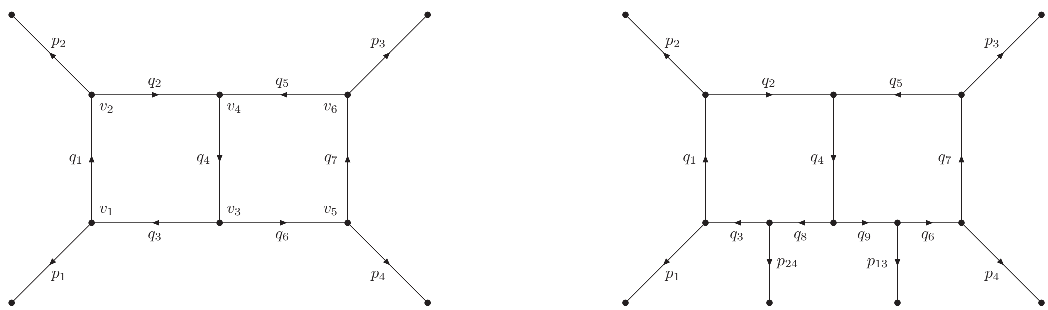
Exercises for the course “Feynman integrals”

Sheet 3

Exercise 5

Consider the following double-box graph G and the auxiliary graph $\tilde{G}$ with nine propagators:



Take

$$k_1 = q_3, \qquad k_2 = q_6$$

as the independent loop momenta. This exercise is about the family of Feynman integrals

$$I_{\nu_1\nu_2\nu_3\nu_4\nu_5\nu_6\nu_7\nu_8\nu_9}$$

with $\nu_8, \nu_9 \leq 0$. Assume that all external momenta are light-like ($p_1^2 = p_2^2 = p_3^2 = p_4^2 = 0$) and that all internal propagators are massless. Use one of the public available computer programs `Kira`, `Reduze` or `Fire` to reduce the Feynman integral

$$I_{1111111(-1)(-1)}$$

to master integrals. For the choice of master integrals you may use the default ordering criteria of the chosen computer program.

Solution:

We are considering the integral

$$I_{\nu_1\nu_2\nu_3\nu_4\nu_5\nu_6\nu_7\nu_8\nu_9} = e^{2\varepsilon\gamma_E} \left(\mu^2\right)^{\nu-D} \int \frac{d^Dk_1}{i\pi^{\frac{D}{2}}} \frac{d^Dk_2}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{9} \frac{1}{\left(-q_j^2\right)^{\nu_j}}$$

with

$$q_1 = k_1 - p_1, \qquad q_2 = k_1 - p_1 - p_2, \qquad q_3 = k_1,$$
$$q_4 = k_1 + k_2, \qquad q_5 = k_2 + p_1 + p_2, \qquad q_6 = k_2,$$
$$q_7 = k_2 + p_1 + p_2 + p_3, \qquad q_8 = k_1 - p_1 - p_3, \qquad q_9 = k_2 + p_1 + p_3.$$

The aim of this exercise is to get acquainted with one of the integration-by-parts reduction programs `Kira`, `Reduze` or `Fire`. We show how each of the three programs can be applied to the problem at hand. The actual syntax may differ for different versions of the same program. The solutions shown below refer to `Kira` version 2.0, `Reduze` version 2.4 and `Fire` version 6.4.2. In addition one should consult the manuals of these programs. Please note that `Kira` and `Reduze` use the convention

$$\frac{1}{q_j^2 - m_j^2} \quad \text{instead of} \quad \frac{1}{-q_j^2 + m_j^2}$$

for propagators. This implies a minus sign for every integral where ν is odd between the `Kira`/`Reduze` notation and the notation used in this book. The CPU timings refer to a standard laptop with 2.6 GHz.

