ETH zürich

Distributing MultiLoop results via UFO

Just an Idea

Valentin Hirschi

Les Houches

9th June 2017

Universal FeynRules Output

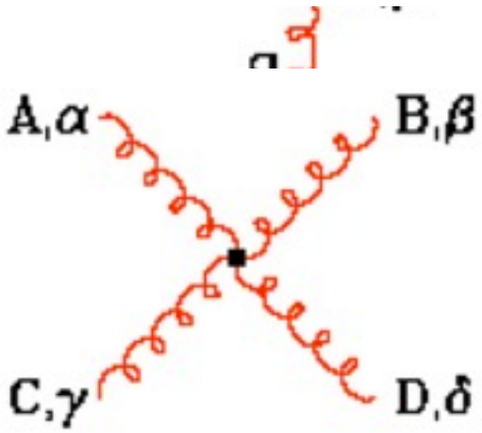
Reminder

[arXiv:1108.2040]

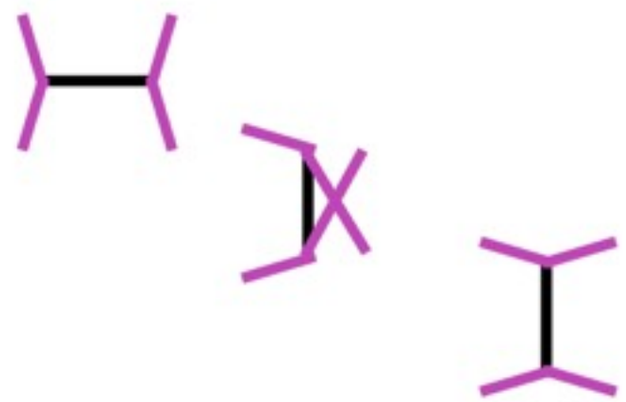
- Python module to specify a model for MC generators

vertices.py couplings.py parameters.py lorentz.py

```
V_37 = Vertex(name = 'V_37',
              particles = [ P.g, P.g, P.g, P.g ],
              color = [ 'f(-1,1,2)*f(3,4,-1)',
                        'f(-1,1,3)*f(2,4,-1)',
                        'f(-1,1,4)*f(2,3,-1)' ],
lorentz = [ L.VVVV1, L.VVVV3, L.VVVV4 ],
couplings = {(1,1):C.GC_12,
             (0,0):C.GC_12,
             (2,2):C.GC_12})
```



$$-ig^2 f^{XAC} f^{XBD} [g^{\alpha\beta} g^{\gamma\delta} - g^{\alpha\delta} g^{\beta\gamma}]$$
$$-ig^2 f^{XAD} f^{XBC} [g^{\alpha\beta} g^{\gamma\delta} - g^{\alpha\gamma} g^{\beta\delta}]$$
$$-ig^2 f^{XAB} f^{XCD} [g^{\alpha\gamma} g^{\beta\delta} - g^{\alpha\delta} g^{\beta\gamma}]$$



Universal FeynRules Output

[arXiv:1108.2040]

vertices.py **couplings.py** parameters.py lorentz.py

```
GC_12 = Coupling(name = 'GC_12',
                 value = 'complex(0,1)*G**2',
                 order = {'QCD':2})
```

coupling_orders.py

```
QCD = CouplingOrder(name = 'QCD',
                    expansion_order = 99,
                    hierarchy = 1)

QED = CouplingOrder(name = 'QED',
                    expansion_order = 99,
                    hierarchy = 2)
```

Universal FeynRules Output

[arXiv:1108.2040]

vertices.py couplings.py **parameters.py** lorentz.py

```
G = Parameter(name = 'G',
              nature = 'internal',
              type = 'real',
              value = '2*cmath.sqrt(aS)*cmath.sqrt(cmath.pi)',
              texname = 'G')
```

```
aS = Parameter(name = 'aS',
               nature = 'external',
               type = 'real',
               value = 0.118,
               texname = '\\alpha _s',
               lhablock = 'SMINPUTS',
               lhacode = [ 3 ])
```

Universal FeynRules Output

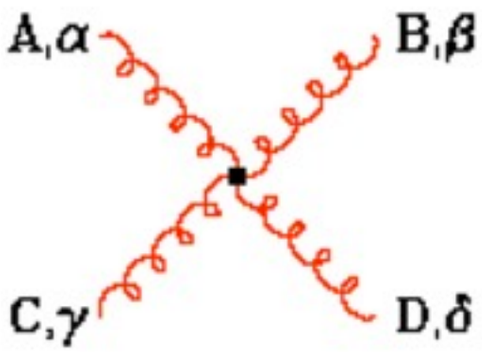
[arXiv:1108.2040]

