Presenting `LiteRed`: a tool for the Loop InTEgrals REDuction

R.N. Lee[a,b]

[a]*Budker Institute of Nuclear Physics, 630090, Novosibirsk, Russia*
[b]*Novosibirsk State University, 630090, Novosibirsk, Russia*

E-mail: R.N.Lee@inp.nsk.su

ABSTRACT: *Mathematica* package `LiteRed` is described. It performs the heuristic search of the symbolic IBP reduction rules for loop integrals. It implements also several convenient tools for the search of the symmetry relations, construction of the differential equations and dimensional recurrence relations.

Dedicated to my father's birthday

Contents

1   Introduction

Demand for the multiloop calculations is constantly growing nowadays. As the number of loops increases, the calculations become more and more complicated and require, almost necessarily, some stages to be done automatically using computers. The reduction of the loop integrals with the help of the integration-by-parts (IBP) identities [1, 2] is an important stage which can be automatized. This reduction allows one to reduce the calculation of the multiloop diagrams to the calculation of a finite set of master integrals. It is important that the reduction also allows one to obtain differential and difference equations, which can be used for the calculation of the loop integrals without explicit itegration.

One of the most successful methods of the IBP reduction is the Laporta algorithm [3]. The algorithm is easy to implement and to use, given a sufficient amount of time, it works flawlessly. It also allows for a number of programming improvements. These advantages explain why many modern most powerful reduction programs heavily rely on this algorithm, in particular, `AIR` [4], `FIRE` [5], `Reduze` [6, 7], and many private versions. However, there are some weak points of this algorithm, which may put some restrictions on its application. First, the reduction generates heavy-weight databases of the discovered rules which can be too expensive to save. Therefore, typically, the reduction is performed each time "from the beginning" which requires the same identities to be solved in each run. Another disadvantage of the Laporta algorithm is due to the huge redundancy of the IBP identities [8]. This redundancy results in many unnecessary calculations which make the reduction slow.

Another approach to the reduction is a derivation of the symbolic rules which can be applied to any problem of a given class. Its advantages are obvious: nothing is being solved in the process of reduction, therefore, the reduction is very fast.

Symbolic rules are small in size, so, they can be easily saved for future calculations. The bottleneck of the approach is the search of the symbolic rules, a stage which seems to require a lot of manual work.

An ideal solution of the reduction problem would be, therefore, designing a program which automatically finds the symbolic reduction rules at the first stage and then uses those rules for the reduction. Much effort has been devoted to the developement of the approach connected with the notion of the Groebner basis [9–12]. This is due to the fact that the problem of IBP reduction is very similar to the reduction of the elements of some algebra with respect to its ideal. In the latter problem there is a known algorithm of the construction of the Groebner basis — the Buchberger's algorithm and its generalizations. When the Groebner basis is constructed, the reduction can be performed unambiguously and very fast. If this approach always worked for the IBP reduction, it would definitely be the most complete solution of the reduction problem. However, there is a small peculiarity of the IBP identities which results in the essential obstacle when generalizing the Buchberger's algorithm to the IBP reduction. As it was shown in Ref. [8], the IBP reduction problem can be reduced to the problem of reduction of some noncommuting polynomial ring with respect to the direct sum of the left and right ideals. While the Buchberger's algorithm seperately works for both left and right ideals, it appears to be difficult to generalize this algorithm to the desired case. Probably, the only, partly successful, attempt of this approach has been made in `FIRE`, where the notion of s-bases [11, 12] have been used.

Of course, there are many peculiar features of the IBP identities which make the IBP reduction not the general-case reduction with respect to the direct sum of the left and right ideals. In particular, as it was shown in Ref. [8], the generating set of the left ideal appears to be equipped with a Lie-algebraic structure, and that of the right ideal consists of the commuting elements. It is quite possible that at some point in the future a general solution of the IBP reduction problem will appear. Meanwhile, one can try to develop some heuristic agorithms which are not guaranteed to work for each case, but, nevertheless, are useful from the practical point of view. With the lack of a systematic approach (i.e., a strict algorithm), this developement can be quite challenging.

This short note describes a *Mathematica* package `LiteRed` which can be considered as an attempt of the implementation of the heuristic approach to the IBP reduction. The package can be downloaded as a zip archive from

    http://www.inp.nsk.su/~lee/programs/LiteRed/

2 General setup

Assume that we are interested in the calculation of the L-loop integral depending on the E external momenta $p_1, \ldots, p_E$. There are $N = L(L+1)/2 + LE$ scalar products

depending on the loop momenta l_i:

$$s_{ij} = l_i \cdot q_j \,, \ 1 \leqslant i \leqslant L, j \leqslant L + E, \tag{2.1}$$

where $q_{1,\ldots,L} = l_{1,\ldots,L}$, $q_{L+1,\ldots,L+E} = p_{1,\ldots,E}$.