We start with `Kira`. We prepare the following files

```
job.yaml
myreduction.in
config/integralfamilies.yaml
config/kinematics.yaml
```

The files `job.yaml` and `myreduction.in` reside in a directory. This directory also contains a subdirectory `config`. The `config`-subdirectory contains the files `integralfamilies.yaml` and `kinematics.yaml`.

The file `job.yaml` isthe main file and specifies what should be done:

```
jobs:
  - reduce_sectors:
      reduce:
         - {topologies: [doublebox], sectors: [127], r: 8, s: 2}
      select_integrals:
        select_mandatory_recursively:
          - {topologies: [doublebox], sectors: [127], r: 8, s: 2}
  - kira2form:
     target:
     - [doublebox,myreductions.in]
     reconstruct_mass: true
```

The file `myreduction.in` contains the integrals, which should be reduced. In our case it only contains a single line

```
doublebox[1,1,1,1,1,1,1,-1,-1]
```

We need to give the information on the family of Feynman integrals we are interested in. This is done in the file `integralfamilies.yaml`:

```
integralfamilies:
  - name: "doublebox"
    loop_momenta: [k1, k2]
    top_level_sectors: [127]
    propagators:
      - [ "k1-p1", 0 ]
```

```
    - [ "k1-p1-p2", 0 ]
    - [ "k1", 0 ]
    - [ "k1+k2", 0 ]
    - [ "k2+p1+p2", 0 ]
    - [ "k2", 0 ]
    - [ "k2+p1+p2+p3", 0 ]
    - [ "k1-p1-p3", 0 ]
    - [ "k2+p1+p3", 0 ]
```

Finally, we need to specify the kinematics. This is done in the file `kinematics.yaml`:

```
kinematics :
  incoming_momenta: []
  outgoing_momenta: [p1, p2, p3, p4]
  momentum_conservation: [p4,-p1-p2-p3]
  kinematic_invariants:
    - [s,  2]
    - [t,  2]
  scalarproduct_rules:
    - [[p1,p1],  0]
    - [[p2,p2],  0]
    - [[p3,p3],  0]
    - [[p1+p2,p1+p2],  s]
    - [[p2+p3,p2+p3],  t]
    - [[p1+p3,p1+p3],  -s-t]
  symbol_to_replace_by_one: s
```

With these preparations we may now run `Kira` with the command

```
kira job.yaml
```

This will produce a file `kira_myreductions.in.inc` in the directory `results/doublebox` with the content

```
id doublebox(1,1,1,1,1,1,1,-1,-1) =
 + doublebox(1,1,1,1,1,1,1,-1,0)*((-4*t-3*s)*den(2))
 + doublebox(1,1,1,1,1,1,1,0,0)*(-t^2+(-s)*t)
 + doublebox(1,0,1,1,1,0,1,0,0)*((-18*t^2+(-27*s)*t-9*s^2)*den((2*s)*t))
 + doublebox(1,1,1,1,0,0,1,0,0)*(((12*d-36)*t+(9*d-27)*s)*den((d-4)*t))
 + doublebox(1,0,0,1,0,0,1,0,0)*(((162*d^3-1458*d^2+4356*d-4320)*t+(99*d^3-891*d^2+2662*d
 -2640)*s)*den(((2*d^3-24*d^2+96*d-128)*s)*t^2))
 + doublebox(1,0,0,1,1,1,0,0,0)*(((12*d^2-76*d+120)*t^2+((66*d^2-418*d+660)*s)*t+(27*d^2
 -171*d+270)*s^2)*den(((2*d^2-16*d+32)*s^2)*t))
 + doublebox(0,1,1,0,1,1,0,0,0)*(((4*d^2-16*d+12)*t+(4*d^2-16*d+12)*s)*den((d^2-8*d+16)
 *s^2))
 + doublebox(0,0,1,1,1,0,0,0,0)*(((-144*d^2+864*d-1280)*t^2+((54*d^3-630*d^2+2316*d-2720)
 *s)*t+(81*d^3-729*d^2+2178*d-2160)*s^2)*den(((2*d^3-24*d^2+96*d-128)*s^3)*t))
;
```

This gives the desired reduction in `FORM` notation. `Kira` uses by default a ISP-basis. There are eight master integrals. The total run time on a standard laptop is about 30 s.

