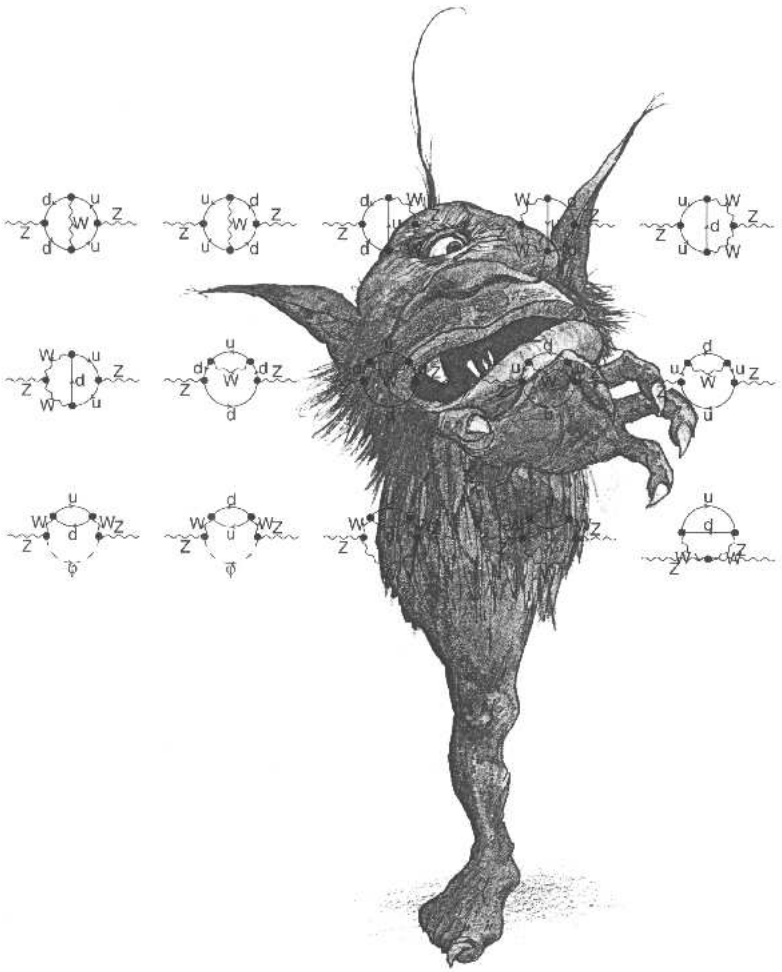
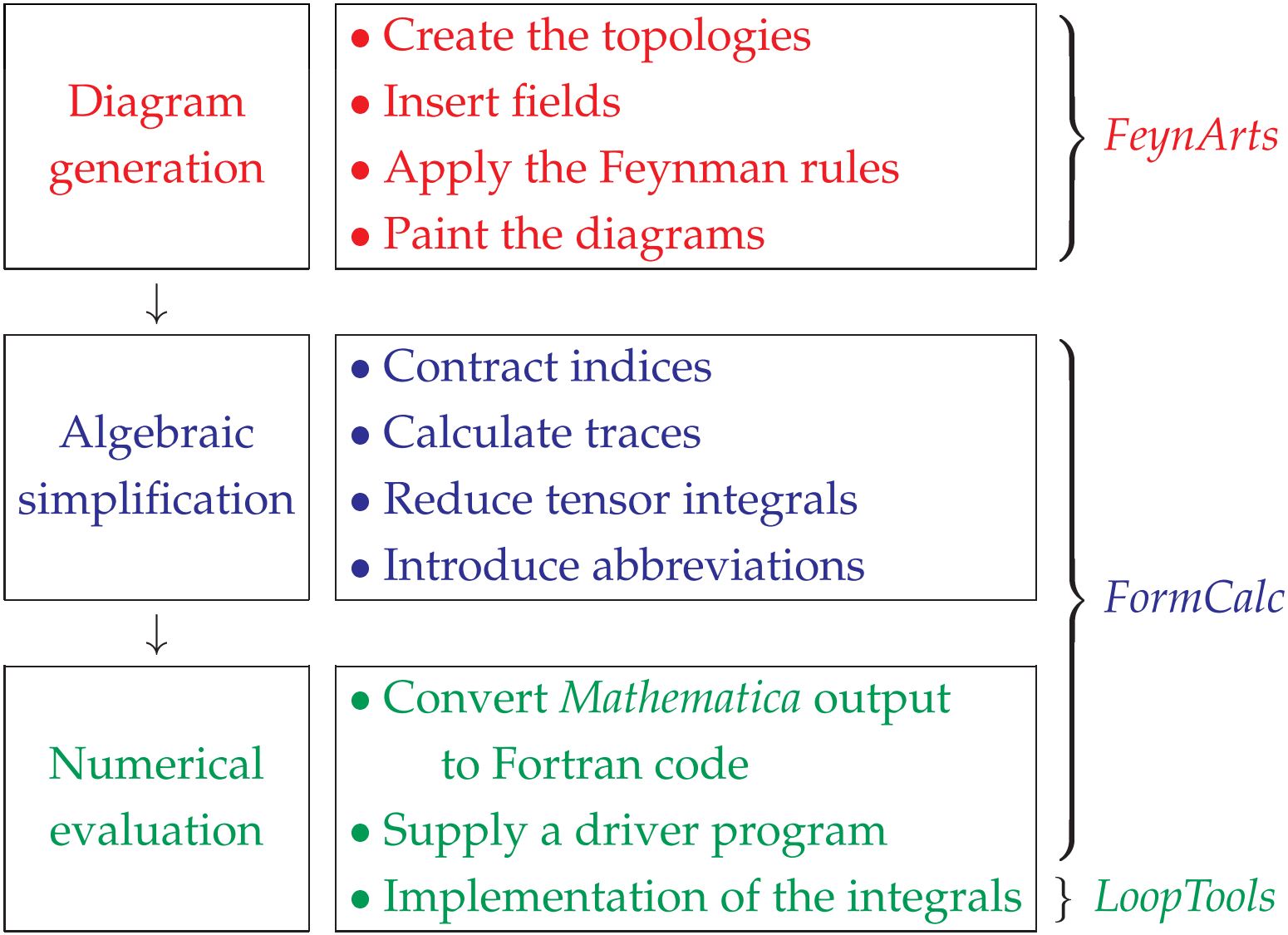
AUTOMATING ONE-LOOP CALCULATIONS WITH *FeynArts*, *FormCalc*, AND *LoopTools*

