Forcer: parametric reduction of 4-loop massless propagators

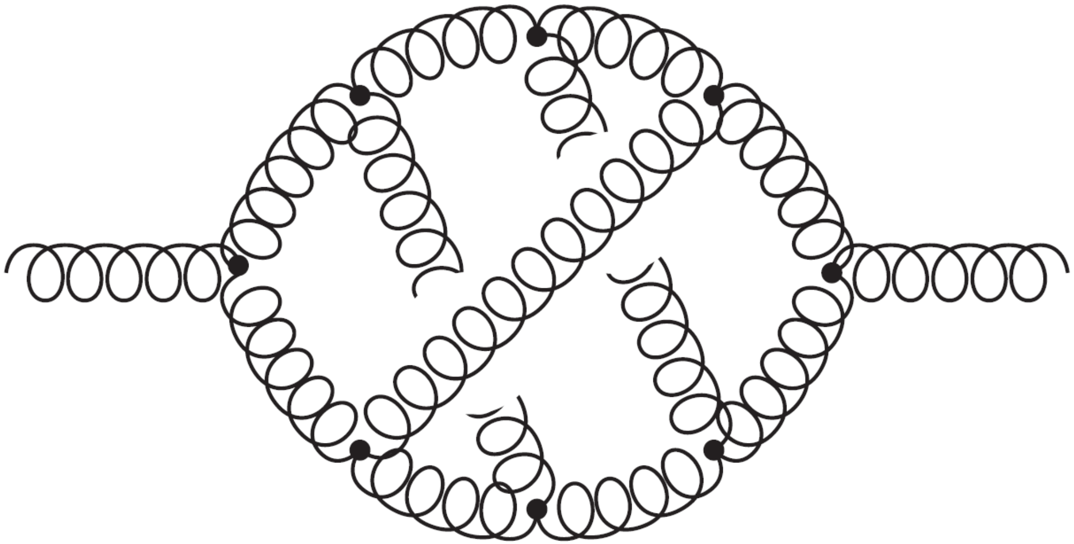
Takahiro Ueda
Nikhef, The Netherlands

Collaboration with:
Ben Ruijl, Jos Vermaseren

25 October 2016, UZH, Zürich

This Talk

- Efficient analytical computation of 4-loop massless propagator-type Feynman integrals. An example with this class of integrals:



- Why efficiency? Physics motivation
- How to achieve this? ⇨ New program Forcer

  [Ben Ruijl, TU, Jos Vermaseren, in preparation]
- Results

Physics motivation

Recent N^{3}LO Higgs calculations

- Much effort has been made on theoretical predictions for the LHC Run2: NLO, NNLO QCD corrections
- N^{3}LO inclusive ggF Higgs production cross section

[Anastasiou et al., JHEP 1605 (2016) 058, arXiv:1602.00695]

$$\sigma = 48.58\,\mathrm{pb}^{+2.22\,\mathrm{pb}\,(+4.56\%)}_{-3.27\,\mathrm{pb}\,(-6.72\%)}\ (\mathrm{theory}) \pm 1.56\,\mathrm{pb}\,(3.20\%)\ (\mathrm{PDF}+\alpha_s)$$

at $m_H = 125\mathrm{GeV}$, $\sqrt{s} = 13\mathrm{TeV}$

  - Reduced the theoretical uncertainty considerably
- N^{3}LO inclusive VBF Higgs [Dreyer and Karlberg, PRL 117 (2016) 072001, arXiv:1606.00840]

N^{3}LO inclusive DIS: Moch, Vermaseren, Vogt '05; for F_3: '08

- Now the N^{3}LO era has come!

N^3LO parton distributions?

- The N^3LO Higgs production cross sections were computed with the NNLO PDFs
- Question: effect of the N^3LO PDFs?

for ignorance of the N^3LO PDFs

δ(PDF-TH)
± 0.56 pb
$\pm 1.16\%$

with the PDF4LHC recommendation

δ(PDF)
± 0.90 pb
$\pm 1.86\%$

taken from arXiv:1602.00695

  - May not affect results...

cf. Forte, Andrea Isgrò, Vita, PLB731 (2014) 136, arXiv:1312.6688

- Ideally they should be computed with the N^3LO PDFs

⇒ 4-loop splitting functions (unknown)

4-loop splitting functions?

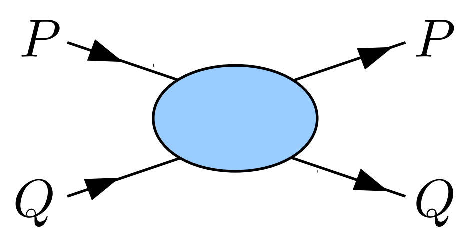
- N-th moment of the splitting function $P_{ab}(x)$

$$\gamma_{ab}(N) = -\int_0^1 dx\, x^{N-1} P_{ab}(x)$$

can be computed from the partonic forward scattering

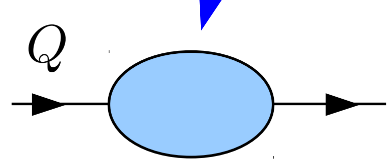
[Gorishnii, Larin, Tkachev '83; Gorishnii, Larin '87]

$$\frac{Q^{\{\mu_1} \ldots Q^{\mu_N\}}}{N!} \frac{\partial^N}{\partial P^{\mu_1} \ldots \partial P^{\mu_N}}$$



$$\Big|_{P=0}$$

$\Rightarrow \sum$



N-dependence

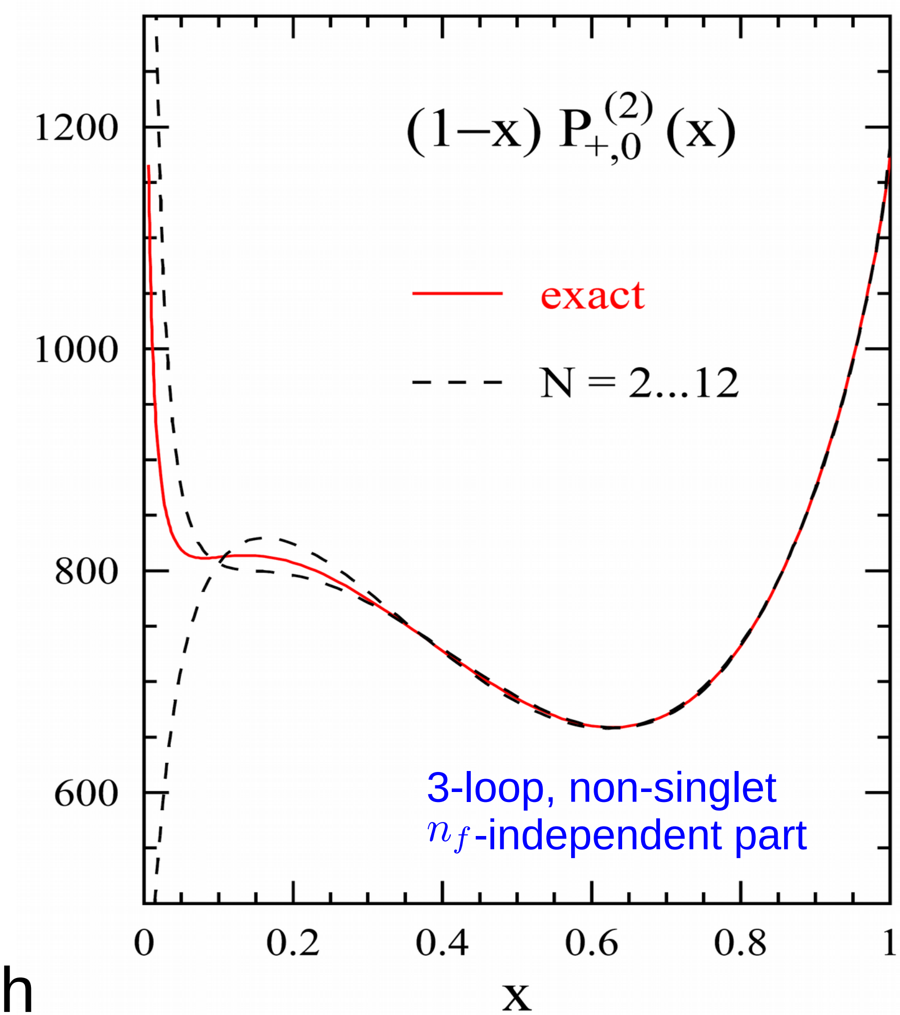
Massless propagator-type integrals
with N-dependence

single pole: splitting function
finite part: coefficient function

3-loop full N-dependence (hence x-dependence): Moch, Vermaseren, Vogt '04

Approximated 4-loop splitting functions: our motivation

- Full N-dependence in 4-loops: out of reach
- Fixed N = 2, 4, 6, …
  Their information gives an approximation to / upper bounds of uncertainty of the result (especially at large x)
- If we go for more higher N, then get a better estimation
- Very time-consuming for high N
  Requires efficiency beyond the reach of, e.g., the Laporta's algorithm

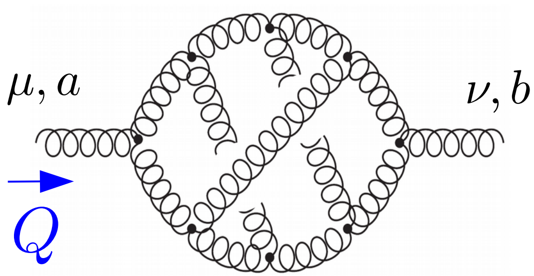


taken from Moch, Vermaseren, Vogt
NPB 688 (2004) 101 [arXiv:hep-ph/0403192]

How to achieve such an efficiency?

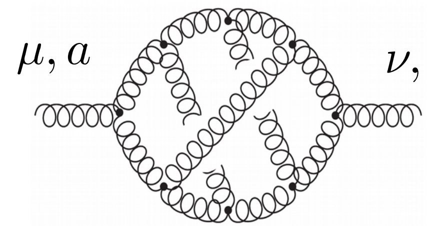
Feynman integral calculus

- Generate diagrams, Feynman rules, projection to scalarize integrals



$$= A(Q^2 g^{\mu\nu} - Q^\mu Q^\nu)\delta^{ab}$$

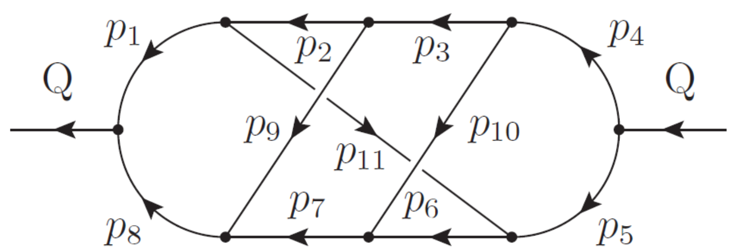
$$A = \frac{1}{N_A}\frac{1}{D-1}\frac{1}{Q^2} g_{\mu\nu}\delta^{ab} \times$$



dimensional regularization

$D = 4 - 2\epsilon$

$$= \sum$$



$$F(n_1, \ldots, n_{14}) = \int d^D p_1 \ldots d^D p_4 \frac{(p_2 \cdot p_4)^{-n_{12}} (Q \cdot p_2)^{-n_{13}} (Q \cdot p_3)^{-n_{14}}}{(p_1^2)^{n_1} \ldots (p_{11}^2)^{n_{11}}}$$

- Many integrals with various $n_1, \ldots, n_{14}$ (indices)

Feynman integral calculus

- The "standard" way to proceed:
  (1) Reduction to master integrals (MIs) via integration-by-parts identities (IBPs) [Chetyrkin, Tkachov '81]
  (2) Evaluation of MIs
- 4-loop massless propagator-type MIs are known [Baikov, Chetyrkin '10; Lee, Smirnov^2 '11]

  The problem is (1) Reduction

- Generic IBP reduction algorithms for any processes, very general
  - Laporta's algorithm, Baikov's method, Lee's "LiteRed", ...
    [Laporta '01] AIR, FIRE, Reduze,... [Baikov '96; '05] [Lee '12; '13]
- "Mincer" algorithm [Chetyrkin, Tkachov '81]
  - Specialized reduction for massless propagator-type integrals (up to 3-loops), very efficient

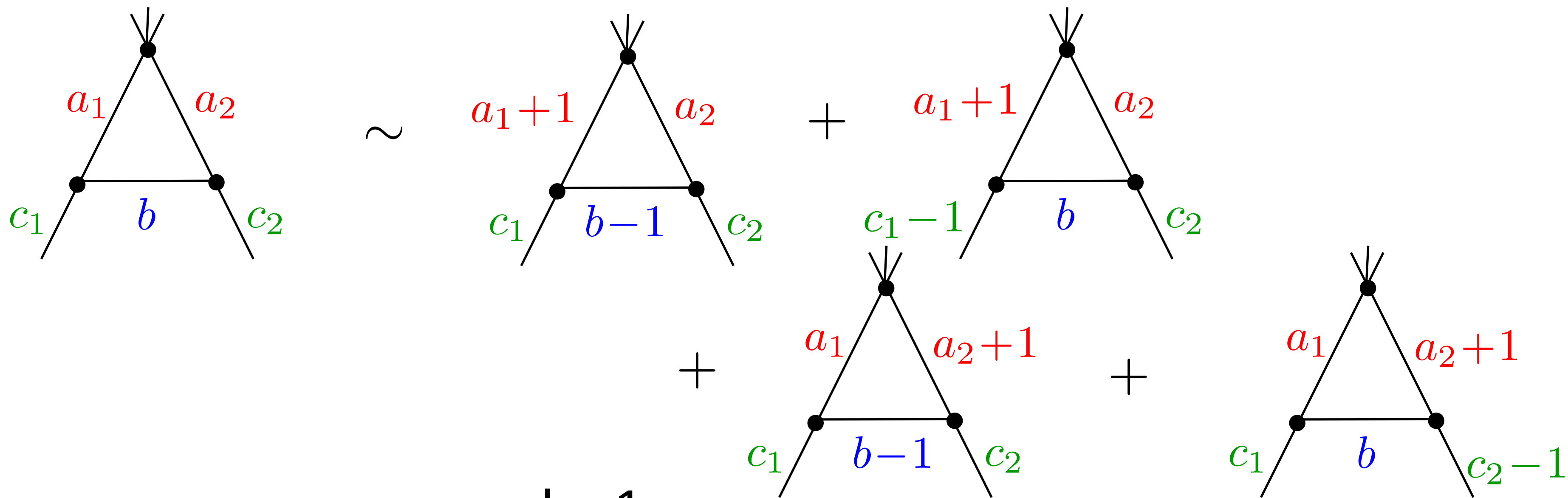
  Can be extended to 4-loops?