The loop integral has the form

$$J\left(\boldsymbol{n}\right) = J(n_1, n_2, \ldots, n_N) = \int d^d l_1 \ldots d^d l_L j(\boldsymbol{n}) = \int \frac{d^d l_1 \ldots d^d l_L}{D_1^{n_1} D_2^{n_2} \ldots D_N^{n_N}}\,, \tag{2.2}$$

where the scalar functions D_α are linear polynomials with respect to s_{ij}. The functions D_α are assumed to be linearly independent and to form a complete basis in the sense that any non-zero linear combination of them depends on the loop momenta, and any s_{ik} can be expressed in terms of D_α. Thus, each integral is associated with a point in $\mathbb{Z}^N$. Some of the functions D_α correspond to the denominators of the propagators, the other correspond to the irreducible numerators. E.g., the K-legged L-loop diagram corresponds to $E = K - 1$ and the maximal number of denominators is $M = E + 3L - 2$, so that the rest $N - M = (L - 1)(L + 2E - 4)/2$ functions correspond to irreducible numerators.

IBP identities The IBP identities [1, 2] are based on the fact that, in the dimensional regularization, the integral of the total derivative is zero. They are derived from the identity

$$0 = \int d^d l_1 \ldots d^d l_L \frac{\partial}{\partial l_i} \cdot q_k j(\boldsymbol{n})\,. \tag{2.3}$$

Performing the differentiation on the right-hand side and expressing the scalar products via D_α, we obtain the recurrence relation for the function J.

LI identities There is also another class of identities, called Lorentz-invariance (LI) identities due to the fact that the integral (2.2) is Lorentz scalar [13]. They have the form

$$p_i^\mu p_j^\nu \left(\sum_k p_{k[\nu} \frac{\partial}{\partial p_k^{\mu]}} \right) J(n_1, n_2, \ldots, n_N) = 0\,. \tag{2.4}$$

The differential operator in braces is nothing but the generator of the Lorentz transformation in the linear space of scalar functions depending on p_k. Again, performing the differentiation on the right-hand side, we obtain LI identity. Though these identities can be represented as a linear combination of the IBP identities [8], they prove to be useful in real-life reduction.

Sectors The notion of sectors can be introduced as follows. The $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$ *sector*, where $\theta_i = 0, 1$, is a set of all points $(n_1, \ldots, n_N)$ in $\mathbb{Z}^N$ whose coordinates obey the condition

$$\Theta\left(n_\alpha - 1/2\right) = \theta_\alpha \tag{2.5}$$

In particular, the point $(\theta_1, \ldots, \theta_N)$ belongs to the $(\theta_1, \ldots, \theta_N)$ sector, and will be referred to as the *corner point of the sector*. Owing to this definition, the integrals of the same sector have the same set of denominators.

Scaleless integrals   The scaleless integral can be defined as the one which gains additional non-unity factor under some linear transformation of the loop momenta. Obviously, if $j(\theta_1, \ldots, \theta_N)$ is scaleless, then all integrals of the sector $(\theta_1, \ldots, \theta_N)$ are zero. We will call such a sector a *zero sector*. A simple and convenient criterion of zero sectors has been formulated in Ref. [8]. According to this criterion, the sector is zero if the solution of the IBP equations in the corner point $(\theta_1, \ldots, \theta_N)$ result in the identity $j(\theta_1, \ldots, \theta_N) = 0$. Note that this criterion may miss some scaleless sectors. Though this seems to be a small problem (undetected zero sectors are simply reduced to lower sectors), let us explain on a simple example why this happens.

Let us consider the massless one-loop onshell propagator integral

$$J(n_1, n_2) = \int \frac{d^d l}{[l^2]^{n_1} \left[(l-k)^2\right]^{n_2}}, \quad k^2 = 0 \,.$$

Obviously, this integral is zero for any n_1 and n_2. However, it can be explicitly checked that there is no linear combination of the operators $\partial_l \cdot l$ and $\partial_l \cdot k$ which acts as identity on the integrand for $n_1 = n_2 = 1$. In other words, the solution of the IBP identities in the corner point of the sector $(1, 1)$ does not result directly to $J(1, 1) = 0$ (though, it results to, e.g., $J(1, 1) \propto J(0, 2)$). In order to prove that the integral $J(1, 1)$ is scaleless, let us consider instead the following operator

$$O = \partial_l \cdot \left(l + (l \cdot k)\, \tilde{k} - \left(l \cdot \tilde{k}\right) k\right),$$

where $\tilde{k}$ is an auxiliary vector chosen to satisfy the conditions $\tilde{k}^2 = 0$ and $\tilde{k} \cdot k = 1$. It is easy to check that $O j(1, 1) = (d - 4)\, j(1, 1)$. Since the operator O is a generator of the linear transformation $l \to l + \epsilon\left(l + (l \cdot k)\, \tilde{k} - \left(l \cdot \tilde{k}\right) k\right)$, the integral $j(1, 1)$ is scaleless. The reason why the IBP identities failed to lead to the identity $J(1, 1) = 0$ is that the construction of this identity required introduction of the auxiliary vector $\tilde{k}$. There is an interesting open question: whether the introduction of the auxiliary vectors (or tensors) may lead to a new kind of the identities independent of IBP identities (unlikely) or dependent on them, but still useful for real-life reduction.