Thomas Hahn, Universität Karlsruhe

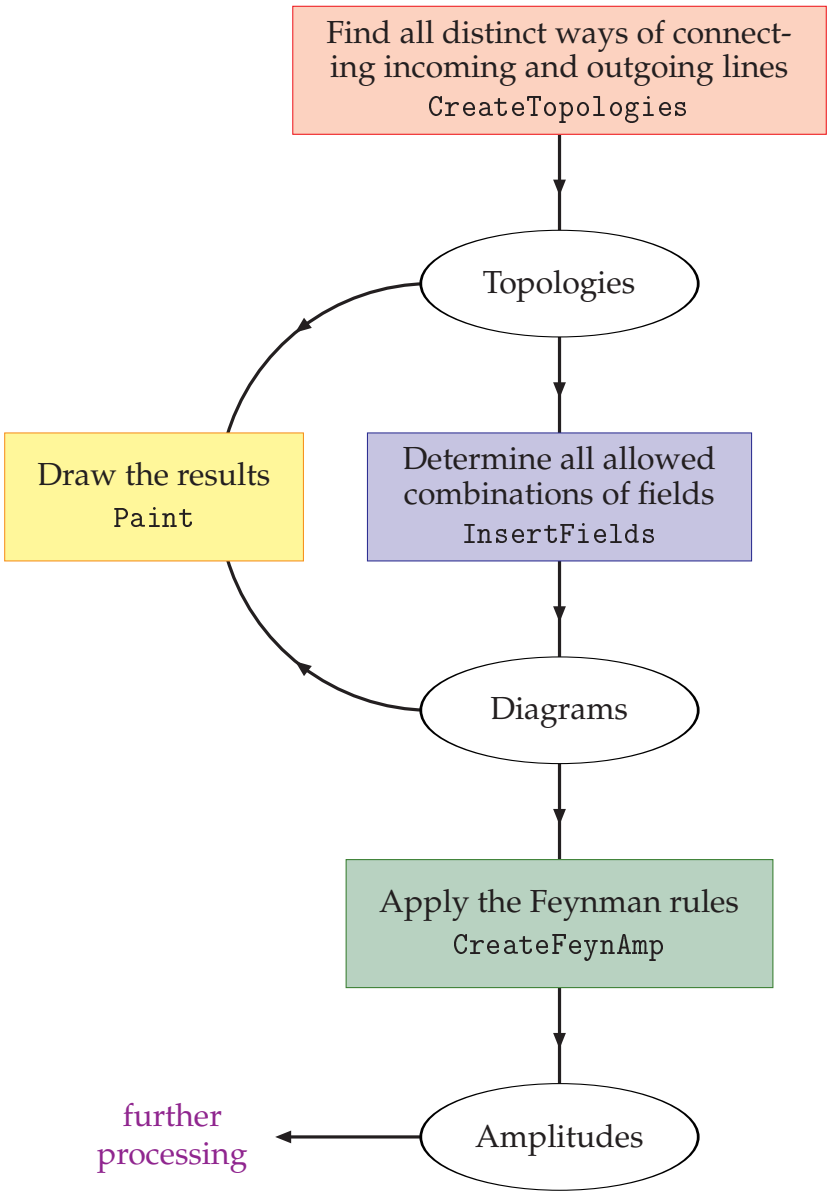


Thomas Hahn: Automatic Loop Calculations with FeynArts and FormCalc; talk given at Loops & Legs 2000, April 9-14, Bastei

