From Lagrangians to events (& more) with FEYNRULES

Benjamin Fuks

LPTHE / Sorbonne Université

Dark Tools 2025 – 16-19 June 2025

Dipartimento di Fisica @ Torino

HEP software and their role

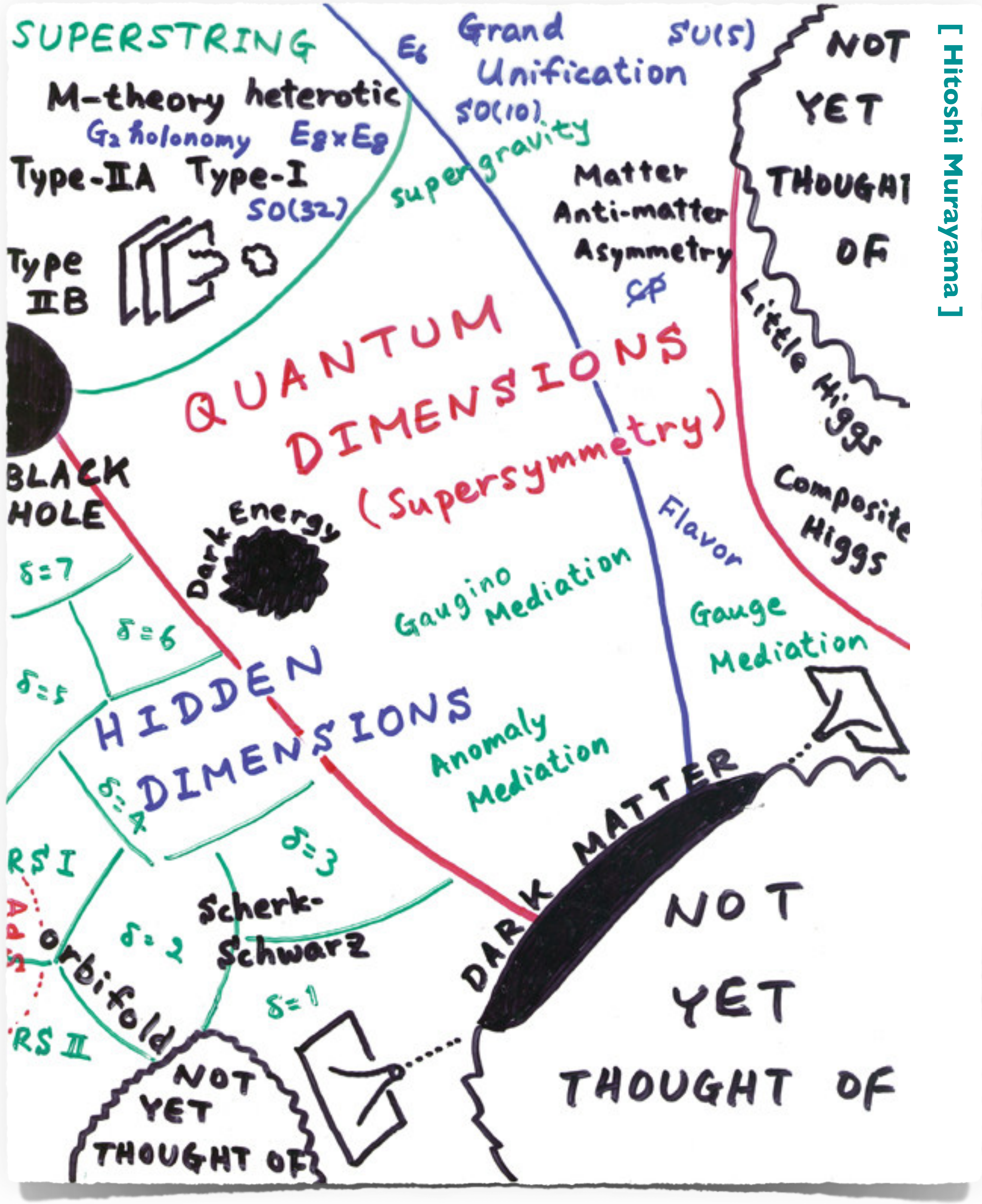
Towards the characterisation of new physics

- About the nature of an observation
  - ➙ Fitting and (re)interpreting deviations
  - ➙ Prospective studies of varied signals
- Final words on the nature of any potential BSM
  - ➙ Accurate measurements
  - ➙ Precise predictions

HEP tools ≡ key role at each step

- Prospective studies (*a priori* preparation)
  - ➙ Study of BSM signals in light of backgrounds
- Reinterpretation of existing results (*a posteriori* reactions)
  - ➙ Recasting in new contexts

BSM implementations in tools



[Hitoshi Murayama]

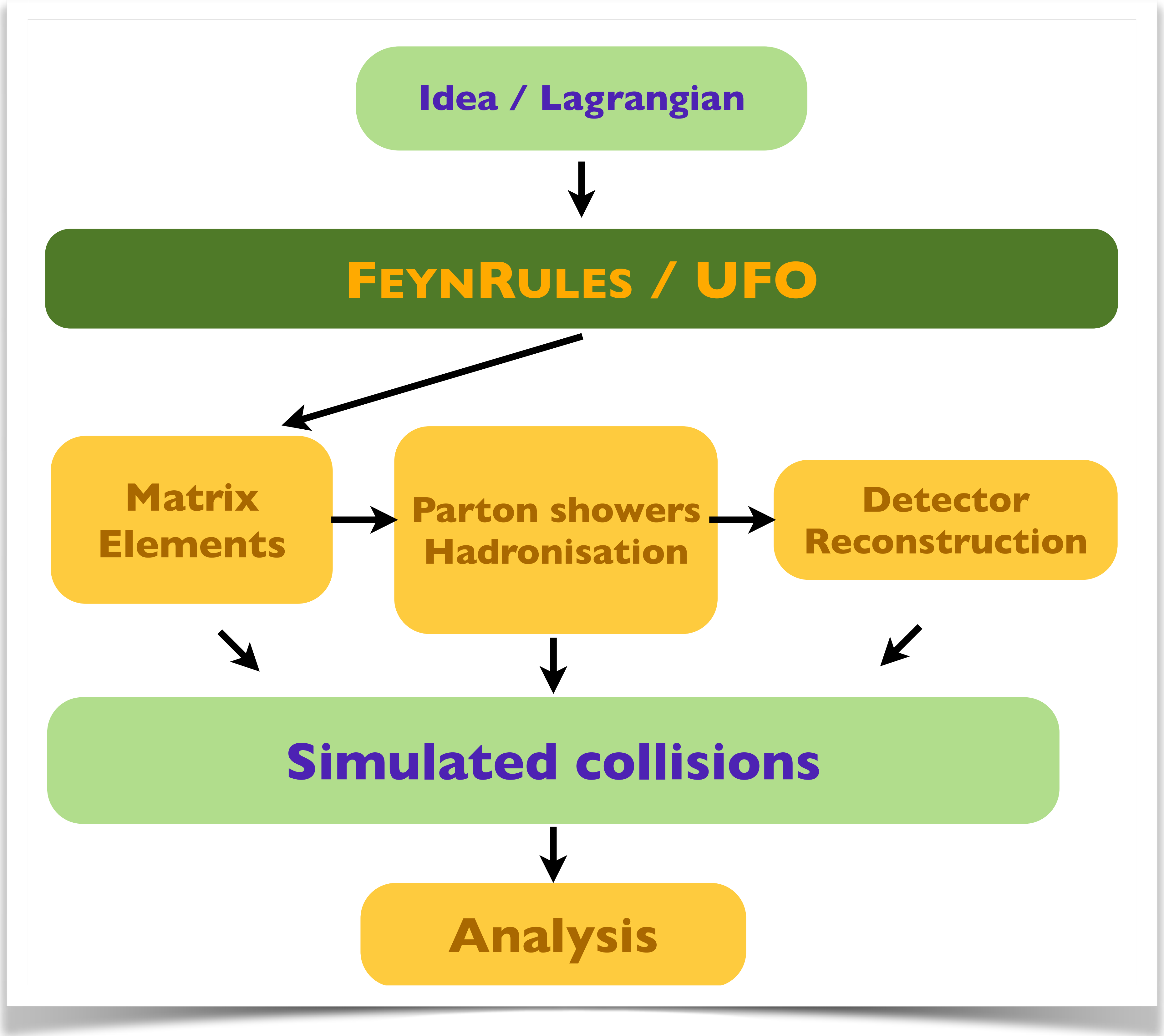
Challenges for new physics

- No sign of new physics
  ➙ no leading candidate theory

- Plethora of models to consider
  ➙ many implementations in tools required

Connecting ideas to software

[Christensen, de Aquino, Degrande, Duhr, BF, Herquet, Maltoni & Schumann (EPJC`11)]



- Model building / spectrum → FEYNRULES/UFO
- Hard scattering
  - ⋆ Feynman diagram and amplitude generation
  - ⋆ Monte Carlo integration
  - ⋆ Event generation

  → MG5_AMC / MADDM
  CALCHEP (*cf.* MICROMEGAS)
- QCD environment
  - ⋆ Parton showering
  - ⋆ Hadronisation
  - ⋆ Underlying event

  → PYTHIA 8
- Detector simulation
  - ⋆ Simulation of the detector response
  - ⋆ Object reconstruction

  → MADANALYSIS 5 – SFS/DELPHES
- Event analysis
  - ⋆ Signal/background analysis
  - ⋆ LHC recasting

