In[∘]:= **Needs["TensorBases`"]**

Mathematica package **TensorBases** loaded
Authors: Andreas Geißel, Franz Richard Sattler
Version: 1.0
Year: 2024

For a list of available bases, call TBInfo[]. For further information on a particular basis, call TBInfo["BasisName"].

This package provides the methods TBGetBasisElement, TBGetInnerProduct, TBGetMetric, TBGetInverseMetric, TBGetProjector for every tensor basis available.
For closer explanations, please call their usage messages, e.g. TBGetProjector::usage.

FormTracer package loaded.

To see all (user-defined and package-defined) FormTracer definitions, call TBInfo["FormTracer"].
Furthermore, **TensorBases** extends **FormTracer**. To see all extensions, call TBInfo["Extensions"]

Lorentz group undefined, using default names.

Group with name color undefined, using default names.

Group with name flavor undefined, using default names.

In[∘]:= **TBInfo::usage**
TBInfo[]
TBInfo["transAqbq"]

Out[∘]= TBInfo[_String]
Return information on a given object.
TBInfo[] prints all available bases with some usage information.
TBInfo[BasisName] prints detailed information provided by this basis.
TBInfo["FormTracer"] prints all defined groups and identites which **FormTracer** currently knows.
TBInfo["Extensions"] prints all extensions to **FormTracer**, defined by the **TensorBases** package.

Name	Vertex	Indices	Inner product	Comment	Author
FierzComplete-Nf2Nc3Pheno	$\bar{q}q\bar{q}q$	p1 d1 A1 F1 p2 d2 A2 F2 p3 d3 A3 F3 p4 d4 A4 F4	2 Tensor1[1,2,3,4] Tensor2[2,1,4,3] -2 Tensor1[1,2,3,4] Tensor2[2,3,4,1]	Fierz-complete, phenomenologically inspired basis for N_f=2, N_c=3	FR Sattler
transAqbq	$A\bar{q}q$	p1 mu a1 p2 d2 A2 F2 p3 d3 A3 F3	Tensor1[1,2,3] Tensor2[1,3,2]	Transversal quark-gluon vertex basis	FR Sattler

Indices:

A: $\{p_1, \mu, a_1\}$

$\bar{q}$: $\{p_2, d_2, A_2, F_2\}$

q: $\{p_3, d_3, A_3, F_3\}$

We use the general form

$\mathcal{L} = (2\pi)^d \delta(p_1+p_2+p_3)\delta_{F_2\,F_3} T^{a_1}[\tau_i]_{\mu_1}$

and the τ_i are listed in the following:

	Tensor
1	$\Pi^{\perp}{}_{\mu\nu}(p_1)\cdot i\,\gamma_\nu$
2	$\Pi^{\perp}{}_{\mu\nu}(p_1)\cdot(p_2-p_3)_\nu$
3	$\Pi^{\perp}{}_{\mu\nu}(p_1)\cdot i\,\sigma_{\nu\rho}(p_2-p_3)_\rho$
4	$i\,\sigma_{\mu\nu}(p_1)_\nu$
5	$i(p_1)_\mu(p_1)_\nu\gamma_\nu$
6	$\Pi^{\perp}{}_{\mu\nu}(p_1, p_2-p_3)\cdot i\,\gamma_\nu\;\; p_1\cdot(p_2-p_3)\;-\;(p1\cdot p3-p1\cdot p2)\cdot\Pi^{\perp}{}_{\mu\nu}(p_1)\cdot i\,\gamma_\nu$
7	$\frac{1}{3}\{\sigma_{\alpha\beta}\gamma_\mu+\sigma_{\beta\mu}\gamma_\alpha+\sigma_{\mu\alpha}\gamma_\beta\}$
8	$\Pi^{\perp}{}_{\mu\nu}(p_1, p_2-p_3)\cdot p_1\cdot(p_2-p_3)\,i\,\sigma_{\nu\rho}(p_1)_\rho$

In[∘]:=
```
TBGetBasisElement::usage
TBGetBasisElement["transAqbq", 1, {p1, mu, a}, {p2, d2, A2, F2}, {p3, d3, A3, F3}]
```

Out[∘]=
```
TBGetBasisElement[BasisName_String,n_Integer,indices__]
Obtains the n-th element of the specified basis. The given
  indices must match the ones specified by the basis, see TBInfo[].
```

Out[∘]=
```
i deltaFundFlav[F2, F3] × gamma[nu$9417, d2, d3] × TCol[a, A2, A3] × transProj[p1, mu, nu$9417]
```

In[∘]:=
```
TBGetInnerProduct::usage
TBGetInnerProduct["transAqbq"][TBGetBasisElement, 1, TBGetBasisElement, 1] // FormTrace //
 Simplify
```

Out[∘]=
```
TBGetInnerProduct[BasisName_String]
Returns the bilinear operator O that
  represents the inner product of the specified basis.
It can be called as O[Tensor1, n, Tensor2, m], where Tensor1 and Tensor2 are
  functions with signatures Tensor[BasisName_String, n_Integer, indices__].
For example, O[TBGetBasisElement, 1, TBGetBasisElement, 1] returns <e_i,e_j>.
```

Out[∘]= $-6\left(-1+Nc^2\right)Nf$

In[∘]:=
```
TBGetMetric::usage
TBGetMetric["transAqbq"][[1, 1]]
```

Out[∘]=
```
TBGetMetric[BasisName_String]
Returns the metric of the specified basis, i.e. the matrix
  g_ij = <e_i,e_j>, where the e_i are the basis elements of the basis.
```

Out[∘]= $-6\left(-1+Nc^2\right)Nf$

In[∘]:=
```
TBGetInverseMetric::usage
TBGetInverseMetric["transAqbq"][[1, 1]]
```

Out[∘]=
```
TBGetInverseMetric[BasisName_String]
Returns the inverse of the metric of the specified basis, i.e. the matrix
  g_ij^-1 = (<e_i,e_j>)^-1, where the e_i are the basis elements of the basis.
```

Out[∘]= $\dfrac{1}{6\,Nf-6\,Nc^2\,Nf}$

In[∘]:=
```
TBGetProjector::usage
TBGetProjector["transAqbq", 1, {p1, mu, a}, {p2, d2, A2, F2}, {p3, d3, A3, F3}]
TBGetInnerProduct["transAqbq"][TBGetProjector, 1, TBGetBasisElement, 1] // FormTrace //
 Simplify
TBGetInnerProduct["transAqbq"][TBGetProjector, 1, TBGetBasisElement, 8] // FormTrace //
 Simplify
```

Out[∘]=
```
TBGetBasisProjector[BasisName_String,n_Integer,indices__]
Returns the n-th projector, which is defined by g_nj^-1 e_j.
```

Out[∘]= $\dfrac{i\,\text{deltaFundFlav[F2, F3]}\times\text{gamma[nu\$11898, d2, d3]}\times\text{TCol[a, A2, A3]}\times\text{transProj[p1, mu, nu\$11898]}}{6\,Nf-6\,Nc^2\,Nf}$

Out[∘]= 1

Out[∘]= 0