STEPS FOR EVALUATING FEYNMAN DIAGRAMS



FeynArts



EXAMPLE: generating the Higgs self-energy

```
top = CreateTopologies[ 1, 1 -> 1 ]
```

one loop
one incoming particle
one outgoing particle

```
Paint[top]
```

```
ins = InsertFields[ top, S[1] -> S[1],
        Model -> SM ]
```

use the Standard Model

the name of the Higgs boson in the "SM" model file

```
Paint[ins]
```

```
amp = CreateFeynAmp[ins]
```

```
amp >> HiggsSelfEnergy.amp
```

FeynArts distinguishes **three levels of diagrams**:

Generic level, e.g.
`S, F, F`
(scalar–fermion–fermion)

Classes level, e.g.
`S[1], -F[2], F[2]`
(Higgs–antilepton–lepton)

Particles level, e.g.
`S[1], -F[2, {1}], F[2, {1}]`
(Higgs–positron–electron)

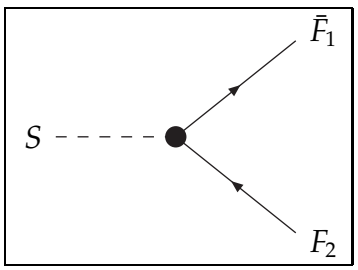
This distinction is actually quite natural when compared to the textbook definition of a coupling.

Why is it useful?

- Most algebraic simplifications can be carried out at Generic level, e.g. taking traces, tensor reduction.
- Summing over open indices of Classes-level diagrams in a loop is more economic than calculating directly at Particles level.

Scalar-Fermion-Fermion coupling in the electroweak Standard Model

Reference: A. Denner, Forts. Physik, **41** (1993) 307.



$= ie(G_{\omega_-}\omega_- + G_{\omega_+}\omega_+)$

Generic Level

Classes Level

where for the following combinations of S, $\bar{F}_1$, and F_2 G_{ω_+} and G_{ω_-} take these values:

$$\mathrm{H}\bar{f}_i f_j \quad : \quad \begin{cases} G_{\omega_+} = -\frac{1}{2s}\frac{1}{M_W}\Big[\delta_{ij} m_{f,i}\Big(1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{f,i}}{m_{f,i}} - \frac{\delta M_W}{M_W} + \frac{1}{2}\delta Z^H\Big) \\ \qquad + \frac{1}{2}(m_{f,i}\delta Z_{ij}^{f,R} + \delta Z_{ij}^{f,L\dagger} m_{f,j})\Big], \\ G_{\omega_-} = -\frac{1}{2s}\frac{1}{M_W}\Big[\delta_{ij} m_{f,i}\Big(1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{f,i}}{m_{f,i}} - \frac{\delta M_W}{M_W} + \frac{1}{2}\delta Z^H\Big) \\ \qquad + \frac{1}{2}(m_{f,i}\delta Z_{ij}^{f,L} + \delta Z_{ij}^{f,R\dagger} m_{f,j})\Big], \end{cases}$$

$$\chi\bar{f}_i f_j \quad : \quad \begin{cases} G_{\omega_+} = \mathrm{i}\frac{1}{2s} 2 I^3_{W,f}\frac{1}{M_W}\Big[\delta_{ij} m_{f,i}\Big(1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{f,i}}{m_{f,i}} - \frac{\delta M_W}{M_W}\Big) \\ \qquad + \frac{1}{2}(m_{f,i}\delta Z_{ij}^{f,R} + \delta Z_{ij}^{f,L\dagger} m_{f,j})\Big], \\ G_{\omega_-} = -\mathrm{i}\frac{1}{2s} 2 I^3_{W,f}\frac{1}{M_W}\Big[\delta_{ij} m_{f,i}\Big(1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{f,i}}{m_{f,i}} - \frac{\delta M_W}{M_W}\Big) \\ \qquad + \frac{1}{2}(m_{f,i}\delta Z_{ij}^{f,L} + \delta Z_{ij}^{f,R\dagger} m_{f,j})\Big], \end{cases}$$

