

Example 11 The RNC connection.

```
1 {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,u#}::Indices(position=independent).
2
3 D{#}::PartialDerivative.
4 \nabla{#}::Derivative.
5
6 g_{a b}::Metric.
7 g^{a b}::InverseMetric.
8 g^{a b}::Weight(label=gnum,value=1).
9
10 \delta{#}::KroneckerDelta.
11
12 R_{a b c d}::RiemannTensor.
13 R_{a b c d}::Depends(\nabla{#}).
14
15 x^{a}::Depends(D{#}).
16 x^{a}::Weight(label=xnum,value=1).
17
18 Dx := D_{a}{x^{b}} -> \delta^{b}_{a}. # cdb (ex-11.000,Dx)
19
20 gab := g_{a b} -> g_{a b}
21 - (1/3) x^{c} x^{d} R_{a c b d}
22 - (1/6) x^{c} x^{d} x^{e} \nabla_{c}{R_{a d b e}}
23 + (1/180) x^{c} x^{d} x^{e} x^{f} ( 8 g^{g h} R_{a c d g} R_{b e f h}
24 - 9 \nabla_{c d}{R_{a e b f}} ). # cdb (ex-11.001,gab)
25
26 iab := g^{a b} -> g^{a b}
27 + (1/3) x^{c} x^{d} g^{a e} g^{b f} R_{c e d f}
28 + (1/6) x^{c} x^{d} x^{e} g^{a f} g^{b g} \nabla_{c}{R_{d f e g}}
29 + (1/60) x^{c} x^{d} x^{e} x^{f} g^{a g} g^{b h}
30 ( 4 g^{i j} R_{c g d i} R_{e h f j}
31 + 3 \nabla_{c d}{R_{e g f h}} ). # cdb(ex-11.002,iab)
32
33 distribute (gab)
34 distribute (iab)
35
36 ChrSym := \Gamma^{a}_{b c} -> 1/2 g^{a d} ( D_{b}{g_{d c}}
```

```

37         + D_{c}{g_{b d}}
38         - D_{d}{g_{b c}} ).    # cdb (ex-11.003,ChrSym)
39
40 Gamma := \Gamma^{a}_{b c}.      # cdb (ex-11.100,Gamma)
41
42 substitute      (Gamma,ChrSym)  # cdb (ex-11.101,Gamma)
43 substitute      (Gamma,gab)    # cdb (ex-11.102,Gamma)
44 substitute      (Gamma,iab)    # cdb (ex-11.103,Gamma)
45 distribute      (Gamma)         # cdb (ex-11.104,Gamma)
46 unwrap         (Gamma)         # cdb (ex-11.105,Gamma)
47 product_rule    (Gamma)         # cdb (ex-11.106,Gamma)
48 distribute      (Gamma)         # cdb (ex-11.107,Gamma)
49 substitute      (Gamma,Dx)     # cdb (ex-11.108,Gamma)
50 eliminate_kronecker (Gamma)    # cdb (ex-11.109,Gamma)
51
52 def truncate (obj,n):
53
54     ans = Ex("0") # create a Cadabra object with value zero
55
56     for i in range (0,n+1):
57         foo := @obj.
58         bah = Ex("xnum = " + str(i))
59         distribute (foo)
60         keep_weight (foo, bah)
61         ans = ans + foo
62
63     return ans
64
65 checkpoint.append (Gamma)
66
67 # sort_product      (Gamma) # 52.3 sec, 49 Mbyte
68 # rename_dummies    (Gamma) # 58.6 sec, 51 Mbyte
69 # canonicalise      (Gamma) # killed after 20 mins and over 500 Mbyte
70
71 Gamma = truncate (Gamma,3) # cdb (ex-11.110,Gamma) # allow up to 3rd order in x^a
72
73 sort_product      (Gamma)
74 rename_dummies    (Gamma)

```

```

75 canonicalise (Gamma)
76
77 checkpoint.append (Gamma)
78
79 # =====
80 # the remaining code is just for pretty printing
81
82 def product_sort (obj):
83     substitute (obj,$ g^{a b}          -> A001^{a b}          $)
84     substitute (obj,$ x^{a}            -> A002^{a}          $)
85     substitute (obj,$ z^{a}            -> A003^{a}          $)
86     substitute (obj,$ R_{a b c d}       -> A004_{a b c d}     $)
87     substitute (obj,$ \nabla_{e}\{R_{a b c d}\} -> A005_{a b c d e}     $)
88     substitute (obj,$ \nabla_{e f}\{R_{a b c d}\} -> A006_{a b c d e f}   $)
89     sort_sum      (obj)
90     sort_product  (obj)
91     rename_dummies (obj)
92     substitute (obj,$ A001^{a b}          -> g^{a b}          $)
93     substitute (obj,$ A002^{a}            -> x^{a}            $)
94     substitute (obj,$ A003^{a}            -> z^{a}            $)
95     substitute (obj,$ A004_{a b c d}       -> R_{a b c d}       $)
96     substitute (obj,$ A005_{a b c d e}     -> \nabla_{e}\{R_{a b c d}\} $)
97     substitute (obj,$ A006_{a b c d e f}   -> \nabla_{e f}\{R_{a b c d}\} $)
98
99 def get_xterm (obj,n):
100
101     foo := @(obj).
102     bah = Ex("xnum = " + str(n))
103     distribute (foo)
104     keep_weight (foo, bah)
105
106     return foo
107
108 def get_gterm (obj,n):
109
110     foo := @(obj).
111     bah = Ex("gnum = " + str(n))
112     distribute (foo)