Let us now turn to `Reduze`. Using `Reduze` we prepare the following files:

```
job.yaml
myreduction.in
config/integralfamilies.yaml
config/kinematics.yaml
config/global.yaml
```

The syntax of Reduze is very similar to the syntax of Kira. The file job.yaml reads now

```
jobs:
  - setup_sector_mappings: {}
  - reduce_sectors:
      conditional: true
      sector_selection:
        select_recursively: [ [doublebox, 127] ]
      identities:
        ibp:
          - { r: [t, 8], s: [0, 2] }
        lorentz:
          - { r: [t, 8], s: [0, 2] }
        sector_symmetries:
          - { r: [t, 8], s: [0, 2] }
  - select_reductions:
      input_file: "myreductions.in"
      output_file: "myreductions.tmp.1"
  - reduce_files:
      equation_files:
          - "myreductions.tmp.1"
      output_file: "myreductions.tmp.2"
  - export:
      input_file: "myreductions.tmp.2"
      output_file: "myreductions.sol"
      output_format: "maple"
```

The file myreduction.in contains again a list of the integrals to be reduced, for the case at hand it is given by

```
{
INT["doublebox",{1,1,1,1,1,1,1,-1,-1}]
}
```

The file integralfamilies.yaml specifies the family of Feynman integrals under consideration:

```
integralfamilies:
  - name: "doublebox"
    loop_momenta: [k1, k2]
    propagators:
      - [ "k1-p1", 0 ]
      - [ "k1-p1-p2", 0 ]
      - [ "k1", 0 ]
      - [ "k1+k2", 0 ]
      - [ "k2+p1+p2", 0 ]
      - [ "k2", 0 ]
```

```
      - [ "k2+p1+p2+p3", 0 ]
      - [ "k1-p1-p3", 0 ]
      - [ "k2+p1+p3", 0 ]
    permutation_symmetries: []
```

The file `kinematics.yaml` is identical to the corresponding file for `Kira`:

```
kinematics :
  incoming_momenta: []
  outgoing_momenta: [p1,p2,p3,p4]
  momentum_conservation: [p4,-p1-p2-p3]
  kinematic_invariants:
    - [s,  2]
    - [t,  2]
  scalarproduct_rules:
    - [[p1,p1],  0]
    - [[p2,p2],  0]
    - [[p3,p3],  0]
    - [[p1+p2,p1+p2],  s]
    - [[p2+p3,p2+p3],  t]
    - [[p1+p3,p1+p3],  -s-t]
  symbol_to_replace_by_one: s
```

In addition, there is a file `global.yaml` containing

```
global_symbols:
  space_time_dimension: d

paths:
  fermat: /usr/local/fermat/ferl64/fer64
```

The last line gives the absolute path to the `Fermat`-executable and should be modified accordingly. Running

```
 reduze job.yaml
```