Symmetry relations   In many cases there exist nontrivial linear transformations of the loop momenta which map the set of the denominators of one sector on the set of the denominators of the same, or another, sector. Those transformations have the form

$$l_i \to M_{ij} q_j,$$

where M_{ij} is a $L \times (L + E)$ matrix. It is easy to understand that, for nonzero sectors, there is only a finite number of such transformations, all subjected to the

condition $\left| \det \left\{ M_{ij} |_{i,j=1,\ldots L} \right\} \right| = 1$. Those transformations induce some mappings of the denominator set and also mappings of the numerators into the linear polynomials of D_i. These mappings give nontrivial identities between the integrals of two different (or one) sectors, which are conventionally called *symmetry relations* (SR).

Ordering   For the reduction procedure to work, it is necessary to define some suitable ordering of the integrals, i.e., the ordering in $\mathbb{Z}^N$. It is natural to consider the integrals with less denominators to be simpler. This defines a partial ordering of the sectors. We can extend this ordering to the complete one by, e.g., saying, that the two sectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ with equal number of denominators are ordered lexicographically. For the integrals in sector $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_N)$ with $K = \sum_{\alpha=1}^N \Theta(n_\alpha - 1/2)$ denominators we choose the folowing ordering. For the integral $J(n_1,\ldots,n_N)$ we determine the ordering weight $\mathbf{w} = (w_{-1}, w_0, w_1,\ldots,w_N)$, where $w_{-1} = \sum_{\alpha=1}^N |n_\alpha|$ is a total power of the denominators and numerators, $w_0 = \sum_{\alpha=1}^N (1-\theta_\alpha) n_\alpha$ is the total power of the numerators, and $(w_1,\ldots,w_N)$ is obtained from $(-|n_1|,\ldots,-|n_N|)$ by shifting powers, corresponding to denominators, to K left-most positions. Then, the two integrals in one sector are compared by comparing the left-most dictinct entries of their ordering weights. In particular, the simplest integral in the sector $(\theta_1,\ldots,\theta_N)$ is $J(\theta_1,\ldots,\theta_N)$.

Of course, the choice of the ordering is not unique. In principle, for the possibility of the reduction, there is only one strictly required property of the ordering. It is that for any integral $J(\mathbf{n})$ there is only finite number of the integrals simpler than it (this number depends, of course, on $\mathbf{n}$). However, the following condition of the chosen ordering is essential for our consideration. For a given sector all components of the ordering weight $\mathbf{w}$ are just linear combinations of n_i. In particular, it leads to the fact that the relation $J(\mathbf{n}_1) \prec J(\mathbf{n}_2)$ between the integrals in the same sector is invariant with respect to the shift of both $\mathbf{n}_1$ and $\mathbf{n}_2$ by some $\delta\mathbf{n}$, provided that $J(\mathbf{n}_1 + \delta\mathbf{n})$ and $J(\mathbf{n}_2 + \delta\mathbf{n})$ also belong to the same sector.

Differential equations   As it was mentioned above, the differential equations can be used for finding the master integrals. The simplest type of such equations is the differential equation with respect to the mass. Probably, the first example of their application is presented in Refs. [14–16]. The differential equations with respect to the invariant constructed of the external momenta have been introduced and applied in Refs. [13, 17, 18]. The peculiarity of the latter case is due to the fact that, though the *integral* depends on the external momenta only via their scalar products, the *integrand* also depends on the scalar products of the external momenta and loop momenta. Therefore, before differentiating the integral, it is necessary to express the derivative with respect to the invariant via the derivatives with respect to the external momenta. For example, if the integral depends on p^2, q^2 and $p \cdot q$, the derivative with respect to $p \cdot q$ at fixed p^2 and q^2 can be expressed in two equivalent

ways

$$\frac{\partial}{\partial\left(p\cdot q\right)}J\left(\mathbf{n}\right)=\frac{\left(p\cdot q\right)p-p^2q}{\left(p\cdot q\right)^2-p^2q^2}\cdot\frac{\partial}{\partial p}J\left(\mathbf{n}\right)=\frac{\left(p\cdot q\right)q-q^2p}{\left(p\cdot q\right)^2-p^2q^2}\cdot\frac{\partial}{\partial q}J\left(\mathbf{n}\right)\,.$$

In general case, when there are $E>2$ external vectors, we have the following formulas:

$$\frac{\partial}{\partial\left(p_1\cdot p_2\right)}J\left(\mathbf{n}\right)=\sum\left[\mathbb{G}^{-1}\right]_{i2}p_i\cdot\partial_{p_1}J\left(\mathbf{n}\right)=\sum\left[\mathbb{G}^{-1}\right]_{i1}p_i\cdot\partial_{p_2}J\left(\mathbf{n}\right)\,,$$
$$\frac{\partial}{\partial\left(p_1^2\right)}J\left(\mathbf{n}\right)=\frac{1}{2}\sum\left[\mathbb{G}^{-1}\right]_{i1}p_i\cdot\partial_{p_1}J\left(\mathbf{n}\right)\,. \tag{2.6}$$

where $\mathbb{G}=\mathbb{G}\left(p_1,\ldots,p_E\right)=\begin{pmatrix}p_1^2&\cdots&p_1\cdot p_E\\\vdots&\ddots&\vdots\\p_1\cdot p_E&\cdots&p_E^2\end{pmatrix}$ is a Gram matrix.

