This notebook accompanies the paper: "Creative Telescoping on Multiple Sums" by Christoph Koutschan and Elaine Wong.
The paper has been submitted for publication to a special issue of the Springer-Birkhauser Journal, Mathematics in Computer Science, as part of the post-proceedings for CASC 2020.

Package Loading

The following packages are available for free download at this link: https://www3.risc.jku.at/research/-combinat/software/ergosum/packages.html
They should be saved in the same directory as the notebook.

In[ ]:=
```
SetDirectory[NotebookDirectory[]];
<< RISC`HolonomicFunctions`;
<< RISC`Guess`;
```

HolonomicFunctions Package version 1.7.3 (21-Mar-2017)
written by Christoph Koutschan
Copyright Research Institute for Symbolic Computation (RISC),
Johannes Kepler University, Linz, Austria

```
--> Type  ?HolonomicFunctions  for help.
```

Package GeneratingFunctions version 0.8 written by Christian Mallinger
Copyright Research Institute for Symbolic Computation (RISC),
Johannes Kepler University, Linz, Austria

Guess Package version 0.52
written by Manuel Kauers
Copyright Research Institute for Symbolic Computation (RISC),
Johannes Kepler University, Linz, Austria

Definitions

In[◦]:=
```
(* Original Formula (1) *)
polyG[b_, m_, s_, x_] :=
 Sum[Binomial[s, r] Binomial[k - 1, r - 1] (b - 1) (b x) ^k / (-b) ^r
   Sum[(-b) ^i Binomial[r - 1, i], {i, 0, r - 1 - Max[k - m, 0]}], {k, m + s - 1}, {r, s}]
(* Split Sums *)
G1[b_, m_, s_, x_] := -Sum[
   Binomial[s, r] Binomial[k - 1, r - 1] (b - 1) ^r (b x) ^k / b^r, {k, m + s - 1}, {r, s}]
G2[b_, m_, s_, x_] := Sum[Binomial[s, r] Binomial[k - 1, r - 1] (1 - b) / (-b) ^r
   Sum[Binomial[r - 1, i] (-b) ^i (b x) ^k, {i, r - (k - m), r - 1}],
  {k, m + 1, m + s - 1}, {r, s}]
(* Triple Sum No Max (8) *)
summand[b_, m_, s_, x_, k_, r_, i_] :=
 Binomial[s, r] Binomial[k - 1, r - 1] Binomial[r - 1, i] (b - 1) (b x) ^k / (-b) ^ (r - i)
tripleG[b_, m_, s_, x_] := Sum[summand[b, m, s, x, k, r, i],
  {k, m + s - 1}, {r, s}, {i, 0, r - 1 - (k - m)}]
```

In[◦]:=
```
(* We do a few experimental sanity checks
 to confirm that our definitions are the same. *)
Timing[(Table[Together[polyG[b, m, s, x] - G1[b, m, s, x] - G2[b, m, s, x]],
      {m, 30}, {s, 30}] // Flatten // Union) === {0}]
Timing[(Table[Together[polyG[b, m, s, x] - tripleG[b, m, s, x]], {m, 30}, {s, 30}] //
     Flatten // Union) === {0}]
```

Out[◦]= {137.95, True}

Out[◦]= {135.61, True}

Toy Example (6)

ALERT: We prefix all functions with "te-" (for toy example) in this section to prevent repeated usage later.

In[◦]:=
```
TraditionalForm[HoldForm[Sum[Binomial[n, k], {k, 5, n}]]]
teSummand[n_, k_] := Binomial[n, k];
```

Out[◦]//TraditionalForm=

$$\sum_{k=5}^{n}\binom{n}{k}$$

Creative Telescoping can take as input the summand OR the annihilator of the summand for the sum to be determined.

```
In[•]:= CreativeTelescoping[Annihilator[teSummand[n, k], {S[k], S[n]}], S[k] - 1, S[n]]
```

$$\text{Out[•]=}\ \left\{\{S_n - 2\},\ \left\{-\frac{k}{-1+k-n}\right\}\right\}$$

```
In[•]:= {tePlist, teQlist} = CreativeTelescoping[teSummand[n, k], S[k] - 1, S[n]]
```

$$\text{Out[•]=}\ \left\{\{S_n - 2\},\ \left\{-\frac{k}{-1+k-n}\right\}\right\}$$

Here, we deal with singularities and commutation.

```
In[•]:= (* Since the sum takes on nonzero values outside of the summation
         bounds AND the operator and the summation does not commute AND
         the certificate has a singularity at one of the boundary values,
        we have a nonzero inhomogeneous part. We collect all such terms here. *)
        teDeltapart[lower_, upper_] :=
         With[{delta = ApplyOreOperator[teQlist[[1]], Binomial[n, k]]},
          (delta /. {k → (upper + 1)}) - (delta /. {k → lower})]
        teCompensated[upper_] := Module[{j = n - upper},
            -Binomial[n + 1, n - j + 1] -
             Sum[(Binomial[n + 1, n + 1 - j + k] - 2 Binomial[n, n - j + k]), {k, 1, j}]
          ];
        Table[teCompensated[i] // Together, {i, n - 3, n + 3, 1}]
```

$$\text{Out[•]=}\ \left\{\frac{1}{6}\left(-2\,n + 3\,n^2 - n^3\right),\ \frac{1}{2}\left(n - n^2\right),\ -n,\ -1,\ 0,\ 0,\ 0\right\}$$

```
In[•]:= (* Such parts disappear if the lower limit of the sum is ≤
         0 or the upper limit is ≥ n+1, with a division by zero error for k=n. *)
        (* We use the term "natural boundaries" to describe the sum
         in the case that the inhomogeneous parts disappear. *)
        teInhomoparts[lower_, upper_] := teCompensated[upper] + teDeltapart[lower, upper];
        teInhomoparts[5, n - 1] // FullSimplify
        Quiet[teInhomoparts[5, n]]
        teInhomoparts[0, n + 1]
```

$$\text{Out[•]=}\ -\frac{1}{24}\,(-3+n)\,(-2+n)\,(-1+n)\,n$$

```
Out[•]= Indeterminate
```

```
Out[•]= 0
```