  → MADANALYSIS 5 / SPEY

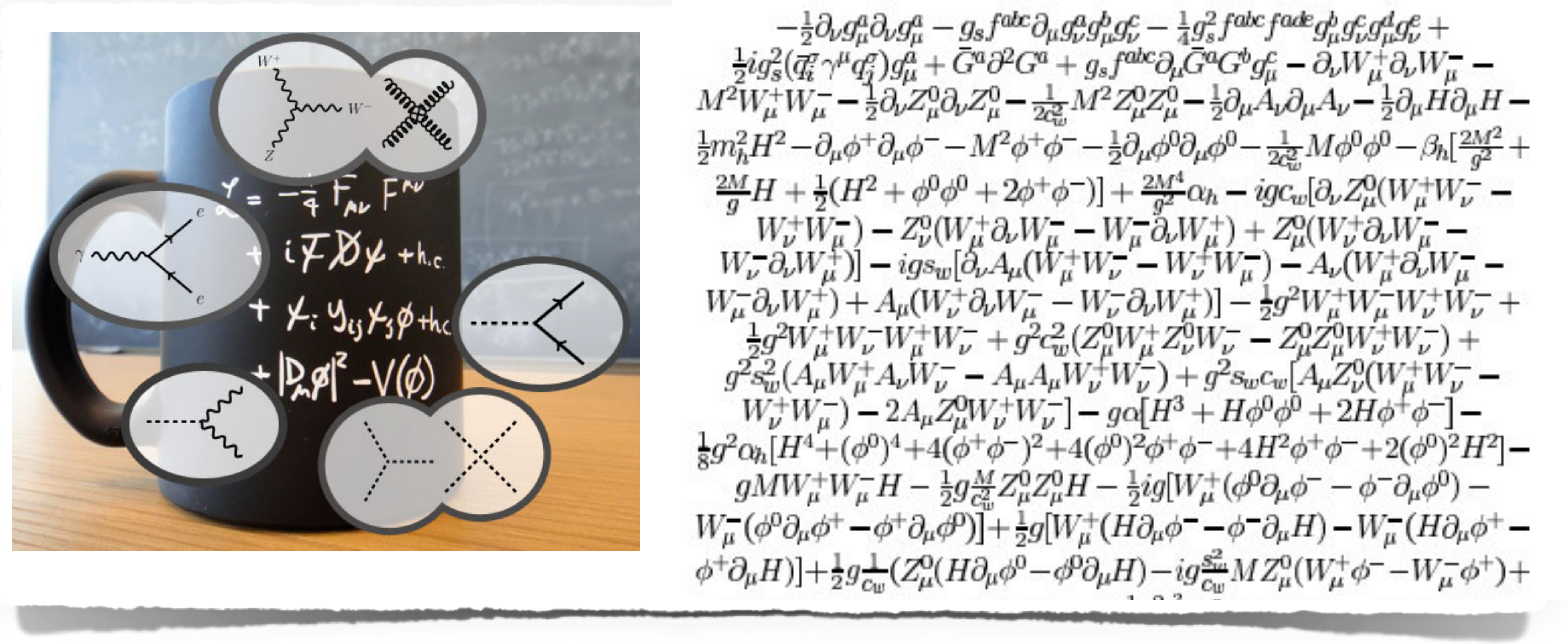
Implementation of a new physics model in a HEP program

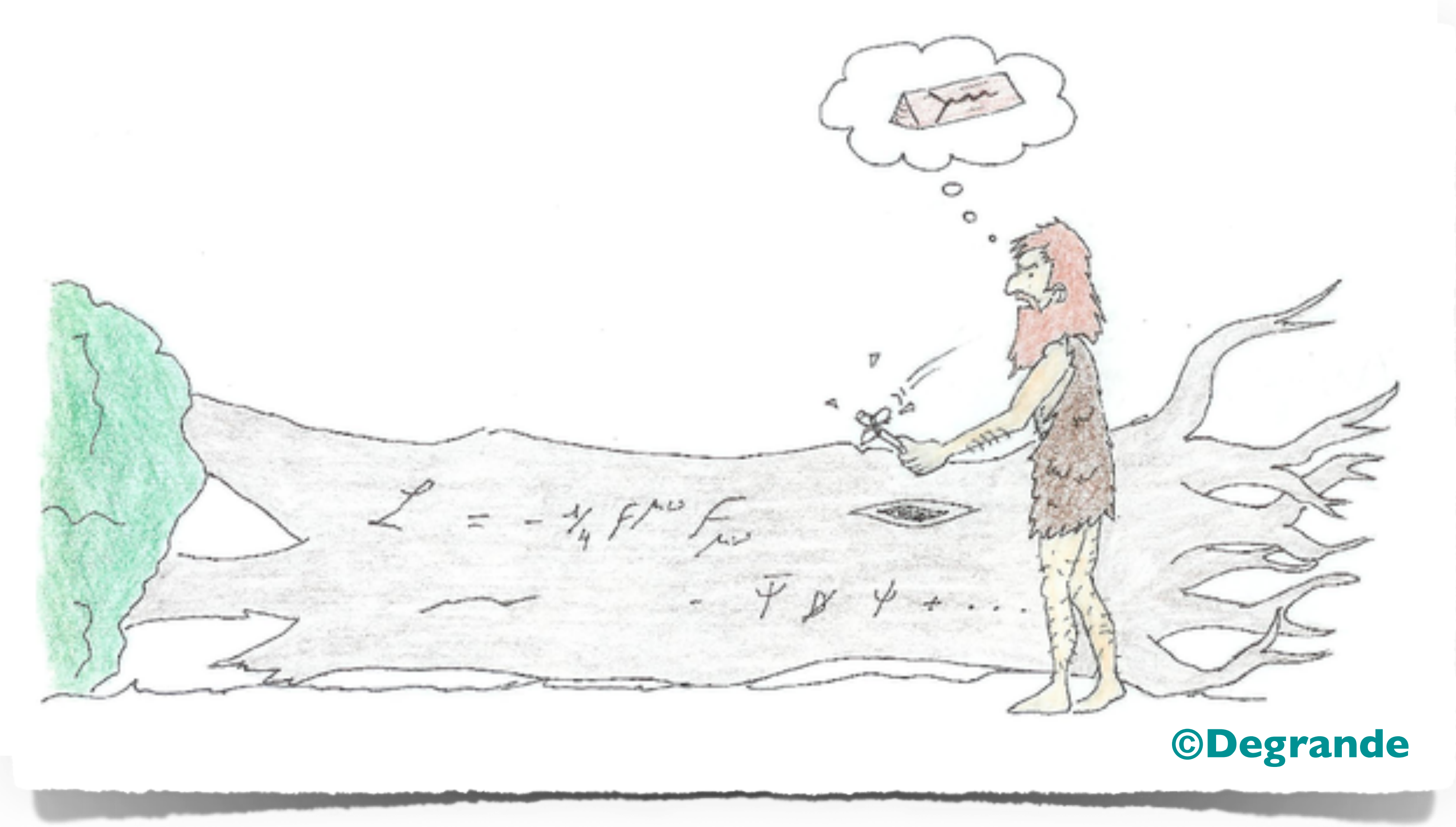
- Definition: particles, parameters and vertices (≡ Lagrangian)
  - ➝ translated in some programming language
- Tedious, time-consuming, error prone
- Beware of restrictions/conventions

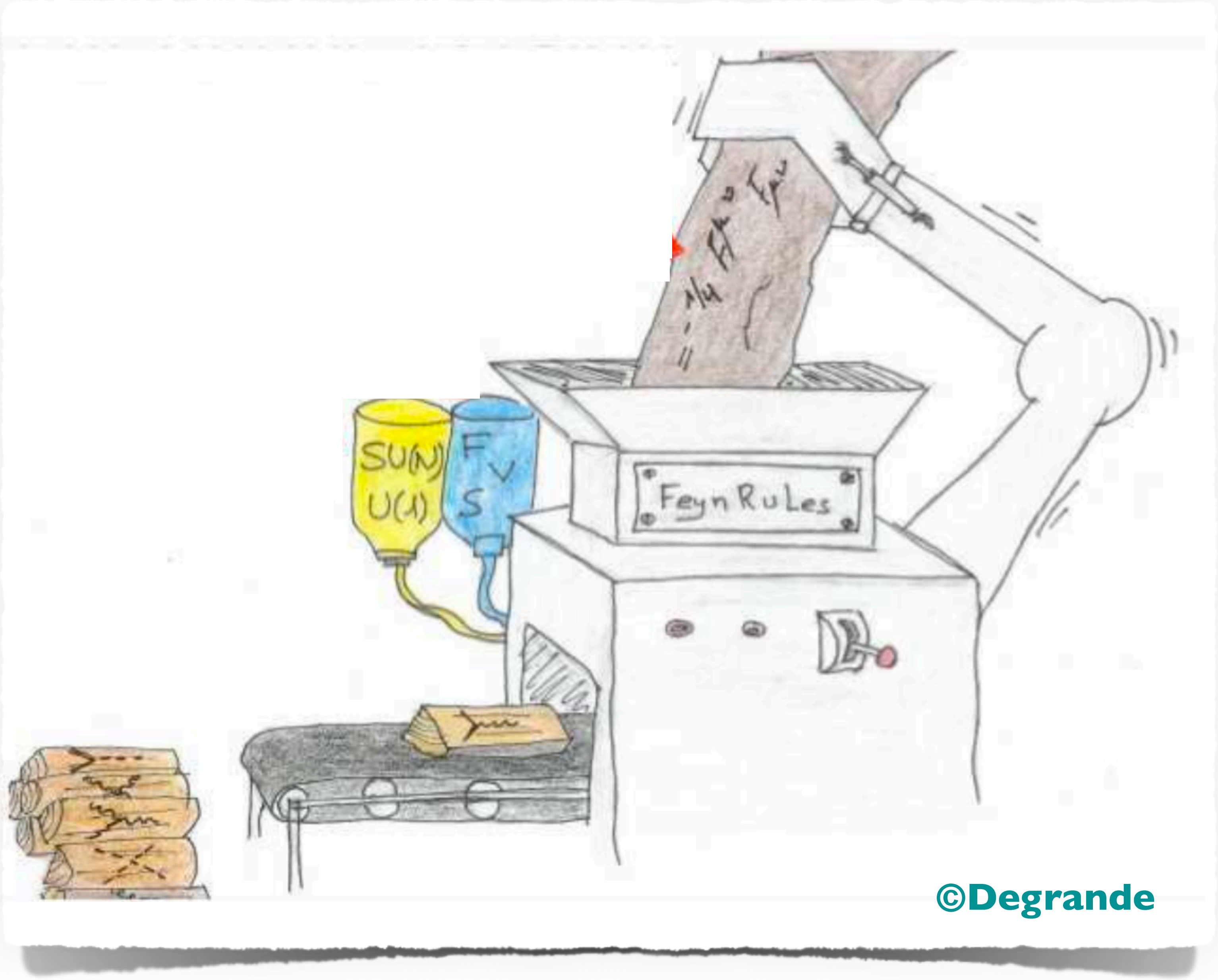
Highly redundant

- Each tool ⊕ each model
- No-brainer tasks (from Feynman rules to pieces of code)