$$\phi^+\bar{u}_i d_j \quad : \quad \begin{cases} G_{\omega_+} = -\frac{1}{\sqrt{2}s}\frac{1}{M_W}\Big[V_{ij} m_{d,j}\Big(1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{d,j}}{m_{d,j}} - \frac{\delta M_W}{M_W}\Big) + \delta V_{ij} m_{d,j} \\ \qquad + \frac{1}{2}\sum_k(\delta Z_{ik}^{u,L\dagger} V_{kj} m_{d,j} + V_{ik} m_{d,k}\delta Z_{kj}^{d,R})\Big], \\ G_{\omega_-} = \frac{1}{\sqrt{2}s}\frac{1}{M_W}\Big[m_{u,i} V_{ij}\Big(1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{u,i}}{m_{u,i}} - \frac{\delta M_W}{M_W}\Big) + m_{u,i}\delta V_{ij} \\ \qquad + \frac{1}{2}\sum_k(\delta Z_{ik}^{u,R\dagger} m_{u,k} V_{kj} + m_{u,i} V_{ik}\delta Z_{kj}^{d,L})\Big], \end{cases}$$

$$\phi^-\bar{d}_j u_i \quad : \quad \begin{cases} G_{\omega_+} = \frac{1}{\sqrt{2}s}\frac{1}{M_W}\Big[V_{ji}^\dagger m_{u,i}\Big(1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{u,i}}{m_{u,i}} - \frac{\delta M_W}{M_W}\Big) + \delta V_{ji}^\dagger m_{u,i} \\ \qquad + \frac{1}{2}\sum_k(\delta Z_{jk}^{d,L\dagger} V_{ki}^\dagger m_{u,i} + V_{jk}^\dagger m_{u,k}\delta Z_{ki}^{u,R})\Big], \\ G_{\omega_-} = -\frac{1}{\sqrt{2}s}\frac{1}{M_W}\Big[m_{d,j} V_{ji}^\dagger\Big(1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{d,j}}{m_{d,j}} - \frac{\delta M_W}{M_W}\Big) + m_{d,j}\delta V_{ji}^\dagger \\ \qquad + \frac{1}{2}\sum_k(\delta Z_{jk}^{d,R\dagger} m_{d,k} V_{ki}^\dagger + m_{d,j} V_{jk}^\dagger\delta Z_{ki}^{u,L})\Big], \end{cases}$$

$$\phi^+\bar{\nu}_i l_j \quad : \quad \begin{cases} G_{\omega_+} = -\frac{1}{\sqrt{2}s}\frac{m_{l,i}}{M_W}\delta_{ij}\Big[1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{l,i}}{m_{l,i}} - \frac{\delta M_W}{M_W} + \frac{1}{2}(\delta Z_{ii}^{\nu,L\dagger} + \delta Z_{ii}^{l,R})\Big], \\ G_{\omega_-} = 0, \end{cases}$$

$$\phi^-\bar{l}_j\nu_i \quad : \quad \begin{cases} G_{\omega_+} = 0, \\ G_{\omega_-} = -\frac{1}{\sqrt{2}s}\frac{m_{l,i}}{M_W}\delta_{ij}\Big[1 + \delta Z_e - \frac{\delta s}{s} + \frac{\delta m_{l,i}}{m_{l,i}} - \frac{\delta M_W}{M_W} + \frac{1}{2}(\delta Z_{ii}^{l,R\dagger} + \delta Z_{ii}^{\nu,L})\Big]. \end{cases}$$

Particles Level: Insert integers for i and j (1, 2, 3 in this case)

Generating diagrams in just a few lines – aren't there any strings attached?

Yes, one has to set up, once and for all, a MODEL FILE containing the couplings.

E.g. the SFF coupling is declared by

- in the Generic model file:

```
AnalyticalCoupling[ s1 F[i, mom1], s2 F[j, mom2],
  s3 S[k, mom3] ] ==
    G[1][ s1 F[i], s2 F[j], s3 S[k] ] .
      { NonCommutative[ ChiralityProjector[-1] ],
        NonCommutative[ ChiralityProjector[+1] ] }
```

coupling vector

kinematic vector

- in the Classes model file (here for $\overline{\ell}_{j_2}\nu_{j_1}\phi^-$):

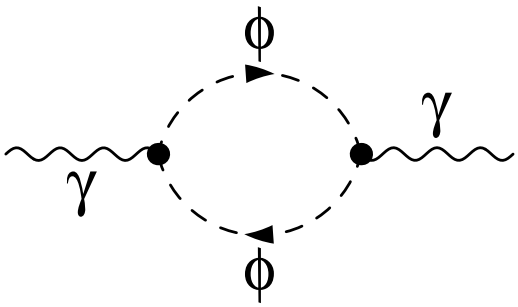
```
C[ -F[2, {j2}], F[1, {j1}], S[3] ] ==
  -I EL/(Sqrt[2] SW) Mass[F[2, {j2}]]/MW *
  IndexDelta[j1, j2] *
  { {1, dZe1 - dSW1/SW + dMWsq1/(2 MW^2) +
    dMf1[2, {j2}]/Mass[F[2, {j2}]] -
    1/2 (Conjugate[dZfR1[2, {j2}]] +
      dZfL1[1, {j1}]) },
    {0, 0} }
```