In[•]:=
```
(* We can now find the annihilator for the inhomogeneous part and
 multiply (on the left) to P to get the annihilator for the whole sum. *)
teAnninhomo = Annihilator[teInhomoparts[5, n - 1], {S[n]}][[1]];
teAnn = teAnninhomo ** tePlist[[1]];
(* Check. *)
ApplyOreOperator[teAnn, Sum[teSummand[n, k], {k, 5, n}]] // Together
```

Out[•]= 0

We solve the recurrence to get a closed form, equivalent to Mathematica's result.

```
(* Here, we illustrate the use of Carsten Schneider's Sigma package,
which has a very nice recurrence solver. *)
Sigma`Summation`Objects`S = SigmaS;
Sigma`Summation`SumProducts`CreativeTelescoping = SigmaCT;
Quiet[Import["Sigma.m"], General::shdw]
Remove[Sigma`Summation`Objects`S,
  Sigma`Summation`SumProducts`CreativeTelescoping];
```

Sigma - A summation package by Carsten Schneider — © RISC — V 2.01 (May 12, 2016) Help

In[•]:=
```
(* Solving the recurrence and comparing closed forms. *)
teInicond = Table[Sum[teSummand[n, k], {k, 5, n}], {n, 0, 15}];
SolveRecurrence[ApplyOreOperator[teAnn, c[n]] == 0, c[n]];
teClosedall = FindLinearCombination[%, teInicond, n, 3, MinInitialValue → 5] /.
  Sigma`Summation`Objects`Private`MyPower → Power
```

Out[•]= $\frac{1}{24}\left(-24+3\times 2^{3+n}-14\,n-11\,n^2+2\,n^3-n^4\right)$

In[•]:=
```
(* The closed form is equivalent to Mathematica's version. *)
teClosedmath = Together[Simplify[Sum[Binomial[n, k], {k, 5, n}]]];
teClosedall === teClosedmath
```

Out[•]= True

Guessing

In[∘]:=
```
(* We generate data necessary for guessing in all four variables. *)
(* We impose what we believe the shape of the recurrence to be,
and if such a recurrence exists, the function will find one. *)
datalist =
  (Table[polyG[b, m, s, x], {x, 2, 10}, {b, 2, 10}, {m, 10}, {s, 10}] // Together);
guessrec = GuessMultRE[datalist, {c[x, b, m, s], c[x, b, m, s + 1], c[x, b, m, s + 2],
     c[x, b, m, s + 3]}, {x, b, m, s}, 4, StartPoint → {2, 2, 1, 1}][[1]]
```

Out[∘]=
$$(1+m+s)\ (-1+x)^2\ c[x,b,m,s] - (-1+x)\ \left(-4-2\,m-3\,s+x+b\,x+m\,x+b\,m\,x+s\,x+b\,s\,x\right)\ c[x,b,m,1+s] + \left(5+m+3\,s-3\,x-3\,b\,x-m\,x-b\,m\,x-2\,s\,x-2\,b\,s\,x+b\,x^2+b\,m\,x^2+b\,s\,x^2\right)\ c[x,b,m,2+s] + (2+s)\ (-1+b\,x)\ c[x,b,m,3+s]$$

```
(* Since the recurrence only depends on s,
we make some subsitutions so that it looks nicer for printing. *)
(* This is the minimal order recurrence
 that generates the annihilating ideal for polyG. *)
guessrec = guessrec /. (Thread[c[x, b, m, s + #] → c[s + #]] & /@ (Range[4] - 1))
```

Out[∘]=
$$(1+m+s)\ (-1+x)^2\ c[s] - (-1+x)\ \left(-4-2\,m-3\,s+x+b\,x+m\,x+b\,m\,x+s\,x+b\,s\,x\right)\ c[1+s] + \left(5+m+3\,s-3\,x-3\,b\,x-m\,x-b\,m\,x-2\,s\,x-2\,b\,s\,x+b\,x^2+b\,m\,x^2+b\,s\,x^2\right)\ c[2+s] + (2+s)\ (-1+b\,x)\ c[3+s]$$

Strategy Implementations for our Problem

The following implementations do not depend on each other and each section can be run on its own. All relevant definitions for that particular section are self-contained.

Treatment of Inhomogeneous Parts: Singularities and Commutation

In[∘]:=
```
(* Summand of G1 *)
f[b_, s_, x_, k_, r_] := Binomial[s, r] Binomial[k - 1, r - 1] (b - 1) ^r (b x) ^k / b^r;
(* Innersum of G1 *)
F1[b_, s_, x_, k_] := Sum[f[b, s, x, k, r], {r, 1, s}];
(* Creative Telescoping: Observe that we have a singularity at r=
 s+1 in the first certificate. *)
{Pinner1, Qinner1} = CreativeTelescoping[f[b, s, x, k, r], S[r] - 1, {S[s], S[k]}]
```

Out[∘]=
$$\left\{\left\{b\,s\,x\,S_s + (-1-k)\,S_k + (k\,x-s\,x),\ (2+k)\,S_k^2 + \left(-x-b\,x-k\,x-b\,k\,x+s\,x-b\,s\,x\right)\,S_k + b\,k\,x^2\right\},\right.$$
$$\left.\left\{-\frac{b\left(-k\,r+k\,r^2+r\,s-r^2\,s\right)x}{(1+k-r)\ (-1+r-s)},\ \frac{b^2\,k\left(-r+r^2\right)x^2}{2+3\,k+k^2-3\,r-2\,k\,r+r^2}\right\}\right\}$$

```
In[•]:= deltapart[lower_, upper_, head_, pos_] :=
         With[{delta = ApplyOreOperator[head[[pos]], f[b, s, x, k, r]]},
          (delta /. {r → (upper + 1)}) - (delta /. {r → lower})]
        (* We determine the delta part for the sum up to r=s-1 (i.e.,
           up to the point that we can avoid the singularity in our evaluation. *)
        deltapart[1, s - 1, Qinner1, 1] // Simplify
```

Out[•]= $\frac{1}{-1-k+s}\left(\frac{-1+b}{b}\right)^{s} b\,(-1+s)\,s\,(-k+s)\,x\,(b\,x)^{k}\,\text{Binomial}[-1+k,-1+s]$

```
In[•]:= (* The original sum must be ``filled
          in'' to account for the missing term at r=s. *)
        (* This gives us the following compensation term. *)
        compensatedSing =
         -ReleaseHold[ApplyOreOperator[Pinner1[[1]], Hold[f[b, s, x, k, r]]] /. r → s]
```