➝ Systematisation / automation







FEYNRULES in a nutshell

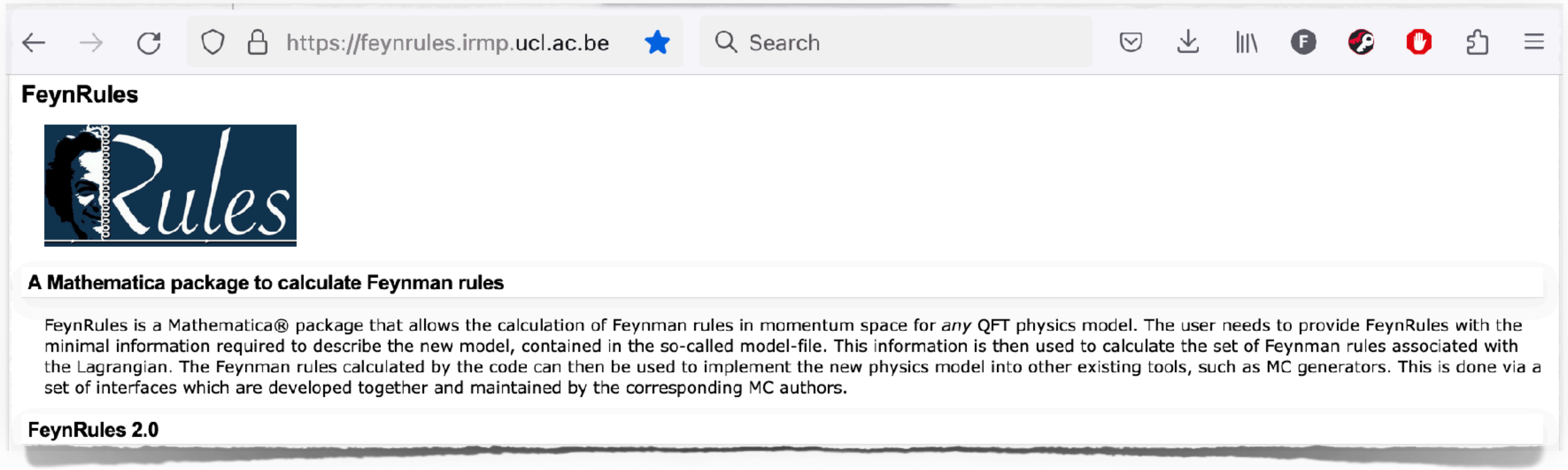
The FEYNRULES platform (since 2009)

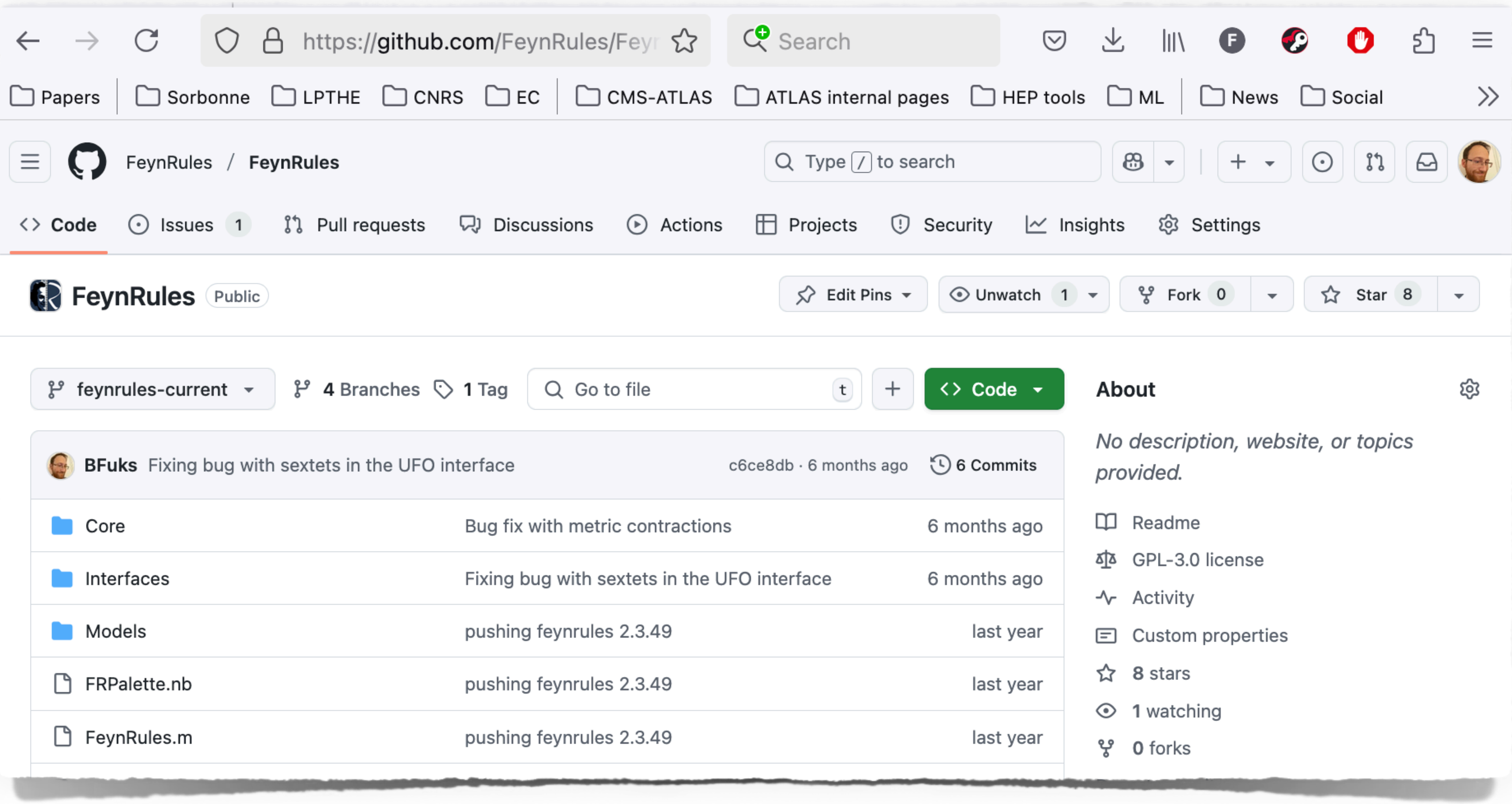
- Working environment: MATHEMATICA
  - ⋆ Flexibility, symbolic manipulations, easy implementation of new methods, *etc.*
  - ⋆ Many plugins (superspace, spectrum, decays, NLO, *etc.*)

- Interfaces to many MC tools
  - ⋆ Dedicated interfaces (CALCHEP, FEYNARTS)
  - ⋆ Generic interface: UFOs (MG5_AMC, HERWIG, SHERPA, WHIZARD, …)

- Very few limitations on models
  - ⋆ Higher-dimensional operators supported
  - ⋆ Spins (up to 2); colour structures (1, 3, 6, 8)

[Christensen & Duhr (CPC '09)]
[Alloul, Christensen, Degrande, Duhr & BF (CPC'14)]



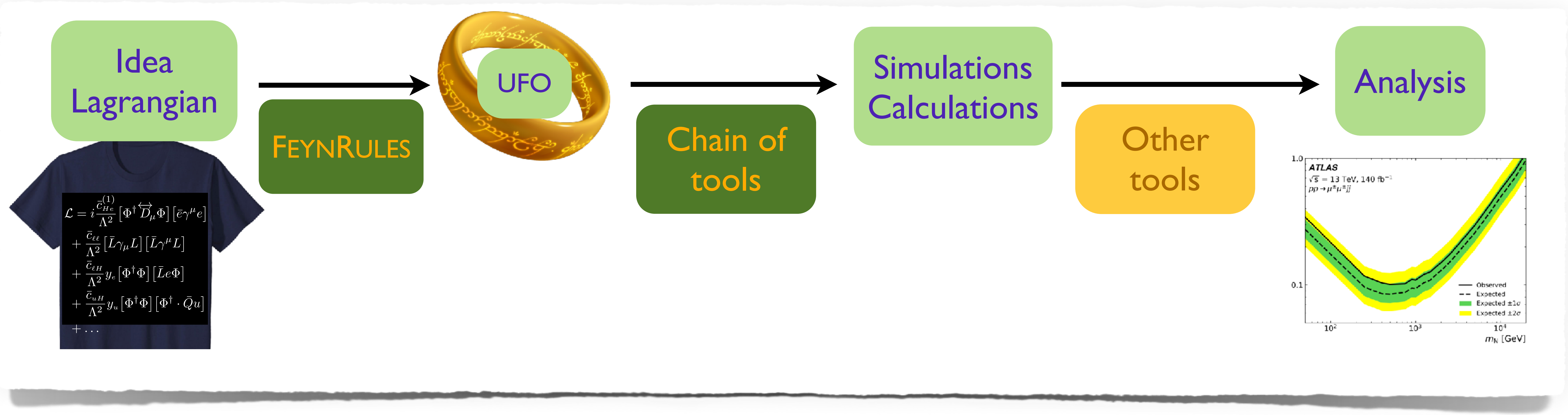


Interfacing Lagrangians and HEP tools

Linking a Lagrangian to various HEP tools

- Derivation of the model's Feynman rules (vertices, particle content, *etc.*)
  - ➞ role of FEYNRULES
- Interface of FEYNRULES to event generators
  - ⋆ Removal of vertices not compliant with the tool (colour and Lorentz structures)
  - ⋆ Translation to a specific format and programming language