Acting by the operator on the right-hand side on the integrand and performing the IBP reduction, one obtains the differential equation for $J\left(\mathbf{n}\right)$.

Dimensional recurrences   Dimensional recurrences have been introduced in Ref. [19]. Since their introduction, they have been successfully applied to the calcultion of the different integrals, see Refs. [19–21]. Recently, a method of calculation of the multiloop master integrals, based on the dimensional recurrences and analytical properties of the multiloop integrals as functions of d, has been introduced in Ref. [22] and successfully applied in Refs. [23–30]. For further purposes it is convenient to introduce the operators A_α and B_α, see Ref. [8], acting as follows

$$\left(A_iJ^{(\mathcal{D})}\right)\left(n_1,\ldots,n_N\right)=n_iJ^{(\mathcal{D})}\left(n_1,\ldots,n_i+1,\ldots,n_N\right),$$
$$\left(B_iJ^{(\mathcal{D})}\right)\left(n_1,\ldots,n_N\right)=J^{(\mathcal{D})}\left(n_1,\ldots,n_i-1,\ldots,n_N\right). \tag{2.7}$$

The original derivation of the dimensional recurrence relation in Ref. [19] relied on the parametric representation. The result of this derivation for the integrals which can be presented as a graph has a nice and compact form

$$J^{(d-2)}\left(\mathbf{n}\right)=\mu^L\sum_{\text{trees}}\left(A_{i_1}\ldots A_{i_L}J^{(d)}\right)\left(\mathbf{n}\right), \tag{2.8}$$

where $i_1,\ldots,i_L$ numerate the chords of the tree, and $\mu=\pm1$ for the Euclidean/Minkovskian case, respectively. For computer implementations, probably, a more convenient formula has been derived in Ref. [26]. It reads

$$J^{(d-2)}\left(\mathbf{n}\right)=\left(\mu/2\right)^L\det\left\{2^{\delta_{ij}}\frac{\partial D_k}{\partial s_{ij}}A_k|_{i,j=1\ldots L}\right\}J^{(d)}\left(\mathbf{n}\right). \tag{2.9}$$

Note that this formula is valid also for the integrals with numerators and dots. We will call Eqs. (2.8) and (2.9) the *raising dimensional recurrence relation*s.

For completeness, we present also the *lowering dimensional recurrence relation*, obtained in Ref. [22]

$$J^{(d+2)}(\mathbf{n}) = \frac{(2\mu)^L [V(p_1, \ldots, p_E)]^{-1}}{(d-E-L+1)_L} P(B_1, \ldots, B_N) J^{(d)}(\mathbf{n}), \qquad (2.10)$$

where $\alpha_L = \alpha(\alpha+1)\ldots(\alpha+L-1)$ is the Pochhammer symbol, $V(v_1, \ldots v_k) = \det \mathbb{G}(v_1, \ldots v_k)$ is the Gram determinant, and $P(D_1, \ldots, D_N) = V(q_1, \ldots q_{L+E})$.

If we consider one of these dimensional recurrence relations for the master integral and make the IBP reduction of the right-hand side, we will obtain the difference equation for this master integral.

3   Using the `LiteRed` package

A typical package usage includes two stages: the search of the reduction rules and their application for the reduction. During the first stage the definitions, related to the basis, in particular, the reduction rules, can be saved to the disk. These definitions should be loaded on the second stage and applied for the reduction. The basic example of a program searching the reduction rules is presented in Fig. 1. This program finds the reduction rules for the two-loop onshell massive propagator.

Let us provide some comments. At each stage the program generates some objects which are then used at the later stages. E.g., the command `NewBasis[p2,{sp[p-l]-1,sp[p-l-r]-1,sp[p-r]-1,l,r},{l,r}]` sets up a new basis `p2`, consisting of the functions D_α depending linearly on the scalar products involving loop momenta `l`, `r` (`sp[a,b]` stands for the scalar product of `a` and `b`, `sp[a]` is a shortcut for `sp[a,a]`). This procedure checks that the set of D_α is linearly independent and complete and generates several objects `Ds[p2]`, `SPs[p2]`, `LMs[p2]`, `EMs[p2]`, `Toj[p2]`. The meaning of these objects should be clear from the output immediately following the `NewBasis` command. The objects `IBP[p2]` and `LI[p2]` generated by `GenerateIBP` call give the functions which return the Integration-By-Parts identities and Lorentz-Invariance identities in a given point, see Eqs. (2.3) and (2.4). E.g., `IBP[p2][n1,n2,n3,n4,n5]` gives a list of IBP identities in a general point. The IBP identities generated by `GenerateIBP` procedure are necessary for the detemination of zero sectors by the `AnalyzeSectors` procedure. This procedure generates the list `ZeroSectors[p2]` of zero sectors as well as the list `SimpleSectors[p2]` of the simplest nonzero sectors. The latter is used in the procedure `FindSymmetries`. This procedure finds internal and mutual symmetries of the nonzero sectors and generates the lists `MappedSectors[p2]` and `UniqueSectors[p2]`. Any sector from the former list can be mapped onto some sector from the latter. The substitution rules for such a mapping can be found in `jRules[p2,...]` for each sector `js[p2,...]` from the list `MappedSectors[p2]`.