Out[•]= $-(-1+b)^{s}\,b^{1-s}\,s\,(1+s)\,x\,(b\,x)^{k}\,\text{Binomial}[-1+k,-1+s]-(-1+b)^{s}\,b^{-s}\,(b\,x)^{k}\,(k\,x-s\,x)\,\text{Binomial}[-1+k,-1+s]-(-1+b)^{s}\,b^{-s}\,(-1-k)\,(b\,x)^{1+k}\,\text{Binomial}[k,-1+s]$

```
In[•]:= (* We also have to concern ourselves with the compensated parts from
         any and all S[s] operators. Luckily, there is only one s-shift. *)
        compensatedSshift[coeff_, upper_, shift_, ind_List] :=
          -coeff * Sum[f[b, s + shift, x, If[ind[[1]] == 1, k + i, k], If[ind[[2]] == 1, r + i, r]],
            {i, upper + 1, upper + shift}];
        (* The annhilator for the inhomogeneous parts, together with the telescoper,
        gives the annihilator for the whole sum. *)
        inhomoparts1 = compensatedSshift[b s x, s - 1, 1, {0, 1}] +
           compensatedSing + deltapart[1, s - 1, Qinner1, 1];
        ann1 = Annihilator[inhomoparts1, {S[s], S[k]}][[1]] ** Pinner1[[1]];
        (* For the second certificate, there are no singularities,
        and the boundaries are natural,
        so the delta part is 0 and we just need to treat the commutation. *)
        inhomoparts2 = compensatedSshift[k + 2, s, 2, {0, 1}] + deltapart[1, s, Qinner1, 2];
        ann2 = Annihilator[inhomoparts2, {S[s], S[k]}][[1]] ** Pinner1[[2]];
        Support[{ann1, ann2}]
```

Out[•]= $\{\{S_s^2 S_k, S_s S_k^2, S_k^3, S_s^2, S_s S_k, S_k^2, S_s, S_k, 1\}, \{S_k^3, S_k^2, S_k, 1\}\}$

```
In[•]:= (* We now confirm that our annihilators
         are correct and indeed annihilates our sum. *)
        FullSimplify[ApplyOreOperator[{ann1, ann2}, Sum[f[b, s, x, k, r], {r, 1, s}]]]
```

Out[•]= {0, 0}

In[∘]:=
```
(* We confirm that our computations are correct. *)
OreReduce[{ann1, ann2}, Pinner1]
```

Out[∘]= {0, 0}

In[∘]:=
```
(* In fact, the telescopers are the correct
 annihilators already due to a natural boundaries argument. *)
Annihilator[F1[b, s, x, k], {S[s], S[k]}] === Pinner1
```

Out[∘]= True

Treatment of Inhomogeneous Parts: Closure

In[∘]:=
```
(* G1 *)
G1[b_, m_, s_, x_] :=
 -Sum[Binomial[s, r] Binomial[k - 1, r - 1] (b - 1) ^ r (b x) ^ k / b ^ r, {k, m + s - 1}, {r, s}]
(* Subs for testing. *)
subs = {b → 31, m → 15, x → 5 / 7};
checksubs = {31, 15, s, x};
(* Summand of G1 *)
f[b_, s_, x_, k_, r_] := Binomial[s, r] Binomial[k - 1, r - 1] (b - 1) ^ r (b x) ^ k / b ^ r;
(* Innersum of G1 *)
F1[b_, s_, x_, k_] := Sum[f[b, s, x, k, r], {r, 1, s}];
(* Creative Telescoping *)
{Pinner1, Qinner1} = CreativeTelescoping[f[b, s, x, k, r], S[r] - 1, {S[s], S[k]}];
{Pouter1, Qouter1} = CreativeTelescoping[Pinner1, S[k] - 1, {S[s]}]
```

Out[∘]= $\left\{\left\{(1 - b\,x)\,S_s + (-1 + x)\right\}, \left\{\frac{1 + k}{b\,s\,x}\,S_k + \frac{-k + s - b\,s}{b\,s}\right\}\right\}$

Symbolic Computation : Here we deal with unnatural boundaries and the fact that the telescoper does not commute with the sum and make the appropriate adjustments

```
In[•]:= (* We first apply the certificates and then
         determine the collapsing sum (up to k=m+s-1). *)
        outercert1 = With[{delta = ApplyOreOperator[Qouter1[[1]], F1[b, s, x, k]]},
           (delta /. k → (s + m)) - (delta /. k → 1)];
        (* Next, we determine the annihilator of the collapsed sum and
         extra stuff that was added to Sum[P*F[s,k],{k,1,m+s-1}]
         so that what remains is P*Sum[F[s,k],{k,1,m+s-1}]. *)
        outerstuff1 = (1 - b x) F1[b, s + 1, x, m + s];
        (* The inhomogeneous part is the combination
         of stuff we want to remove and the telescoped sum. *)
        inhom1 = FunctionExpand[-outerstuff1 + outercert1]
```

Out[•]= $-\frac{(-1+b)\ (-1+s-b\,s)\ x}{b}-\frac{(-1+b)\ (1+b-s+b\,s)\ x}{b}+\frac{1}{b^2\,x}$

$(-1+b)\ (1+m+s)\ (b\,x)^{1+m+s}\ \text{Hypergeometric2F1}\left[1-s,\,-m-s,\,2,\,\frac{-1+b}{b}\right]+\frac{1}{b^2}$

$(-1+b)\ (-m-b\,s)\ (b\,x)^{m+s}\ \text{Hypergeometric2F1}\left[1-s,\,1-m-s,\,2,\,\frac{-1+b}{b}\right]-\frac{1}{b}$

$(-1+b)\ (1+s)\ (b\,x)^{m+s}\ (1-b\,x)\ \text{Hypergeometric2F1}\left[1-m-s,\,-s,\,2,\,\frac{-1+b}{b}\right]$