- The UFO: one format to rule them all
  - ⋆ Too many interfaces not efficient (maintenance, versioning, *etc.*)
  - ⋆ Design of a unique intermediate layer



The Universal FeynRules Output

[Degrande et al. (CPC '12)]
[Darmé et al. (EPJC`24)]

The UFO in a nutshell

- UFO ≡ Universal FeynRules output ➞ Universal Feynman Output
  - ⋆ Universal as not tied to any specific programme
- Set of Python files to be linked to any code with full information
  - ⋆ Generic colour and Lorentz structures
  - ⋆ Restrictions on acceptable colour/Lorentz structures enforced at the software level

Initially designed as the MG5aMC model format, UFOs now standard

FeynRules | Sarah | LanHEP

Achilles | Comix | Contur | GoSam | Herwig 7 | MadAnalysis 5

MadDM | MadGraph5_aMC@NLO | Recola | Sherpa | Whizard

The UFO in practice

The UFO ≡ set of PYTHON files

- Factorisation of the information in mandatory and optional files
  - ➙ particles
  - ➙ parameters
  - ➙ interactions
  - ➙ extra stuff (NLO, decays, propagators, functions, *etc.*)
- Economical implementation of vertices and structures through recycling across the model

Particles   Parameters

```
[fuks@NewMouth /Users/fuks/Documents/HEPTools/MG5_aMC/2.9.13/models/DMSimp_t$] ls
CT_couplings.py          decays.py            param_s3d_ur.dat       restrict_F3C_ur.dat    restrict_F3V_uni.dat   restrict_S3D_ur.dat
CT_parameters.py         function_library.py  param_s3m_ur.dat       restrict_F3S_3rd.dat   restrict_F3V_ur.dat    restrict_S3M_3rd.dat
CT_vertices.py           lorentz.py           parameters.py          restrict_F3S_VLQ.dat   restrict_F3W_3rd.dat   restrict_S3M_uni.dat
DMSimpt_NLO_v1_2_UFO.log object_library.py    particles.py           restrict_F3S_cr.dat    restrict_F3W_uni.dat   restrict_S3M_ur.dat
__init__.py              param_f3s_cr.dat     propagators.py         restrict_F3S_uni.dat   restrict_F3W_ur.dat    vertices.py
coupling_orders.py       param_f3s_ur.dat     restrict_F3C_3rd.dat   restrict_F3S_ur.dat    restrict_S3D_3rd.dat   write_param_card.py
couplings.py             param_f3v_ur.dat     restrict_F3C_uni.dat   restrict_F3V_3rd.dat   restrict_S3D_uni.dat
```

Interactions

Restrictions: from a general model to a less general one

The UFO: particles & parameters

Particles ≡ instances of the particle class

- Attributes: spin, colour representation, mass, width, *etc.*
- Antiparticles automatically derived

```
Xs = Particle(pdg_code = 51,
              name = 'Xs',
              antiname = 'Xs',
              spin = 1,
              color = 1,
              mass = Param.MXs,
              width = Param.WXs,
              texname = 'Xs',
              antitexname = 'Xs',
              charge = 0,
              GhostNumber = 0,
              LeptonNumber = 0,
              Y = 0)
```

Parameters ≡ instances of the parameter class

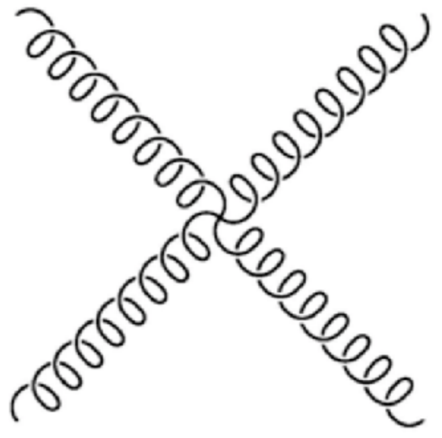
- External parameters: Les Houches-like structure
- PYTHON-compliant formula for the internal parameters

```
lamF3Q3x3 = Parameter(name = 'lamF3Q3x3',
                      nature = 'external',
                      type = 'real',
                      value = 0.22,
                      texname = '\\text{lamF3Q3x3}',
                      lhablock = 'DMF3Q',
                      lhacode = [ 3, 3 ])

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

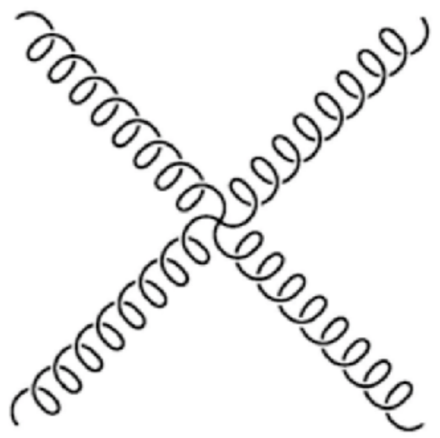
Decomposition in a spin x colour basis (coupling strengths ≡ coordinates)

- Example: the quartic gluon vertex



$$
\begin{aligned}
& i g_s^2\, f^{a_1 a_2 b} f^{b a_3 a_4}\ (\eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4}) \\
&\quad + i g_s^2\, f^{a_1 a_3 b} f^{b a_2 a_4}\ (\eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4}) \\
&\quad + i g_s^2\, f^{a_1 a_4 b} f^{b a_2 a_3}\ (\eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4})
\end{aligned}
$$

- UFO version



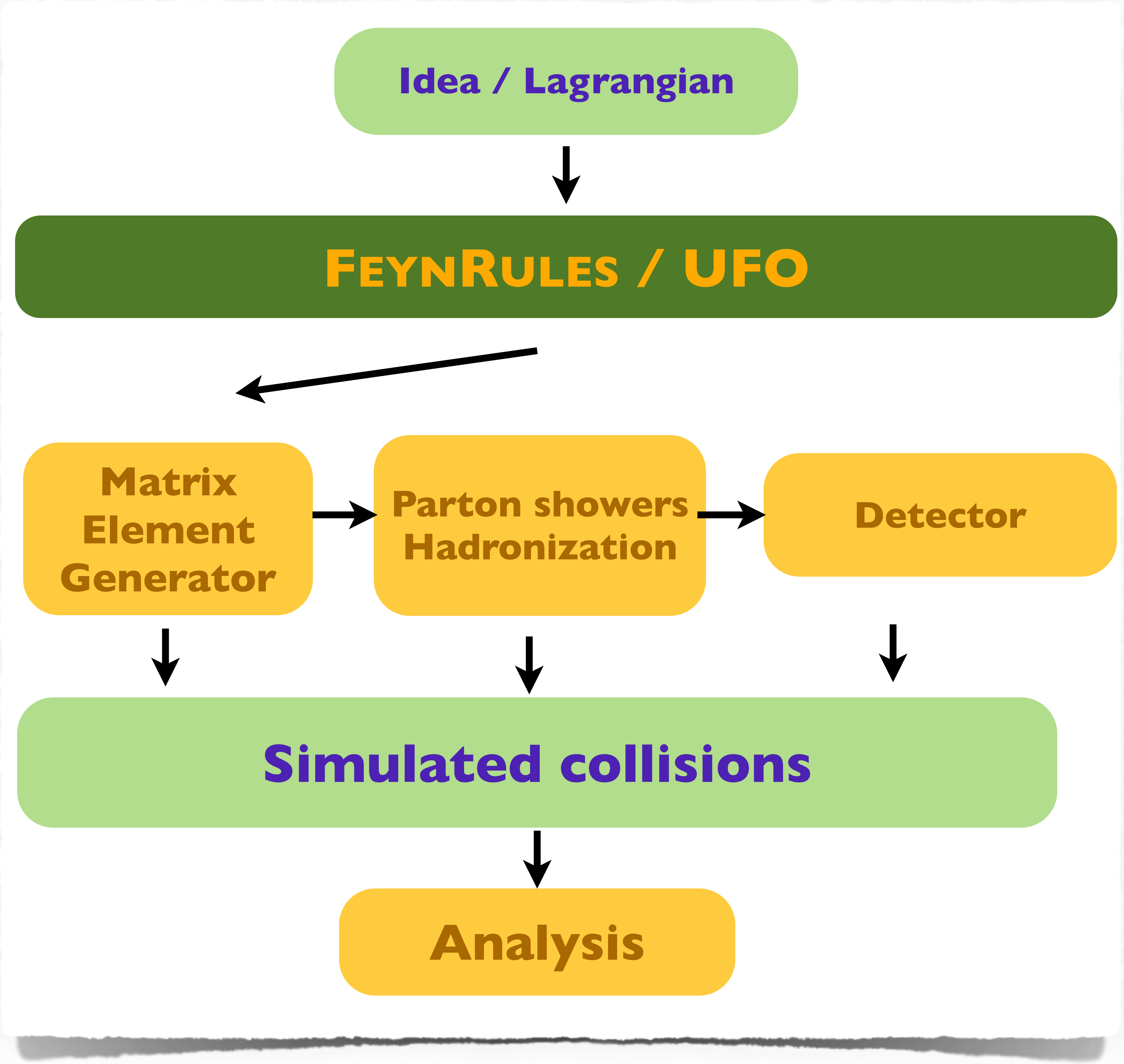
$$
\left(f^{a_1 a_2 b} f^{b a_3 a_4},\ \ f^{a_1 a_3 b} f^{b a_2 a_4}, f^{a_1 a_4 b} f^{b a_2 a_3} \right)
\times \begin{pmatrix} i g_s^2 & 0 & 0 \\ 0 & i g_s^2 & 0 \\ 0 & 0 & i g_s^2 \end{pmatrix}
\begin{pmatrix} \eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} \\ \eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4} \\ \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4} \end{pmatrix}
$$

  - ⋆ 3 elements for the colour basis
  - ⋆ 3 elements for the spin (Lorentz structure) basis
  - ⋆ 9 coordinates (6 are zero, only 1 encoded)

Information split across several files

