Compute $\Gamma^a{}_{bcd}$ directly

Here we will compute one of generalised connections directly from the connection. That is, we will compute

$$\Gamma^a{}_{bcd} = \Gamma^a{}_{(bc,d)} - 2\Gamma^a{}_{p(c}\Gamma^p{}_{bd)} \tag{1}$$

given an explict expression for the RNC connection $\Gamma^a{}_{bc}$.

This code was written as a check for the `genGamma.tex` code. I had a discrepency between my newly created Cadabra v2.0 codes and my old Cadabra v1.0 codes (which were the basis of my lcb09-03 paper). I found that my new codes agreed with this code and thus my old codes were wrong. I found the errors in the old code (see the updated codes in v1.0/rnc-new/gen-gamma.cdbp, see also the file v1.0/rnc-new/NOTES.txt).

The head-on approach shown in this code works for this simple compuation but for the higher generalised connections it's almost certainly going to be way too slow to be use. So this code is only a check of the results from `genGamma.tex` code. The good news is – they agree.

```
{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w#}::Indices(position=independent).

D{#}::Derivative.
\nabla{#}::Derivative.
\partial{#}::PartialDerivative.

g_{a b}::Metric.
g^{a b}::InverseMetric.
g_{a}^{b}::KroneckerDelta.
g^{a}_{b}::KroneckerDelta.
\delta^{a}_{b}::KroneckerDelta.
\delta_{a}^{b}::KroneckerDelta.

R_{a b c d}::RiemannTensor.
R_{a b c d}::Depends(\nabla{#}).

x^{a}::Depends(\partial{#}).

\Gamma^{a}_{b c}::Depends(\partial{#}).
\Gamma^{a}_{b c}::TableauSymmetry(shape={2}, indices={1,2}).

Q_{a b c d}::Weight(label=numR,value=2).
Q_{a b c d e}::Weight(label=numR,value=3).
Q_{a b c d e f}::Weight(label=numR,value=4).
Q_{a b c d e f g}::Weight(label=numR,value=5).

def product_sort (obj):
    substitute (obj,$ A^{a}                              -> A001^{a}              $)
    substitute (obj,$ x^{a}                              -> A002^{a}              $)
    substitute (obj,$ g^{a b}                            -> A003^{a b}            $)
    substitute (obj,$ R_{a b c d}                        -> A004_{a b c d}        $)
    substitute (obj,$ \nabla_{e}{R_{a b c d}}            -> A005_{a b c d e}      $)
    substitute (obj,$ \nabla_{e f}{R_{a b c d}}          -> A006_{a b c d e f}    $)
    substitute (obj,$ \nabla_{e f g}{R_{a b c d}}        -> A007_{a b c d e f g}  $)
    substitute (obj,$ \nabla_{e f g h}{R_{a b c d}}      -> A008_{a b c d e f g h} $)
    sort_product   (obj)
    rename_dummies (obj)
```

```
    substitute (obj,$ A001^{a}                  -> A^{a}                          $)
    substitute (obj,$ A002^{a}                  -> x^{a}                          $)
    substitute (obj,$ A003^{a b}                -> g^{a b}                        $)
    substitute (obj,$ A004_{a b c d}            -> R_{a b c d}                    $)
    substitute (obj,$ A005_{a b c d e}          -> \nabla_{e}{R_{a b c d}}        $)
    substitute (obj,$ A006_{a b c d e f}        -> \nabla_{e f}{R_{a b c d}}      $)
    substitute (obj,$ A007_{a b c d e f g}      -> \nabla_{e f g}{R_{a b c d}}    $)
    substitute (obj,$ A008_{a b c d e f g h}    -> \nabla_{e f g h}{R_{a b c d}} $)

    return obj

def truncate (obj,n):