We now illustrate how inhom1 can be simplified and rewritten so that we can use the closure property of applying an operator.

```
In[•]:= (* The first two terms do not contain 2F1s so they will be
         easy to treat. We therefore separate inhom1 into two parts. *)
        inhom1first = inhom1[[1 ;; 2]];
        inhom1last = inhom1[[3 ;; 5]];
        (* We rewrite inhom1last as a combination of some shifts (by observation). *)
        hg[s_, m_] := (b x) ^ (m + s) * Hypergeometric2F1[1 - s, 2 - m - s, 2, (b - 1) / b]
        pf1 = HoldForm[(-1 + b) (1 + m + s) / (b ^ 2 x * b x )];
        pf2 = HoldForm[(-1 + b) (-m - b s) / (b ^ 2 * b x)];
        pf3 = HoldForm[(-1 + b) (1 + s) (-1 + b x) / (b * b x)];
        (* This is the new operator constructed from the above observations. *)
        op = pf1 S[m] ^ 2 + pf2 S[m] + pf3 S[s];
        applyop = ReleaseHold[ApplyOreOperator[op, hg[s, m]]];
        (* We can see that our operator gives us the correct expression. *)
        Simplify[inhom1last] === Simplify[applyop]

Out[•]= True
```

This allows us to compute the annihilator for inhom1 by computing the separate annihilators and applying the closure property.

```
In[•]:= anninhom1first = Annihilator[inhom1first, {S[s]}];
anninhom1last =
  DFiniteOreAction[Annihilator[hg[s, m], {S[m], S[s]}], ReleaseHold[op]];
R1 = DFinitePlus[anninhom1first,
   {ToOrePolynomial[anninhom1last[[2]], OreAlgebra[S[s]]]}];
(* We now finally have a good annihilator for polyfirst. *)
annouter1 = ((# ** Pouter1[[1]]) & /@ R1);
(* The support gives an idea of the operators
 that we are dealing with in our annihilating ideal. *)
Support[annouter1]
```

Out[•]= $\{\{S_s^4, S_s^3, S_s^2, S_s, 1\}\}$

```
In[•]:= (* Another quick check confirms that we are on the right track. *)
Table[ApplyOreOperator[#, (G1 @@ checksubs) /. x → 5 / 7] & /@ annouter1, {s, 30}];
(% /. subs // Flatten // Union) === {0}
```

Out[•]= True

Speedup: DFiniteSub

```
In[•]:= (* G2 *)
G2[b_, m_, s_, x_] :=
  Sum[Binomial[s, r] Binomial[k - 1, r - 1] (1 - b) / (-b) ^ r Sum[Binomial[r - 1, i]
      (-b) ^ i (b x) ^ k, {i, r - (k - m), r - 1}], {k, m + 1, m + s - 1}, {r, s}];
subs = {b → 31, m → 15, x → 5 / 7};
checksubs = {31, 15, s, x};
(* Inner Summand of G2 *)
h[s_, k_, r_] := Binomial[s, r] Binomial[k - 1, r - 1] (1 - b) / (-b) ^ r
   Sum[Binomial[r - 1, i] (-b) ^ i (b x) ^ k, {i, r - (k - m), r - 1}];
(* Inner Sum of G2 *)
H[s_, k_] := Sum[h[s, k, r], {r, 1, s}];
(* Creative Telescoping *)
{Pinner2, Qinner2} = CreativeTelescoping[h[s, k, r], S[r] - 1, {S[k], S[s]}];
(* Timing[{Pouter2,Qouter2}=CreativeTelescoping[Pinner2,S[k]-1,{S[s]}];] *)
(* This took about 70 seconds so we save
 it and then import it for use in this notebook. *)
{Pouter2, Qouter2} = Import["ctouter2.m"];
```

Symbolic Computation I: Here we deal with the fact that the telescoper does not commute with the sum and make the appropriate adjustments.

In[•]:=
```
(* We first apply the certificates and then
 determine the collapsing sum (up to k=m+s-1). *)
outercert2 = With[{delta = ApplyOreOperator[Qouter2, HH[s, k]]},
   (delta /. k → (m + s)) - (delta /. k → (m + 1))];
(* Next we determine the stuff that must be removed to take into account
 the extra shifts when moving the telescopers outside of the sum. *)
Pouter2coeff = OrePolynomialListCoefficients[Pouter2[[1]]];
outerstuff2 = Total[Pouter2coeff[[1 ;; 3]] *
    Table[Sum[HH[s + i, k], {k, m + s, m + s + i - 1}], {i, 3, 1, -1}]];
(* The inhomogeneous part is a combination of stuff we want to
 remove and the telescoped sums (from the delta). *)
inhom2 = -outerstuff2 + outercert2;
```

Symbolic Computation II: Here we try to avoid Mathematica's use of 2F1s and DifferenceRoots by rewriting all the inhomogeneous parts as an operator.

In[•]:=
```
(* We first note that the pieces that DO NOT have 2F1s evaluate to 0:
 this corresponds to evaluating the sum
  with the certificates at the lower boundary. *)
posfree = Position[FreeQ[#, Hypergeometric2F1] & /@
     (List @@ inhom2[[1]] /. HH → H), True] // Flatten;
inhom2[[1, posfree]] /. HH → H // Cancel // FunctionExpand
```

Out[•]= 0

In[•]:=
```
(* We remove those zero pieces and proceed to
 determine an annihilator for the rest of inhom2. *)
inhom2rest = (inhom2[[1, Complement[Range[9], posfree]]]);
(* We construct an operator for these parts (this is done by observation). *)
newop = Qouter2 - Sum[Coefficient[Pouter2[[1]], S[s], i] **
     S[s] ^ i ** Sum[S[k] ^ j, {j, 0, i - 1}], {i, 3}];
(* Let's make sure our operator is correct after the
 application and substitution. *)
Together[inhom2rest - ApplyOreOperator[newop, HH[s, k]] /. k → (m + s)] === {0}
```