```
[fuks@NewMouth /Users/fuks/Documents/HEPTools/MG5_aMC/2.9.13/models/DMSimp_t$] ls
CT_couplings.py          decays.py              param_s3d_ur.dat        restrict_F3C_ur.dat     restrict_F3V_uni.dat    restrict_S3D_ur.dat
CT_parameters.py         function_library.py    param_s3m_ur.dat        restrict_F3S_3rd.dat    restrict_F3V_ur.dat     restrict_S3M_3rd.dat
CT_vertices.py           lorentz.py             parameters.py           restrict_F3S_VLQ.dat    restrict_F3W_3rd.dat    restrict_S3M_uni.dat
DMSimpt_NLO_v1_2_UFO.log object_library.py      particles.py            restrict_F3S_cr.dat     restrict_F3W_uni.dat    restrict_S3M_ur.dat
__init__.py              param_f3s_cr.dat       propagators.py          restrict_F3S_uni.dat    restrict_F3W_ur.dat     vertices.py
coupling_orders.py       param_f3s_ur.dat       restrict_F3C_3rd.dat    restrict_F3S_ur.dat     restrict_S3D_3rd.dat    write_param_card.py
couplings.py             param_f3v_ur.dat       restrict_F3C_uni.dat    restrict_F3V_3rd.dat    restrict_S3D_uni.dat
```

Summary



Challenges for new physics

- No sign of new physics
  → no leading candidate theory
- Plethora of models to consider
  → many implementations in tools required

Automation

- Connecting models (Lagrangians) to tools
- Translation of Feynman rules to dedicated programming languages
- One UFO to rule them all!

The FeynRules TUTORIAL

Benjamin Fuks

LPTHE / Sorbonne Université

Dark Tools 2025 – 16-19 June 2025

Dipartimento di Fisica @ Torino

The model – compositeness and dark matter

Modelling composite theories with dark matter and partial compositeness

- Top mass achieved from mixing with vector-like partners
  ➙ bottom quark massless

$$Q^0_{L,R} = \begin{pmatrix} T^0_{L,R} \\ B^0_{L,R} \end{pmatrix} \quad \text{and} \quad \tilde{T}^0_{L,R}$$

- A scalar dark matter candidate X

- Quite simple Lagrangian:
  ➙ Yukawa couplings y between the partners and the Higgs $\boldsymbol{\Phi}$
  ➙ Mass mixings $\boldsymbol{\Delta}$ [origin not relevant]
  ➙ Yukawa couplings λ between X, the partners and the SM

$$\begin{aligned}\mathcal{L}_{\rm BSM} = & - M_Q \overline{Q^0_L} Q^0_R - M_{\tilde{T}} \overline{\tilde{T}^0_L} \tilde{T}^0_R - \frac{1}{2} M_X X^2 \\ & - \Big(y^* (\overline{Q^0_L} \cdot \Phi^\dagger) \widetilde{T}^0_R + \Delta_L \overline{q^0_L} Q^0_R + \Delta_R \overline{t^0_R} \widetilde{T}^0_L + \text{H.c.} \Big) \\ & + \Big(\hat{\lambda}_{\mathbf{Q}}\ \overline{Q^0_R}\ q^0_L X + \hat{\lambda}_{\mathbf{T}}\ \overline{\tilde{T}^0_L}\ t^0_R X + \text{H.c.} \Big)\end{aligned}$$

Simplified model

- 3 mediators and 1 dark matter (mass eigenstates ≡ gauge eigenstates)

$$T_{L,R}\,, \quad \tilde{T}_{L,R}\,, \quad B_{L,R} \quad \text{and} \quad X$$

- Lagrangian ➙ Free parameters: 4 masses and 2 couplings

$$\begin{aligned}\mathcal{L}_{\rm BSM} = \mathcal{L}_{\rm kin} - M_T \overline{T}\ T - M_B \overline{B}\ B - M_{\tilde{T}} \overline{\tilde{T}}\ \tilde{T} - \frac{1}{2} M_X X^2 \\ + \Big(\lambda_Q \big[\ \overline{T_R}\ t_L + \overline{B_R}\ b_L\ \big] X + \lambda_T\ \overline{\tilde{T}_L}\ t_R\ X + \text{H.c.} \Big)\end{aligned}$$

Model implementation

A FEYNRULES model file

A FEYNRULES model file is an .fr file written in MATHEMATICA syntax

Preamble
- ⋆ Author information
- ⋆ Model information
- ⋆ Index definitions

Declaration of the gauge group
- ⋆ Abelian or not
- ⋆ Representation matrices
- ⋆ Structure constants
- ⋆ Coupling constant
- ⋆ Gauge boson or vector superfield

Declaration of fields
- ⋆ Names, spins, PDG codes
- ⋆ Indices, quantum numbers
- ⋆ Masses, widths
- ⋆ Classes and class members

Declaration of parameters
- ⋆ External and internal
- ⋆ Scalar and tensor

A Lagrangian

Check the SM implementation on GITHUB

- To be updated during the tutorial

https://github.com/BFuks/DMSimpt/blob/379b14d6436dc02dac9d6664af2256c529c462b4/sm.fr

Preamble: general information

An electronic signature for the model implementation

- Important for traceability, documentation, contact with the authors, *etc*.
- Reference publications
- Webpage information

```
M$ModelName = "DMsimp_t";

M$Information = {
  Authors->{"Benjamin Fuks"}, Emails->{"fuks@lpthe.jussieu.fr"},
  Institutions->{"LPTHE / Sorbonne U."},
  Version->"2.0", Date->"27.03.24",
  URLs->"http://feynrules.irmp.ucl.ac.be/wiki/DMsimpt/"
};
```

Preamble: indices

Index dimensions

- Fundamental $SU(3)_C$ indices for squarks and quarks: *Colour*, dimension 3
- Adjoint $SU(3)_C$ indices for gluons and gluinos: *Gluon*, dimension 8
- Lorentz and spin indices automatically handled

```
IndexRange[Index[Gluon ]] = NoUnfold[Range[8]];
IndexRange[Index[Colour]] = NoUnfold[Range[3]];
```

The style of the indices can be specified

- Fundamental indices starting with *m*
- Adjoint indices starting with *a*

```
IndexStyle[Colour, m];
IndexStyle[Gluon,  a];
```

QCD ≡ special role in many tools → special names

- *Colour, Sextet and Gluon* for colour indices
- *G* for the gluon field
- *T* for the fundamental representation matrices
- *f* and *d* for the structure constants
- G/*gs* and aS for the coupling constants

Gauge groups

Each element the group is declared in the *M$GaugeGroups* list

- One group ≡ one set of MATHEMATICA replacement rules
  - ➞ Examples: SU(3)$_C$ (name: *SU3C*) or SU(2)$_L$ (name: *SU2L*)

- One rule ≡ one group property
  - ⋆ *Abelian*: abelian or non-abelian group
  - ⋆ *GaugeBoson*: associated gauge boson
  - ⋆ *CouplingConstant, StructureConstant*: coupling and structure constants
  - ⋆ *Representations*: list of 2-tuples linking indices to representations

Advantages of a proper gauge group declaration

- Writing Lagrangians more easily
  - ⋆ Covariant derivatives (*DC[field, Lorentz index]*)
  - ⋆ Field strength tensors (*FS[field, Lorentz index 1, Lorentz index 2]*)

$$-\frac{1}{4}g^a_{\mu\nu}g^{\mu\nu}_a + i\bar{u}_R \not{D} u_R$$

```
- 1/4 FS[G,mu,nu,aa] FS[G,mu,nu,aa]

   + I* uRbar.Ga[mu].DC[uR, mu]
```

```
SU2L == {
  Abelian           -> False,
  CouplingConstant  -> gw,
  GaugeBoson        -> Wi,
  StructureConstant -> Eps,
  Representations   -> {Ta,SU2D},
  Definitions       -> {
    Ta[a_,b_,c_]->PauliSigma[a,b,c]/2,
    FSU2L[i_,j_,k_]:> I Eps[i,j,k]
  }
},
SU3C == {
  Abelian           -> False,
  CouplingConstant  -> gs,
  GaugeBoson        -> G,
  StructureConstant -> f,
  Representations   -> {T,Colour},
  SymmetricTensor   -> dSUN
}
```

See the manual for more details on gauge groups

Fields – some gauge fields

Each field is an element of the *M$ClassesDescription* list

- A declaration ≡ a set of MATHEMATICA replacement rules
- One rule ≡ one property of the field
  - ⋆ Vector field
    - ➙ label = V[4],V[12] (with V and not F, S, R, T, *etc.*)
  - ⋆ ClassName: defines a symbol for Lagrangians
    - ➙ *G*, *Wi*
  - ⋆ Indices: gluon/Wi fields in the adjoint representation
    - ➙ *G* as gauge boson of $SU(3)_c$ ≡ *Gluon* internally linked to the adjoint
    - ➙ *Wi* as gauge boson of $SU(2)_L$ ≡ *SU2W* internally linked to the adjoint
  - ⋆ Unphysical and definitions:
    - ➙ Unphysical fields in the Lagrangian (easier)
    - ➙ Rotations to the physical basis automatic (FlavorIndex)
    - ➙ To the *W*/*Wbar* and *A*/*Z* fields
  - ⋆ Others: mass, width, PDG code, self-conjugate, *etc.*