vertices.py couplings.py parameters.py **lorentz.py**

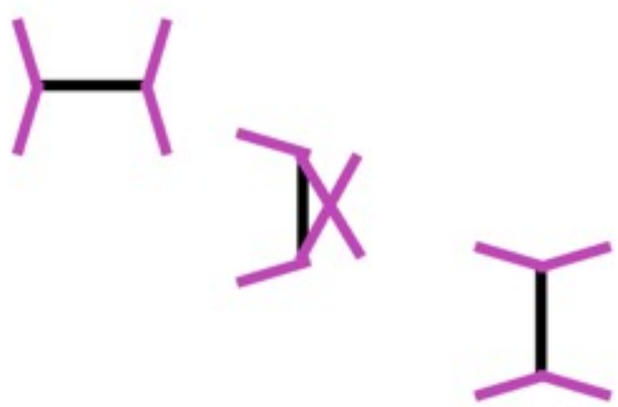
```
VVVV1 = Lorentz(name = 'VVVV1',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,4)*Metric(2,3) - Metric(1,3)*Metric(2,4)')

VVVV3 = Lorentz(name = 'VVVV3',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,4)*Metric(2,3) - Metric(1,2)*Metric(3,4)')

VVVV4 = Lorentz(name = 'VVVV4',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,3)*Metric(2,4) - Metric(1,2)*Metric(3,4)')
```



$$-ig^2 f^{XAC} f^{XBD} [g^{\alpha\beta} g^{\gamma\delta} - g^{\alpha\delta} g^{\beta\gamma}]$$
$$-ig^2 f^{XAD} f^{XBC} [g^{\alpha\beta} g^{\gamma\delta} - g^{\alpha\gamma} g^{\beta\delta}]$$
$$-ig^2 f^{XAB} f^{XCD} [g^{\alpha\gamma} g^{\beta\delta} - g^{\alpha\delta} g^{\beta\gamma}]$$

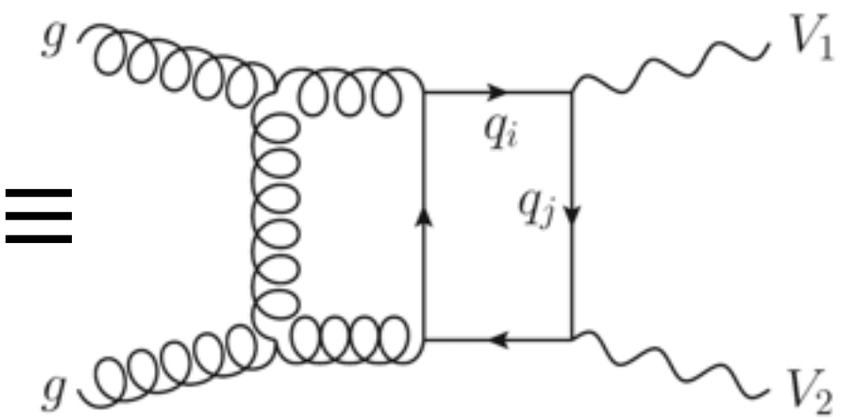


MultiLoop form factors

- IDEA: use a similar format for distributing multi-loop form-factors:

$$
\begin{aligned}
S^{\mu\nu\rho\sigma}(p_1,p_2,p_3) = {} & a_1 g^{\mu\nu} g^{\rho\sigma} + a_2 g^{\mu\rho} g^{\nu\sigma} + a_3 g^{\mu\sigma} g^{\nu\rho} \\
& + \sum_{j_1,j_2=1}^{3} \Big(b^{(1)}_{j_1 j_2}\, g^{\mu\nu}\, p^{\rho}_{j_1}\, p^{\sigma}_{j_2} + b^{(2)}_{j_1 j_2}\, g^{\mu\rho}\, p^{\nu}_{j_1}\, p^{\sigma}_{j_2} + b^{(3)}_{j_1 j_2}\, g^{\mu\sigma}\, p^{\nu}_{j_1}\, p^{\rho}_{j_2} \\
& \quad + b^{(4)}_{j_1 j_2}\, g^{\nu\rho}\, p^{\mu}_{j_1}\, p^{\sigma}_{j_2} + b^{(5)}_{j_1 j_2}\, g^{\nu\sigma}\, p^{\mu}_{j_1}\, p^{\rho}_{j_2} + b^{(6)}_{j_1 j_2}\, g^{\rho\sigma}\, p^{\mu}_{j_1}\, p^{\nu}_{j_2} \Big) \\
& + \sum_{j_1,j_2,j_3,j_4=1}^{3} c_{j_1 j_2 j_3 j_4} p^{\mu}_{j_1}\, p^{\nu}_{j_2} p^{\rho}_{j_3}\, p^{\sigma}_{j_4},
\end{aligned}
$$