# I would like to assign different weights to \nabla_{a}, \nabla_{a b}, \nabla_{a b c} etc. but no matter
# what I do it appears that Cadabra assigns the same weight to all of these regardless of the number of subscripts.
# It seems that the weight is assigned to the symbol \nabla alone. So I'm forced to use the following substitution trick.

    tmp := @(obj).

    substitute (tmp, $\nabla_{e f g}{R_{a b c d}} -> Q_{a b c d e f g}$)
    substitute (tmp, $\nabla_{e f}{R_{a b c d}} -> Q_{a b c d e f}$)
    substitute (tmp, $\nabla_{e}{R_{a b c d}} -> Q_{a b c d e}$)
    substitute (tmp, $R_{a b c d} -> Q_{a b c d}$)

    ans = Ex(0)

    for i in range (0,n+1):
       foo := @(tmp).
       bah = Ex("numR = " + str(i))
       keep_weight (foo, bah)
       ans = ans + foo

    substitute (ans, $Q_{a b c d e f g} -> \nabla_{e f g}{R_{a b c d}}$)
    substitute (ans, $Q_{a b c d e f} -> \nabla_{e f}{R_{a b c d}}$)
    substitute (ans, $Q_{a b c d e} -> \nabla_{e}{R_{a b c d}}$)
    substitute (ans, $Q_{a b c d} -> R_{a b c d}$)

    return ans
```

```
def get_term (obj,n):

    bah := @(obj).
    distribute (bah)

    substitute (bah, $\nabla_{e f g}{R_{a b c d}} -> Q_{a b c d e f g}$)
    substitute (bah, $\nabla_{e f}{R_{a b c d}} -> Q_{a b c d e f}$)
    substitute (bah, $\nabla_{e}{R_{a b c d}} -> Q_{a b c d e}$)
    substitute (bah, $R_{a b c d} -> Q_{a b c d}$)

    foo = Ex("numR = " + str(n))
    keep_weight (bah, foo)

    substitute (bah, $Q_{a b c d e f g} -> \nabla_{e f g}{R_{a b c d}}$)
    substitute (bah, $Q_{a b c d e f} -> \nabla_{e f}{R_{a b c d}}$)
    substitute (bah, $Q_{a b c d e} -> \nabla_{e}{R_{a b c d}}$)
    substitute (bah, $Q_{a b c d} -> R_{a b c d}$)

    return bah

def tidy (obj,number):
   bah  = Ex(str(number))
   tmp := @(bah) @(obj).
   distribute  (tmp)
   factor_out  (tmp,$A^{a?},x^{b?}$)
   ans := @(tmp) / @(bah).
   return ans

import cdblib

Gamma = cdblib.get ('Gamma','../connection.json')

defGamma := \Gamma^{d}_{a b} -> @(Gamma).

genGam := A^{d} A^{b} A^{c} (\partial_{d}{\Gamma^{a}_{b c}} - 2 \Gamma^{a}_{p c} \Gamma^{p}_{b d}).

substitute     (genGam,defGamma)   # cdb (genGam.001,genGam)
```

```
distribute     (genGam)
unwrap         (genGam)            # cdb (genGam.002,genGam)
distribute     (genGam)            # cdb (genGam.003,genGam)
product_rule   (genGam)            # cdb (genGam.004,genGam)
distribute     (genGam)            # cdb (genGam.005,genGam)
substitute     (genGam,$\partial_{a}{x^{b}}->\delta_{a}^{b}$)   # cdb (genGam.006,genGam)
eliminate_kronecker (genGam)       # cdb (genGam.007,genGam)
genGam = truncate     (genGam,5)   # cdb (genGam.008,genGam)
genGam = product_sort (genGam)     # cdb (genGam.009,genGam)
rename_dummies (genGam)            # cdb (genGam.010,genGam)
canonicalise   (genGam)            # cdb (genGam.011,genGam)

# all done, now for some housekeeping

term3 = get_term (genGam,3)   # cdb (term3.001,term3)
term4 = get_term (genGam,4)   # cdb (term4.001,term4)
term5 = get_term (genGam,5)   # cdb (term5.001,term5)

term3 = tidy (term3,2)
term4 = tidy (term4,120)
term5 = tidy (term5,180)

genGam := @(term3) + @(term4) + @(term5).

scaledGamma := 360 @(genGam).  # cdb (scaledGamma.001,scaledGamma)
```