The central procedure which tries to construct the complete set of symbolic rules for a given sector is `SolvejSector` which we apply to each unique sector in this example. The result of its work is the set of rules for each `js[p2,...]` from the `UniqueSectors[p2]` list. All rules found are saved in the directory "p2 dir" and ready to use for the reduction. The typical program performing the reduction is shown in Fig. 2.

Drawing graphs There is a possibility to draw graphs, corresponding to the integrals and sectors. We remark that, in principle, the graph is determined, up to some equivalences, by the set of internal lines. The possibility to automatically determine the graph, corresponding to the set of denominators, is planned in the future versions of the package. Meanwhile, the present version implements the following. After defining the basis, one can attach a graph to the highest sector(s) (which can be depicted as a graph) by the command `AttachGraph`. Then the graph for all subsectors is determined automatically.

For the above example there are no irreducible numerators, so the sector `js[p2,1,1,1,1,1]` can be depicted by the graph. Then, the graph is attached by the command

```
AttachGraph[js[p2, 1, 1, 1, 1, 1], {
    (*5 internal lines*)
    {1 → 2, "1"},
    {2 → 3, "1"},
    {3 → 4, "1"},
    {1 → 3, "0"},
    {2 -> 4, "0"},
    (*2 external lines*)
    {0 → 1, "p"},
    {4 → 0, "p"}
}];
```

Then, a graph of, say, sector `js[p2,1,1,1,1,0]`, can be drawn with the command `GraphPlot[jGraph[js[p2,1,1,1,1,0]]]`. Note that `GraphPlot` is a standard Mathematica function and the presentation of the graph can be altered using its options, as can be found in the examples distributed with the package.

Additional tools Several additional tools are included in the package:

1. `Dinv[j[...],sp[p,q]]` — the derivative with respect to the invariant constructed of the external momenta, Eq. (2.6).

2. `RaisingDRR[basis,...]` — the right-hand side of the dimensional recurrence relation $j^{(d-2)}$ (basis, ...) = ... , Eq. (2.9). Note that the factor $\mu^L = -1$ for Minkovskian metrics and odd number of loops should be taken into account manually.

3. `LoweringDRR[basis,...]` — the right-hand side of the dimensional recurrence relation $j^{(d+2)}$ (basis, ...) = ..., Eq. (2.10). Note that the factor $\mu^L = -1$

for Minkovskian metrics and odd number of loops should be taken into account manually.

4. `FeynParUF[js[basis,...]]` — the functions U and F entering the Feynman parametrization of the integrals in the given sector.

Learning more   There are several reduction examples in the directory **Examples** of the archive file. One is encouraged to examine these examples for some hints of the package usage. Another good starting point to know more about the functions of the package is to submit a command `?LiteRed'*`.

4   Implementation notes

The `LiteRed` package depends on small packages `Types, Numbers, Vectors,` and `LinearFunctions`, which are also included in the distributive. The `Types` package allowas one to define types and their transformation rules (e.g. vector plus vector is a vector). The packages `Numbers, Vectors` introduce specific types for number and vector variables. The package `LinearFunctions` contains the function `LFDistribute` which distributes linear functions over sum and pulls out the numbers and expressions having type `Number`.

`AnalyzeSectors` uses Criterion 1 from Ref. [8] for the determination of zero sectors. `FindSymmetries` uses a combined approach based on the Feynman parametrization and on the loop momenta shifts. In the first stage `FindSymmetries` finds mappings between the simple sectors using approach based on Feynman parametrization very similar to the one described in Ref. [31]. This procedure works sufficiently fast even for complicated examples. E.g., for the four-loop onshell mass operator topologies the typical working time is a few minutes.

As it was stressed above, the package performs a heuristic search of the reduction rules. The result of this search may strongly depend on the order in which the functions D_α are listed, as well as on the choice of the irreducible numerators. Therefore, in case the program fails to find reduction rules, it makes sense to try changing the irreducible numerators, as well as the listing order of D_α.

5   Conclusion

In this short note we have presented a *Mathematica* package `LiteRed` performing the IBP reduction of the multiloop integrals. The package is based on the heuristic search of the reduction rules (the procedure `SolvejSector`) and further application of the rules found to the reduction problem. If the heuristic search finishes successfully, the rules found present a very effective solution of the reduction problem for the given class of the integrals. Similar to the algorithm of costruction of s-bases in Ref. [12],

the heuristic search of the reduction rules is not proved to terminate (and, in fact, seems to be not terminating in some complicated cases). However, it appears that the search of the reduction rules, as implemented in `LiteRed`, succeeds for a larger class of physically interesting cases, see **Examples** folder in the distributive.

The package can be useful also for some other purposes. In particular, the procedure `FindSymmetries` can be used to find mapping between equivalent sectors.