Mincer approach

Triangle rule

[Chetyrkin, Tkachov '81]

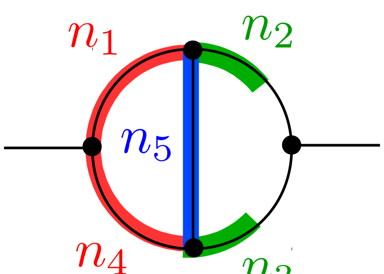
- Topology substructure tells how to reduce integrals into simpler ones via IBP identities



- Decreases b or c_1 or c_2 by 1

- From positive integer indices, recursive use of the triangle rule makes $b=0$ or $c_1=0$ or $c_2=0$ (removal of a line)

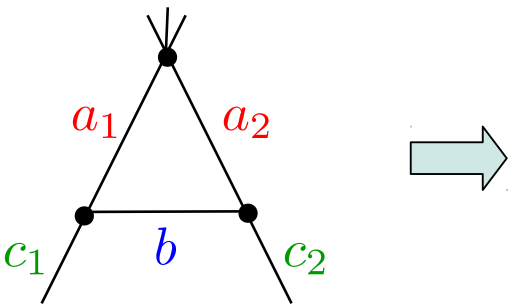
  sums of integrals in simpler topologies

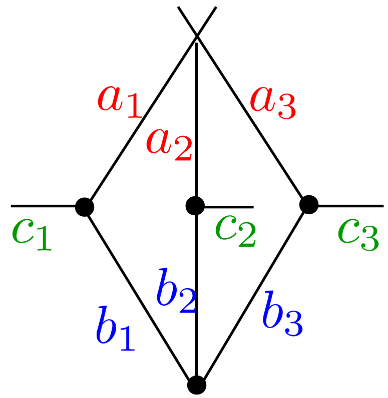


Diamond rule

[Ruijl, TU, Vermaseren ’15]

- Extension of the triangle rule to multi-loop diamond-shaped (sub-)diagrams:

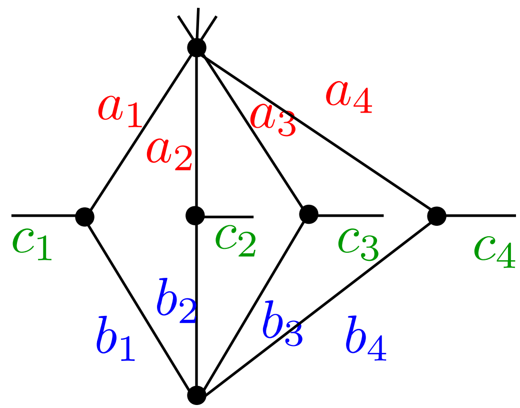




Increases a_1, a_2, a_3

Decreases b_1, b_2, b_3

c_1, c_2, c_3



Increases a_1, a_2, a_3, a_4

Decreases b_1, b_2, b_3, b_4

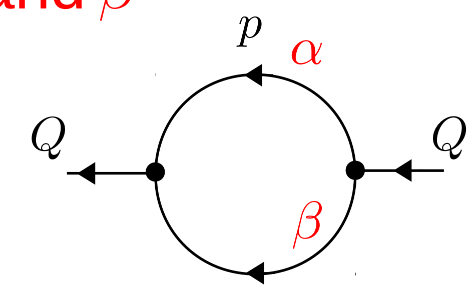
c_1, c_2, c_3, c_4

Appear from 4-loops

1-loop insertion integrals

- Perform massless 1-loop integrals with arbitrary α and β

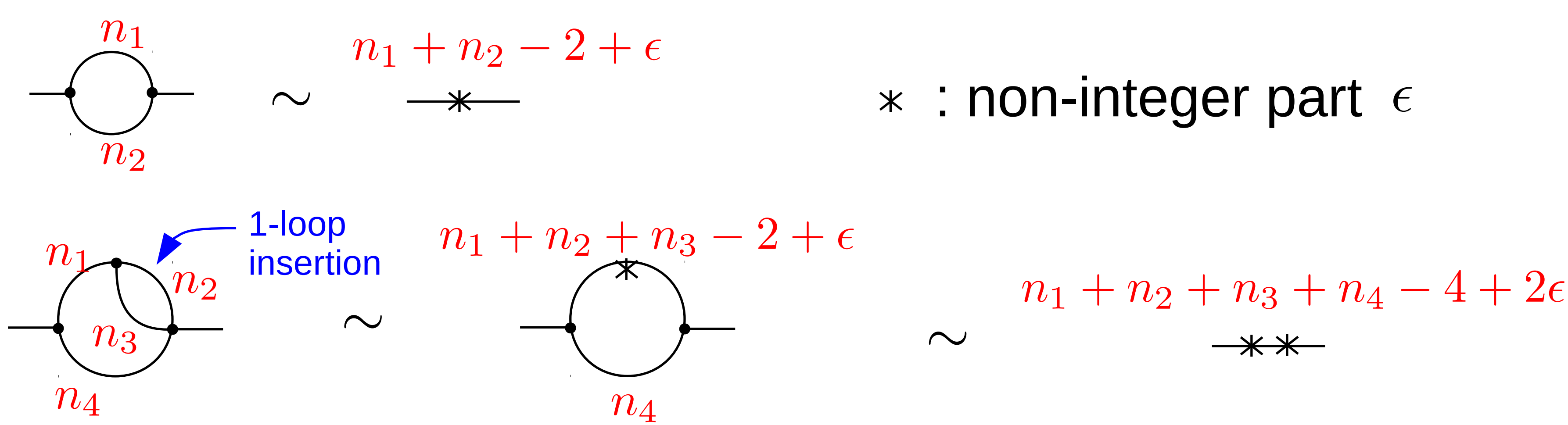
$$\int d^D p \frac{1}{(p^2)^{\alpha}[(Q-p)^2]^{\beta}} \sim \frac{1}{(Q^2)^{\alpha+\beta-2+\epsilon}}$$



General formula: Chetyrkin, Kataev, Tkachov ’80; Chetyrkin, Tkachov ’81

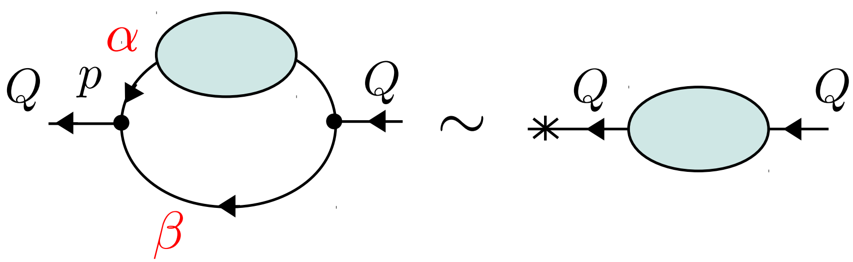
$D = 4 - 2\epsilon$

- The result gets a non-integer power $1/(Q^2)^{\epsilon}$



1-loop “carpet” integral

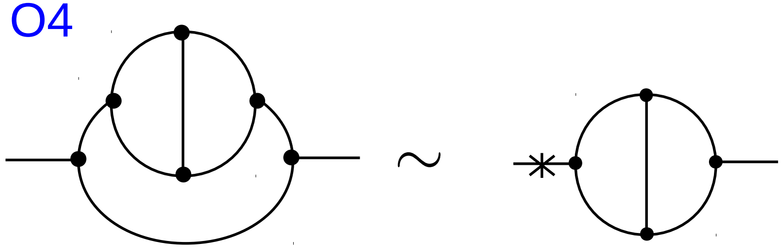
- We can perform another type of 1-loop integral (outer loop of p) for arbitrary indices α and β



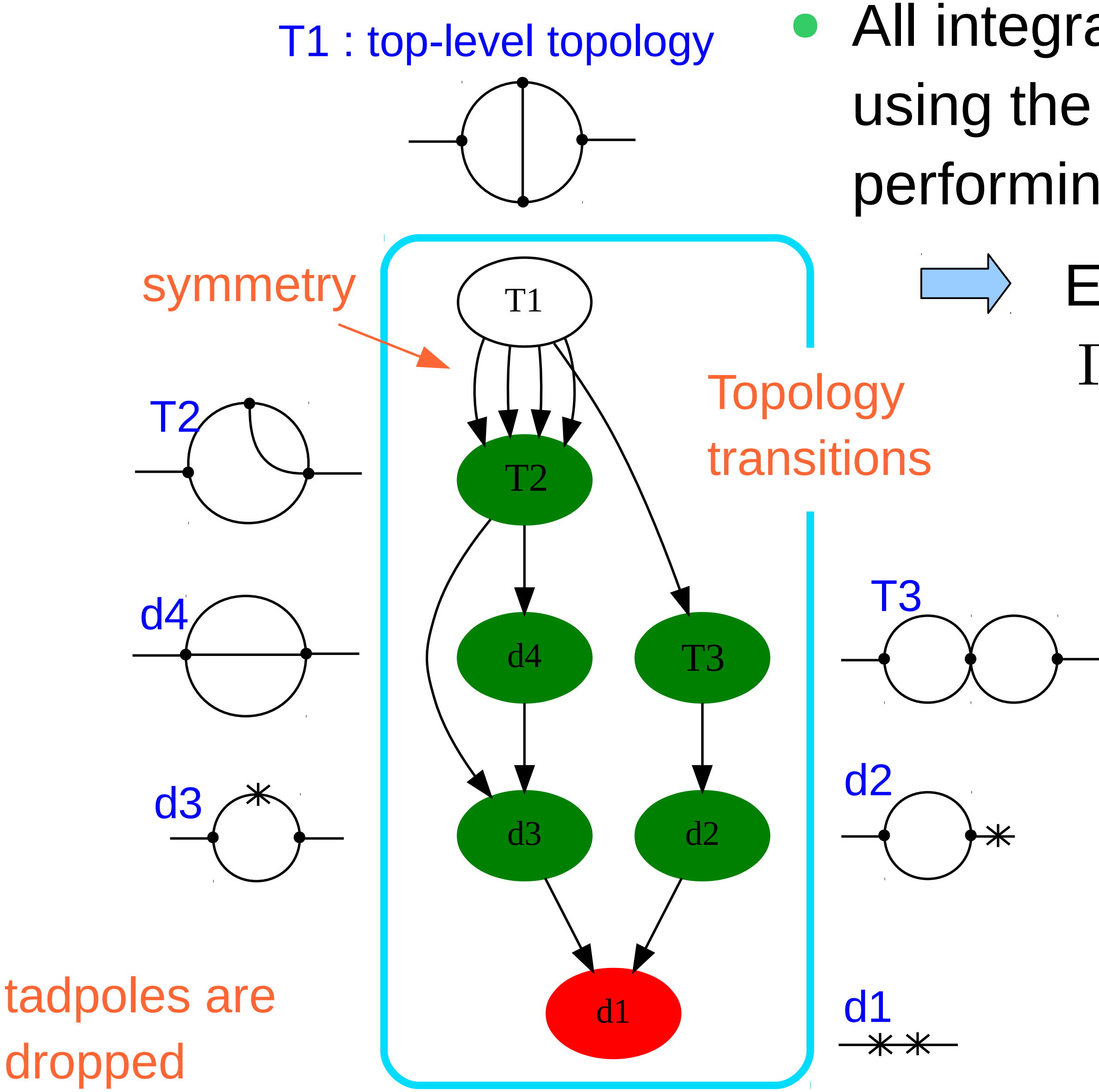
- Used for the following topology at the 3-loop level

[Chetyrkin, Tkachov ’81]

O4



2-loop reductions



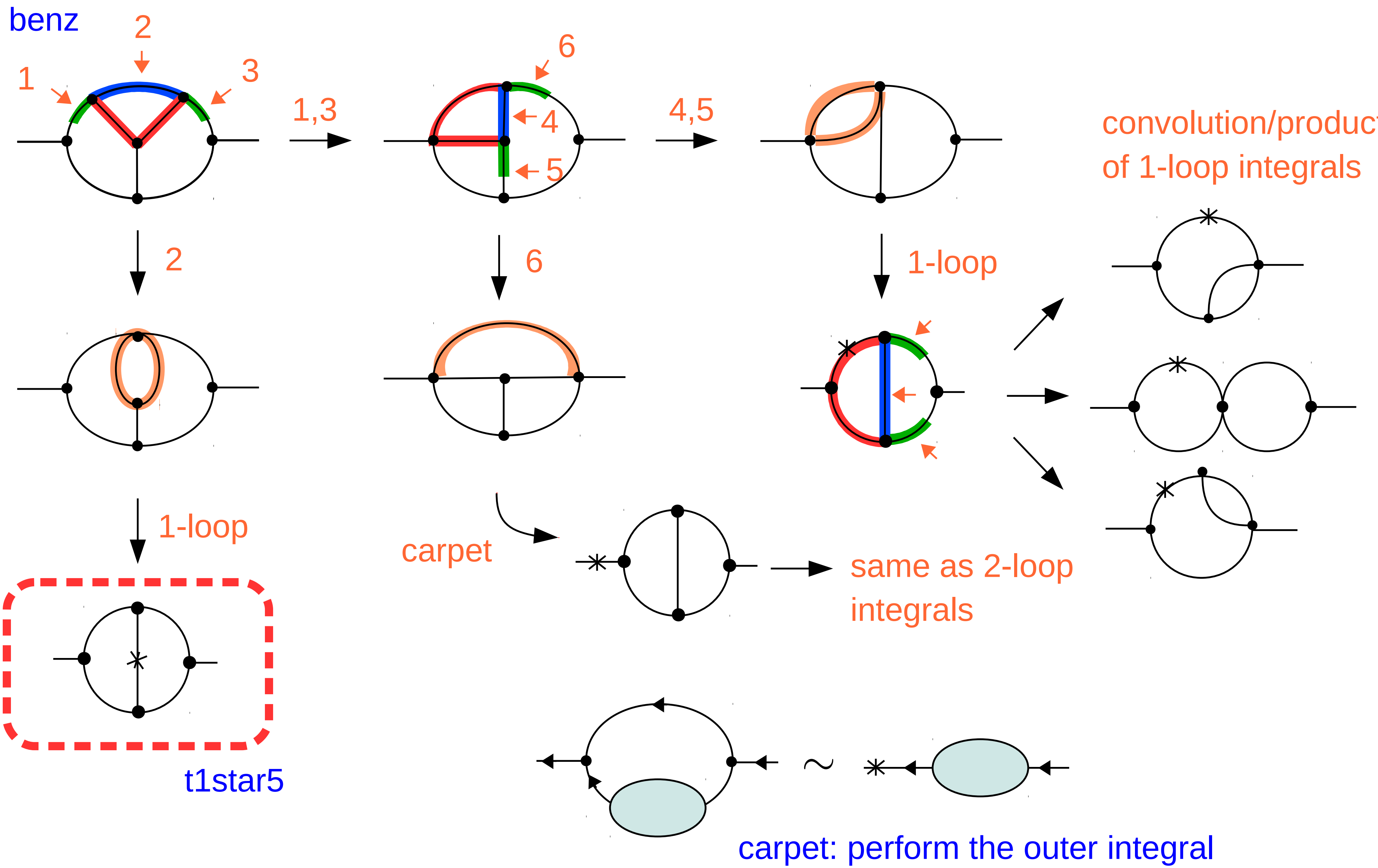
- All integrals can be evaluated by using the triangle rule and performing one-loop integrals