```
M$ClassesDescription = {

(* Gauge bosons: physical vector fields *)
  V[4] == {
    ClassName       -> G,
    SelfConjugate   -> True,
    Indices         -> {Index[Gluon]},
    Mass            -> 0,
    Width           -> 0,
    ParticleName    -> "g",
    PDG             -> 21,
    PropagatorLabel -> "G",
    PropagatorType  -> C,
    PropagatorArrow -> None,
    FullName        -> "G"
  },

   V[12] == {
     ClassName     -> Wi,
     Unphysical    -> True,
     SelfConjugate -> True,
     Indices       -> {Index[SU2W]},
     FlavorIndex   -> SU2W,
     Definitions   -> {
       Wi[mu_,1] -> (Wbar[mu]+W[mu])/Sqrt[2],
       Wi[mu_,2] -> (Wbar[mu]-W[mu])/(I*Sqrt[2]),
       Wi[mu_,3] -> cw Z[mu] + sw A[mu]
     }
   },
```

See the manual for more options for field declarations

Fields – the quark fields

Example : three generations of up-type quarks

- Differences with the gauge field declaration
  - ⋆ Four-component fermionic field
    - ➙ F[3], F[13] (starting with an F)
  - ⋆ Classname: two symbols
    - ➙ *uq* and *uqbar*
  - ⋆ Mass/width:
    - ➙ Symbols and associated numerical values
  - ⋆ Varied indices
    - ➙ *Generation, Colour, SU2D*
  - ⋆ Unphysical and definitions:
    - ➙ Unphysical gauge fields in the Lagrangian (easier)
    - ➙ Automatic rotations to the mass basis
    - ➙ FlavorIndex: from *QL* to *uq*/*dq*
- New options related to flavour
  - ⋆ ClassMembers: all the members of the class
    - ➙ A *Generation* index as the FlavorIndex
  - ⋆ Mass, Width, PDG:
    - ➙ one / class member (+ generic mass symbol)

```
F[3] == {
  ClassName         -> uq,
  ClassMembers      -> {u, c, t},
  Indices           -> {Index[Generation], Index[Colour]},
  FlavorIndex       -> Generation,
  SelfConjugate     -> False,
  Mass              -> {Mu, {MU, 2.55*^-3}, {MC,1.27}, {MT,172}},
  Width             -> {0, 0, {WT,1.50833649}},
  QuantumNumbers    -> {Q -> 2/3},
  PropagatorLabel   -> {"uq", "u", "c", "t"},
  PropagatorType    -> Straight,
  PropagatorArrow   -> Forward,
  PDG               -> {2, 4, 6},
  ParticleName      -> {"u",  "c",  "t" },
  AntiParticleName  -> {"u~", "c~", "t~"},
  FullName          -> {"u-quark", "c-quark", "t-quark"}
},

F[13] == {
  ClassName       -> QL,
  Unphysical      -> True,
  Indices         -> {Index[SU2D], Index[Generation], Index[Colour]},
  FlavorIndex     -> SU2D,
  SelfConjugate   -> False,
  QuantumNumbers  -> {Y -> 1/6},
  Definitions     -> {
   QL[sp1_,1,ff_,cc_] :> Module[{sp2}, ProjM[sp1,sp2] uq[sp2,ff,cc]],
   QL[sp1_,2,ff_,cc_] :> Module[{sp2,ff2}, CKM[ff,ff2] ProjM[sp1,sp2] ... }
},
```

See the manual for more options for field declarations

Parameters

The model requires the implementation of various parameters

- Masses and widths handled automatically
- Parameters can be external or internal
  ➞ QCD coupling: g_s and α_s special and not independent

Parameters are declared as elements of the list *M$Parameters*

- One declaration ≡ one set of MATHEMATICA replacement rules
  - ⋆ Internal and External parameters
    - ➞ External ≡ free param. (numerical)
    - ➞ Internal ≡ dependent param (formula)
  - ⋆ InteractionOrder: specific to some tools
    - ➞ more efficient diagram generation

More options

- Tensor parameters can be implemented too
  - ➞ Indices, Unitary
- Simplifications in the Lagrangian
  - ➞ Definitions, Value
- Complex and real parameters
  - ➞ Externals real, internal complex
  - ➞ ComplexParameter if needed

See the manual for more details on parameter declarations

```
aS    == {
  ParameterType    -> External,
  BlockName        -> SMINPUTS,
  OrderBlock       -> 3,
  Value            -> 0.1184,
  InteractionOrder -> {QCD,2},
  TeX              -> Subscript[\[Alpha],s],
  Description      -> "Strong coupling constant at the Z pole"
},
gs == {
  ParameterType    -> Internal,
  Value            -> Sqrt[4 Pi aS],
  InteractionOrder -> {QCD,1},
  TeX              -> Subscript[g,s],
  ParameterName    -> G,
  Description      -> "Strong coupling constant at the Z pole"
},
yd == {
  ParameterType    -> Internal,
  Indices          -> {Index[Generation], Index[Generation]},
  Definitions      -> {
    yd[i_?NumericQ, j_?NumericQ] :> 0  /; (i =!= j)
  },
  Value            -> {
    yd[1,1] -> Sqrt[2] ymdo/vev,
    yd[2,2] -> Sqrt[2] yms/vev,
    yd[3,3] -> Sqrt[2] ymb/vev
  },
  InteractionOrder -> {QED, 1},
  ParameterName    -> {yd[1,1] -> ydo, yd[2,2] -> ys, yd[3,3] -> yb},
  TeX              -> Superscript[y, d],
  Description      -> "Down-type Yukawa couplings"
},
```

Parameter organisation - Les Houches structure

Les Houches blocks to organise external parameters

- Les Houches parameter card by MC programmes
  - ➙ BlockName / OrderBlock
  - ➙ Unspecified: the 'FRBlock' block

```
###################################
## INFORMATION FOR SMINPUTS
###################################
Block sminputs
    1 1.279000e+02 # aEWM1
    2 1.166370e-05 # Gf
    3 1.184000e-01 # aS

###################################
## INFORMATION FOR YUKAWA
###################################
Block yukawa
    5 4.700000e+00 # ymb
    6 1.720000e+02 # ymt
   15 1.777000e+00 # ymtau
```

```
(* External parameters *)
aEWM1 == {
  ParameterType    -> External,
  BlockName        -> SMINPUTS,
  OrderBlock       -> 1,
  Value            -> 127.9,
  InteractionOrder -> {QED,-2},
  Description      -> "Inverse of the EW coupling constant at the Z pole"
},
Gf == {
  ParameterType    -> External,
  BlockName        -> SMINPUTS,
  OrderBlock       -> 2,
  Value            -> 1.16637*^-5,
  InteractionOrder -> {QED,2},
  TeX              -> Subscript[G,f],
  Description      -> "Fermi constant"
},
aS    == {
  ParameterType    -> External,
  BlockName        -> SMINPUTS,
  OrderBlock       -> 3,
  Value            -> 0.1184,
  InteractionOrder -> {QCD,2},
  TeX              -> Subscript[\[Alpha],s],
  Description      -> "Strong coupling constant at the Z pole"
},
```

See the manual for more details on parameter declarations

The Mathematica session

Starting a FEYNRULES MATHEMATICA session

Step 1: loading FEYNRULES into MATHEMATICA

- Setting the FEYNRULES path
- Loading FEYNRULES

```
In[1]:= $FeynRulesPath = SetDirectory["/Users/fuks/Documents/HEPTools/FeynRules/fr-2.4.91"];
        << FeynRules`;

         - FeynRules -

        Version: 2.4.91 ( 29 September 2021).

        Authors: A. Alloul, N. Christensen, C. Degrande, C. Duhr, B. Fuks


        Please cite:

            - Comput.Phys.Commun.185:2250-2300,2014 (arXiv:1310.1921;

            - Comput.Phys.Commun.180:1614-1641,2009 (arXiv:0806.4194).


        http://feynrules.phys.ucl.ac.be


        The FeynRules palette can be opened using the command FRPalette[].
```

The output

- Information on FEYNRULES, authors, version numbers, references, *etc.*

Loading the model implementation

Step 2: loading the model implementation into FeynRules

- Implemented the BSM part as an extension of the SM
  - ➙ No need to reinvent the wheel
- Loading a merged model
  - ➙ The SM implementation as the first bit
  - ➙ Your task : implement the new physics part of the model file
- Restrictions: optional simplifications
  - ➙ Some parameters (definitely) set to 0 or 1

```
M$Restrictions = {
   CKM[i_,i_] -> 1,
   CKM[i_?NumericQ, j_?NumericQ] :> 0 /; (i =!= j),
   cabi -> 0
}
```

The output:

- Preamble of the model are printed to the screen
- More information from ModelInformation[ ]

```
In[3]:= SetDirectory[NotebookDirectory[]];
        LoadModel["sm.fr", "dmsimpt_v2.0.fr"];
        LoadRestriction["Massless_4f.rst"];
        LoadRestriction["DiagonalCKM.rst"];

        Merging model-files...
        This model implementation was created by
        Benjamin Fuks
        Model Version: 2.0
        http://feynrules.irmp.ucl.ac.be/wiki/DMsimpt/
        For more information, type ModelInformation[].