tree-level coupling

counter term

neutrinos have no right-handed coupling

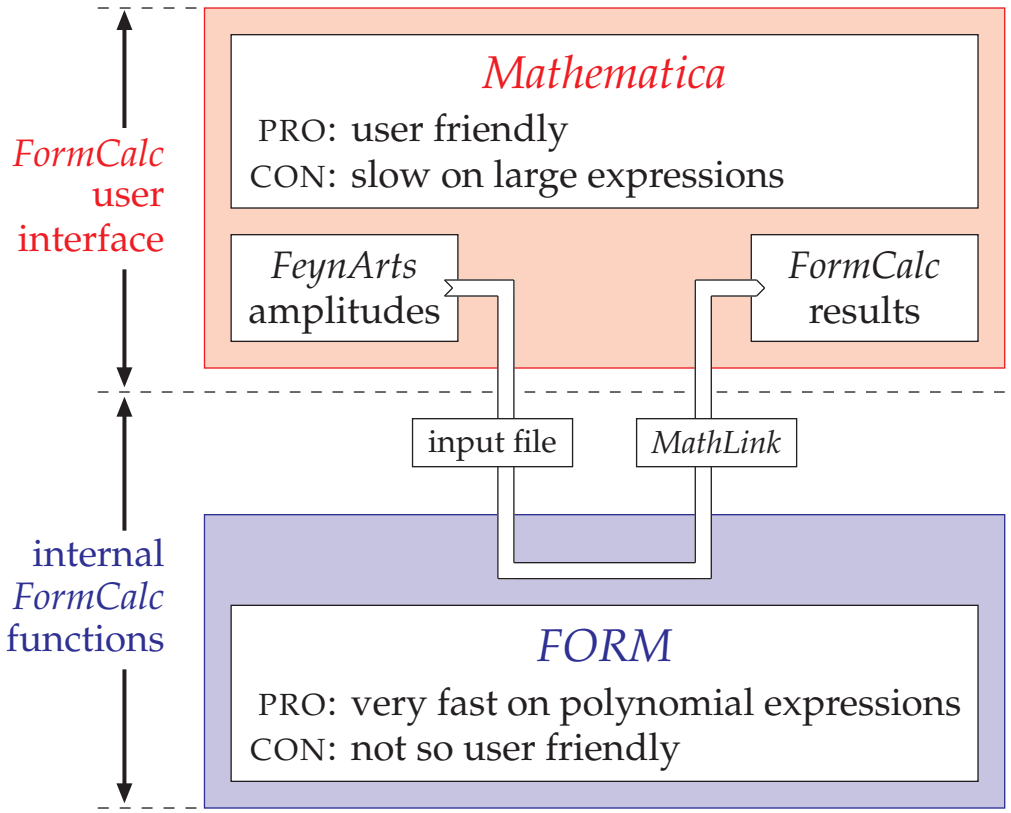
Sample `CreateFeynAmp` output:




```
FeynAmp[
  GraphName[V1V1, Topology == 1, Generic == 1], ..... name

  Integral[q1], ........................... integration momentum

  ( I/32 RelativeCF ..................................... prefactor
    FeynAmpDenominator[1/(q1^2 - Mass[S[G3]]^2), ... loop denominators
      1/((-p1 + q1)^2 - Mass[S[G4]]^2)]
    (p1 - 2 q1)[L1] (-p1 + 2 q1)[L2] .....kin. coupling structure
    ep[V[1], p1, L1] ep*[V[1], k1, L2] ..... polarisation vectors
    G(0)SSV[(Mom[1] - Mom[2])[KI1[3]]] .....coupling constants
    G(0)SSV[(Mom[1] - Mom[2])[KI1[3]]] ) / Pi^4,

  {Mass[S[G3]], Mass[S[G4]], ..................... insertions
   G(0)SSV[(Mom[1] - Mom[2])[KI1[3]]],
   G(0)SSV[(Mom[1] - Mom[2])[KI1[3]]], RelativeCF} ->
     Insertions[Classes][{MW, MW, I EL, -I EL, 2}] ]
```

Thomas Hahn: Automatic Loop Calculations with FeynArts and FormCalc; talk given at Loops & Legs 2000, April 9-14, Bastei