  ⇨ Expressed in terms of Γ -functions

Topologies in which
- One loop-insertion
- Triangle rule

can be applied

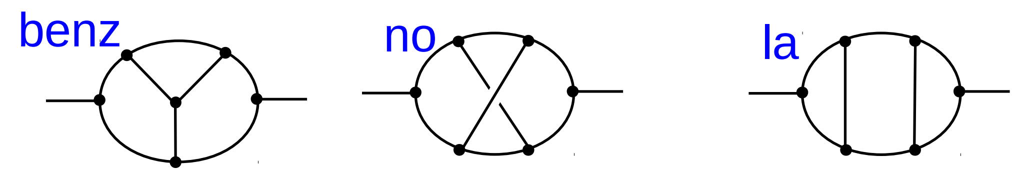
3-loop example: Benz topology

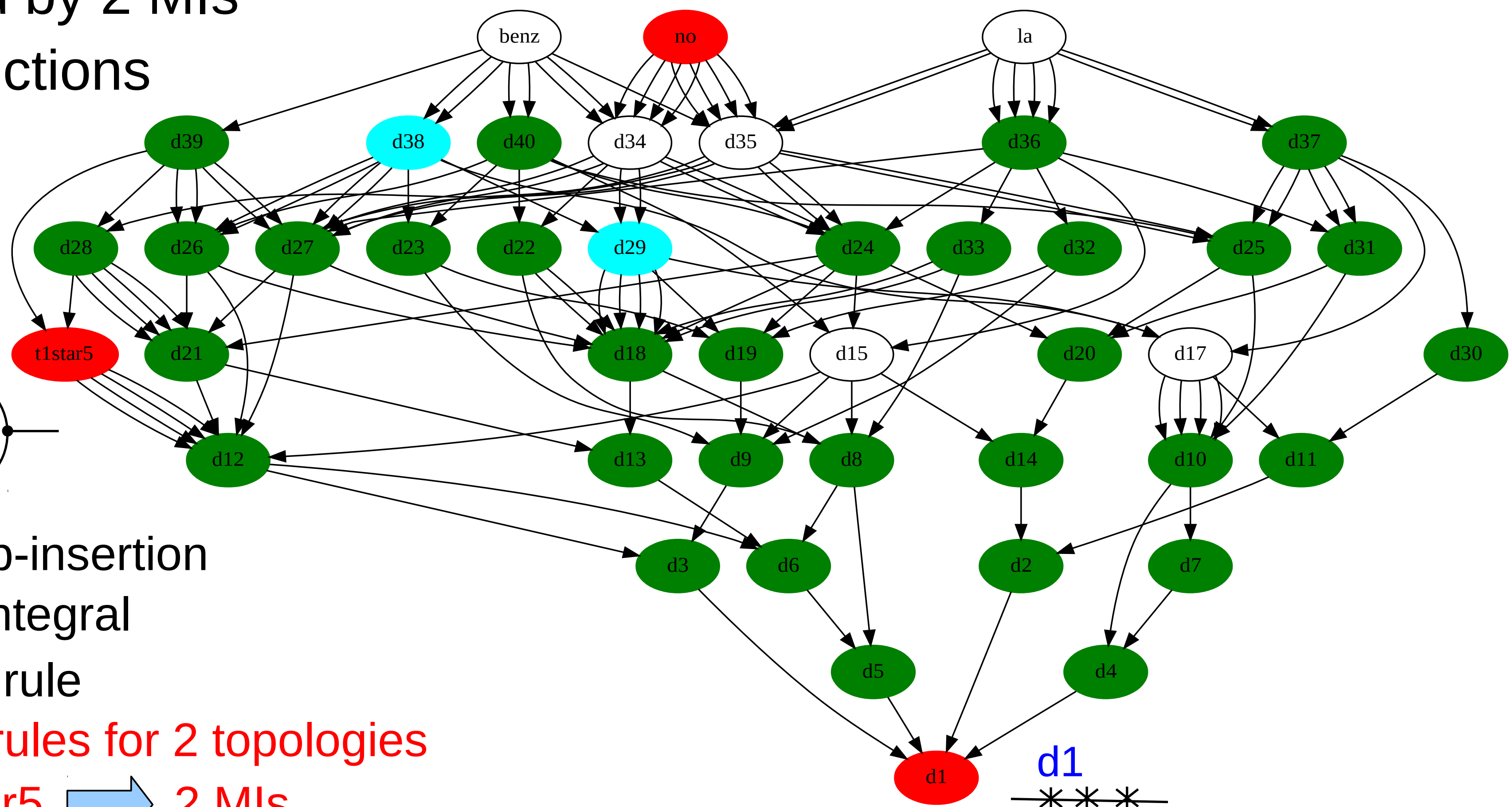


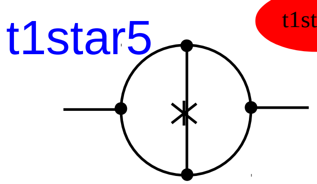
3-loop reductions

- All integrals are expressed by 2 MIs and Γ-functions

3 top-level topologies







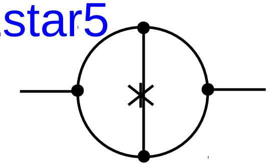
- One loop-insertion
- Carpet integral
- Triangle rule
- Special rules for 2 topologies

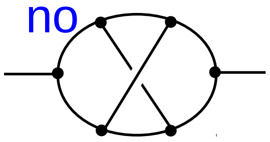
no, t1star5 ⇨ 2 MIs

Mincer approach

Algorithm: [Chetyrkin, Tkachov '81]
Schoonschip implementation: [Gorishny, Larin, Surguladze, Tkachov '89]
Form implementation: [Larin, Tkachov, Vermaseren '91]

- Many topologies for massless propagator-type integrals can be reduced to simpler ones by
  - performing one-loop integrals
  - use of triangle rules to remove one of lines
- Special cases (where we need to solve IBPs as recursion relations) are not too many
  - Only 2 topologies up to 3-loops



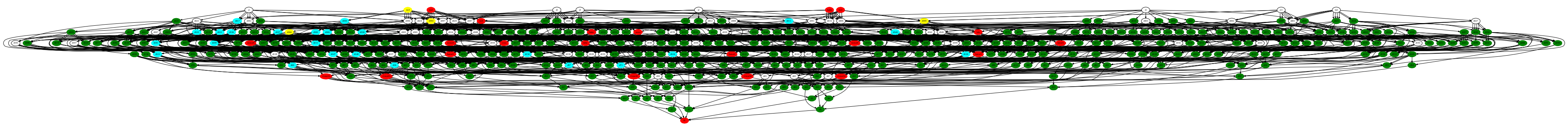


Let's consider an extension to 4-loops!

Extension to 4-loops

4-loop reductions

11 top-level topologies

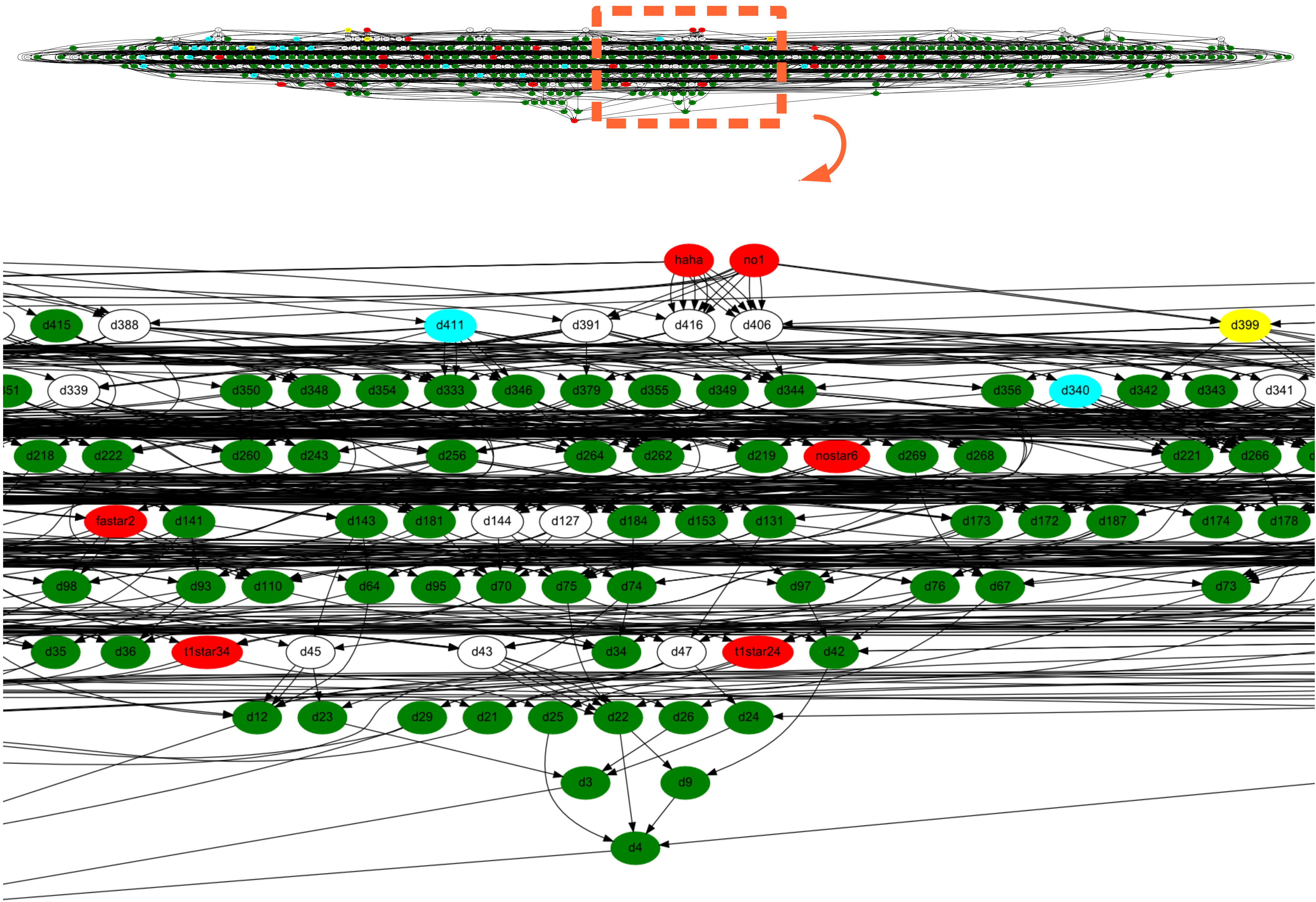


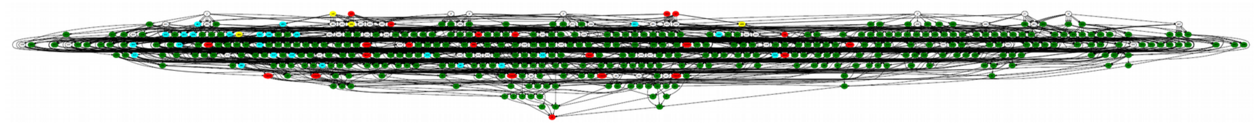
One loop-insertion	(335)	
Carpet integral	(24)	
Triangle rule	(53)	
Diamond rule	(4)	
Special rules	(21)	(pure 4-loop: 9)

Total: 437

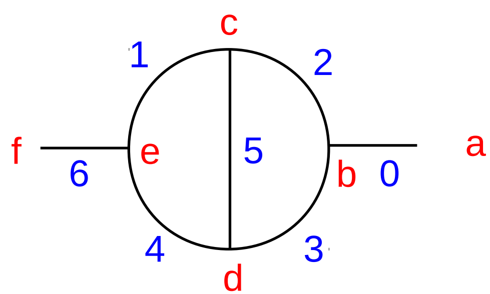
- Good news: # of special topologies only 21 out of 437
- Bad news: enormous number of cases!
- Coding such a reduction by hand is impractical

⇒ Automatization


haha
no1
d415
d388
d411
d391
d416
d406
d399
d339
d350
d348
d354
d333
d346
d379
d355
d349
d344
d356
d340
d342
d343
d341
d218
d222
d260
d243
d256
d264
d262
d219
nostar6
d269
d268
d221
d266
fastar2
d141
d143
d181
d144
d127
d184
d153
d131
d173
d172
d187
d174
d178
d98
d93
d110
d64
d95
d70
d75
d74
d97
d76
d67
d73
d35
d36
t1star34
d45
d43
d34
d47
t1star24
d42
d12
d23
d29
d21
d25
d22
d26
d24
d3
d9
d4

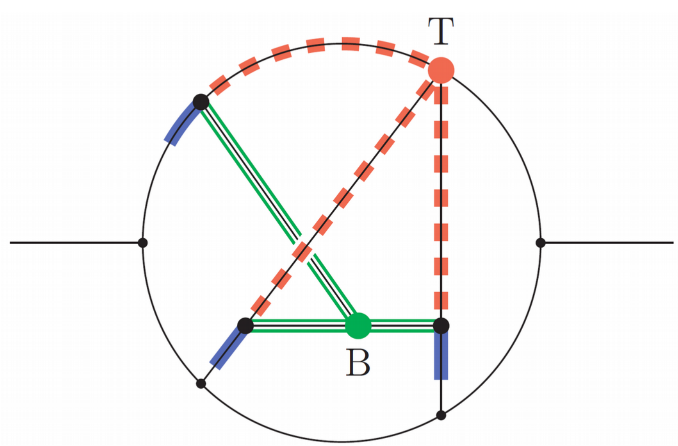
How to handle 

- Each topology as an “undirected graph” in graph theory

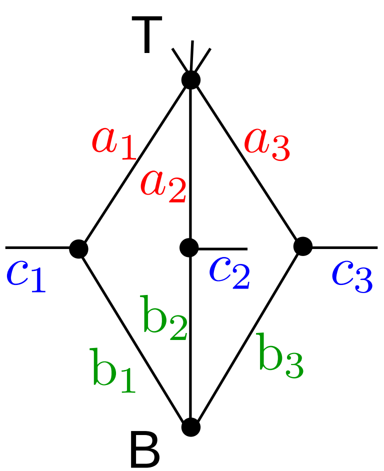


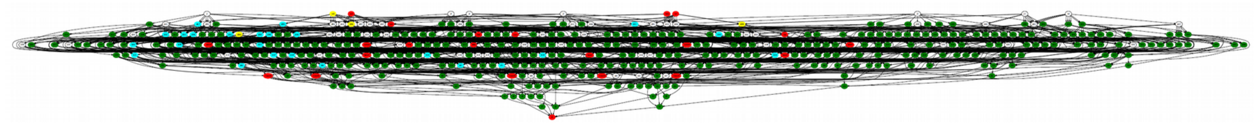
Vertices and edges are labeled
A graph is represented by connections of them

- Easy to detect one-loop insertion, carpet, triangles/diamonds and tadpoles

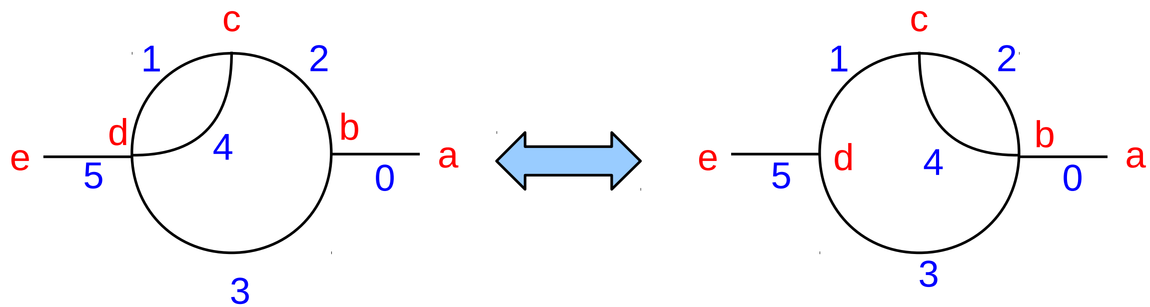


could be difficult by human eyes (diamond example)

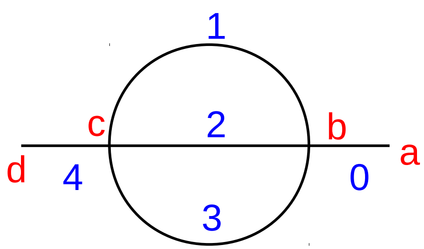


How to handle 

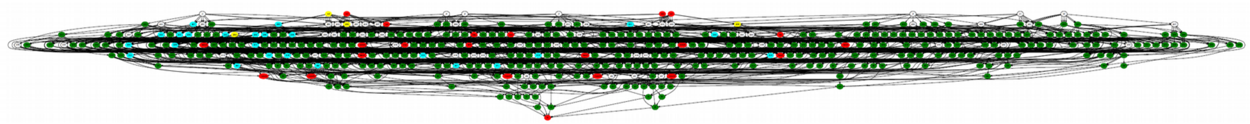
- Graph isomorphism
  - Detects equivalent graphs and finds mappings among them