$$\texttt{term3.001} := \frac{1}{2} A^b A^c A^d x^e g^{af} \nabla_b R_{cedf}$$

$$\begin{aligned}\texttt{term4.001} := {} & \frac{8}{15} A^b A^c A^d x^e x^f g^{ag} g^{hi} R_{bech} R_{dgfi} + \frac{2}{15} A^b A^c A^d x^e x^f g^{ag} g^{hi} R_{bech} R_{difg} - \frac{2}{15} A^b A^c A^d x^e x^f g^{ag} g^{hi} R_{befh} R_{cgdi} \\ & + \frac{1}{10} A^b A^c A^d x^e x^f g^{ag} \nabla_{bc} R_{defg} + \frac{3}{20} A^b A^c A^d x^e x^f g^{ag} \nabla_{be} R_{cfdg} + \frac{3}{20} A^b A^c A^d x^e x^f g^{ag} \nabla_{eb} R_{cfdg} \\ & + \frac{2}{5} A^b A^c A^d x^e x^f g^{ag} g^{hi} R_{bech} R_{dfgi} + \frac{1}{40} A^b A^c A^d x^e x^f g^{ag} \nabla_{gb} R_{cedf} + \frac{1}{40} A^b A^c A^d x^e x^f g^{ag} \nabla_{bg} R_{cedf}\end{aligned}$$

$$\begin{aligned}\texttt{term5.001} := {} & \frac{8}{45} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{beci} \nabla_d R_{fhgj} + \frac{4}{15} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{beci} \nabla_f R_{dhgj} + \frac{1}{15} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{beci} \nabla_f R_{djgh} \\ & + \frac{1}{10} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{bhei} \nabla_c R_{dfgj} + \frac{1}{90} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{bieh} \nabla_c R_{dfgj} + \frac{11}{90} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{ehfi} \nabla_b R_{cgdj} \\ & + \frac{4}{15} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{bhei} \nabla_f R_{cgdj} + \frac{1}{15} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{bieh} \nabla_f R_{cgdj} + \frac{1}{12} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{bhei} \nabla_j R_{cfdg} \\ & + \frac{1}{36} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{bieh} \nabla_j R_{cfdg} - \frac{1}{15} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{bhci} \nabla_e R_{dfgj} - \frac{1}{15} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{befi} \nabla_c R_{dhgj} \\ & - \frac{2}{45} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{befi} \nabla_c R_{djgh} - \frac{1}{15} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{befi} \nabla_g R_{chdj} + \frac{1}{45} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{bce} R_{dfgh} \\ & + \frac{1}{45} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{bec} R_{dfgh} + \frac{1}{30} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{bef} R_{cgdh} + \frac{1}{45} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{ebc} R_{dfgh} + \frac{1}{30} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{ebf} R_{cgdh} \\ & + \frac{1}{30} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{efb} R_{cgdh} + \frac{4}{45} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{behi} \nabla_c R_{dfgj} + \frac{1}{5} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{behi} \nabla_f R_{cgdj} \\ & + \frac{4}{45} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{beci} \nabla_h R_{dfgj} - \frac{1}{45} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{befi} \nabla_h R_{cgdj} + \frac{1}{5} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{beci} \nabla_f R_{dghj} \\ & - \frac{1}{45} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{befi} \nabla_c R_{dghj} + \frac{1}{180} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{hbe} R_{cfdg} + \frac{1}{180} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{heb} R_{cfdg} \\ & + \frac{1}{180} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{bhe} R_{cfdg} + \frac{1}{180} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{ehb} R_{cfdg} + \frac{1}{180} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{beh} R_{cfdg} \\ & + \frac{1}{180} A^b A^c A^d x^e x^f x^g g^{ah} \nabla_{ebh} R_{cfdg} + \frac{1}{18} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{behi} \nabla_j R_{cfdg} - \frac{1}{9} A^b A^c A^d x^e x^f x^g g^{ah} g^{ij} R_{beci} \nabla_j R_{dfgh}\end{aligned}$$

