minusguess = {guess[[1, 1]], s[0] == -1/2, s[1] == -1/2 (1 + 3 x)}
```

Out[19]= $\left\{(2+n)(5+2n)\,s[n]+\left(-1-15x-16nx-4n^2x\right)s[1+n]+(2+n)(3+2n)\,s[2+n]==0,\ s[0]==-\frac{1}{2},\ s[1]==\frac{1}{2}(-1-3x)\right\}$

In[20]:=
```
zero = REPlus[sum, minusguess, s[n]]
```

Out[20]= $\left\{-(2+n)(7+2n)\,s[n]+(7+2n)(2+n+3x+2nx)\,s[1+n]-(3+2n)(3+n+7x+2nx)\,s[2+n]+(3+n)(3+2n)\,s[3+n]==0,\ s[0]==0,\ s[1]==\frac{1}{2}(-1-3x)+\frac{1}{2}(1+3x),\ s[2]==\frac{3}{4}-\frac{3x}{2}-\frac{15x^2}{4}+\frac{3}{4}\left(-1+2x+5x^2\right)\right\}$

In[21]:=
```
Factor[Last /@ Take[zero, -3]]
```

Out[21]= {0, 0, 0}

In[22]:=
```
(* end of proof *)
```

In[23]:= **? GuessRE**

```
GuessRecurrenceEquation[list,a[n],{minorder,maxorder},{mindeg,maxdeg}]
tries to guess a linear recurrence equation (RE) in a[n] with polynomial
coefficients, which is satisfied by the elements in the input list.
The orders that are tried range from "minorder" to "maxorder",
the coefficient polynomials are tried with degrees "mindeg" up to
"maxdeg". The output contains a RE ( or FAIL, if no recurrence could be
found) together with a transformation that had to be performed on the
generating function of the input list. Short forms for the function call are

GuessRecurrenceEquation[list,a[n]] and

GuessRecurrenceEquation[list,a[n],maxorder,maxdeg],

where the default values minorder=1, maxorder=2, mindeg=0, maxdeg=3 are used.

GuessRecurrenceEquation has the following options (and default values):

AdditionalEquations ("All")     In order to avoid accidental results,

     "All" elements in the input list are used

     to build the equations for the coefficients

     of the RE. Setting this parameter to

     a positive integer k, causes the function

     to build just d+k equations, where d is the

     number of indeterminants. This option can

     be used to get a speed up.

Hypergeom           (False)         whether to search for m-hypergeometric

     recurrences only.

Transform           ({"ogf","egf"}) transformations that are tried.

Note: The first element in the list gives the term a[0] in the sequence.

Alias: GuessRE

See also: GuessDE, GuessAE, GuessRatF, ListOfTransformations
```

In[24]:= **(* Guessing: a simple example *)**

In[25]:=
```
Clear[c, ans];
c[i_, n_, d_] := c[i, 0] + Sum[c[i, i1] n^i1, {i1, 1, d}];
ans[a_, n_, {order_, deg_}] := Sum[c[i2, n, deg] a[n + i2], {i2, 0, order}];
```

In[28]:= **(* ansatz for a recurrence of order 2 with coefficients of maximal degree 2 satisfied by a given sequence a[n] *)**

In[29]:= **ans[a, n, {2, 2}]**

Out[29]= $a[n]\,(c[0,0] + n\,c[0,1] + n^2\,c[0,2]) + a[1+n]\,(c[1,0] + n\,c[1,1] + n^2\,c[1,2]) + a[2+n]\,(c[2,0] + n\,c[2,1] + n^2\,c[2,2])$

In[30]:= **b[n_] := Binomial[2 n, n] (-4)^n;**

```
In[31]:= data = Table[b[n], {n, 0, 20}]

Out[31]= {1, -8, 96, -1280, 17 920, -258 048, 3 784 704, -56 229 888, 843 448 320, -12 745 441 280,
  193 730 707 456, -2 958 796 259 328, 45 368 209 309 696, -697 972 450 918 400, 10 768 717 814 169 600,
  -166 556 168 859 156 480, 2 581 620 617 316 925 440, -40 091 049 586 568 724 480,
  623 638 549 124 402 380 800, -9 715 632 133 727 531 827 200, 151 563 861 286 149 496 504 320}

In[32]:= sys = Table[ans[b, n, {1, 1}] == 0, {n, 0, 6}]

Out[32]= {c[0, 0] - 8 c[1, 0] == 0, -8 (c[0, 0] + c[0, 1]) + 96 (c[1, 0] + c[1, 1]) == 0,
  96 (c[0, 0] + 2 c[0, 1]) - 1280 (c[1, 0] + 2 c[1, 1]) == 0,
  -1280 (c[0, 0] + 3 c[0, 1]) + 17 920 (c[1, 0] + 3 c[1, 1]) == 0,
  17 920 (c[0, 0] + 4 c[0, 1]) - 258 048 (c[1, 0] + 4 c[1, 1]) == 0,
  -258 048 (c[0, 0] + 5 c[0, 1]) + 3 784 704 (c[1, 0] + 5 c[1, 1]) == 0,
  3 784 704 (c[0, 0] + 6 c[0, 1]) - 56 229 888 (c[1, 0] + 6 c[1, 1]) == 0}

In[33]:= sol = First[Solve[sys]]

         Solve::svars : Equations may not give solutions for all "solve" variables. >>

Out[33]= {c[0, 0] → 8 c[1, 1], c[1, 0] → c[1, 1], c[0, 1] → 16 c[1, 1]}

In[34]:= rec = ans[B, n, {1, 1}] /. sol

Out[34]= B[1 + n] (c[1, 1] + n c[1, 1]) + B[n] (8 c[1, 1] + 16 n c[1, 1])

In[35]:= rec = Collect[rec, B[_], Factor]

Out[35]= 8 (1 + 2 n) B[n] c[1, 1] + (1 + n) B[1 + n] c[1, 1]

In[36]:= First[GuessRE[Table[b[n], {n, 0, 20}], B[n]]]

Out[36]= {8 (1 + 2 n) B[n] + (1 + n) B[1 + n] == 0, B[0] == 1}
```