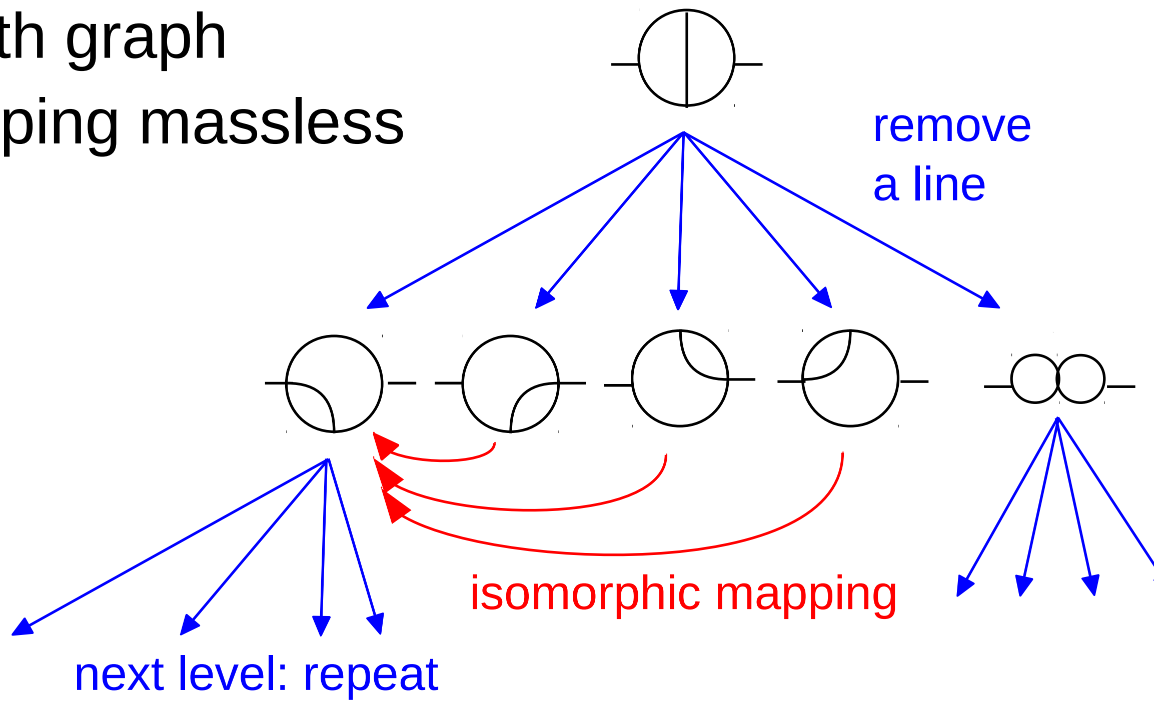
- Graph automorphism
  - Finds symmetry / mappings in each graph



$1 \leftrightarrow 2 \leftrightarrow 3$

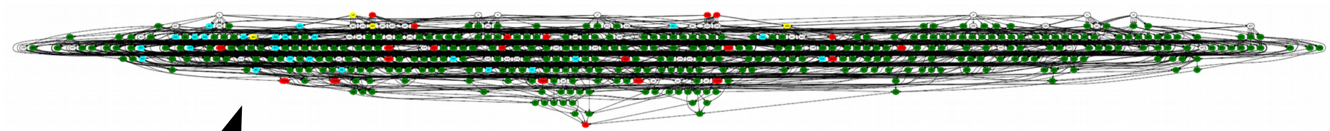
How to handle 

- Input : top-level topologies (11 for 4-loops)
- From each topology, remove a line in all possible ways with graph isomorphism and dropping massless tadpoles



- For each topology, the next action is decided from its substructure (one-loop, carpet, triangle, diamond, otherwise special rule needed)

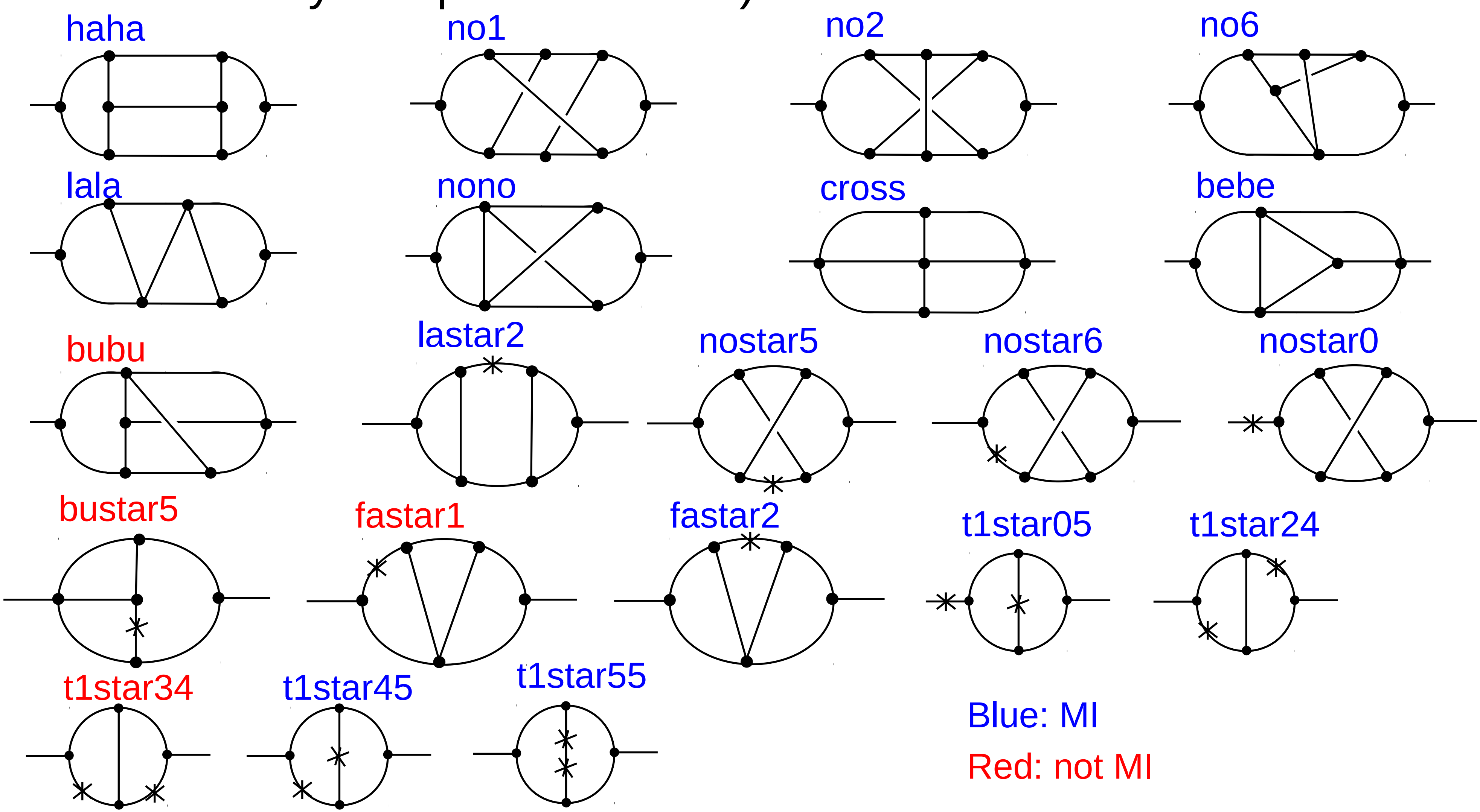
Code generation

- In the end, we get 
- Code generation from
  - Adequate subroutines are called at each topology
  - Symmetries from graph automorphism
  - Python3 program with 2793 lines using a graph library "igraph" generates FORM code with 39406 lines (+ auxiliary routines) http://igraph.org for 4-loops (as of 25.10.2016)
- Works even for 5-loops
  (64 top-topologies, 6570 topologies in total,
  284 special topologies to be implemented, currently out of range)

Special topologies

Special reduction rules

- 21 topologies require special rules, constructed manually (but considerably computer-assisted)



Finding manual reduction rules

- Shift an index by 1 (or -1 or more) in IBPs in all possible ways

$$n_1 \to n_1 \pm 1, \ n_2 \to n_2 \pm 1, \ n_3 \to n_3 \pm 1, \ \ldots$$

  - Combined with the original IBPs ⇨ new set of equations

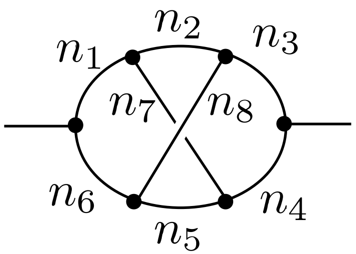
- Eliminate “complicated” integrals that increase indices for propagators (and/or decrease those for irreducible numerators) from the system of equations as possible

- In general, this is not complete, but helps a lot. Human eyes for finding “good” rules for reducing integrals

  ← less number of terms, decrease of complexity, short coefficients, no spurious poles, etc.

- Expressions may be very complicated. A rule can easily be 10-1000 lines or more. Use of computer algebra systems. (Don't do that by hand!)

Example: 3-loop NO



$n_9 : p_2 \cdot Q$

positive: $n_1, \ldots, n_8$
negative: n_9

- 3 loop momenta and 1 external momentum
  8 propagators and 1 irreducible numerator

  ⇨ 3×4 = 12 IBPs

- Shift an index by -1 in the IBPs in all possible ways

$$n_1 \to n_1 - 1,\ n_2 \to n_2 - 1,\ \ldots, n_9 \to n_9 - 1$$

  ⇨ 12×9 = 108 ⇨ 12 + 108 = 120 combined equations

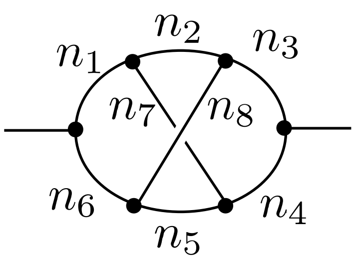
- Eliminate "complicated" integrals from the system of the equations
  A solution is (after overall $n_9 \to n_9 + 1$)

```
id Z(n1?{>=1},n2?{>=1},n3?{>=1},n4?{>=1},n5?{>=1},n6?{>=1},n7?{>=1},n8?{>=1},n9?{<=-1}) =
+Z(n1,n2-1,n3+1,n4,n5,n6,n7,n8,n9+1)*rat(1,2)*rat(n3,-7+n9+n8+n7+n6+n5+n4+n3+n2+n1+4*ep)
+Z(n1,n2,n3+1,n4,n5,n6,n7,n8-1,n9+1)*rat(-1,2)*rat(n3,-7+n9+n8+n7+n6+n5+n4+n3+n2+n1+4*ep)
+Z(n1+1,n2-1,n3,n4,n5,n6,n7,n8,n9+1)*rat(1,2)*rat(n1,-7+n9+n8+n7+n6+n5+n4+n3+n2+n1+4*ep)
+Z(n1+1,n2,n3,n4,n5,n6,n7-1,n8,n9+1)*rat(-1,2)*rat(n1,-7+n9+n8+n7+n6+n5+n4+n3+n2+n1+4*ep)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,n9+1)*rat(1,2)*rat(-3+n9+n3+2*n2+n1+2*ep,-7+n9+n8+n7+n6+n5+n4+n3+n2+n1+4*ep)
+Z(n1,n2-1,n3,n4,n5,n6,n7,n8,n9+2)*rat(1,2)*rat(1+n9,-7+n9+n8+n7+n6+n5+n4+n3+n2+n1+4*ep)
;
```

Repeated use until $n_9 = 0$ (removal of the irreducible numerator)

Example: 3-loop NO

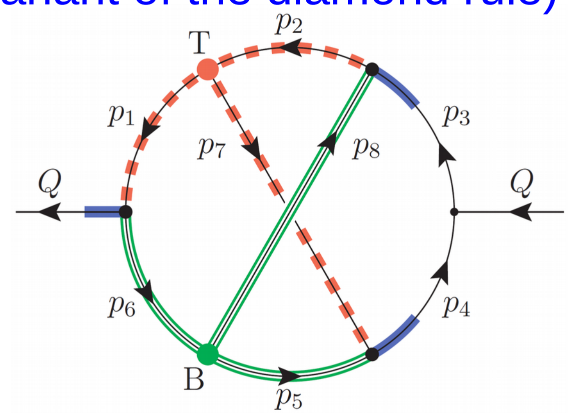
- Substitute $n_9 = 0$ to the equations
- Eliminate “complicated” integrals
  (with overall $n_1 \to n_1 - 1$, the 12 IBPs are enough)



$n_9 = 0$

```
id Z(n1?{>=2},n2?{>=1},n3?{>=1},n4?{>=1},n5?{>=1},n6?{>=1},n7?{>=1},n8?{>=1},0) =
+Z(n1-1,n2,n3,n4-1,n5,n6,n7+1,n8,0)*rat(-n7,-1+n1)
+Z(n1-1,n2,n3,n4,n5-1,n6,n7+1,n8,0)*rat(n7,-1+n1)
+Z(n1-1,n2+1,n3-1,n4,n5,n6,n7,n8,0)*rat(-n2,-1+n1)
+Z(n1-1,n2+1,n3,n4,n5,n6,n7,n8-1,0)*rat(n2,-1+n1)
+Z(n1-1,n2,n3,n4,n5,n6,n7,n8,0)*rat(-9+2*n8+n7+2*n6+2*n5+n2+n1+4*ep,-1+n1)
+Z(n1,n2,n3,n4,n5,n6-1,n7,n8,0)
;
```

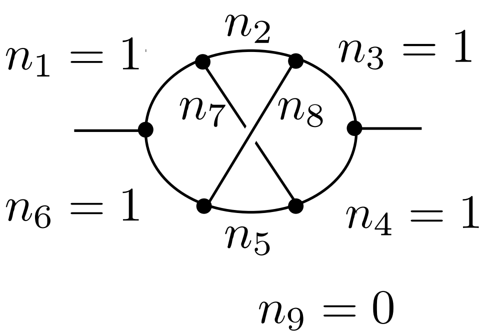
(a variant of the diamond rule)



Decrease the complexity until $n_1 = 1$

The symmetry of the topology gives similar rules for n_3, n_4 and n_6

Example: 3-loop NO



- Substitute $n_1 = n_3 = n_4 = n_6 = 1$ and $n_9 = 0$ to the 120 combined equations
- Eliminate "complicated" integrals