Out[•]= True

Symbolic Computation III: Finally, we take control of the substitution by deciding which monomial to use first.

```
(* This is the long way and we keep
 it here so that the timings can be compared. *)
(* Timing[R2act=DFiniteOreAction[Pinner2,newop];] *)
(* The above took 331 seconds. *)
(* Timing[R2=DFiniteSubstitute[R2act,{k→(m+s)}];] *)
(* This above took 107860 seconds!! *)
(* annouter2={R2[[1]]**Pouter2[[1]]}; *)
```

```
In[•]:= (* Here, we are taking control of the substitution. *)
(* First, we compute the operators that act on our inhomogeneous parts. *)
(* Timing[R2act=DFiniteOreAction[Pinner2,newop];] *)
(* The above took 331 seconds so we save and reload. *)
R2act = Import["ctR2act.m"];
(* The following information allows us to determine
 the holonomic rank and shape/size of our operator. *)
(# /@ R2act) & /@ {ByteCount, Support}
```

Out[•]= $\{\{19\,199\,856, 9\,133\,192, 22\,585\,680\}, \{\{S_s^2, S_k, S_s, 1\}, \{S_k S_s, S_k, S_s, 1\}, \{S_k^2, S_k, S_s, 1\}\}\}$

```
In[•]:= (* We also identify the set of irreducible monomials
 of the Groebener basis for the annihilating ideal. *)
UnderTheStaircase[R2act]
```

Out[•]= $\{1, S_s, S_k\}$

```
In[•]:= (* We now do DFiniteSub by hand. *)
(* To do this we will construct an operator T with the desired properties. *)
(* In particular, it should be generated by the Groebner basis of R2act and have
 support {1,S[s]S[k],(S[s]S[k])^2}. We do this using the command OreReduce. *)
(* We also apply the substitution already. *)
red00 = {0, 1};
red11 = OrePolynomialListCoefficients[
   OreReduce[S[k] S[s], Together[R2act], OrePolynomialSubstitute → {k → m + s}]];
(* Speedup actually occurs here. *)
Timing[red22a = OreReduce[S[k] ^2 S[s] ^2, {Together[R2act[[2]]]},
   OrePolynomialSubstitute → {k → m + s}]; Print["Timing for red22a"]; ]
red22a >> "red22a.m";
coeffs = Together[OrePolynomialListCoefficients[R2act[[3]]] /. k → m + s];
coeffs1 = Together[coeffs[[{2, 4}]] / coeffs[[1]]];
coeffs2 = Together[LeadingCoefficient[red22a] * coeffs1];
Timing[red22 = Together[Rest[OrePolynomialListCoefficients[red22a]] - coeffs2];
 Print["Timing for red22"];]
red22 >> "red22.m";
```

Timing for red22a

Out[•]= {1004.67, Null}

Timing for red22

Out[•]= {3211.57, Null}

In[•]:=
```
(* Next, from our ansatz for T,
we solve the corresponding system for the coefficients. *)
sys = Transpose[{red00, red11, red22}];
Timing[R2 = NormalizeVector[NullSpace[sys][[1]]] // Reverse;
 Print["Timing for solving the linear system."];]
R2 = ToOrePolynomial[R2.{S[s]^2, S[s], 1}, OreAlgebra[S[s]]];
(* The whole procedure takes about 4200 seconds so we save it and this
 can be imported by removing the comments from the following code. *)
(* R2=Import["ctR2.m"]; *)
(* This now gives us the desired annihilator for G2. *)
annouter2 = {R2 ** Pouter2[[1]]};
UnderTheStaircase[annouter2]
```

```
Timing for solving the linear system.
```

Out[•]= {1.10798, Null}

Out[•]= $\{1, S_s, S_s^2, S_s^3, S_s^4\}$

In[•]:=
```
(* We perform a quick sanity check to see
 that this is indeed the correct annihilator for G2. *)
Table[ApplyOreOperator[#, (G2 @@ checksubs) /. x → 5 / 7] & /@ annouter2, {s, 30}];
((% /. subs) // Flatten // Union) === {0}
```