$\equiv$

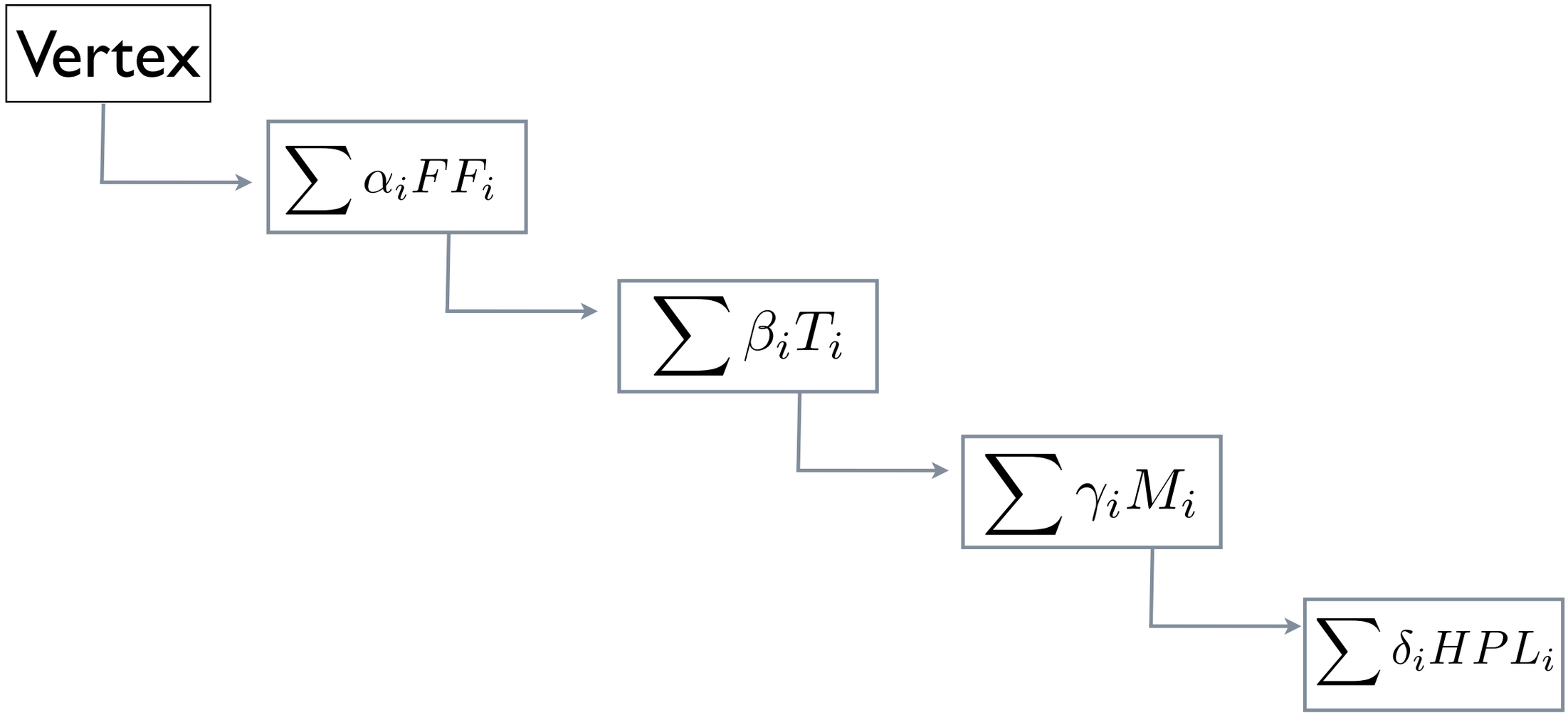



```
GGAA = Vertex(name = 'GGAA',
              particles = [ P.G, P.G, P.A, P.A ],
              color = [ 'Identity(1,2)' ],
              lorentz = [ L.A, L.B, L.C, L.D, L.E,
                          L.F, L.G, L.H, L.I, L.J,
                          L.K, L.L, L.M, L.N, L.O,
                          L.P, L.Q, L.R, L.S, L.T
                          ],
               couplings = {(0,0):C.GGAA_C1, (0,1):C.GGAA_C2, (0,2):C.GGAA_C3, (0,3):C.GGAA_C4, (0,4):C.GGAA_C5,
                            (0,5):C.GGAA_C6, (0,6):C.GGAA_C7, (0,7):C.GGAA_C8, (0,8):C.GGAA_C9, (0,9):C.GGAA_C10,
                            (0,10):C.GGAA_C11, (0,11):C.GGAA_C12, (0,12):C.GGAA_C13, (0,13):C.GGAA_C14, (0,14):C.GGAA_C15,
                            (0,15):C.GGAA_C16, (0,16):C.GGAA_C17, (0,17):C.GGAA_C18, (0,18):C.GGAA_C19, (0,19):C.GGAA_C20
                           })
```

- Allows a tool like MG5_aMC to generate arbitrary 2-loop amplitudes containing this loop (with any decay or vector quantum numbers, but with still onshell gluons)

Beyond Form Factors

- Push the idea further: extend the UFO format so as to further break-down the form factor into Topologies and scalar masters.



- UFO extension to accommodate the above?