will produce a file `myreductions.sol`

```
myreductions := [

INT("doublebox",7,127,7,2,[1,1,1,1,1,1,1,-1,-1]) =
  INT("doublebox",7,127,7,1,[1,1,1,1,1,1,1,-1,0]) *
      (-2*t-3/2*s) +
  INT("doublebox",7,127,7,0,[1,1,1,1,1,1,1,0,0]) *
      (-t^2-t*s) +
  INT("doublebox",5,93,5,0,[1,0,1,1,1,0,1,0,0]) *
      (-9/2*(2*t^2+3*t*s+s^2)*t^(-1)*s^(-1)) +
  INT("doublebox",5,79,5,0,[1,1,1,1,0,0,1,0,0]) *
      (-3*(4*t-t*d)^(-1)*(3*d*s-12*t+4*t*d-9*s)) +
  INT("doublebox",4,57,4,0,[1,0,0,1,1,1,0,0,0]) *
      (1/2*(12*t^2*d^2+66*t*d^2*s+120*t^2-171*d*s^2-76*t^2*d+660*t*s+27*d^2*s^2+270*s^2
      -418*t*d*s)*(16*t-8*t*d+t*d^2)^(-1)*s^(-2)) +
  INT("doublebox",4,54,4,0,[0,1,1,0,1,1,0,0,0]) *
```

```
      (-4*(16+d^2-8*d)^(-1)*(4*d*s-3*t+4*t*d-d^2*s-3*s-t*d^2)*s^(-2)) +
  INT("doubleboxx123",3,28,3,0,[0,0,1,1,1,0,0,0,0]) *
      (-1/2*(2662*d*s-4320*t+4356*t*d+99*d^3*s-891*d^2*s+162*t*d^3-2640*s-1458*t*d^2)
      *(12*t^2*d^2+64*t^2-t^2*d^3-48*t^2*d)^(-1)*s^(-1)) +
  INT("doublebox",3,28,3,0,[0,0,1,1,1,0,0,0,0]) *
      (1/2*(144*t^2*d^2+630*t*d^2*s+1280*t^2-54*t*d^3*s-2178*d*s^2-864*t^2*d+2720*t*s
      +729*d^2*s^2+2160*s^2-81*d^3*s^2-2316*t*d*s)*(64*t-48*t*d-t*d^3+12*t*d^2)^(-1)
      *s^(-3))
];
```

This gives the reduction in `Maple` format. `Reduze` uses by default a ISP-basis. The running time on a standard laptop is about 1270 s.

Let us now turn to `Fire`. In order to get the same number of master integrals we use it in combination with `Litered` The program `Litered` provides symmetry relations to `Fire` and ensures that we end up in the example under consideration with eight master integrals as above. We prepare the following files:

```
 prepare1.m
 prepare2.m
 prepare3.m
 readout.m
 work/data.m
 work/myreductions.m
 work/doublebox.config
```

The file `data.m` defines the family of Feynman integrals

```
Internal = {k1, k2};
External = {p1, p2, p3};
Propagators = { -(k1-p1)^2, -(k1-p1-p2)^2, -k1^2, -(k1+k2)^2, -(k2+p1+p2)^2, -k2^2,
-(k2+p1+p2+p3)^2, -(k1-p1-p3)^2, -(k2+p1+p3)^2 };
Replacements = { p1^2 -> 0, p2^2 -> 0, p3^2 -> 0, p1 p2 -> s/2, p2 p3 -> t/2,
p1 p3 -> (-s-t)/2 };
```

The file `myreductions.m` contains the list of Feynman integrals, which we would like to reduce. For the case at hand

```
{
 {1,{1,1,1,1,1,1,1,-1,-1}}
}
```

`Fire` runs partly within `Mathematica` and partly in `C++`. First, three preparation steps are done within `Mathematica`, specified by the files `prepare1.m`, `prepare2.m` and `prepare3.m`. The file `prepare1.m` reads

```
Get["FIRE6.m"];
Get["work/data.m"];
PrepareIBP[];
Prepare[AutoDetectRestrictions -> True];
SaveStart["work/doublebox"];
```

Issuing in `Mathematica` the command

```
 Get["prepare1.m"];
```

will generate the file

```
work/doublebox.start
```

The file `prepare2.m` reads

```
SetDirectory["extra/LiteRed/Setup/"];
Get["LiteRed.m"];
SetDirectory["../../../"];
Get["FIRE6.m"];
Get["work/data.m"];
CreateNewBasis[doublebox, Directory -> "work/litered"];
GenerateIBP[doublebox];
AnalyzeSectors[doublebox, {__,0,0}];
FindSymmetries[doublebox, EMs->True];
DiskSave[doublebox];
```