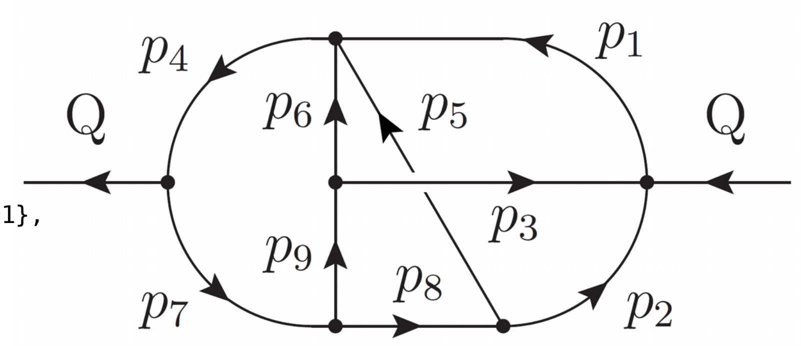
```
id Z(1,n2?{>=2},1,1,n5?{>=1},1,n7?{>=1},n8?{>=1},0) =
+Z(1,n2-1,1,1,n5-1,1,n7,n8+1,0)*rat(-n8,-1+n2)*rat(-5+n8+2*n7+2*n5+4*ep,-3+n8+n5+n2+2*ep)
+Z(1,n2-1,1,1,n5-1,1,n7+1,n8,0)*rat(-n7,-1+n2)*rat(-7+n8+2*n7+2*n5+2*n2+4*ep,-3+n8+n5+n2+2*ep)
+Z(1,n2-1,1,1,n5,1,n7,n8,0)*rat(-1,-3+n8+n5+n2+2*ep)*rat(30-12*n8+n8^2-15*n7+3*n7*n8+2*n7^2-22*n5
    +4*n5*n8+6*n5*n7+4*n5^2-11*n2+3*n2*n8+2*n2*n7+4*n2*n5+n2^2-32*ep+6*ep*n8+8*ep*n7+12*ep*n5+6*ep*n2
    +8*ep^2,-1+n2)
+Z(1,n2,1,1,n5-1,1,n7,n8,0)*rat(2-n8-n5-2*ep,-3+n8+n5+n2+2*ep)
+Z(1,n2,1,1,n5,1,n7-1,n8,0)
+Z(1,n2,1,1,n5,1,n7,n8-1,0)*rat(7-n8-2*n7-2*n5-2*n2-4*ep,-3+n8+n5+n2+2*ep)
+Z(1,n2-1,1,0,n5,1,n7+1,n8,0)*rat(n7,-1+n2)*rat(-7+n8+2*n7+2*n5+2*n2+4*ep,-3+n8+n5+n2+2*ep)
+Z(1,n2-1,1,1,n5,0,n7,n8+1,0)*rat(n8,-1+n2)*rat(-7+n8+2*n7+2*n5+2*n2+4*ep,-3+n8+n5+n2+2*ep)
+Z(1,n2,0,1,n5,0,n7,n8+1,0)*rat(-2*n8,-3+n8+n5+n2+2*ep)
+Z(1,n2,0,1,n5,1,n7,n8,0)*rat(-9+2*n8+2*n7+3*n5+2*n2+6*ep,-3+n8+n5+n2+2*ep)
+Z(1,n2,1,1,n5,0,n7,n8,0)*rat(-9+2*n8+2*n7+2*n5+3*n2+6*ep,-3+n8+n5+n2+2*ep)
;
```

Decrease the complexity until $n_2 = 1$
Similar rules for n_5, n_7 and n_8

⇨ The MI: NO(1,1,1,1,1,1,1,1,0)

Example: 4-loop BUBU

- A part of rules (reduction of n_{11} to 0)
  (short relations fitted in a page)



$$2Q\cdot p_2, 2Q\cdot p_8, 2p_2\cdot p_3$$
$$2p_2\cdot p_9, 2p_3\cdot p_8$$

One of difficult topologies to solve, though no MI

```
id Z(n1?{>=1},n2?{>=1},n3?{>=1},n4?{>=2},n5?{>=1},n6?{>=1},n7?{>=1},n8?{>=1},n9?{>=1},
     n10?{<=0},n11?{<=-1},n12?{<=0},n13?{<=0},n14?{<=0}) =
+Z(n1,n2-1,n3,n4-1,n5+1,n6,n7,n8,n9,n10,n11+1,n12,n13,n14)*rat(n5,-1+n4)
+Z(n1,n2,n3,n4-1,n5,n6,n7+1,n8-1,n9,n10,n11+1,n12,n13,n14)*rat(-n7,-1+n4)
+Z(n1,n2,n3,n4-1,n5,n6,n7+1,n8,n9-1,n10,n11+1,n12,n13,n14)*rat(n7,-1+n4)
+Z(n1,n2,n3,n4-1,n5+1,n6,n7,n8-1,n9,n10,n11+1,n12,n13,n14)*rat(-n5,-1+n4)
+Z(n1,n2,n3,n4-1,n5,n6,n7,n8,n9,n10,n11+1,n12,n13,n14)*rat(3-2*n8-n7-n5-n14-n11-2*ep,-1+n4)
+Z(n1,n2,n3,n4,n5,n6,n7-1,n8,n9,n10,n11+1,n12,n13,n14)*rat(-1,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8-1,n9,n10,n11+1,n12,n13,n14)*rat(-1,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,n9-1,n10,n11+1,n12,n13,n14)
;
id Z(n1?{>=1},n2?{>=1},n3?{>=1},1,n5?{>=1},n6?{>=1},n7?{>=1},n8?{>=1},n9?{>=1},
     n10?{<=0},n11{<=-1},n12?{<=0},n13?{<=0},n14?{<=0}) =
+Z(n1,n2,n3,1,n5,n6,n7,n8,n9,n10,n11+1,n12+1,n13-1,n14)*rat(n12,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7,n8,n9,n10+1,n11+1,n12,n13-1,n14)*rat(-n10,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7+1,n8-1,n9,n10,n11+1,n12,n13,n14)*rat(-n7,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7+1,n8,n9-1,n10,n11+1,n12,n13,n14)*rat(n7,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3+1,1,n5,n6-1,n7,n8,n9,n10,n11+1,n12,n13,n14)*rat(-n3,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3+1,1,n5,n6,n7,n8,n9-1,n10,n11+1,n12,n13,n14)*rat(n3,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7,n8,n9,n10,n11+1,n12,n13,n14)*rat(4+n9-n8-n6-n5-n2-n14-n12-n11-n10-n1-3*ep,
    -8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7-1,n8,n9,n10,n11+1,n12,n13,n14)*rat(-1,1)
+Z(n1,n2,n3,1,n5,n6,n7-1,n8,n9,n10,n11+1,n12,n13,n14+1)*rat(n14,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7-1,n8,n9,n10,n11+2,n12,n13,n14)*rat(-1-n11,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7,n8-1,n9,n10,n11+1,n12,n13,n14+1)*rat(-n14,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7,n8-1,n9,n10,n11+2,n12,n13,n14)*rat(1+n11,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7,n8,n9-1,n10,n11+1,n12,n13,n14+1)*rat(-n14,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,1,n5,n6,n7,n8,n9-1,n10,n11+2,n12,n13,n14)*rat(1+n11,-8+n9+n8+n7+n6+n5+n3+n2+n14+n13+n12+n11+n10+n1+5*ep)
+Z(n1,n2,n3,0,n5,n6,n7,n8,n9,n10,n11+1,n12,n13,n14)
;
```

3098 lines for whole BUBU reduction

(and 176 + 4770 + 1741 lines unused by default)

Forcer

New program: Forcer

[Ben Ruijl, TU, Jos Vermaseren, in preparation]

- We developed a program package Forcer for 4-loop massless propagator-type integrals by using the explained techniques
- Analytical results obtained in terms of MIs / ϵ-expansion
- Feedback / improvements from Fixed-N DIS [Forcer team with Andreas Vogt]
- Recomputed known results as strong checks
  - 4-loop QCD β-function [Ritbergen, Vermaseren, Larin '97; Czakon '04]
  - Checked the gauge invariance, all powers of ξ

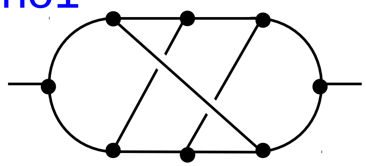
with no gauge parameter	10 minutes
with one power	38 minutes
with all gauge parameters	8.5 hours

(background field method)

on a decent 24 core machine (6 "tform -w4" jobs)

Running Forcer

no1



one of the top-level topology
11 propagators
3 irreducible numerators

benchmark: complexity 14 relative to the MI no1(1,1,1,1,1,1,1,1,1,1,1,0,0,0)

```
[no1(2,2,2,2,2,2,2,2,2,2,2,-1,-1,-1)] =

-10/9*num(1+2*ep)^2*num(2+5*ep)*num(3+2*ep)^2*num(3+5*ep)*num(4+5*ep)*num(6+5*ep)*num(7+5*ep)*num(8+5*ep
)*num(9+5*ep)*num(36141384+167650024*ep+369157793*ep^2+504389598*ep^3+470560515*ep^4+312347786*ep^5+1497
70838*ep^6+51734214*ep^7+12600912*ep^8+2060632*ep^9+203648*ep^10+9216*ep^11)*den(1+ep)^2*den(2+ep)^2*den
(2+3*ep)*den(3+ep)^2*den(4+3*ep)*den(5+3*ep)*den(7+3*ep)*den(8+3*ep)*Master(no1)

+2/9*num(1+2*ep)*num(1+4*ep)^2*num(3+2*ep)*num(39337381531008+422506983834144*ep+2126828256064272*ep^2+6
682923999248124*ep^3+14715479582570820*ep^4+24142592323497543*ep^5+30606097414180459*ep^6+30663886432898
386*ep^7+24613240685251374*ep^8+15944111435166437*ep^9+8353498138352567*ep^10+3530865125153640*ep^11+119
5052526807120*ep^12+319518770241334*ep^13+66015276933500*ep^14+10173244132808*ep^15+1101566031376*ep^16+
74820000000*ep^17+2400000000*ep^18)*den(1+ep)^2*den(2+ep)^2*den(2+3*ep)*den(3+ep)^2*den(4+3*ep)*den(5+3*
ep)*den(7+3*ep)*den(8+3*ep)*Master(no6)

-4/9*num(ep)^2*num(1+2*ep)*num(3+2*ep)*num(4970268890523930816+65981799142896214872*ep+41636699974387213
0470*ep^2+1663841711414114631648*ep^3+4730696245772216998455*ep^4+10190116418687033776180*ep^5+172831512
25628577356718*ep^6+23674869565869007631757*ep^7+26649665048245779930527*ep^8+24945440138420091798098*ep
^9+19571616320498135318722*ep^10+12933081950816489701881*ep^11+7214358008133792648788*ep^12+339681000556
8803286172*ep^13+1346680080420400500352*ep^14+447318716827968227680*ep^15+123500240869574621248*ep^16+28
008584917867939712*ep^17+5129371482425778688*ep^18+739826312414941184*ep^19+80910145880457216*ep^20+6306
386119458816*ep^21+312125440000000*ep^22+7372800000000*ep^23)*den(1+ep)^2*den(2+ep)^2*den(3+ep)^2*den(3+
4*ep)*den(4+3*ep)*den(5+3*ep)*den(5+4*ep)*den(7+3*ep)*den(7+4*ep)*den(8+3*ep)*den(9+4*ep)*den(11+4*ep)*d
en(13+4*ep)*Master(lala)
```

(cont'd on next page)

(cont'd)

```
+ ( other 17 terms )

+1/69984*num(-1+2*ep)^3*num(724191620194497621650964480000+319933970273126101430280830976000*ep+6756458
704694283224428114990694400*ep^2+91350459184391655670944774398730240*ep^3+891289886775817303600596179144
718336*ep^4+6693037596934954757286105298423578624*ep^5+40235415927130336283161325668026920448*ep^6+19871
3617144519275979305008924953882368*ep^7+820916970447314603727085836746702354688*ep^8+2873881635171369729
280438878868838131008*ep^9+8607952779410951522413494094313728793280*ep^10+22214777336844473013242467001
94857424432*ep^11+49635643135503998398733139623853337720392*ep^12+96271970320109519554706474095019058884
972*ep^13+162092004243336736948422894086915245188710*ep^14+236011025514494738395477852326524380246208*ep
^15+294116625122127952136996687381588045466897*ep^16+306225945611093340988941157877804946565622*ep^17+25
0395537138862785223188155623245534964532*ep^18+127849096823093076621319723934762735418774*ep^19-32311698
658641642165998037752400422510997*ep^20-183932395284770686557606510999638263724434*ep^21-284795600836347
457186578491353430578870326*ep^22-315022919192483812286675434478208160276910*ep^23-282305580037611102151
358311949984849163164*ep^24-212936199022871691907574075383055371432706*ep^25-136445993518657866963478040
015820540166838*ep^26-73490078278879345786210757806097518867018*ep^27-3189846583021924021328559833408903
5466304*ep^28-9651382484951439893472757109220693984414*ep^29-421236561806300181608330799113484613884*ep^
30+2017002468644720352265271521884033262822*ep^31+1832874761562866686698305090568644702899*ep^32+1096707
581456919232171368580075436795092*ep^33+524523702662773366611140017451441153706*ep^34+212149752741671710
649385032559095034796*ep^35+74250901029010212070735988610887333749*ep^36+2272051252243184038118975458520
5484412*ep^37+6103292399690879903999959233123243684*ep^38+1439736393111277351174910769581576800*ep^39+29
7528530322975798096839220046747248*ep^40+53606182091916632708972765495066048*ep^41+835965908902794893568
4267151014592*ep^42+1117001205642226519475032390006784*ep^43+126129308732809465322496308824064*ep^44+118
10033418325211214154334027776*ep^45+892823848935298884179601309696*ep^46+52384336126816611070473928704*e
p^47+2238933537865339140484104192*ep^48+62029021136981852160000000*ep^49+836264176857907200000000*ep^50)
*den(ep)^6*den(-1+ep)*den(1+ep)^6*den(1+2*ep)^2*den(1+3*ep)^2*den(-2+ep)*den(2+ep)^6*den(2+3*ep)^2*den(3
+ep)^3*den(3+2*ep)^2*den(3+4*ep)*den(4+3*ep)^2*den(5+2*ep)^2*den(5+3*ep)^2*den(5+4*ep)*den(7+3*ep)^2*den
(7+4*ep)*den(8+3*ep)*den(9+4*ep)*den(11+4*ep)*den(13+4*ep)*G10*G20*G30
```

This is exact in ϵ

4.3 hours on a desktop PC with “tform -w4”

Running Forcer

- The previous result is exact in ϵ. Instead, one can make Forcer perform ϵ-expansion and truncate the result at each step (PolyRatFun,expand)