FormCalc



EXAMPLE: Calculating the Higgs self-energy

```
In[1]:= << FormCalc`

FormCalc 2
by Thomas Hahn
last revision: 6 Mar 00

In[2]:= OneLoop[<< HiggsSelfEnergy.amp]
preparing FORM code in t25886.frm
I2 I4 I5 I1 I2 I1 I3
running FORM...  ok
```

$$\text{Out[2]=}\ \frac{\text{-3 Alfa Pair1 A0[MW2]}}{\text{2 Pi}} + \frac{\text{3 Alfa Pair1 B00[0,MW2,MW2]}}{\text{Pi}} + \left(\frac{\text{Alfa Pair1 A0[MLE2[Gen1]]}}{\text{Pi}} + \frac{\text{Alfa Pair1 A0[MQD2[Gen1]]}}{\text{3 Pi}} + \frac{\text{4 Alfa Pair1 A0[MQU2[Gen1]]}}{\text{3 Pi}} - \frac{\text{2 Alfa Pair1 B00[0,MLE2[Gen1],MLE2[Gen1]]}}{\text{Pi}} - \frac{\text{2 Alfa Pair1 B00[0,MQD2[Gen1],MQD2[Gen1]]}}{\text{3 Pi}} - \frac{\text{8 Alfa Pair1 B00[0,MQU2[Gen1],MQU2[Gen1]]}}{\text{3 Pi}}\right) * \text{SumOver[Gen1,3]}$$

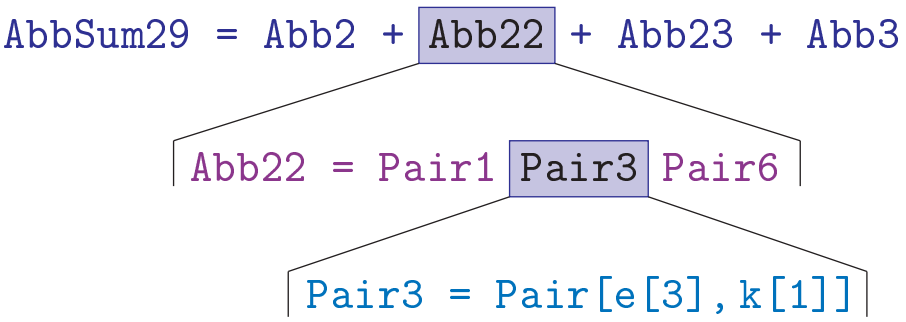
Typical term in the *FormCalc* output:

```
C0i[cc12, MW2, MW2, S, MW2, MZ2, MW2] *
  ( -4 Alfa2 CW2 MW2/SW2 S AbbSum16 +
    32 Alfa2 CW2/SW2 S^2 AbbSum28 +
    4 Alfa2 CW2/SW2 S^2 AbbSum30 -
    8 Alfa2 CW2/SW2 S^2 AbbSum7 +
    Alfa2 CW2/SW2 S (T - U) Abb1 +
    8 Alfa2 CW2/SW2 S (T - U) AbbSum29 )
```

= loop integral = automatically introduced abbreviations

= constants = kinematical variables

Introducing abbreviations is important — just look:



Without abbreviations, the result would for each `AbbSum29` contain

```
Pair[e[1], e[2]] Pair[e[3], k[1]] Pair[e[4], k[1]] +
Pair[e[1], e[2]] Pair[e[3], k[2]] Pair[e[4], k[1]] +
Pair[e[1], e[2]] Pair[e[3], k[1]] Pair[e[4], k[2]] +
Pair[e[1], e[2]] Pair[e[3], k[2]] Pair[e[4], k[2]]
```