Issuing in `Mathematica` the command

```
Get["prepare2.m"];
```

will generate a subdirectory

```
work/litered
```

containing several files. The file `prepare3.m` reads

```
Get["FIRE6.m"];
LoadStart["work/doublebox"];
TransformRules["work/litered", "work/doublebox.lbases", 1];
SaveSBases["work/doublebox"];
```

Issuing in `Mathematica` the command

```
Get["prepare3.m"];
```

will generate the two files

```
work/doublebox.lbases
work/doublebox.sbases
```

After these preparation step the `C++` program can be called. We need a configuration file `doublebox.config` containing

```
#threads          1
#variables        d, s, t
#start
#folder           work/
#problem          1 doublebox.sbases
#lbases           doublebox.lbases
#integrals        myreductions.m
#output           myreductions.tables
```

Running

```
bin/FIRE6 -c work/doublebox
```

will generate the file

```
work/myreductions.tables
```

The readout is again done within `Mathematica`. The file `readout.m` reads

```
Get["FIRE6.m"];
LoadStart["work/doublebox", 1];
Burn[];
LoadTables["work/myreductions.tables"];
res = F[1, {1,1,1,1,1,1,1,-1,-1}];
Save["work/myreductions.out",res];
```

Issuing in Mathematica the command

```
Get["readout.m"];
```

will generate the file `myreductions.out` with

```
res = ((-10 + 3*d)*(-8 + 3*d)*(-1350*s^2 + 990*d*s^2 - 234*d^2*s^2 +
        18*d^3*s^2 - 238*s*t - 56*d*s*t + 65*d^2*s*t - 9*d^3*s*t + 1616*t^2 -
        1448*d*t^2 + 404*d^2*t^2 - 36*d^3*t^2)*
       G[1, {0, 0, 1, 1, 1, 0, 0, 0, 0}])/(2*(-5 + d)^2*(-4 + d)^3*s^3*t) -
     (2*(-3 + d)*(46*s - 33*d*s + 5*d^2*s + 58*t - 40*d*t + 6*d^2*t)*
       G[1, {0, 1, 1, 0, 1, 1, 0, 0, 0}])/((-5 + d)*(-4 + d)^2*s^2) -
     ((-3 + d)*(-10 + 3*d)*(-5760*s^2 + 2772*d*s^2 - 414*d^2*s^2 +
        18*d^3*s^2 - 21744*s*t + 12352*d*s*t - 2326*d^2*s*t + 145*d^3*s*t -
        13152*t^2 + 8368*d*t^2 - 1768*d^2*t^2 + 124*d^3*t^2)*
       G[1, {0, 1, 1, 1, 0, 0, 1, 0, 0}])/(8*(-6 + d)*(-5 + d)^2*(-4 + d)^2*
       s^2*t) + (5*(-3 + d)*(-10 + 3*d)*(-8 + 3*d)*(2*s + 3*t)*
       G[1, {1, 0, 0, 1, 0, 0, 1, 0, 0}])/((-4 + d)^3*s*t^2) +
     (3*(-3 + d)*(64 - 18*d + d^2)*(3*s + 4*t)*
       G[1, {1, 0, 0, 1, 1, 1, 1, 0, 0}])/(2*(-6 + d)*(-5 + d)*(-4 + d)*t) -
     (3*(s + t)*(3*s + 5*t)*G[1, {1, 0, 1, 1, 1, 0, 1, 0, 0}])/(s*t) +
     ((-42*s^2 + 9*d*s^2 - 64*s*t + 14*d*s*t - 16*t^2 + 4*d*t^2)*
       G[1, {1, 1, 1, 1, 1, 1, 1, 0, 0}])/(4*(-4 + d)) -
     ((-6 + d)*s*t*(3*s + 4*t)*G[1, {1, 1, 1, 1, 1, 1, 2, 0, 0}])/
      (4*(-5 + d)*(-4 + d))
```

`Fire` uses by default a dot-basis. The running time for the individual parts sum up to about 30 s on a standard laptop.