Acknowledgments

References

[1] K. G. Chetyrkin and F. V. Tkachov, *Integration by parts: The algorithm to calculate β-functions in 4 loops*, Nucl. Phys. B **192** (1981) 159.

[2] F. V. Tkachov, *A theorem on analytical calculability of 4-loop renormalization group functions*, Physics Letters B **100** (Mar., 1981) 65–68.

[3] S. Laporta, *High precision calculation of multiloop Feynman integrals by difference equations.*, Int. J. Mod. Phys. A **15** (2000) 5087.

[4] C. Anastasiou and A. Lazopoulos, *Automatic integral reduction for higher order perturbative calculations*, JHEP **0407** (2004) 046, [hep-ph/0404258].

[5] A. V. Smirnov, *Algorithm FIRE – Feynman Integral REduction*, JHEP **10** (2008) 107, [arXiv:0807.3243].

[6] C. Studerus, *Reduze-Feynman integral reduction in c++*, Comput.Phys.Commun. **181** (2010) 1293–1300, [arXiv:0912.2546].

[7] A. von Manteuffel and C. Studerus, *Reduze 2 - Distributed Feynman Integral Reduction*, arXiv:1201.4330.

[8] R. N. Lee, *Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals*, Journal of High Energy Physics **07** (2008) 031.

[9] O. V. Tarasov, *Reduction of Feynman graph amplitudes to a minimal set of basic integrals*, Acta Phys. Polon. B **29** (1998) 2655.

[10] V. P. Gerdt, *Grobner bases in perturbative calculations*, Nucl.Phys.Proc.Suppl. **135** (2004) 232–237, [hep-ph/0501053].

[11] A. V. Smirnov and V. A. Smirnov, *S-bases as a tool to solve reduction problems for feynman integrals*, 2006.

[12] A. V. Smirnov, *An algorithm to construct Grobner bases for solving integration by parts relations*, *JHEP* **0604** (2006) 026, [`hep-ph/0602078`].

[13] T. Gehrmann and E. Remiddi, *Differential equations for two-loop four-point functions*, *Nucl. Phys. B* **580** (2000) 485, [`hep-ph/9912329`].

[14] A. V. Kotikov, *Differential equations method: New technique for massive Feynman diagrams calculation*, *Phys. Lett.* **B254** (1991) 158–164.

[15] A. V. Kotikov, *Differential equations method: The calculation of vertex type Feynman diagrams*, *Phys. Lett.* **B259** (1991) 314–322.

[16] A. V. Kotikov, *Differential equation method: The calculation of N point Feynman diagrams*, *Phys. Lett.* **B267** (1991) 123–127.

[17] T. Gehrmann and E. Remiddi, *Two loop master integrals for $\gamma^* \to 3$ jets: The planar topologies*, *Nucl.Phys.* **B601** (2001) 248–286, [`hep-ph/0008287`].

[18] T. Gehrmann and E. Remiddi, *Two loop master integrals for $\gamma^* \to 3$ jets: The nonplanar topologies*, *Nucl.Phys.* **B601** (2001) 287–317, [`hep-ph/0101124`].

[19] O. V. Tarasov, *Connection between Feynman integrals having different values of the space-time dimension*, *Phys. Rev. D* **54** (1996) 6479, [`hep-th/9606018`].

[20] O. V. Tarasov, *Application and explicit solution of recurrence relations with respect to space-time dimension*, *Nucl. Phys. Proc. Suppl.* **89** (2000) 237–245, [`hep-ph/0102271`].

[21] O. V. Tarasov, *Hypergeometric representation of the two-loop equal mass sunrise diagram*, *Phys. Lett.* **B638** (2006) 195–201, [`hep-ph/0603227`].

[22] R. N. Lee, *Space-time dimensionality D as complex variable: Calculating loop integrals using dimensional recurrence relation and analytical properties with respect to D*, *Nuclear Physics B* **830** (2010) 474, [`arXiv:0911.0252`].

[23] R. N. Lee, A. V. Smirnov, and V. A. Smirnov, *Analytic results for massless three-loop form factors*, *Journal of High Energy Physics* **2010** (Apr., 2010) 1–12, [`arXiv:1001.2887`].

[24] R. N. Lee and V. A. Smirnov, *Analytic epsilon expansions of master integrals corresponding to massless three-loop form factors and three-loop g-2 up to four-loop transcendentality weight*, *JHEP* **1102** (2011) 102, [`arXiv:1010.1334`].

[25] R. N. Lee, A. V. Smirnov, and V. A. Smirnov, *Dimensional recurrence relations: an easy way to evaluate higher orders of expansion in ϵ*, *Nucl. Phys. Proc. Suppl.* **205-206** (2010) 308–313, [`arXiv:1005.0362`].

[26] R. N. Lee, *Calculating multiloop integrals using dimensional recurrence relation and D-analyticity*, *Nucl. Phys. Proc. Suppl.* **205-206** (2010) 135–140, [`arXiv:1007.2256`].

[27] R. N. Lee and I. S. Terekhov, *Application of the DRA method to the calculation of the four-loop QED-type tadpoles*, *JHEP* **1101** (2011) 068, [`arXiv:1010.6117`].