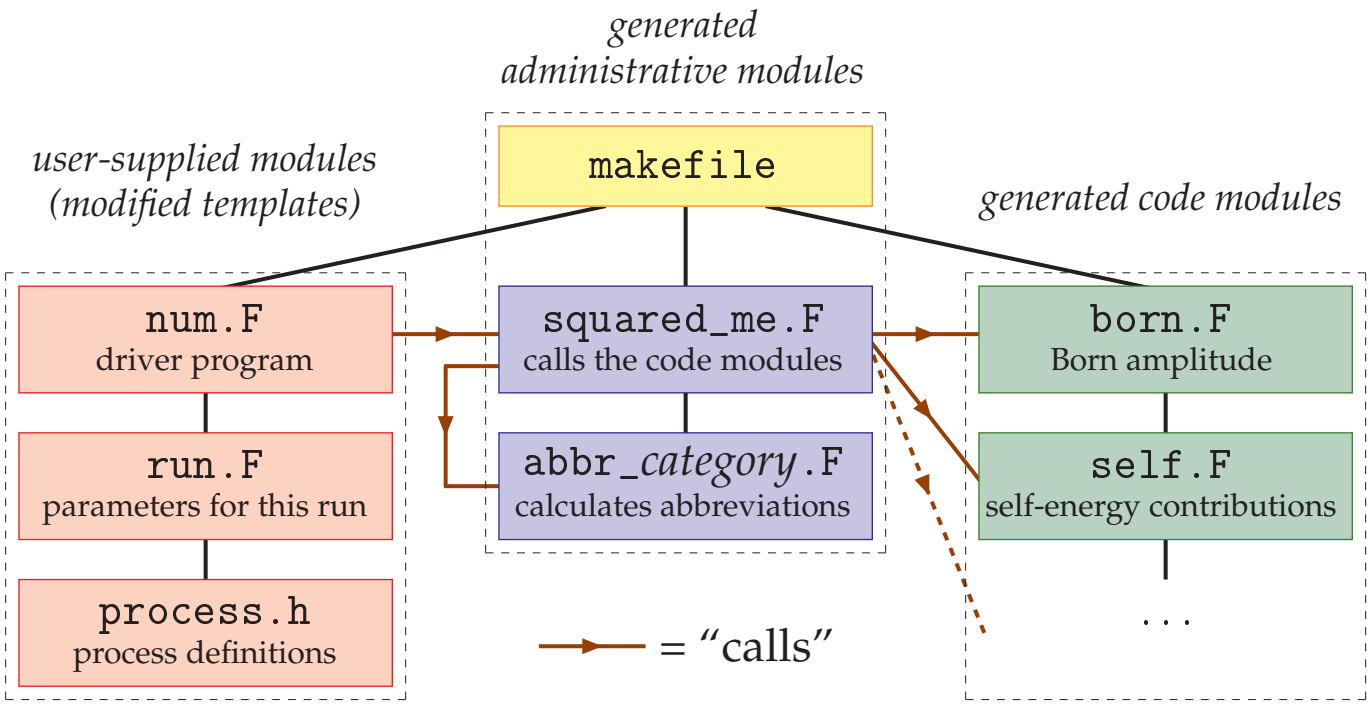
The Numerical Evaluation

. . . could be done in *Mathematica* directly, but quickly becomes too slow for large amplitudes.

Solution: translate into a Fortran program.

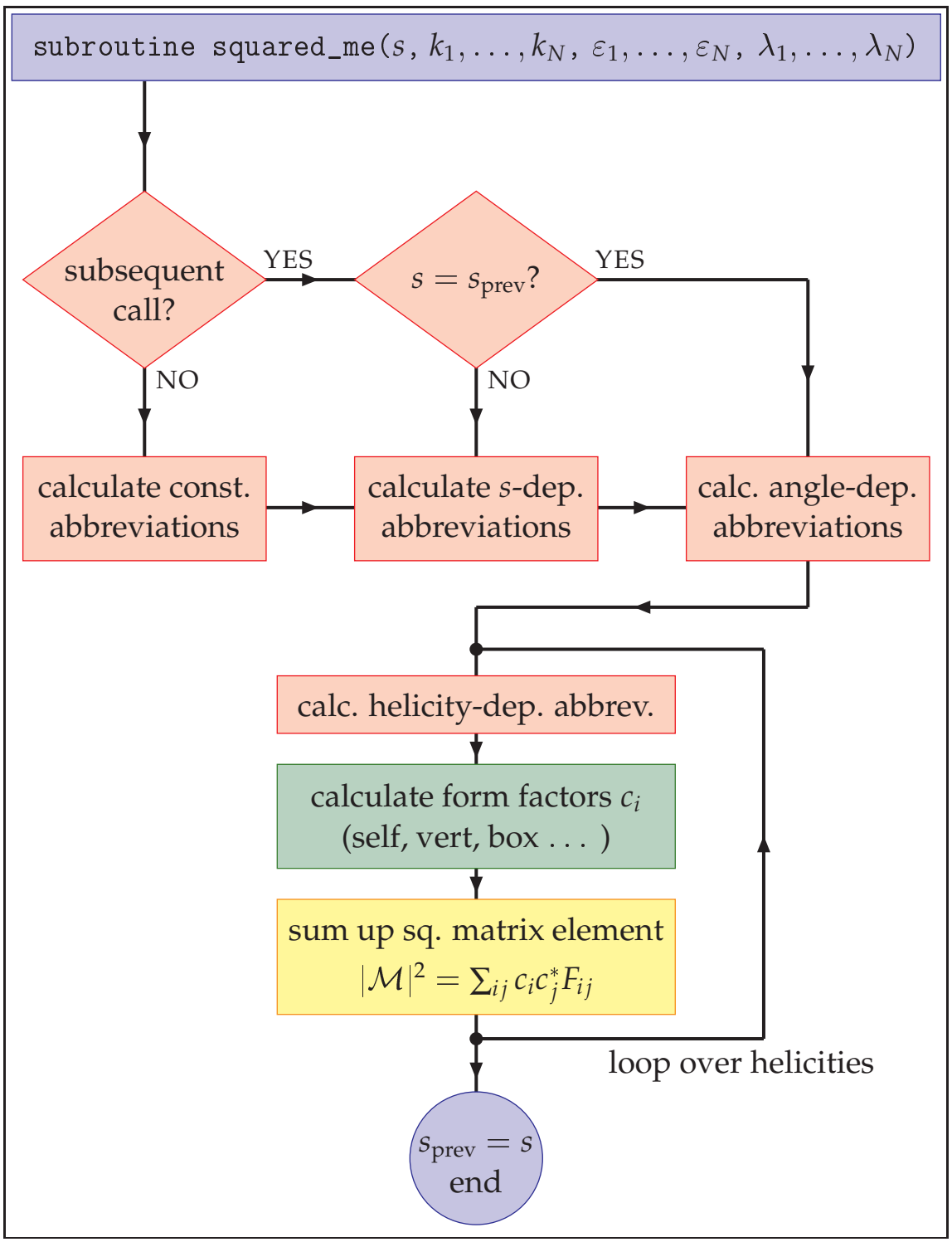
Philosophy: the user should not have to modify the generated code.