Out[•]= True

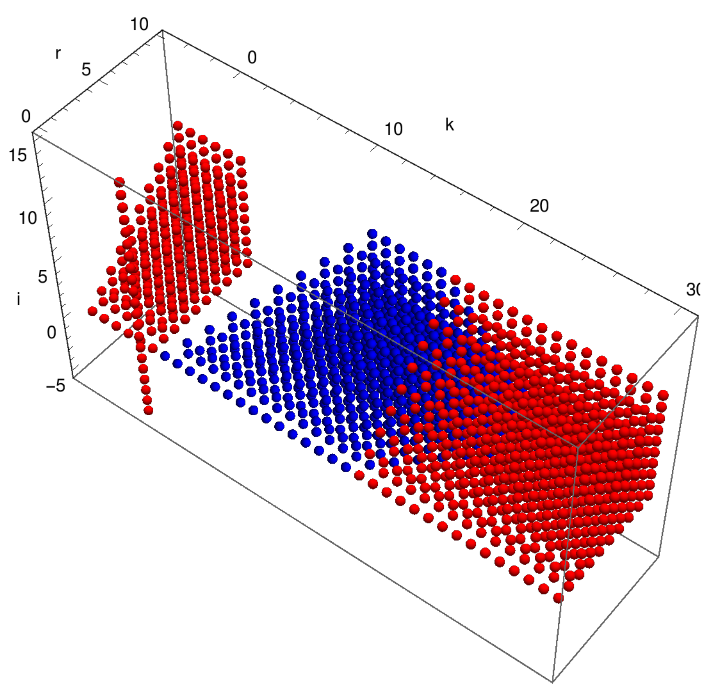
Speedup: Gamma Insertions

In[∘]:=
```
(* Here is the guessed recurrence that we will use to compare our result. *)
guessrecpoly = ToOrePolynomial[
   (1 + m + s) (-1 + x)^2 c[s] - (-1 + x) (-4 - 2 m - 3 s + x + b x + m x + b m x + s x + b s x) c[1 + s] +
    (5 + m + 3 s - 3 x - 3 b x - m x - b m x - 2 s x - 2 b s x + b x^2 + b m x^2 + b s x^2) c[2 + s] +
    (2 + s) (-1 + b x) c[3 + s], c[s]];
(* Here is the summand for the triple sum. *)
summand[m_, s_, b_, x_, k_, r_, i_] := Binomial[s, r] * Binomial[k - 1, r - 1] *
   (- (1 - b) ^ (1 - r)) * (-b) ^ (i) * Binomial[r - 1, i] * ((b - 1) / b) ^ r * (b x) ^ k;
(* New summand with one or two gamma factors. *)
summandTwoGammas[m_, s_, b_, x_, k_, r_, i_] :=
 Binomial[s, r] * Binomial[k - 1, r - 1] * (- (1 - b) ^ (1 - r)) * (-b) ^ (i) *
  Binomial[r - 1, i] * ((b - 1) / b) ^ r * (b x) ^ k * (Gamma[k + eps] / Gamma[k])
  (Gamma[r - i - (k - m) + eps] / Gamma[r - i - (k - m)])
summandOneGamma[m_, s_, b_, x_, k_, r_, i_] := Binomial[s, r] *
  Binomial[k - 1, r - 1] * (- (1 - b) ^ (1 - r)) * (-b) ^ (i) * Binomial[r - 1, i] *
  ((b - 1) / b) ^ r * (b x) ^ k (Gamma[r - i - (k - m) + eps] / Gamma[r - i - (k - m)])
(* Triple sum: this is the same as G and Gnomax. *)
T[m_, s_] := Hold[
   Sum[summand[m, s, b, x, k, r, i], {k, 1, m + s - 1}, {r, 1, s}, {i, 0, r - 1 - (k - m)}]];
(* This is our corrected sum with Gammas on a slightly enlarged region. *)
Tnew2[m_, s_] := Hold[Sum[summandTwoGammas[m, s, b, x, k, r, i],
    {k, -1, m + s + 1}, {r, -1, s + 2}, {i, -2, r + 1 - (k - m)}]];
Tnew1[m_, s_] := Hold[Sum[summandOneGamma[m, s, b, x, k, r, i],
    {k, -1, m + s + 1}, {r, -1, s + 2}, {i, -2, r + 1 - (k - m)}]];
```

Figures

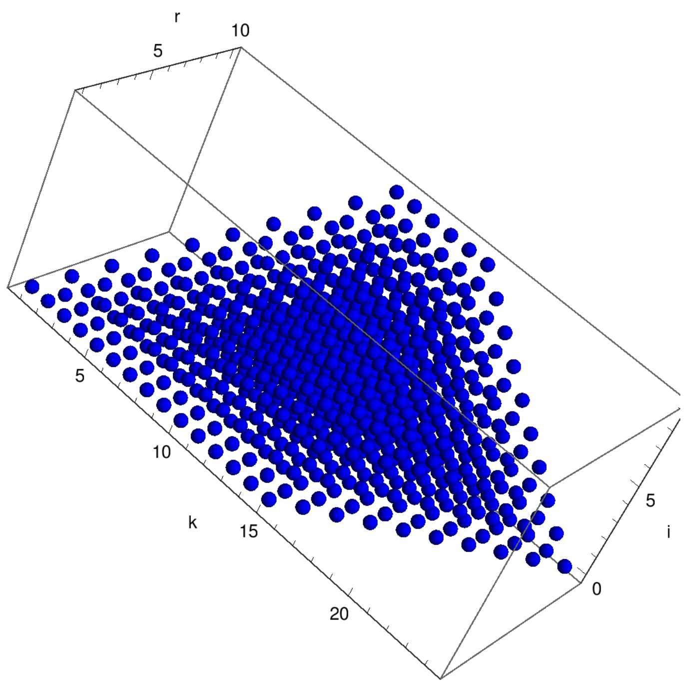
In[•]:=
```
(* The following figure gives a visual representation of how we
 distinguish the points we like from the points we don't like. *)
(* Blue dots: the summand is non-
   zero AND properly within the given boundaries. *)
(* Red dots: the summand is non-zero and outside of the given boundaries. *)
(* Empty: The summand is zero. *)
(* Goal: Manipulate the summand with
  gamma corrections until we only see the blue dots. *)
limit = 5;
With[{s = 10, m = 15, b = 7, x = 1/2},
 Graphics3D[Table[
   If[summand[m, s, b, x, k, r, i] === 0, {},
    {If[1 ≤ k ≤ m + s - 1 && 1 ≤ r ≤ s && 0 ≤ i ≤ r - 1 - (k - m), Blue, Red],
     Sphere[{k, r, i}, 0.3]}]
   , {k, -limit, m + s + limit}, {r, -limit, s + limit}, {i, -limit, s + limit}],
  Axes → True, AxesLabel → {"k", "r", "i"}
 ]]
```

Out[•]=



In[•]:=
```
(* We can see that our "bad" (red) points go away after applying the gammas. *)
With[{s = 10, m = 15, b = 7, x = 1/2},
 Graphics3D[Table[
   If[(summandTwoGammas[m, s, b, x, k, r, i] /. eps → 0) === 0,
    {}, {If[1 ≤ k ≤ m + s - 1 && 1 ≤ r ≤ s && 0 ≤ i ≤ r - 1 - (k - m), Blue, Red],
     Sphere[{k, r, i}, 0.3]}]
   , {k, -limit, m + s + limit}, {r, -limit, s + limit}, {i, -limit, s + limit}],
  Axes → True, AxesLabel → {"k", "r", "i"}
 ]]
```

Out[•]=



Result and Timings for Two Gamma Insertions (+ Treatment of Singularities)

First we derive the annihilator for the triple sum by applying creative telescoping three times.

In[•]:=
```
Timing[
 cti2 = CreativeTelescoping[
   summandTwoGammas[m, s, b, x, k, r, i], S[i] - 1, {S[r], S[k], S[s]}];
 ctr2 = CreativeTelescoping[cti2[[1]], S[r] - 1];
 ctk2 = FindCreativeTelescoping[ctr2[[1]], {S[k] - 1}];
]
```

Out[•]= {666.56, Null}

```
In[◦]:= (* The result is not too bad: a fourth-order recurrence in s. *)
Factor[ctk2[[1, 1]] /. eps → 0]
(* And this recurrence lies in the ideal generated by the guessed one,
and this completes the proof. *)
OreReduce[ctk2[[1, 1]] /. eps → 0, {guessrecpoly}]
```