          - Loading particle classes.
          - Loading gauge group classes.
          - Loading parameter classes.

        Model DMsimp_t loaded.
        Loading restrictions from Massless_4f.rst : 3 / 18
        Restrictions loaded.
        Loading restrictions from DiagonalCKM.rst : 3 / 3
        Restrictions loaded.
```

Lagrangian – gauge-invariant kinetic terms

The gauge sector Lagrangian ≡ sum of field strength tensors squared

- One term for each gauge group
- From textbook to FEYNRULES: straightforward through gauge-covariant objects

$$-\frac{1}{4}g^a_{\mu\nu}\, g_a^{\mu\nu} - \frac{1}{4}W^a_{\mu\nu}\, W_a^{\mu\nu} - \frac{1}{4}B_{\mu\nu}\, B^{\mu\nu}$$

$$i\bar{Q}_L \not{D} Q_L + i\bar{L}_L \not{D} L_L + i\bar{u}_R \not{D} u_R + i\bar{d}_R \not{D} d_R + i\bar{e}_R \not{D} e_R$$

```
LGauge := Block[{mu,nu,ii,aa},
   - 1/4 FS[B,mu,nu] FS[B,mu,nu] \
   - 1/4 FS[Wi,mu,nu,ii] FS[Wi,mu,nu,ii] \
   - 1/4 FS[G,mu,nu,aa] FS[G,mu,nu,aa]
];

LFermions := Block[{mu}, I*(
    QLbar.Ga[mu].DC[QL, mu]\
  + LLbar.Ga[mu].DC[LL, mu]\
  + uRbar.Ga[mu].DC[uR, mu]\
  + dRbar.Ga[mu].DC[dR, mu]\
  + lRbar.Ga[mu].DC[lR, mu]
)];
```

Strengths of a proper gauge-group implementation

- Shortcut functions (DC, FS, *etc.*)
- Very compact
  - ⋆ Indices understood (automatic handling)

Lagrangian in the gauge basis

Printing the Lagrangians in the gauge basis

In[7]:= **LGauge**

Out[7]= $$-\frac{1}{4}\left(-\partial_{nu}[B_{mu}]+\partial_{mu}[B_{nu}]\right)^2-\frac{1}{4}\left(-\partial_{nu}\left[G_{mu,aa}\right]+\partial_{mu}\left[G_{nu,aa}\right]+g_s\, f_{aa,bb\$9586,cc\$9586}\, G_{mu,bb\$9586}\, G_{nu,cc\$9586}\right)\left(-\partial_{nu}\left[G_{mu,aa}\right]+\partial_{mu}\left[G_{nu,aa}\right]+g_s\, f_{aa,bb\$9587,cc\$9587}\, G_{mu,bb\$9587}\, G_{nu,cc\$9587}\right)-\frac{1}{4}\left(-\partial_{nu}\left[Wi_{mu,ii}\right]+\partial_{mu}\left[Wi_{nu,ii}\right]+g_w\, \epsilon_{ii,bb\$9584,cc\$9584}\, Wi_{mu,bb\$9584}\, Wi_{nu,cc\$9584}\right)\left(-\partial_{nu}\left[Wi_{mu,ii}\right]+\partial_{mu}\left[Wi_{nu,ii}\right]+g_w\, \epsilon_{ii,bb\$9585,cc\$9585}\, Wi_{mu,bb\$9585}\, Wi_{nu,cc\$9585}\right)$$

In[8]:= **LFermions**

Out[8]= $$i\left(\overline{dR}.\gamma^{mu}.\left(\frac{1}{3}\, i\, dR\, g_1\, B_{mu}+\partial_{mu}[dR]-i\, g_s\, T^{PRIVATE`a\$9607}.dR\, G_{mu,PRIVATE`a\$9607}\right)+\overline{LL}.\gamma^{mu}.\left(\frac{1}{2}\, i\, g_1\, LL\, B_{mu}+\partial_{mu}[LL]-i\, g_w\, Ta^{PRIVATE`a\$9605}.LL\, Wi_{mu,PRIVATE`a\$9605}\right)+\overline{lR}.\gamma^{mu}.\left(i\, g_1\, lR\, B_{mu}+\partial_{mu}[lR]\right)+\overline{QL}.\gamma^{mu}.\left(-\frac{1}{6}\, i\, g_1\, QL\, B_{mu}+\partial_{mu}[QL]-i\, g_s\, T^{PRIVATE`a\$9604}.QL\, G_{mu,PRIVATE`a\$9604}-i\, g_w\, Ta^{PRIVATE`a\$9603}.QL\, Wi_{mu,PRIVATE`a\$9603}\right)+\overline{uR}.\gamma^{mu}.\left(-\frac{2}{3}\, i\, g_1\, uR\, B_{mu}+\partial_{mu}[uR]-i\, g_s\, T^{PRIVATE`a\$9606}.uR\, G_{mu,PRIVATE`a\$9606}\right)\right)$$

- Covariant derivatives and field strength tensors automatically evaluated
- Structure constants and representation matrices added automatically
  ➞ Proper gauge-group and field (indices) declarations

Lagrangian in the mass basis

All field rotations automatically handled

- Example : EM interactions
  - ➞ Lagrangian read as in textbooks

Many checks possible

- Hermiticity: FeynmanRules[$\mathscr{L} - \mathscr{L}^\dagger$] $\equiv 0$
- Kinetic and mass terms normalisation

See the manual for more details

```
CheckHermiticity[LFermions]

Checking for hermiticity by calculating
  the Feynman rules contained in L-HC[L].
If the lagrangian is hermitian,
  then the number of vertices should be zero.
Starting Feynman rule calculation.
Expanding the Lagrangian...
Expanding the indices over 8 cores
Collecting the different structures that enter the vertex.
No vertices found.
0 vertices obtained.
The lagrangian is hermitian.
{}
```

```
ExpandedLag = ExpandIndices[LFermions, FlavorExpand → {SU2W, SU2D}];
OptimizeIndex /@ (ExpandedLag /. {(G | W | Wbar | Z)[__] → 0, CKM → IndexDelta})
```

$$
\begin{aligned}
&-\frac{1}{3}\, e\, A_{mu\$1}\, \overline{dq}_{SP\$1,Generation\$1,Colour\$1} . dq_{SP\$2,Generation\$1,Colour\$1}\, \gamma^{mu\$1} . P_{-SP\$1,SP\$2} + \\
&i\, \overline{l}_{SP\$1,Generation\$1} . \partial_{mu\$1}\left[l_{SP\$2,Generation\$1}\right] \gamma^{mu\$1} . P_{-SP\$1,SP\$2} - \\
&e\, A_{mu\$1}\, \overline{l}_{SP\$1,Generation\$1} . l_{SP\$2,Generation\$1}\, \gamma^{mu\$1} . P_{-SP\$1,SP\$2} + \\
&i\, \overline{uq}_{SP\$1,Generation\$1,Colour\$1} . \partial_{mu\$1}\left[uq_{SP\$2,Generation\$1,Colour\$1}\right] \gamma^{mu\$1} . P_{-SP\$1,SP\$2} + \\
&\frac{2}{3}\, e\, A_{mu\$1}\, \overline{uq}_{SP\$1,Generation\$1,Colour\$1} . uq_{SP\$2,Generation\$1,Colour\$1}\, \gamma^{mu\$1} . P_{-SP\$1,SP\$2} + \\
&i\, \overline{vl}_{SP\$1,Generation\$1} . \partial_{mu\$1}\left[vl_{SP\$2,Generation\$1}\right] \gamma^{mu\$1} . P_{-SP\$1,SP\$2} + \\
&i\, \overline{dq}_{SP\$1,Generation\$1,Colour\$1} . \partial_{mu\$1}\left[dq_{SP\$2,Generation\$2,Colour\$1}\right] \delta_{Generation\$1,Generation\$2}\, \gamma^{mu\$1} . P_{-SP\$1,SP\$2} + \\
&i\, \overline{dq}_{SP\$1,Generation\$1,Colour\$1} . \partial_{mu\$1}\left[dq_{SP\$2,Generation\$1,Colour\$1}\right] \gamma^{mu\$1} . P_{+SP\$1,SP\$2} - \\
&\frac{1}{3}\, e\, A_{mu\$1}\, \overline{dq}_{SP\$1,Generation\$1,Colour\$1} . dq_{SP\$2,Generation\$1,Colour\$1}\, \gamma^{mu\$1} . P_{+SP\$1,SP\$2} + \\
&i\, \overline{l}_{SP\$1,Generation\$1} . \partial_{mu\$1}\left[l_{SP\$2,Generation\$1}\right] \gamma^{mu\$1} . P_{+SP\$1,SP\$2} - \\
&e\, A_{mu\$1}\, \overline{l}_{SP\$1,Generation\$1} . l_{SP\$2,Generation\$1}\, \gamma^{mu\$1} . P_{+SP\$1,SP\$2} + \\
&i\, \overline{uq}_{SP\$1,Generation\$1,Colour\$1} . \partial_{mu\$1}\left[uq_{SP\$2,Generation\$1,Colour\$1}\right] \gamma^{mu\$1} . P_{+SP\$1,SP\$2} + \\
&\frac{2}{3}\, e\, A_{mu\$1}\, \overline{uq}_{SP\$1,Generation\$1,Colour\$1} . uq_{SP\$2,Generation\$1,Colour\$1}\, \gamma^{mu\$1} . P_{+SP\$1,SP\$2}
\end{aligned}
$$

Main method: Feynman rules

Extraction of all N-point interactions from the Lagrangian (with N>2)

In[50]:= `FeynmanRules[LGauge] // Simplify`