POLYRATFUNEXPAND=10

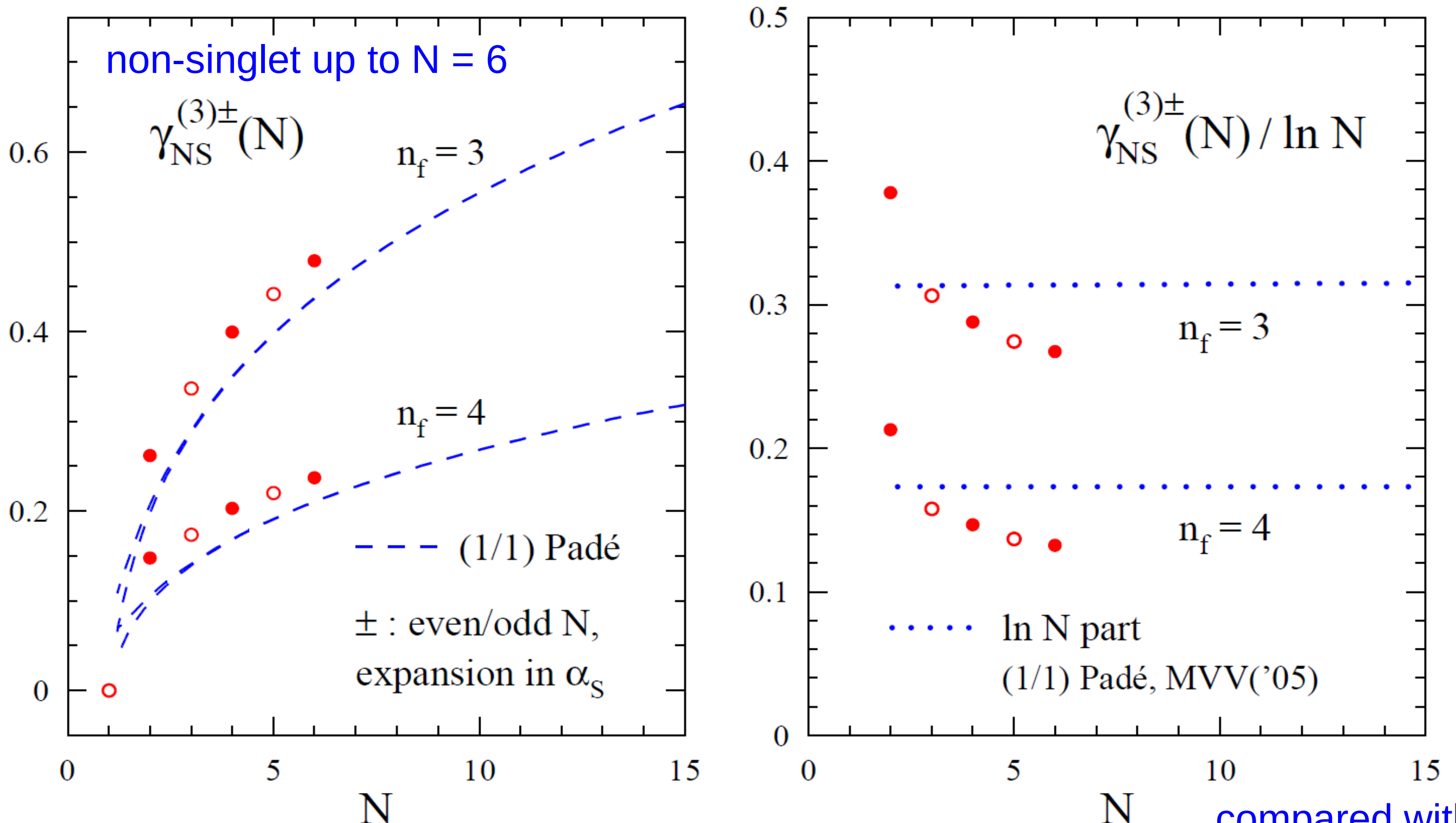
```
[no1(2,2,2,2,2,2,2,2,2,2,2,-1,-1,-1)] =

+acc(-325272456-122915007174/35*ep-79942488536109/4900*ep^2-59396124673689563/1372000*ep^3-
25478841314609132386 9/3457440000*ep^4-8287881708597550061188 3/968083200000*ep^5-
56874361842550044744220343/813189888000000*ep^6-2757489233977860222118599400 3/683079505920000000*ep^7-
929509949505831155067149925686 3/573786784972800000000*ep^8-
20799632223638109919570582007129 23/48198089937715200000000 0*ep^9-
285677421844272606875261556596546183/404863955476807680000000000*ep^10)*Master(no1)

+ ...
```

4.3 hours ⇒ 16 minutes on the same desktop PC

- Caveat: the final result is not guaranteed to be correct up to the specified order (in this case ϵ^{10}) if there are poles $1/\epsilon^n$. Spurious poles in the Forcer reduction are in many cases unpredictable. One needs to run Forcer at least twice with different orders of truncation

Fixed-N 4-loop splitting functions

[Forcer team with Andreas Vogt]



compared with Padé extrapolation from LO, NLO, NNLO

- Reproduced low-N NS splitting functions

N=1 (GLS), 2,3,4 [Baikov, Chetyrkin, Kühn, Rittinger; Velizhanin]

- New: N = 5 and N = 6

(More results: singlet up to N=4, coeff. funcs.: arXiv:1605.08408)

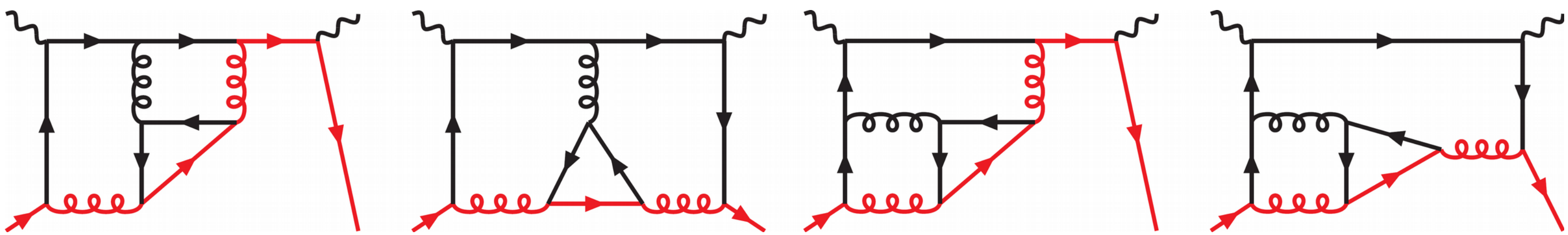
Full-N result for non-singlet n_f^2

[Forcer team with Joshua Davies and Andreas Vogt, arXiv:1610.07477, today]

- n_f^2 -contribution to $\gamma_{NS}^{(3)}$

n_f^3: J.A. Gracey '94 (leading n_f for all orders)

  - 3-loop graphs with a 1-loop insertion on one of gluon lines



  - Simple topologies: Forcer can do $N \sim \mathcal{O}$(20 – 40)
  - Analytic form in N reconstructed with the LLL method

Example:

$$\gamma_{NS}^{(3)\pm}(N)\Big|_{2C_F^2 n_f^2} = \frac{16}{27}\Big\{ -12S_{1,3,1}(N) + 6S_{1,4}(N) - 12S_{2,3}(N) - 24S_{3,2}(N) - 30S_{4,1}(N) + 36S_5(N) + 20S_{1,3}(N)$$
$$+ 40S_{2,2}(N) + 6S_{3,1}(N)(10+\eta) - 3/2S_4(N)(53+2\eta) - 38/3S_{1,2}(N) - 38/3S_{2,1}(N) + 1/3S_3(N)(287 - 12\eta + 18\eta^2$$
$$- 36D_1^2) - 1/12S_2(N)[416\eta - 12\eta^2 - 144\eta^3 - 768D_1^2 + (1259 + 216\zeta_3)] + 1/48S_1(N)[3392\eta - 3656\eta^2 + 432\eta^3$$
$$+ 720\eta^4 - 3392D_1^2 - 576D_1^3 - 1728D_1^4 + (2119 + 2880\zeta_3 - 1296\zeta_4)] + 1/96[944\eta^3 - 864\eta^5 - 7088D_1^3 - 2736D_1^4$$
$$- 1728D_1^5 + 9(127 - 264\zeta_3 + 216\zeta_4) - 24(1705 + 72\zeta_3)D_1^2 - 2(2275 - 432\zeta_3)\eta^2 + (20681 - 2880\zeta_3 + 1296\zeta_4)\eta]\Big\}$$

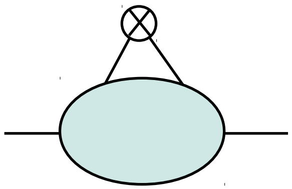
$$D_a = \frac{1}{N+a}, \quad \eta = \frac{1}{N(N+1)}$$

(and n_f^3 for singlet: New)

Further applications

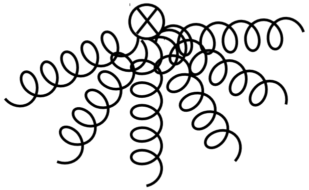
- Splitting functions from operator matrix elements

[Forcer team with Sven Moch and Andreas Vogt, work in progress]



$$\mathcal{O}_{\psi}^{\mu_1,\ldots,\mu_N} = i^{N-1}\bar{\psi}\gamma^{\{\mu_1}D^{\mu_2}\ldots D^{\mu_N\}}\psi$$

pros: milder dependence of the complexity on N
N → N + 2: +2 (instead of +4 in forward scatterings)
cons: only splitting functions (coefficient functions not accessible)
complicated (especially gluon) operators



- 5-loop YM β -function using global IRR?

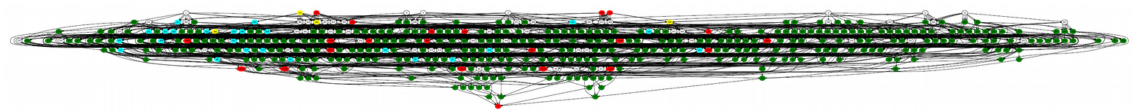
Extension of 5-loop SU(3) β-function to general gauge groups

[Baikov, Chetyrkin, Kühn ’16]

- 5-loop YM β -function using termwise R^*-operations?

Conclusion

- Forcer: a “4-loop extension” of Mincer
  for massless propagator-type Feynman integrals
  - Efficiency beyond general IBP solvers (e.g., Laporta)
  - Highly complicated structure of the program / equations
    ⇒ Automatization: write a program for generating a code

for 

Special rules are derived with the aid of computers

- Correctness of the program was checked for known results
- Applications: DIS and more

Backup

FORM today

- Binaries downloadable in the official site is rather old (and buggy)
  while we are improving code / catching bugs before the next release...
- Recommended way: build your own binary from the source in the Git repository: https://github.com/vermaseren/form

  - Prerequisites: gcc/g++ (compilers), gmp, zlib (for speed), git (for download), autoconf/automake (for build), ruby, rubygem-test-unit (for test)

```
git clone https://github.com/vermaseren/form.git
cd form
autoreconf -i
./configure --prefix=/your/favorite/place/to/install
make
make check
make install
```

```
# with Homebrew/Linuxbrew
brew install tueda/loops/form --HEAD
```

Parametric vs. explicit

- Parametric reduction (Mincer)
  - Solve the IBPs as a set of recurrence relations

$$n_1 F(n_1+1, n_2-1) - n_1 F(n_1+1, n_2) + (n_1 + 2n_2 - 4 + 2\epsilon) F(n_1, n_2) = 0$$
$$n_2 F(n_1-1, n_2+1) - n_2 F(n_1, n_2+1) + (2n_1 + n_2 - 4 + 2\epsilon) F(n_1, n_2) = 0$$

- Explicit reduction (Laporta)
  - Set the indices by explicit numbers to generate a linear system
  - Gaussian elimination with ordering to solve the system

$$F(2,0) - F(2,1) + (-1+2\epsilon) F(1,1) = 0$$
$$F(0,2) - F(1,2) + (-1+2\epsilon) F(1,1) = 0$$
$$F(2,1) - F(2,2) + (1+2\epsilon) F(1,2) = 0$$
$$2F(0,3) - 2F(1,3) + 2\epsilon F(1,2) = 0$$
$$2F(3,0) - 2F(3,1) + 2\epsilon F(2,1) = 0$$
$$F(1,2) - F(2,2) + (1+2\epsilon) F(2,1) = 0$$

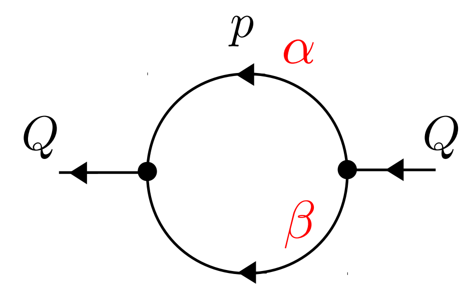
and so on

Suitable on computers
But can be a large system
and a bottleneck

General 1-loop formula

- General formula for arbitrary indices α and β

$$\int \frac{d^D p}{(2\pi)^D} \frac{p^{\mu_1} \dots p^{\mu_n}}{(p^2)^\alpha [(Q-p)^2]^\beta}$$ allowing numerators ($n \geq 0$)

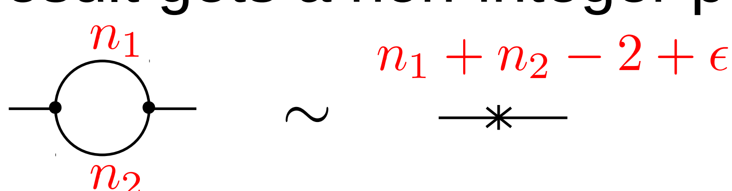


$$= \frac{1}{(4\pi)^2} \frac{1}{(Q^2)^{\alpha+\beta-2+\epsilon}} \sum_{\sigma=0}^{\lfloor n/2 \rfloor} G(\alpha,\beta,n,\sigma)(Q^2)^\sigma \left[\frac{1}{\sigma!} \left(\frac{\Box_p}{4} \right)^\sigma p^{\mu_1} \dots p^{\mu_n} \right]_{p=Q}$$

$$G(\alpha,\beta,n,\sigma) = (4\pi)^\epsilon \frac{\Gamma(\alpha+\beta-\sigma-2+\epsilon)}{\Gamma(\alpha)\Gamma(\beta)} B(2-\epsilon-\alpha+n-\sigma, 2-\epsilon-\beta+\sigma)$$

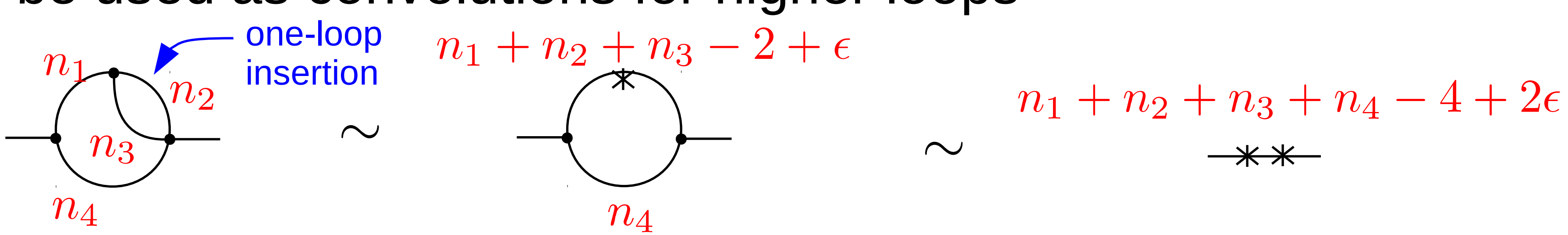
[Chetyrkin, Kataev, Tkachov ’80; Chetyrkin, Tkachov ’81]

- The result gets a non-integer power $1/(Q^2)^\epsilon$



$*$: non-integer part ϵ

- Can be used as convolutions for higher loops



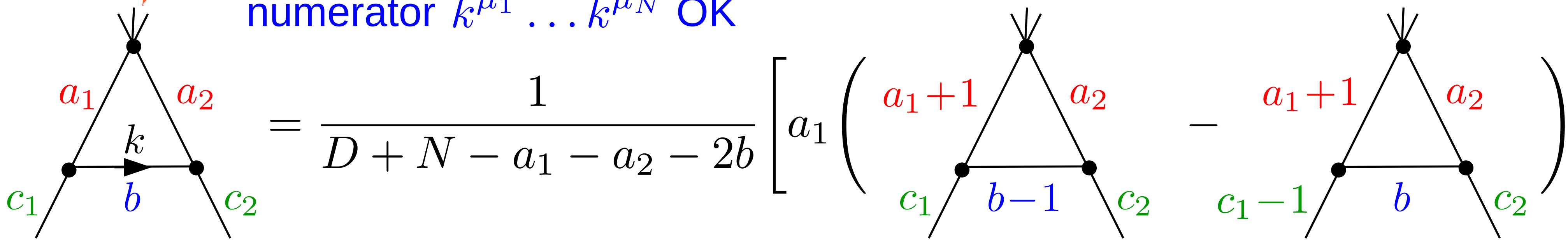
Triangle rule

[Chetyrkin, Tkachov '81]