```

```

113     keep_weight (foo, bah)
114
115     return foo
116
117 def reformat (obj,scale):
118
119     foo = Ex(str(scale))
120     bah := @(foo) @(obj).
121
122     distribute      (bah)
123     product_sort    (bah)
124     rename_dummies  (bah)
125     canonicalise     (bah)
126     factor_out      (bah,$x^{a?},g^{b? c?}$)
127     ans := @(bah) / @(foo).
128
129     return ans
130
131 gam1 = get_xterm (Gamma, 1)           # cdb (ex-11.200,gam1)
132 gam2 = get_xterm (Gamma, 2)           # cdb (ex-11.201,gam2)
133 gam3 = get_xterm (Gamma, 3)           # cdb (ex-11.202,gam3)
134
135 gam31 = get_gterm (gam3, 1)           # cdb (ex-11.210,gam31)
136 gam32 = get_gterm (gam3, 2)           # cdb (ex-11.211,gam31)
137
138 gam1 = reformat (gam1, 3)             # cdb (ex-11.220,gam1)
139 gam2 = reformat (gam2, 12)            # cdb (ex-11.221,gam2)
140
141 gam31 = reformat (gam31, 40)          # cdb (ex-11.222,gam31)
142 gam32 = reformat (gam32, 45)          # cdb (ex-11.223,gam32)
143
144 Gamma := @(gam1) + @(gam2) + @(gam31) + @(gam32). # cdb (ex-11.230,Gamma)
145 Scaled := 360 @(Gamma).               # cdb (ex-11.231,Scaled)
146
147 checkpoint.append (Gamma)

```

$$\begin{aligned}
\Gamma^a{}_{bc}(x) = & \frac{1}{3}g^{ad}x^e(R_{bdce} + R_{becd}) + \frac{1}{12}g^{ad}x^ex^f(-\nabla_c R_{bedf} + \nabla_d R_{becf} + 2\nabla_e R_{bdcf} + 2\nabla_e R_{bfcd} - \nabla_b R_{cedf}) \\
& + \frac{1}{40}g^{ad}x^ex^fx^g(-\nabla_{ce}R_{bfdg} - \nabla_{ec}R_{bfdg} + \nabla_{de}R_{bfcg} + \nabla_{ed}R_{bfcg} + 2\nabla_{ef}R_{bdcg} + 2\nabla_{ef}R_{bgcd} - \nabla_{be}R_{cfdg} - \nabla_{eb}R_{cfdg}) \\
& + \frac{1}{45}g^{ad}g^{ef}x^gx^hx^i(4R_{becg}R_{dhfi} + 4R_{bgce}R_{dhfi} - 2R_{bdeg}R_{chfi} - R_{bedg}R_{chfi} + R_{bgde}R_{chfi} - 2R_{bgeh}R_{cdfi} - R_{bgeh}R_{cfdi} \\
& \hspace{15em} + R_{bgeh}R_{cidf}) \quad (\text{ex-11.230})
\end{aligned}$$

$$\begin{aligned}
360\Gamma^a{}_{bc}(x) = & 120g^{ad}x^e(R_{bdce} + R_{becd}) + 30g^{ad}x^ex^f(-\nabla_c R_{bedf} + \nabla_d R_{becf} + 2\nabla_e R_{bdcf} + 2\nabla_e R_{bfcd} - \nabla_b R_{cedf}) \\
& + 9g^{ad}x^ex^fx^g(-\nabla_{ce}R_{bfdg} - \nabla_{ec}R_{bfdg} + \nabla_{de}R_{bfcg} + \nabla_{ed}R_{bfcg} + 2\nabla_{ef}R_{bdcg} + 2\nabla_{ef}R_{bgcd} - \nabla_{be}R_{cfdg} - \nabla_{eb}R_{cfdg}) \\
& + 8g^{ad}g^{ef}x^gx^hx^i(4R_{becg}R_{dhfi} + 4R_{bgce}R_{dhfi} - 2R_{bdeg}R_{chfi} - R_{bedg}R_{chfi} + R_{bgde}R_{chfi} - 2R_{bgeh}R_{cdfi} - R_{bgeh}R_{cfdi} \\
& \hspace{15em} + R_{bgeh}R_{cidf}) \quad (\text{ex-11.231})
\end{aligned}$$

```
1 jsonfile = 'example-11.json'
2 cdblib.create (jsonfile)
3 cdblib.put ('Gamma',Gamma,jsonfile)
```