```
Starting Feynman rule calculation.
Expanding the Lagrangian...
Collecting the different structures that enter the vertex.
8 possible non-zero vertices have been found -> starting the computation: 8 / 8.
8 vertices obtained.
```

Out[50]=

$$\Big\{\Big\{\{\{G,1\},\{G,2\},\{G,3\}\},\ g_s\, f_{a_1,a_2,a_3}\left(p_1^{\mu_3}\eta_{\mu_1,\mu_2}-p_2^{\mu_3}\eta_{\mu_1,\mu_2}-p_1^{\mu_2}\eta_{\mu_1,\mu_3}+p_3^{\mu_2}\eta_{\mu_1,\mu_3}+p_2^{\mu_1}\eta_{\mu_2,\mu_3}-p_3^{\mu_1}\eta_{\mu_2,\mu_3}\right)\Big\},$$

$$\Big\{\{\{G,1\},\{G,2\},\{G,3\},\{G,4\}\},\ i\, g_s^2\big(f_{a_1,a_2,\mathrm{Gluon\$1}}f_{a_3,a_4,\mathrm{Gluon\$1}}\left(\eta_{\mu_1,\mu_4}\eta_{\mu_2,\mu_3}-\eta_{\mu_1,\mu_3}\eta_{\mu_2,\mu_4}\right)+ f_{a_1,a_3,\mathrm{Gluon\$1}}f_{a_2,a_4,\mathrm{Gluon\$1}}\left(\eta_{\mu_1,\mu_4}\eta_{\mu_2,\mu_3}-\eta_{\mu_1,\mu_2}\eta_{\mu_3,\mu_4}\right)+ f_{a_1,a_4,\mathrm{Gluon\$1}}f_{a_2,a_3,\mathrm{Gluon\$1}}\left(\eta_{\mu_1,\mu_3}\eta_{\mu_2,\mu_4}-\eta_{\mu_1,\mu_2}\eta_{\mu_3,\mu_4}\right)\big)\Big\},$$

$$\Big\{\{\{A,1\},\{W,2\},\{W^\dagger,3\}\},\ -i\,e\left(p_1^{\mu_3}\eta_{\mu_1,\mu_2}-p_2^{\mu_3}\eta_{\mu_1,\mu_2}-p_1^{\mu_2}\eta_{\mu_1,\mu_3}+p_3^{\mu_2}\eta_{\mu_1,\mu_3}+p_2^{\mu_1}\eta_{\mu_2,\mu_3}-p_3^{\mu_1}\eta_{\mu_2,\mu_3}\right)\Big\},$$

$$\Big\{\{\{A,1\},\{A,2\},\{W,3\},\{W^\dagger,4\}\},\ i\,e^2\left(\eta_{\mu_1,\mu_4}\eta_{\mu_2,\mu_3}+\eta_{\mu_1,\mu_3}\eta_{\mu_2,\mu_4}-2\,\eta_{\mu_1,\mu_2}\eta_{\mu_3,\mu_4}\right)\Big\},$$

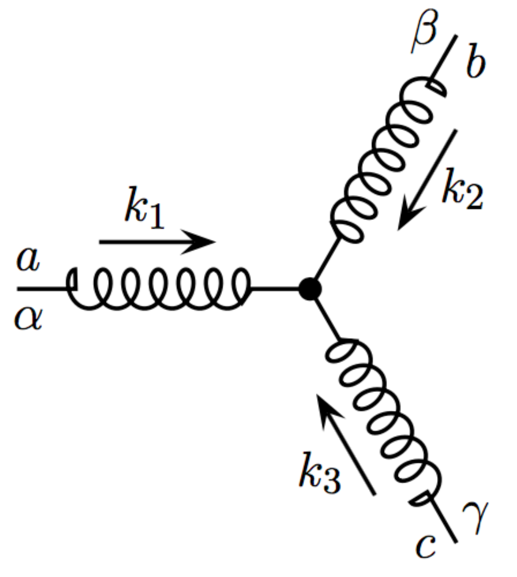
$$\Big\{\{\{W,1\},\{W^\dagger,2\},\{Z,3\}\},\ -\frac{i\,c_w\,e\left(p_1^{\mu_3}\eta_{\mu_1,\mu_2}-p_2^{\mu_3}\eta_{\mu_1,\mu_2}-p_1^{\mu_2}\eta_{\mu_1,\mu_3}+p_3^{\mu_2}\eta_{\mu_1,\mu_3}+p_2^{\mu_1}\eta_{\mu_2,\mu_3}-p_3^{\mu_1}\eta_{\mu_2,\mu_3}\right)}{s_w}\Big\},$$

$$\Big\{\{\{W,1\},\{W,2\},\{W^\dagger,3\},\{W^\dagger,4\}\},\ -\frac{i\,e^2\left(\eta_{\mu_1,\mu_4}\eta_{\mu_2,\mu_3}+\eta_{\mu_1,\mu_3}\eta_{\mu_2,\mu_4}-2\,\eta_{\mu_1,\mu_2}\eta_{\mu_3,\mu_4}\right)}{s_w^2}\Big\},$$

$$\Big\{\{\{A,1\},\{W,2\},\{W^\dagger,3\},\{Z,4\}\},\ -\frac{i\,c_w\,e^2\left(2\,\eta_{\mu_1,\mu_4}\eta_{\mu_2,\mu_3}-\eta_{\mu_1,\mu_3}\eta_{\mu_2,\mu_4}-\eta_{\mu_1,\mu_2}\eta_{\mu_3,\mu_4}\right)}{s_w}\Big\},$$

$$\Big\{\{\{W,1\},\{W^\dagger,2\},\{Z,3\},\{Z,4\}\},\ \frac{i\,c_w^2\,e^2\left(\eta_{\mu_1,\mu_4}\eta_{\mu_2,\mu_3}+\eta_{\mu_1,\mu_3}\eta_{\mu_2,\mu_4}-2\,\eta_{\mu_1,\mu_2}\eta_{\mu_3,\mu_4}\right)}{s_w^2}\Big\}\Big\}$$

- All 3-point and 4-point vertices found
- Example:



$$g_s f_{abc}\Big[(k_1-k_2)^\gamma\eta^{\alpha\beta}+(k_3-k_1)^\beta\eta^{\gamma\alpha}+(k_2-k_3)^\alpha\eta^{\beta\gamma}\Big]$$

⚠️ FEYNRULES index naming scheme

- ⋆ Colour: index a_i related to the i^{th} gluon (a = style for adjoint SU(3)$_C$ indices)
- ⋆ Spin: index μ_i = Lorentz index of the i^{th} gluon

The FeynmanRules function

The function FeynmanRules built with many options

- Restriction on the interactions to display: MaxParticles, MaxCanonicalDimension, *etc*.
- Selection of specific particles: Free, Contains, *etc*.
- ScreenOutput: displaying the vertices to the screen or not
- FlavorExpand: perform a flavour expansion (otherwise classes used)

```
In[51]:= Options[FeynmanRules]

Out[51]= {TeXOutput → MR$Null, ScreenOutput → False, Name → MR$Null, FlavorExpand → {},
  IndexExpand → False, ConservedQuantumNumbers :→ MR$QuantumNumbers, FermionFlow → False,
  MaxParticles → Automatic, MinParticles → Automatic, MaxCanonicalDimension → Automatic,
  MinCanonicalDimension → Automatic, SelectParticles → Automatic, Free → Automatic,
  Contains → Automatic, Exclude4Scalars → False, ApplyMomCons → True, NoDefinitions → False}
```

See the manual for more details on the function *FeynmanRules*

To phenomenology

From FEYNRULES to phenomenology

New model implemented in FEYNRULES

- Lagrangian checked
- Feynman rules extracted and checked
  ➙ Ready to export the model to other tools

Ready to export the model to MC tools

- CALCHEP / COMPHEP / MICROMEGAS
- FEYNARTS / FORMCALC
- UFO: MADGRAPH5_AMC@NLO / MADDM / HERWIG++ / WHIZARD / …

```
SetDirectory[NotebookDirectory[]];
lagr = ExpandIndices[LSM + LDMSIMPt];
WriteUFO[lagr, Output → "DMSimpt_v2_0_LOmix", ConservedQuantumNumbers → Q];
WriteCHOutput[lagr, Output → "DMSimpt_v2_0_CH", ConservedQuantumNumbers → Q]
WriteFeynArtsOutput[lagr, Output → "DMSimpt_v2_0_4FNS_FA", ConservedQuantumNumbers → Q];
```

Summary

The quest for new physics is ongoing…

- Heavily relies on HEP tools for (often automated) calculations
- FEYNRULES facilitates the implementation of new physics models

FEYNRULES: http://feynrules.irmp.ucl.ac.be & https://github.com/FeynRules/

- Straightforward implementation of new physics model in many tools
  ➙ O(100) models in the FEYNRULES online model database
- Shipped with its own computational modules
  - ⋆ A superspace module
  - ⋆ A decay package
  - ⋆ A mass diagonalisation module (ASPERGE)
  - ⋆ An NLO module

See the manual for more details on these modules

Try it on with your favorite model!

The model – compositeness and dark matter

Simplified modelling of composite dark matter (with partial compositeness)

- 3 mediators and 1 dark matter (mass eigenstates ≡ gauge eigenstates)

$$T_{L,R}\,,\quad \tilde{T}_{L,R}\,,\quad B_{L,R}\quad \text{and}\quad X$$

- Lagrangian → Free parameters: 4 masses and 2 couplings

$$\mathcal{L}_{\rm BSM} = \mathcal{L}_{\rm kin} - M_T \overline{T}\, T - M_B \overline{B}\, B - M_{\tilde{T}} \overline{\tilde{T}}\, \tilde{T} - \frac{1}{2} M_X X^2 + \Big(\lambda_Q \big[\, \overline{T_R}\, t_L + \overline{B_R}\, b_L \,\big] X + \lambda_T\, \overline{\tilde{T}_L}\, t_R\, X + \text{H.c.} \Big)$$