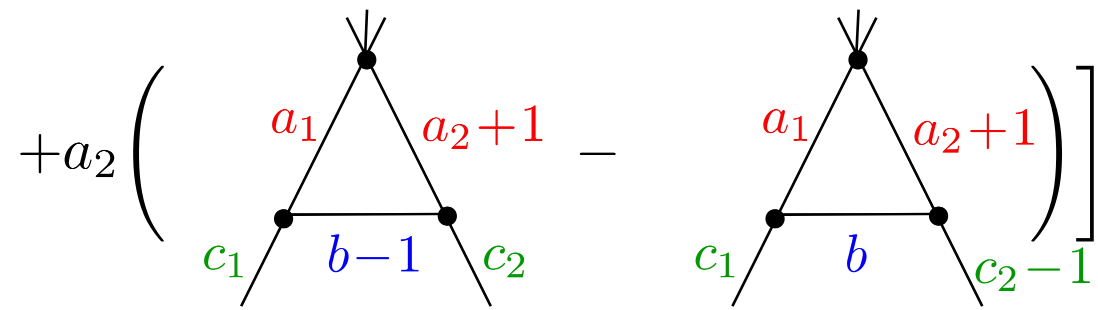
- IBP ($\dfrac{\partial}{\partial k} \cdot k$) in one-loop triangle-shaped (sub-)diagrams

any # of lines

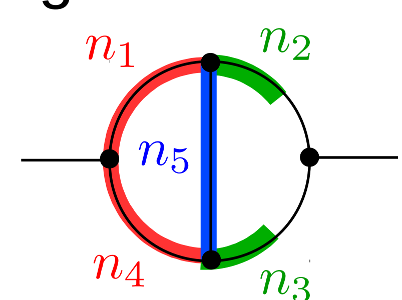
numerator $k^{\mu_1} \ldots k^{\mu_N}$ OK



Decreases b or c_1 or c_2 by 1
at the cost of increasing a_1 or a_2
in the right-hand side



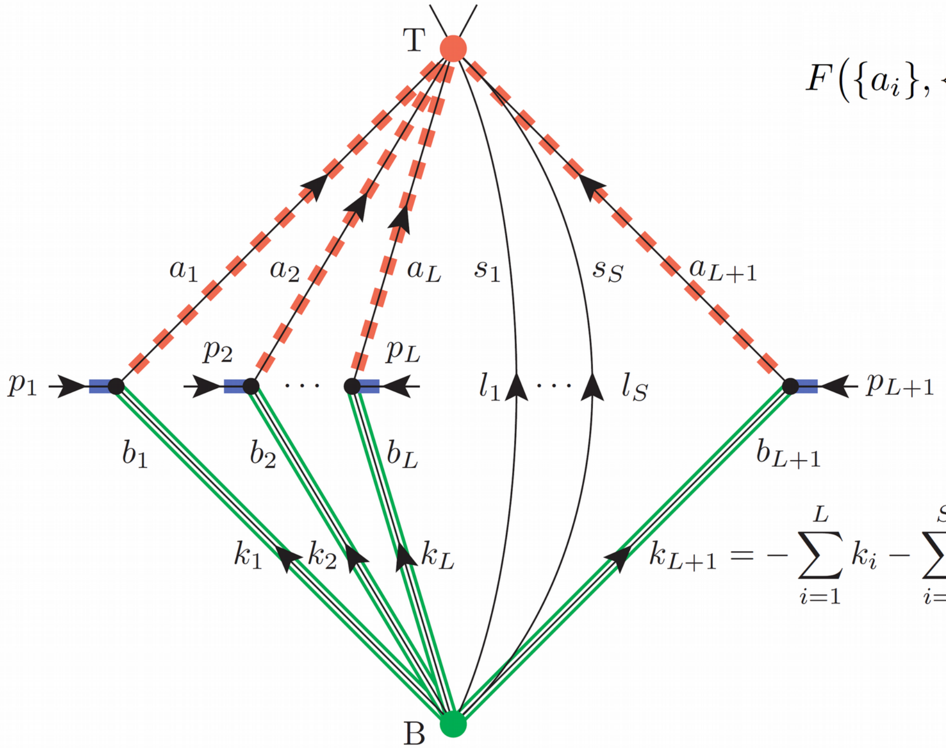
- From positive integer indices, recursive use of the triangle rule makes $b = 0$ or $c_1 = 0$ or $c_2 = 0$ (removal of a line)



⇨ sums of integrals in simpler topologies

Diamond rule

[Ruijl, TU, Vermaseren ’15]



$$F(\{a_i\},\{b_i\}) = \left[\prod_{i=1}^{L}\int d^D k_i\right]\left[\prod_{i=1}^{S}\int d^D l_i\right] \times \left[\prod_{i=1}^{L+1}\frac{k_i^{\mu_1^{(i)}}\dots k_i^{\mu_{N_i}^{(i)}}}{[(k_i+p_i)^2+m_i^2]^{a_i}(k_i^2)^{b_i}}\right]\left[\prod_{i=1}^{S}\frac{l_i^{\nu_1^{(i)}}\dots l_i^{\nu_{R_i}^{(i)}}}{(l_i^2)^{s_i}}\right]$$

$$(L+S)D+\sum_{i=1}^{L+1}(N_i-a_i-2b_i)+\sum_{i=1}^{S}(R_i-2s_i) = \sum_{i=1}^{L+1} a_i \boldsymbol{A}_i^+\left[\boldsymbol{B}_i^- - (p_i^2+m_i^2)\right]$$

Diamond rule

- Explicit summation formula

$$
\begin{aligned}
&F(\{a_i\},\{b_i\},\{c_i\}) = \\
&\quad \sum_{r=1}^{L+1}\Bigg[\left(\prod_{\substack{i=1\\ i\neq r}}^{L+1}\sum_{k_i^+=0}^{b_i-1}\right)\left(\prod_{i=1}^{L+1}\sum_{k_i^-=0}^{c_i-1}\right)(-1)^{k^-}\frac{k_r^+(k^++k^--1)!}{\prod_{i=1}^{L+1}k_i^+!k_i^-!}(E+k^+)_{-k^+-k^-} \\
&\quad\times\left(\prod_{i=1}^{L+1}(a_i)_{k_i^++k_i^-}\right)F\left(\{a_i+k_i^++k_i^-\},\{b_i-k_i^+\},\{c_i-k_i^-\}\right)\Bigg]_{k_r^+=b_r} \\
&\quad+\sum_{r=1}^{L+1}\Bigg[\left(\prod_{i=1}^{L+1}\sum_{k_i^+=0}^{b_i-1}\right)\left(\prod_{\substack{i=1\\ i\neq r}}^{L+1}\sum_{k_i^-=0}^{c_i-1}\right)(-1)^{k^-}\frac{k_r^-(k^++k^--1)!}{\prod_{i=1}^{L+1}k_i^+!k_i^-!}(E+k^++1)_{-k^+-k^-} \\
&\quad\times\left(\prod_{i=1}^{L+1}(a_i)_{k_i^++k_i^-}\right)F\left(\{a_i+k_i^++k_i^-\},\{b_i-k_i^+\},\{c_i-k_i^-\}\right)\Bigg]_{k_r^-=c_r}
\end{aligned}
$$

$$E=(L+S)D+\sum_{i=1}^{L+1}(N_i-a_i-2b_i)+\sum_{i=1}^{S}(R_i-2s_i)$$

$$k^+=\sum_{i=1}^{L+1}k_i^+ \qquad k^-=\sum_{i=1}^{L+1}k_i^-$$

Avoids spurious poles

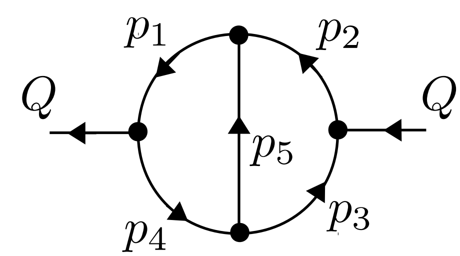
Integration-by-parts identities (IBPs)

[Chetyrkin, Tkachov '81]

- The Gauss theorem in D-dimension (the surface term vanishes)

$$\int d^D p \frac{\partial}{\partial p^\mu} X^\mu = 0$$

- Example:



$$p_3 = p_2 - Q$$
$$p_4 = p_1 - Q$$
$$p_5 = p_1 - p_2$$

$$F(n_1, n_2, n_3, n_4, n_5) = \int \frac{d^D p_1}{(2\pi)^D} \frac{d^D p_2}{(2\pi)^D} \frac{1}{(p_1^2)^{n_1} (p_2^2)^{n_2} (p_3^2)^{n_3} (p_4^2)^{n_4} (p_5^2)^{n_5}}$$

$$\int \frac{d^D p_1}{(2\pi)^D} \frac{d^D p_2}{(2\pi)^D} \frac{\partial}{\partial p_1^\mu} \frac{p_2^\mu}{(p_1^2)^{n_1} (p_2^2)^{n_2} (p_3^2)^{n_3} (p_4^2)^{n_4} (p_5^2)^{n_5}} = 0$$

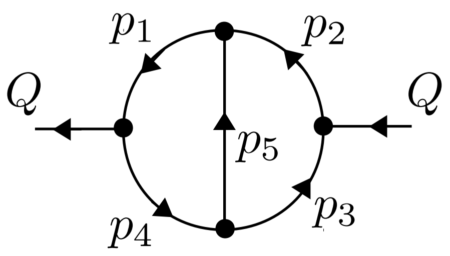
- Performing $\frac{\partial}{\partial p_1^\mu}$ and multiplying p_2^μ give various dot products (e.g., $p_1 \cdot p_2$), which are all decomposed to sums of the propagators ⇨ linear identity among F

Integration-by-parts identities (IBPs)

- $\{p_1, p_2; Q\}$ ⟹ $2 \times 3 = 6$ identities (and their linear combination)

$$\frac{\partial}{\partial p_1}\cdot p_1 \qquad \frac{\partial}{\partial p_1}\cdot p_2 \qquad \frac{\partial}{\partial p_1}\cdot Q$$

$$\frac{\partial}{\partial p_2}\cdot p_1 \qquad \frac{\partial}{\partial p_2}\cdot p_2 \qquad \frac{\partial}{\partial p_2}\cdot Q$$



$$\begin{aligned}
&F(n_1-1,n_2,n_3,n_4+1,n_5)\times(-n_4)\\
&+F(n_1-1,n_2,n_3,n_4,n_5+1)\times(-n_5)\\
&+F(n_1,n_2-1,n_3,n_4,n_5+1)\times(n_5)\\
&+F(n_1,n_2,n_3,n_4+1,n_5)\times(Q^2n_4)\\
&+F(n_1,n_2,n_3,n_4,n_5)\times(-2n_1-n_4-n_5+4-2\epsilon)\\
&=0
\end{aligned}$$

$$\frac{\partial}{\partial p_1}\cdot p_1-\frac{\partial}{\partial p_1}\cdot p_2 = \text{?}$$

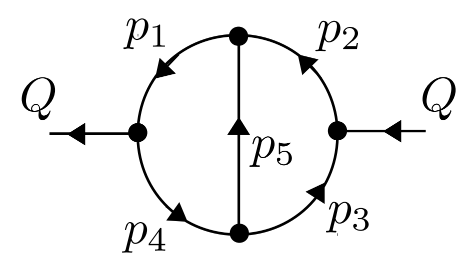
$$\begin{aligned}
&F(n_1-1,n_2,n_3,n_4+1,n_5)\times(-n_4)\\
&+F(n_1-1,n_2,n_3,n_4,n_5+1)\times(-n_5)\\
&+F(n_1,n_2-1,n_3,n_4,n_5+1)\times(n_5)\\
&+F(n_1,n_2,n_3,n_4+1,n_5)\times(Q^2n_4)\\
&+F(n_1+1,n_2-1,n_3,n_4,n_5)\times(-n_1)\\
&+F(n_1+1,n_2,n_3,n_4,n_5-1)\times(n_1)\\
&+F(n_1,n_2,n_3-1,n_4+1,n_5)\times(-n_4)\\
&+F(n_1,n_2,n_3,n_4+1,n_5-1)\times(n_4)\\
&+F(n_1,n_2,n_3,n_4,n_5)\times(-n_1+n_5)\\
&=0
\end{aligned}$$

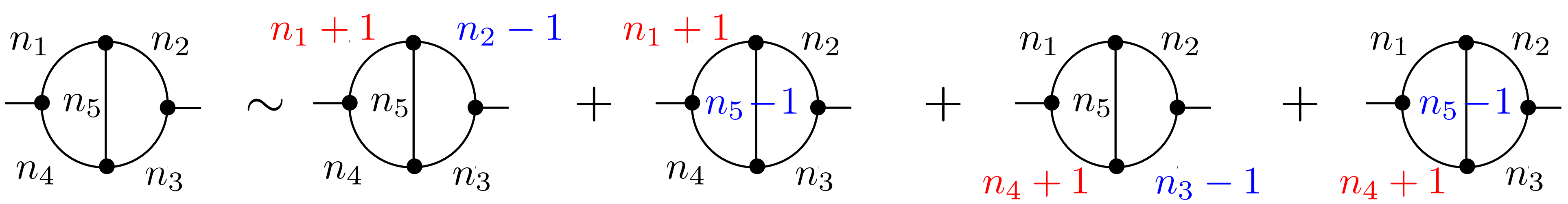
and other 4 identities

Integration-by-parts identities (IBPs)

$$\frac{\partial}{\partial p_1}\cdot p_1 - \frac{\partial}{\partial p_1}\cdot p_2 \implies$$

$$\begin{aligned}
&F(n_1,n_2,n_3,n_4,n_5)\times\underbrace{(-n_1-n_4-2n_5+4-2\epsilon)}_{\text{never be 0}}\\
&+F(n_1+1,n_2-1,n_3,n_4,n_5)\times(n_1)\\
&+F(n_1+1,n_2,n_3,n_4,n_5-1)\times(-n_1)\\
&+F(n_1,n_2,n_3-1,n_4+1,n_5)\times(n_4)\\
&+F(n_1,n_2,n_3,n_4+1,n_5-1)\times(-n_4)=0
\end{aligned}$$

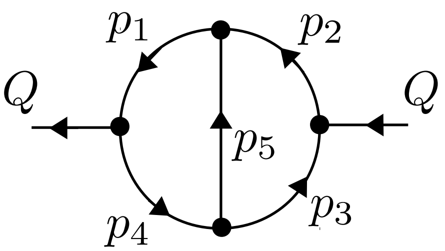




- Decreases $(n_2+n_3+n_5)$ by 1 at the cost of increasing (n_1+n_4)
- Repeated use of this rule gives integrals with $n_2=0$, $n_3=0$ or $n_5=0$ $\implies$ simpler topologies with removing one of the lines