[28] R. N. Lee, A. V. Smirnov, and V. A. Smirnov, *On epsilon expansions of four-loop non-planar massless propagator diagrams*, *Eur.Phys.J.* **C71** (2011) 1708, [arXiv:1103.3409].

[29] R. N. Lee, A. V. Smirnov, and V. A. Smirnov, *Master integrals for four-loop massless propagators up to transcendentality weight twelve*, *Nucl.Phys.* **B856** (2012) 95–110, [arXiv:1108.0732].

[30] R. N. Lee and V. A. Smirnov, *The Dimensional Recurrence and Analyticity method for multicomponent master integrals: Using unitarity cuts to construct homogeneous solutions*, arXiv:1209.0339. accepted for publication in JHEP.

[31] A. Pak, *The toolbox of modern multi-loop calculations: novel analytic and semi-analytic techniques*, *J.Phys.Conf.Ser.* **368** (2012) 012049, [arXiv:1111.0868].

```
In[1]:=  << LiteRed`(*Loading the package*)

         **************** LiteRed v1.0 ********************
         Author: Roman N. Lee, Budker Institute of Nuclear Physics, Novosibirsk.
         Release Date: 12.12.2012
         LiteRed stands for Loop InTEgrals REDuction.
         The package is designed for the search and application of the
             Integration-By-Part reduction rules. It also contains some other useful tools.
         See ?LiteRed`* for a list of functions.

In[2]:=  SetDirectory[NotebookDirectory[]];(*Setting working directory*)
         SetDim[d]; (*d stands for the dimensionality*)
         Declare[{l, r, p}, Vector]; (*vector variables*)
         sp[p, p] = 1; (*constraint*)

In[4]:=  NewBasis[p2, {sp[p - l] - 1, sp[p - l - r] - 1, sp[p - r] - 1, l, r},
             {l, r}, Directory → "p2 dir"];(*Basis definition. The option Directory→
          "p2 dir" determines the directory where all definitions for the basis will be saved*)

         Valid basis.
             Ds[p2]    denominators,
             SPs[p2]   scalar products involving loop momenta,
             LMs[p2]   loop momenta,
             EMs[p2]   external momenta,
             Toj[p2]   rules to transform scalar products to denominators.
         The definitions of the basis will be saved in p2 dir

         DiskSave::dir : The directory p2 dir has been created.

In[5]:=  GenerateIBP[p2](*IBP generation*)

         Integration-By-Part&Lorentz-Invariance identities are generated.
             IBP[p2]   integration-by-part identities,
             LI[p2]    Lorentz invariance identities.

In[6]:=  AnalyzeSectors[p2](*Zero and simple sectors determination*)

         Found 15 zero sectors out of 32.
             ZeroSectors[p2]    zero sectors,
             NonZeroSectors[p2]    nonzero sectors,
             SimpleSectors[p2]    simple sectors (no nonzero subsectors),
             BasisSectors[p2]    basis sectors (at least one immediate subsector is zero),
             ZerojRule[p2]   a rule to nullify all zero j[p2 ].

In[7]:=  FindSymmetries[p2](*Finding unique and mapped sectors*)

         Found 8 mapped sectors and 9 unique sectors.
             UniqueSectors[p2]    unique sectors.
             MappedSectors[p2]    mapped sectors.
             SR[p2][ ]    symmetry relations for j[p2, ] from UniqueSectors[p2].
             jSymmetries[p2, ]    symmetry rules for the sector js[p2, ] in UniqueSectors[p2].
             jRules[p2, ]    reduction rules for j[p2, ] from MappedSectors[p2].

In[8]:=  Timing[SolvejSector /@ UniqueSectors[p2]]

         Sector js[p2, 0, 1, 1, 0, 0]

             Master integrals found: j[p2, 0, 1, 1, 0, 0].
             jRules[p2, 0, 1, 1, 0, 0]   reduction rules for the sector.
             MIs[p2]   updated list of the masters.

         ...

         Sector js[p2, 1, 1, 1, 1, 1]

             Master integrals found: none.
             jRules[p2, 1, 1, 1, 1, 1]   reduction rules for the sector.
             MIs[p2]   updated list of the masters.

Out[8]= {10.624, {1, 1, 0, 0, 1, 0, 0, 0, 0}}

In[9]:=  DiskSave[p2];
         Quit[]

         DiskSave::overwrite : The file p2 dir/p2 has been overwritten.
```

Figure 1. A simple program of finding IBP rules.

```
In[1]:= << LiteRed`(*Loading the package*)

         **************** LiteRed v1.0 *********************
         Author: Roman N. Lee, Budker Institute of Nuclear Physics, Novosibirsk.
         Release Date: 12.12.2012
         LiteRed stands for Loop InTEgrals REDuction.
         The package is designed for the search and application of the
             Integration-By-Part reduction rules. It also contains some other useful tools.
         See ?LiteRed`* for a list of functions.

In[2]:= SetDirectory[NotebookDirectory[]];(*Setting working directory*)
         SetDim[d]; (*d stands for the dimensionality*)
         Declare[{l, r, p}, Vector]; (*vector variables*)
         sp[p, p] = 1; (*constraint*)

In[4]:= << "p2 dir/p2"(*Loading basis*)

Out[4]= p2 dir

In[5]:= Timing[IBPReduce[j[p2, 1, 2, 3, 4, 5], "file1"]]
         (*Reduction. Second argument is the file name to save the result*)
```