Out[◦]=

$$-(2+s)(3+s)(-1+b x) S_s^4 - (2+s)(9+m+4s-5x-6bx-mx-bmx-2sx-3bsx+2bx^2+bmx^2+bsx^2) S_s^3 + (19+5m+21s+3ms+6s^2-21x-7bx-7mx-3bmx-22sx-9bsx-4msx-2bmsx-6s^2x-3bs^2x+4x^2+5bx^2+2mx^2+3bmx^2+4sx^2+6bsx^2+msx^2+2bmsx^2+s^2x^2+2bs^2x^2) S_s^2 - (-1+x)(-8-4m-11s-3ms-4s^2+5x+bx+3mx+bmx+6sx+2bsx+2msx+bmsx+2s^2x+bs^2x) S_s + (1+s)(1+m+s)(-1+x)^2$$

Out[◦]= 0

Next, we certify that the recurrence is correct by showing that the inhomogeneous parts are 0 (expected since we have our natural boundaries). However, there are still singularities in the certificates so it is still a nontrivial (and not yet automated task) to certify.

```
In[◦]:= (* Here are the operators we have to worry about
 in the telescopers and location of singularities. *)
FindSing[var_, ops_] :=
 {#[ops[[1]] /. eps → 0] & /@ {Variables, Support}, Solve[# == 0, var] & /@
   (Denominator /@ (ApplyOreOperator[Flatten[ops[[2]]], c[var]] // Together))}
FindSing[i, cti2]
FindSing[r, ctr2]
FindSing[k, ctk2]
```

Out[◦]= $\{\{\{b, k, m, r, s, x, S[k], S[r], S[s]\}, \{\{S_s, 1\}, \{S_r, S_k, 1\}, \{S_k^2, S_k, 1\}\}\}, \{\{\}, \{\{i \to r\}, \{i \to -k+m+r\}\}, \{\{i \to -1+eps-k+m+r\}\}\}\}$

Out[◦]= $\{\{\{b, k, m, s, x, S[k], S[s]\}, \{\{S_s^2, S_k, S_s, 1\}, \{S_k S_s, S_k, 1\}, \{S_k^2, S_k, S_s, 1\}\}\}, \{\{\{r \to 1+s\}, \{r \to 2+s\}\}, \{\{r \to 1+s\}\}, \{\{r \to 1+k\}, \{r \to 2+k\}, \{r \to 1+s\}\}\}\}$

Out[◦]= $\{\{\{b, m, s, x, S[s]\}, \{\{S_s^4, S_s^3, S_s^2, S_s, 1\}\}\}, \{\{\{k \to -eps\}, \{k \to 1+m+s\}, \{k \to 2+m+s\}, \{k \to 3+m+s\}\}\}\}$

```
(* Here we certify the first CT by confirming that the inhomogeneous
 parts are 0. We only need to check it for the second certificate. *)
deltaparti[certs_, lower_, upper_] :=
 (With[{delta = ApplyOreOperator[certs, S2C[m, s, b, x, k, r, i]]},
    (delta /. {i → (upper + 1)}) - (delta /. {i → lower})]) /. S2C → summandTwoGammas
comppartsi2 = ((ApplyOreOperator[cti2[[1, 2]], S2C[m, s, b, x, k, r, i]] /.
       {i → r - 1 - (k - m)}) +
     ApplyOreOperator[cti2[[1, 2]], S2C[m, s, b, x, k, r, i] /. {i → r - 1 - (k - m)}][[
      2]]) /. S2C → summandTwoGammas;
FullSimplify[(deltaparti[cti2[[2, 2]], 0, r - 2 - (k - m)] - comppartsi2) /. eps → 0] === 0
```

Out[∘]= True

In[∘]:=
```
(* Here we certify the second CT by
 confirming that the inhomogeneous parts are 0. *)
deltapartr[certs_, lower_, upper_] := (With[{delta =
     ApplyOreOperator[certs, Sum[S2C[m, s, b, x, k, r, i], {i, 0, r - 1 - (k - m)}]]},
    (delta /. {r → (upper + 1)}) - (delta /. {r → lower})])
comppartsr[1] = (ApplyOreOperator[ctr2[[1, 1]],
       Sum[S2C[m, s, b, x, k, r, i], {i, 0, r - 1 - (k - m)}]] /. {r → s}) +
    (ApplyOreOperator[ctr2[[1, 1]], Sum[S2C[m, s, b, x, k, r, i],
         {i, 0, r - 1 - (k - m)}]] /. {r → s + 1}) [[3 ;; 4]] +
    (ApplyOreOperator[ctr2[[1, 1]], Sum[S2C[m, s, b, x, k, r, i],
         {i, 0, r - 1 - (k - m)}]] /. {r → s + 2}) [[4]];
comppartsr[2] = (ApplyOreOperator[ctr2[[1, 2]],
       Sum[S2C[m, s, b, x, k, r, i], {i, 0, r - 1 - (k - m)}]] /. {r → s}) +
    (ApplyOreOperator[ctr2[[1, 2]], Sum[S2C[m, s, b, x, k, r, i],
         {i, 0, r - 1 - (k - m)}]] /. {r → s + 1}) [[3]];
comppartsr[3] = (ApplyOreOperator[ctr2[[1, 3]],
       Sum[S2C[m, s, b, x, k, r, i], {i, 0, r - 1 - (k - m)}]] /. {r → s}) +
    (ApplyOreOperator[ctr2[[1, 3]], Sum[S2C[m, s, b, x, k, r, i],
         {i, 0, r - 1 - (k - m)}]] /. {r → s + 1}) [[4]];
(Table[(Simplify[deltapartr[ctr2[[2, i]], 1, s - 1] - comppartsr[i]] /.
          {S2C → summandTwoGammas} /. {eps → 0}) // FunctionExpand //
       FullSimplify, {i, 3}] // Flatten // Union) === {0}
```

Out[∘]= True

Result and Timings for One Gamma Insertion (+ Treatment of Singularities)

First we derive the annihilator for the triple sum by applying creative telescoping three times.