This means: all necessary files have to be generated.



Example of code module:

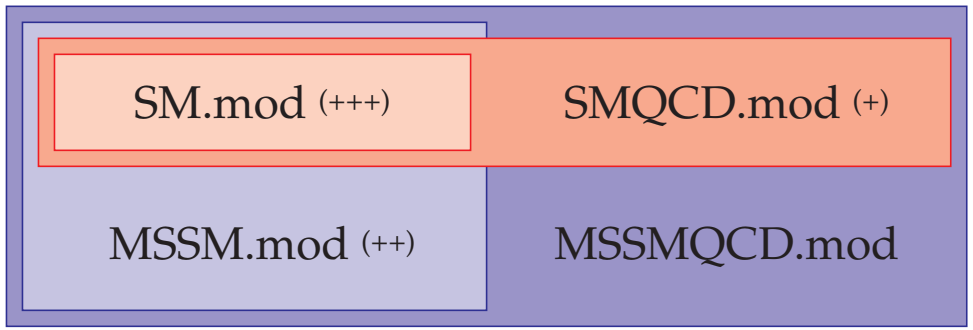
```
        subroutine self
        implicit character (a-s,u-z)
        implicit double complex (t)
#include "vars.h"
        Cloop(1) = Cloop(1) +
     -   ((3*Alfa2)/(8.D0*SW2**2)*(Abb1*cint44*MH2) +
     -      (9*Alfa2)/(8.D0*SW2**2)*(Abb1*cint5*MH2**2) +
     -      Alfa2/(4.D0*SW2**2)*(Abb1*cint41*(MH2 + 6*MW2)) +
     -      (Alfa2*MW2)/SW2**2*(Abb1*cint43*S) +
     -      (Alfa2*MW2)/(2.D0*CW2*SW2**2)*(Abb1*cint46*S) + ...
```


subroutine squared_me(s, k1, ..., kN, ε1, ..., εN, λ1, ..., λN)
subsequent call?
YES
s = sprev?
YES
NO
NO
calculate const. abbreviations
calculate s-dep. abbreviations
calc. angle-dep. abbreviations
calc. helicity-dep. abbrev.
calculate form factors ci (self, vert, box ...)
sum up sq. matrix element |M|2 = Σij ci cj* Fij
loop over helicities
sprev = s
end

Supersymmetric Calculations with *FeynArts* and *FormCalc*

FeynArts	✘ fermion-number violating vertices	✘ Denner/Eck/Hahn/Küblbeck algorithm (flipping rules)
	✘ actual couplings	✘ MSSM.mod
FormCalc	✘ dim. reG. violates SUSY	✘ resort to CDR ≡ dim. reD.
Fortran code	✘ numerical values for the MSSM parameters	✘ new initialization module `mssm_ini.F`

Current Status of the Model Files



Some Recent Checks:

$e^+e^- \to \bar{t}\,t$ (cs, th),

$e^+e^- \to H^+H^-$ (ak, jg, cs, th),

$\bar{q}\,q \to \bar{t}\,t$ (sb, th),

$e^+e^- \to \tilde{\chi}^{0,+}\tilde{\chi}^{0,-}$ (tb),

various self-energies (cs, mf, tb, th).

The three programs

FeynArts, *FormCalc*, and *LoopTools*

can be used for one-loop calculations and they work together smoothly. Their main features are:

FeynArts

- Three-level field structure.
- Support for fermion-number violating theories.
- Model files can be supplied/modified by the user. (Currently included: QED, SM, QCD; in preparation: MSSM)
- Sophisticated graphics routines, output in PS or LaTeX.
- `http://www.feynarts.de`

FormCalc

- Fast, because of *FORM*.
- Analytical calculation as far as possible.
- Numerics package supplies among other things the translation into Fortran code and the Fortran driver programs.
- `http://www.feynarts.de/formcalc`

LoopTools

- Stable, easy-to-use implementation of the one-loop scalar and tensor integrals in Fortran, C++, and *Mathematica*.
- `http://www.feynarts.de/looptools`