$$
\text{Out[5]}= \Big\{ 30.093, \, - \Big(9 \, \big(-19\,405\,749\,861\,751\,770\,316\,800 + 63\,337\,852\,789\,213\,794\,140\,160\,d \,-
$$
$$
96\,838\,493\,039\,514\,249\,854\,976\,d^2 + 92\,253\,915\,806\,057\,780\,477\,952\,d^3 -
$$
$$
61\,441\,859\,881\,785\,836\,896\,256\,d^4 + 30\,423\,441\,751\,985\,908\,285\,440\,d^5 -
$$
$$
11\,626\,976\,809\,896\,282\,329\,088\,d^6 + 3\,513\,911\,887\,978\,934\,254\,592\,d^7 -
$$
$$
853\,504\,812\,879\,165\,864\,192\,d^8 + 168\,392\,515\,692\,831\,666\,432\,d^9 -
$$
$$
27\,160\,529\,195\,747\,468\,416\,d^{10} + 3\,592\,160\,347\,775\,198\,784\,d^{11} -
$$
$$
389\,515\,934\,714\,662\,752\,d^{12} + 34\,517\,755\,931\,398\,912\,d^{13} - 2\,483\,176\,910\,815\,800\,d^{14} +
$$
$$
143\,450\,814\,863\,448\,d^{15} - 6\,546\,228\,381\,621\,d^{16} + 230\,251\,567\,545\,d^{17} -
$$
$$
6\,011\,210\,070\,d^{18} + 109\,500\,174\,d^{19} - 1\,240\,029\,d^{20} + 6561\,d^{21} \big) \, j[p2, 0, 1, 0, 1, 1] \Big) \Big/
$$
$$
(67\,108\,864\,(-13+d)\,(-12+d)\,(-11+d)\,(-9+d)\,(-23+2\,d)\,(-21+2\,d)\,(-19+2\,d)
$$
$$
(-17+2\,d)\,(-15+2\,d)\,(-13+2\,d)\,(-11+2\,d)\,(-9+2\,d)\,(-7+2\,d)) +
$$
$$
\Big(\big(-283\,807\,832\,792\,840\,965\,324\,800 + 948\,786\,748\,942\,911\,447\,490\,560\,d -
$$
$$
1\,462\,663\,235\,537\,746\,082\,583\,552\,d^2 + 1\,385\,351\,619\,026\,760\,469\,495\,296\,d^3 -
$$
$$
905\,881\,158\,492\,949\,979\,133\,120\,d^4 + 435\,489\,259\,998\,064\,093\,702\,272\,d^5 -
$$
$$
159\,957\,413\,669\,416\,877\,168\,624\,d^6 + 46\,033\,302\,705\,355\,987\,593\,184\,d^7 -
$$
$$
10\,555\,172\,064\,614\,400\,902\,260\,d^8 + 1\,949\,560\,973\,441\,170\,302\,024\,d^9 -
$$
$$
291\,940\,360\,748\,911\,454\,721\,d^{10} + 35\,537\,943\,518\,009\,683\,032\,d^{11} -
$$
$$
3\,513\,393\,262\,609\,951\,175\,d^{12} + 280\,771\,095\,481\,687\,696\,d^{13} - 17\,972\,249\,382\,323\,578\,d^{14} +
$$
$$
907\,850\,680\,750\,800\,d^{15} - 35\,362\,115\,157\,090\,d^{16} + 1\,024\,180\,145\,448\,d^{17} -
$$
$$
20\,759\,687\,525\,d^{18} + 262\,701\,688\,d^{19} - 1\,561\,555\,d^{20} \big) \, j[p2, 0, 1, 1, 0, 0] \big) \Big/
$$
$$
\big(536\,870\,912\,(-15+d)\,(-14+d)\,(-13+d)^2\,(-12+d)^2\,(-11+d)^2\,(-10+d)
$$
$$
(-9+d)^2\,(-8+d)\,(-7+d)\,(-6+d)\,(-5+d)\,(-4+d)\,(-3+d) \big) +
$$
$$
\big(81\,\big(1\,056\,184\,729\,600 - 1\,829\,751\,275\,520\,d + 1\,379\,912\,937\,472\,d^2 - 597\,241\,499\,904\,d^3 +
$$
$$
164\,475\,088\,128\,d^4 - 30\,146\,949\,360\,d^5 + 3\,728\,160\,432\,d^6 - 307\,422\,216\,d^7 +
$$
$$
16\,189\,632\,d^8 - 492\,075\,d^9 + 6561\,d^{10} \big) \, j[p2, 1, 1, 1, 0, 0] \big) \Big/
$$
$$
\big(268\,435\,456\,(-14+d)\,(-12+d)^2\,(-10+d)\,(-8+d)\,(-6+d)\,(-4+d) \big) \Big\}
$$

```
In[6]:= Quit[]
```

Figure 2. A typical program of applying IBP rules.