In[∘]:=
```
Timing[
 cti1 = CreativeTelescoping[
   summandOneGamma[m, s, b, x, k, r, i], S[i] - 1, {S[r], S[k], S[s]}];
 ctr1 = CreativeTelescoping[cti1[[1]], S[r] - 1];
 ctk1 = FindCreativeTelescoping[ctr1[[1]], {S[k] - 1}];
]
```

Out[∘]= {26.6068, Null}

In[∘]:=
```
(* The result is exactly the third order recurrence that we guessed. *)
ctk1[[1, 1]] /. eps → 0
-% === guessrecpoly
```

Out[∘]= $\left(2+s-2\,b\,x-b\,s\,x\right) S_s^3 + \left(-5-m-3\,s+3\,x+3\,b\,x+m\,x+b\,m\,x+2\,s\,x+2\,b\,s\,x-b\,x^2-b\,m\,x^2-b\,s\,x^2\right) S_s^2 + \left(4+2\,m+3\,s-5\,x-b\,x-3\,m\,x-b\,m\,x-4\,s\,x-b\,s\,x+x^2+b\,x^2+m\,x^2+b\,m\,x^2+s\,x^2+b\,s\,x^2\right) S_s + \left(-1-m-s+2\,x+2\,m\,x+2\,s\,x-x^2-m\,x^2-s\,x^2\right)$

Out[∘]= True

In[∘]:=
```
(* So of course this recurrence lies in the ideal generated by the guessed one,
and this completes the proof. *)
OreReduce[ctk1[[1, 1]] /. eps → 0, {guessrecpoly}]
```

Out[∘]= 0

Next, we certify that the recurrence is correct by showing that the inhomogeneous parts are 0 (expected since we have our natural boundaries). However, there are still singularities in the certificates so it is still a nontrivial (and not yet automated task) to certify.

In[∘]:=
```
(* Here are the operators we have to worry about
 in the telescopers and location of singularities. *)
FindSing[var_, ops_] :=
 {#[ops[[1]] /. eps → 0] & /@ {Variables, Support}, Solve[# == 0, var] & /@
   (Denominator /@ (ApplyOreOperator[Flatten[ops[[2]]], c[var]] // Together))}
FindSing[i, cti1]
FindSing[r, ctr1]
FindSing[k, ctk1]
```

Out[∘]= $\{\{\{b, k, m, r, s, x, S[k], S[r], S[s]\}, \{\{S_s, 1\}, \{S_r, S_k, 1\}, \{S_k^2, S_k, 1\}\}\}, \{\{\}, \{\{i \to r\}, \{i \to -k+m+r\}\}, \{\{i \to -1+eps-k+m+r\}\}\}\}$

Out[∘]= $\{\{\{b, k, m, s, x, S[k], S[s]\}, \{\{S_s^2, S_k, S_s, 1\}, \{S_k\,S_s, S_k, 1\}, \{S_k^2, S_k, S_s, 1\}\}\}, \{\{\{r \to 1+s\}, \{r \to 2+s\}\}, \{\{r \to 1+s\}\}, \{\{r \to 1+k\}, \{r \to 2+k\}, \{r \to 1+s\}\}\}\}$

Out[∘]= $\{\{\{b, m, s, x, S[s]\}, \{\{S_s^3, S_s^2, S_s, 1\}\}\}, \{\{\{k \to 1+m+s\}, \{k \to 2+m+s\}\}\}\}$

In[∘]:=
```
(* Here we certify the first CT by confirming that the inhomogeneous
 parts are 0. We only need to check it for the second certificate. *)
deltaparti[certs_, lower_, upper_] :=
 (With[{delta = ApplyOreOperator[certs, S1C[m, s, b, x, k, r, i]]},
    (delta /. {i → (upper + 1)}) - (delta /. {i → lower})]) /. S1C → summandOneGamma
comppartsi2 = ((ApplyOreOperator[cti1[[1, 2]], S1C[m, s, b, x, k, r, i]] /.
       {i → r - 1 - (k - m)}) +
      ApplyOreOperator[cti1[[1, 2]], S1C[m, s, b, x, k, r, i] /. {i → r - 1 - (k - m)}][[
       2]]) /. S1C → summandOneGamma;
FullSimplify[(deltaparti[cti1[[2, 2]], 0, r - 2 - (k - m)] - comppartsi2) /. eps → 0] === 0
```

Out[∘]= True

In[∘]:=
```
(* Here we certify the second CT by
 confirming that the inhomogeneous parts are 0. *)
deltapartr[certs_, lower_, upper_] := (With[{delta =
      ApplyOreOperator[certs, Sum[S1C[m, s, b, x, k, r, i], {i, 0, r - 1 - (k - m)}]]},
    (delta /. {r → (upper + 1)}) - (delta /. {r → lower})])
comppartsr[1] = (ApplyOreOperator[ctr1[[1, 1]],
       Sum[S1C[m, s, b, x, k, r, i], {i, 0, r - 1 - (k - m)}]] /. {r → s}) +
    (ApplyOreOperator[ctr1[[1, 1]], Sum[S1C[m, s, b, x, k, r, i],
         {i, 0, r - 1 - (k - m)}]] /. {r → s + 1}) [[3 ;; 4]] +
    (ApplyOreOperator[ctr1[[1, 1]], Sum[S1C[m, s, b, x, k, r, i],
         {i, 0, r - 1 - (k - m)}]] /. {r → s + 2}) [[4]];
comppartsr[2] = (ApplyOreOperator[ctr1[[1, 2]],
       Sum[S1C[m, s, b, x, k, r, i], {i, 0, r - 1 - (k - m)}]] /. {r → s}) +
    (ApplyOreOperator[ctr1[[1, 2]], Sum[S1C[m, s, b, x, k, r, i],
         {i, 0, r - 1 - (k - m)}]] /. {r → s + 1}) [[3]];
comppartsr[3] = (ApplyOreOperator[ctr1[[1, 3]],
       Sum[S1C[m, s, b, x, k, r, i], {i, 0, r - 1 - (k - m)}]] /. {r → s}) +
    (ApplyOreOperator[ctr1[[1, 3]], Sum[S1C[m, s, b, x, k, r, i],
         {i, 0, r - 1 - (k - m)}]] /. {r → s + 1}) [[4]];
(Table[(Simplify[deltapartr[ctr1[[2, i]], 1, s - 1] - comppartsr[i]] /.
          {S1C → summandOneGamma} /. {eps → 0}) // FunctionExpand //
      FullSimplify, {i, 3}] // Flatten // Union) === {0}
```

Out[∘]= True

Export to PDF

In[5]:=
```
SetDirectory[NotebookDirectory[]];
Export["sums_casc2020.pdf", Get["sums_casc2020.nb"] /. ButtonBox[__] → " Help"];
```