

## Exercise 4.1 Differentiate a polynomial – a limited method

```
1 {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).
2
3 def deriv (poly):
4
5     \delta^{a}::Weight(label=\epsilon).
6
7     bah := @(poly).
8
9     substitute      (bah,$x^{a} -> x^{a} + \delta^{a}$)
10    distribute      (bah)
11
12    foo := @(bah) - @(poly).
13
14    keep_weight      (foo, $\epsilon = 1$)
15    sort_product     (foo)
16    rename_dummies   (foo)
17    factor_out        (foo, $\delta^{a?}$)
18    substitute        (foo, $\delta^{a} -> 1$)
19
20    return foo
21
22 # -----
23
24 poly := c^{a}
25       + c^{a}_{b} x^b
26       + c^{a}_{b c} x^b x^c.    # cdb (ex-0401.100,poly)
27
28 dpoly = deriv (poly)           # cdb (ex-0401.101,dpoly)
```

$$p = c^a + c^a_b x^b + c^a_{bc} x^b x^c \quad (\text{ex-0401.100})$$

$$dp = c^a_b + c^a_{cb} x^c + c^a_{bc} x^c \quad (\text{ex-0401.101})$$

## Exercise 4.1 Differentiate a polynomial – a better method

```

1 {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).
2
3 def deriv (poly):
4
5     \partial{#}::PartialDerivative.
6     \delta^{a}_{b}::KroneckerDelta.
7
8     x^{a}::Depends(\partial{#}).
9
10    bah := \partial_{b}{@(poly)}.
11
12    distribute      (bah)
13    unwrap          (bah)  # drop all terms that don't explicitly depend on a derivative operator
14    product_rule    (bah)
15    distribute      (bah)
16    substitute      (bah,$\partial_{b}{x^{a}}->\delta^{a}_{b}$)
17    eliminate_kronecker (bah)
18
19    sort_product    (bah)
20    rename_dummies  (bah)
21
22    return bah
23
24 poly := c^{a}
25       + c^{a}_{b} x^{b}
26       + c^{a}_{b c} x^{b} x^{c}.    # cdb (ex-0401.200,poly)
27
28 dpoly = deriv (poly)                # cdb (ex-0401.201,dpoly)

```

$$p = c^a + c^a_b x^b + c^a_{bc} x^b x^c \quad (\text{ex-0401.200})$$

$$dp = c^a_b + c^a_{bc} x^c + c^a_{cb} x^c \quad (\text{ex-0401.201})$$