T1-topology: IBPs

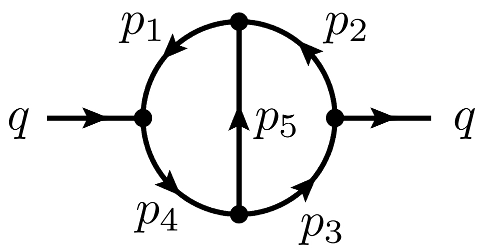
```
A1=
   +F(n1-1,n2,n3,n4+1,n5)*(-n4)
   +F(n1-1,n2,n3,n4,n5+1)*(-n5)
   +F(n1,n2-1,n3,n4,n5+1)*(n5)
   +F(n1,n2,n3,n4+1,n5)*(Q2*n4)
   +F(n1,n2,n3,n4,n5)*(-2*ep-2*n1-n4-n5+4);

A2=
   +F(n1-1,n2,n3,n4+1,n5)*(-n4)
   +F(n1-1,n2,n3,n4,n5+1)*(-n5)
   +F(n1+1,n2-1,n3,n4,n5)*(-n1)
   +F(n1+1,n2,n3,n4,n5-1)*(n1)
   +F(n1,n2-1,n3,n4,n5+1)*(n5)
   +F(n1,n2,n3-1,n4+1,n5)*(-n4)
   +F(n1,n2,n3,n4+1,n5-1)*(n4)
   +F(n1,n2,n3,n4+1,n5)*(Q2*n4)
   +F(n1,n2,n3,n4,n5)*(-n1+n5);

A3=
   +F(n1-1,n2,n3,n4+1,n5)*(-n4)
   +F(n1-1,n2,n3,n4,n5+1)*(-n5)
   +F(n1+1,n2,n3,n4-1,n5)*(n1)
   +F(n1+1,n2,n3,n4,n5)*(-Q2*n1)
   +F(n1,n2-1,n3,n4,n5+1)*(n5)
   +F(n1,n2,n3-1,n4,n5+1)*(-n5)
   +F(n1,n2,n3,n4-1,n5+1)*(n5)
   +F(n1,n2,n3,n4+1,n5)*(Q2*n4)
   +F(n1,n2,n3,n4,n5)*(-n1+n4);

A4=
   +F(n1-1,n2+1,n3,n4,n5)*(-n2)
   +F(n1-1,n2,n3,n4,n5+1)*(n5)
   +F(n1,n2-1,n3+1,n4,n5)*(-n3)
   +F(n1,n2-1,n3,n4,n5+1)*(-n5)
   +F(n1,n2+1,n3,n4,n5-1)*(n2)
   +F(n1,n2,n3+1,n4-1,n5)*(-n3)
   +F(n1,n2,n3+1,n4,n5-1)*(n3)
   +F(n1,n2,n3+1,n4,n5)*(Q2*n3)
   +F(n1,n2,n3,n4,n5)*(-n2+n5);

A5=
   +F(n1-1,n2,n3,n4,n5+1)*(n5)
   +F(n1,n2-1,n3+1,n4,n5)*(-n3)
   +F(n1,n2-1,n3,n4,n5+1)*(-n5)
   +F(n1,n2,n3+1,n4,n5)*(Q2*n3)
   +F(n1,n2,n3,n4,n5)*(-2*ep-2*n2-n3-n5+4);

A6=
   +F(n1-1,n2,n3,n4,n5+1)*(n5)
   +F(n1,n2-1,n3+1,n4,n5)*(-n3)
   +F(n1,n2-1,n3,n4,n5+1)*(-n5)
   +F(n1,n2+1,n3-1,n4,n5)*(n2)
   +F(n1,n2+1,n3,n4,n5)*(-Q2*n2)
   +F(n1,n2,n3-1,n4,n5+1)*(n5)
   +F(n1,n2,n3+1,n4,n5)*(Q2*n3)
   +F(n1,n2,n3,n4-1,n5+1)*(-n5)
   +F(n1,n2,n3,n4,n5)*(-n2+n3);

[A1-A2]=
   +F(n1+1,n2-1,n3,n4,n5)*(n1)
   +F(n1+1,n2,n3,n4,n5-1)*(-n1)
   +F(n1,n2,n3-1,n4+1,n5)*(n4)
   +F(n1,n2,n3,n4+1,n5-1)*(-n4)
   +F(n1,n2,n3,n4,n5)*(-2*ep-n1-n4-2*n5+4);

[A4-A5]=
   +F(n1-1,n2+1,n3,n4,n5)*(-n2)
   +F(n1,n2+1,n3,n4,n5-1)*(n2)
   +F(n1,n2,n3+1,n4-1,n5)*(-n3)
   +F(n1,n2,n3+1,n4,n5-1)*(n3)
   +F(n1,n2,n3,n4,n5)*(2*ep+n2+n3+2*n5-4);
```



Laporta algorithm

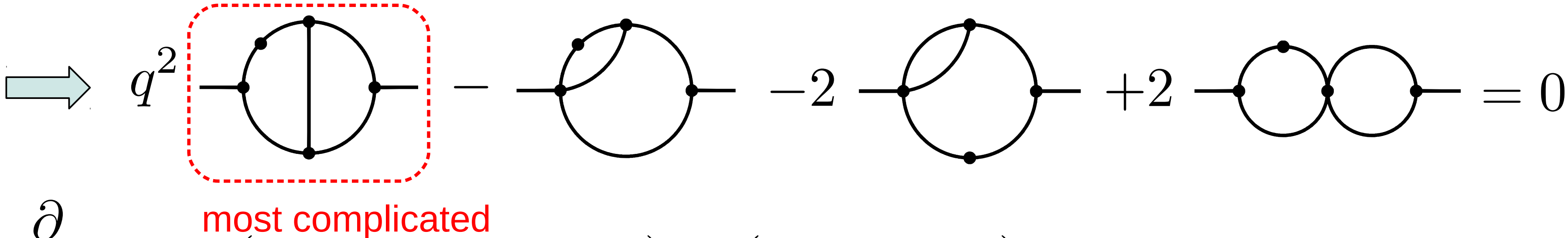
[Laporta '01]

- Substitute integers (seeds) to all indices of integrals in IBPs
- Introduce an "ordering" to decide which integrals are more complicated/simpler
- Gaussian elimination to express complicated integrals in terms of simpler integrals as possible
- Example: 2-loop massless self-energy type, again
  - IBPs: $$\frac{\partial}{\partial p_1}\cdot p_1,\ \frac{\partial}{\partial p_1}\cdot p_2,\ \frac{\partial}{\partial p_1}\cdot q\ ,$$ $$\frac{\partial}{\partial p_2}\cdot p_1,\ \frac{\partial}{\partial p_2}\cdot p_2,\ \frac{\partial}{\partial p_2}\cdot q$$
  - seeds: $(n_1,n_2,n_3,n_4,n_5)=(1,1,1,1,1),(2,1,1,1,1),$ etc.

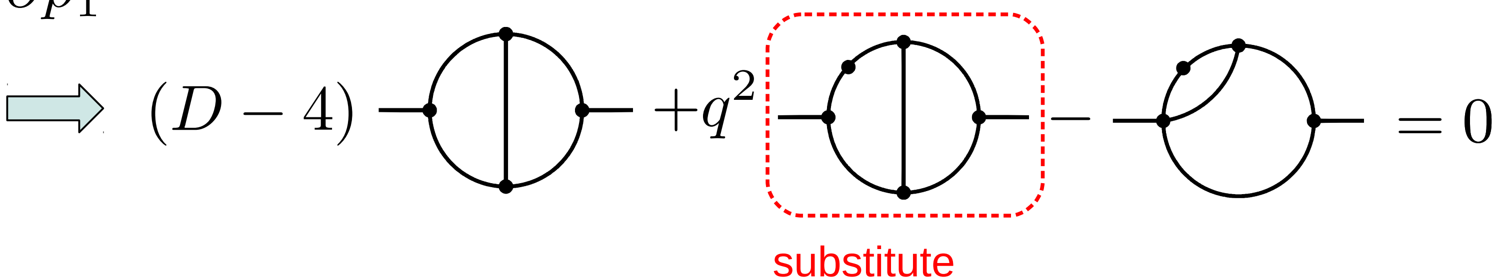


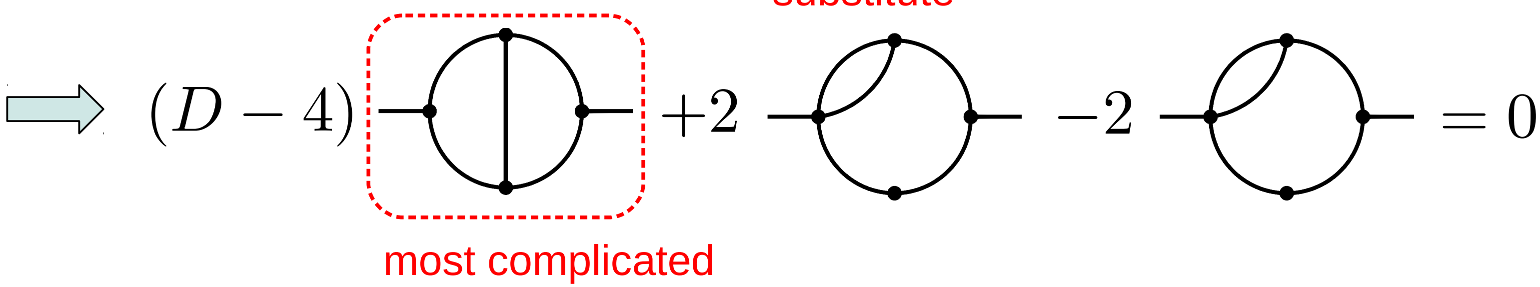
Laporta algorithm

$$\frac{\partial}{\partial p_1} \cdot p_2, \quad (n_1, n_2, n_3, n_4, n_5) = (1, 1, 1, 1, 1)$$

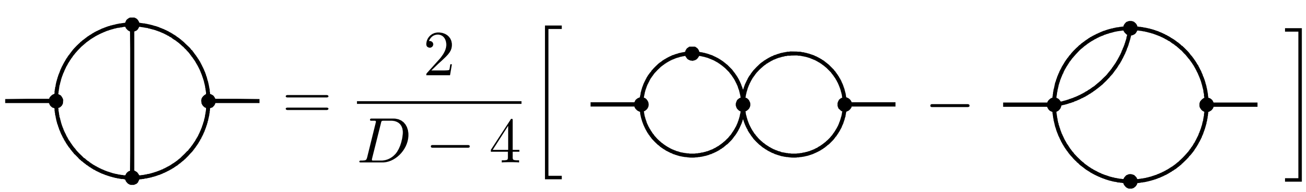


$$\frac{\partial}{\partial p_1} \cdot p_1, \quad (n_1, n_2, n_3, n_4, n_5) = (1, 1, 1, 1, 1)$$



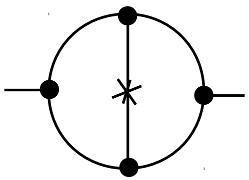


to get



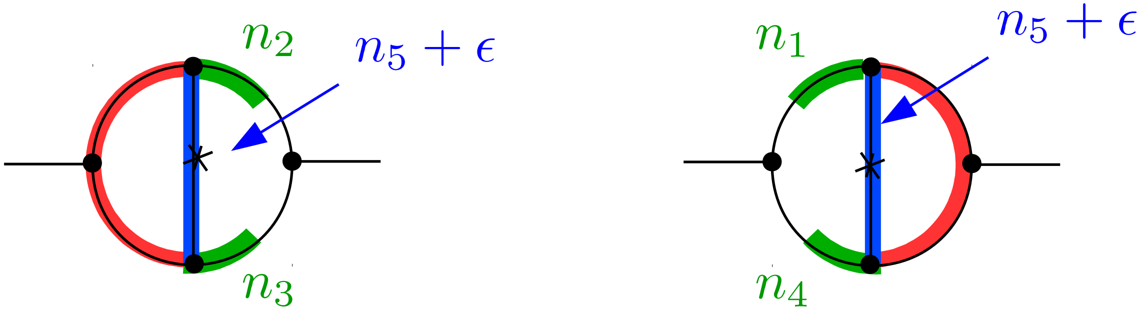
Special topologies at 3-loop

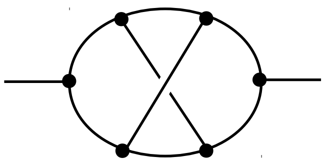
- At the 3-loop level, there are 2 topologies that require special treatment



Repeated use of the triangle rule cannot remove the central line because of the non-integer $n_5 + \epsilon$

2-loop topology with ϵ at the central line





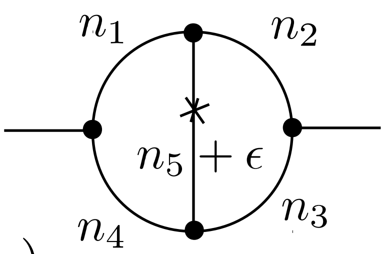
No triangle

generic “non-planer” topology

- Need to “solve” IBPs: solve a system of recurrence relations to reduce these integrals as much as possible

Special topologies at 3-loop

- Fortunately, this is possible:



$$F(n_1,n_2,n_3,n_4,n_5+\epsilon)\times\underbrace{(n_1-1)}$$ valid for $n_1 \neq 1$

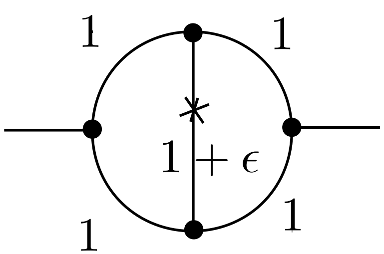
$$+F(n_1-1,n_2,n_3,n_4,n_5+\epsilon)\times(-1)\times(n_1+2n_4+n_5-5+3\epsilon)$$
$$+F(n_1,n_2,n_3,n_4-1,n_5+\epsilon)\times(-1)\times(n_1-1)$$
$$+F(n_1-1,n_2,n_3-1,n_4,n_5+1+\epsilon)\times(n_5+\epsilon)$$
$$+F(n_1-1,n_2,n_3,n_4-1,n_5+1+\epsilon)\times(-1)\times(n_5+\epsilon)=0$$

Decreases n_1 or n_4 (or both) until $n_1 = 1$. Similar rules for n_2, n_3, n_4

* They can be derived from the triangle rule identity [Ruijl, TU, Vermaseren '15]

Then n_5 can be increased/decreased until $n_5 = 1$ by

$$F(1,1,1,1,n_5+\epsilon)\times(-1)\times(n_5-1+2\epsilon)$$
$$+F(1,1,1,1,n_5-1+\epsilon)\times(-1)\times(n_5-2+3\epsilon)$$
$$+F(0,1,1,1,n_5+\epsilon)\times(-3+2n_5+5\epsilon)$$
$$+F(1,1,1,0,n_5+\epsilon)\times(-3+2n_5+5\epsilon)=0$$

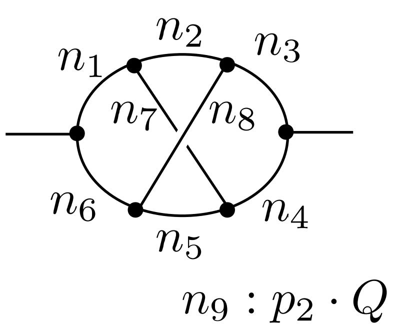


- "Master" integral $F(1,1,1,1,1+\epsilon)$ cannot be reduced any more

Special topologies at 3-loop

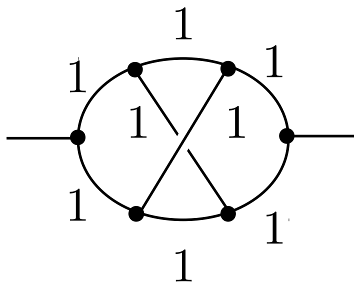
- One can make a reduction scheme also for the non-planer topology

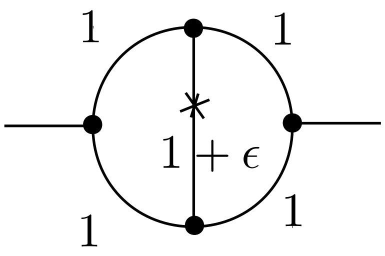
  * Rules for reducing n_1, n_3, n_4, n_6 to 1 can be derived from the diamond rule identity [Ruijl, TU, Vermaseren '15]

  - Master integral $F(1,1,1,1,1,1,1,1,0)$

- The 2 master integrals (MIs) at the 3-loop level





- They cannot be expressed in terms of Γ-functions
  But their expansions with respect to ϵ are known
  (at least, up to enough orders)