Summary

$$
\begin{aligned}
360A^bA^cA^d\Gamma^a_{bcd} = {} & 180A^bA^cA^dx^eg^{af}\nabla_bR_{cedf} + 3A^bA^cA^dx^ex^f\left(64g^{ag}g^{hi}R_{bech}R_{dgfi} + 16g^{ag}g^{hi}R_{bech}R_{difg} - 16g^{ag}g^{hi}R_{befh}R_{cgdi} + 12g^{ag}\nabla_{bc}R_{defg}\right. \\
& + 18g^{ag}\nabla_{be}R_{cfdg} + 18g^{ag}\nabla_{eb}R_{cfdg} + 48g^{ag}g^{hi}R_{bech}R_{dfgi} + 3g^{ag}\nabla_{gb}R_{cedf} + 3g^{ag}\nabla_{bg}R_{cedf}\left.\right) + 2A^bA^cA^dx^ex^fx^g\left(32g^{ah}g^{ij}R_{beci}\nabla_dR_{fhgj}\right. \\
& + 48g^{ah}g^{ij}R_{beci}\nabla_fR_{dhgj} + 12g^{ah}g^{ij}R_{beci}\nabla_fR_{djgh} + 18g^{ah}g^{ij}R_{bhei}\nabla_cR_{dfgj} + 2g^{ah}g^{ij}R_{bieh}\nabla_cR_{dfgj} + 22g^{ah}g^{ij}R_{ehfi}\nabla_bR_{cgdj} \\
& + 48g^{ah}g^{ij}R_{bhei}\nabla_fR_{cgdj} + 12g^{ah}g^{ij}R_{bieh}\nabla_fR_{cgdj} + 15g^{ah}g^{ij}R_{bhei}\nabla_jR_{cfdg} + 5g^{ah}g^{ij}R_{bieh}\nabla_jR_{cfdg} - 12g^{ah}g^{ij}R_{bhci}\nabla_eR_{dfgj} \\
& - 12g^{ah}g^{ij}R_{befi}\nabla_cR_{dhgj} - 8g^{ah}g^{ij}R_{befi}\nabla_cR_{djgh} - 12g^{ah}g^{ij}R_{befi}\nabla_gR_{chdj} + 4g^{ah}\nabla_{bce}R_{dfgh} + 4g^{ah}\nabla_{bec}R_{dfgh} + 6g^{ah}\nabla_{bef}R_{cgdh} \\
& + 4g^{ah}\nabla_{ebc}R_{dfgh} + 6g^{ah}\nabla_{ebf}R_{cgdh} + 6g^{ah}\nabla_{efb}R_{cgdh} + 16g^{ah}g^{ij}R_{behi}\nabla_cR_{dfgj} + 36g^{ah}g^{ij}R_{behi}\nabla_fR_{cgdj} + 16g^{ah}g^{ij}R_{beci}\nabla_hR_{dfgj} \\
& - 4g^{ah}g^{ij}R_{befi}\nabla_hR_{cgdj} + 36g^{ah}g^{ij}R_{beci}\nabla_fR_{dghj} - 4g^{ah}g^{ij}R_{befi}\nabla_cR_{dghj} + g^{ah}\nabla_{hbe}R_{cfdg} + g^{ah}\nabla_{heb}R_{cfdg} + g^{ah}\nabla_{bhe}R_{cfdg} \\
& \left. + g^{ah}\nabla_{ehb}R_{cfdg} + g^{ah}\nabla_{beh}R_{cfdg} + g^{ah}\nabla_{ebh}R_{cfdg} + 10g^{ah}g^{ij}R_{behi}\nabla_jR_{cfdg} - 20g^{ah}g^{ij}R_{beci}\nabla_jR_{dfgh}\right)
\end{aligned}
$$