# On the Reduction and Evaluation of Generalized Polylogarithms

**Hjalte Frellesvig***

*Institute of Nuclear and Particle Physics, NCSR "Demokritos",*
*Patriarchou Gregoriou E. & Neapoleos 27, Agia Paraskevi, 15310, Greece*
*E-mail:* frellesvig@inp.demokritos.gr

**Damiano Tommasini**

*Institute of Nuclear and Particle Physics, NCSR "Demokritos",*
*Patriarchou Gregoriou E. & Neapoleos 27, Agia Paraskevi, 15310, Greece*
*E-mail:* tommasini@inp.demokritos.gr

**Christopher Wever**

*Institute of Nuclear and Particle Physics, NCSR "Demokritos",*
*Patriarchou Gregoriou E. & Neapoleos 27, Agia Paraskevi, 15310, Greece &*
*Institute for Theoretical Particle Physics (TTP), Karlsruhe Institute of Technology,*
*Engesserstraße 7, D-76128 Karlsruhe, Germany &*
*Institute for Nuclear Physics (IKP), Karlsruhe Institute of Technology,*
*Hermann-von-Helmholtz-Platz 1, D-76344 Eggenstein-Leopoldshafen, Germany*
*E-mail:* christopher.wever@kit.edu

The talk and the summary below was based on the paper "On the reduction of Generalized Polylogarithms to $\text{Li}_n$ and $\text{Li}_{2,2}$ and on the reduction thereof" [1] by the three authors, published in March 2016 in JHEP.

---

*Speaker.

## 1. Introduction

The Generalized Polylogarithmic function (GPL) [2, 3] is defined recursively as

$$G(a_1,\ldots,a_n;x) = \int_0^x \frac{dz}{z-a_1} G(a_2,\ldots,a_n;z),$$

(1.1)

with

$$G(\underbrace{0,\ldots,0}_{n};x) = \frac{\log^n(x)}{n!} \qquad \text{and} \qquad G(;x) = 1,$$

(1.2)

with the integration path being a straight line from 0 to $x$. This function generalizes a large number of other functions, such as the logarithm $\log(x)$, the classical polylogarithm $\text{Li}_n(x)$, and the harmonic polylogarithm $H_{\vec{m}}(x)$ [4].

At the one-loop order, all Feynman integrals may be expressed in terms of GPLs, and at two and more loops this remains true for a large class of the Feynman integrals, see e.g. the reviews [5, 6].

The utility of GPLs may also be appreciated from a purely mathematical point of view, since an integral of elemental functions, such as

$$I = \int_0^x \frac{\log(t)\log(1-t)\log(1-t/a)}{t-b} dt$$

(1.3)

has no result expressible in terms of more standard functions, but if results in terms of GPLs are allowed for, the result may be expressed as

$$\begin{aligned} I =\ & G(b,0,1,a;x) + G(b,0,a,1;x) + G(b,1,0,a;x) \\ & + G(b,1,a,0;x) + G(b,a,0,1;x) + G(b,a,1,0;x). \end{aligned}$$

(1.4)

These considerations should motivate the further investigation of GPLs.

Many relations among GPLs are known. Primary are the rescaling relation

$$G(a_1,\ldots,a_n;x) = G(za_1,\ldots,za_n;zx) \quad \text{with} \quad a_n,z \neq 0$$

(1.5)

and the shuffle relation

$$G(a_1,\ldots,a_m;x)G(b_1,\ldots,b_n;x) = \sum_{c\in a\amalg b} G(c_1,\ldots,c_{m+n};x),$$

(1.6)

where $a\amalg b$ denotes the "shuffle" of the lists $a$ and $b$, see e.g. [7, 8].

Many additional relations exist, such as the equivalence to the multiple polylogarithms which are defined as iterated sums, the "stuffle relation" between the multiple polylogarithms, various inversion and duplication relations, and many more [9, 10, 3, 7, 8, 5, 11, 1] which are beyond the scope of these proceedings.

The scope of the current project is, however, slightly different. Where the relations mentioned above relate GPLs to each other, our goal is to express them in terms of some minimal set of simpler functions. Such a minimal set was conjectured by Goncharov as described in ref. [7], and

specifically is was conjectured that all GPLs of weight[1] $\leq 4$, which is all that is needed at the level of two-loop Feynman integrals, can be expressed in terms of the functions

$$\log(x), \; \text{Li}_2(x), \; \text{Li}_3(x), \; \text{Li}_4(x), \; \text{Li}_{2,2}(x,y). \tag{1.7}$$

Here the $\text{Li}_n(x)$ denote the classical polylogarithms, which are defined recursively as

$$\text{Li}_n(x) = \int_0^x \frac{dz}{z} \text{Li}_{n-1}(z) \quad \text{with} \quad \text{Li}_1(x) = -\log(1-x), \tag{1.8}$$

and the function $\text{Li}_{2,2}$ may be expressed in a similar way as

$$\text{Li}_{2,2}(x,y) = \int_0^x \frac{\log(z/x)\text{Li}_2(yz)}{z-1} dz. \tag{1.9}$$

It is believed by the authors that most people in the mathematical as well as in the physical communities considered this conjuncture to be true. In any case a lot of expressions for physical quantities such as scattering amplitudes and Wilson loops, expressible in terms of GPLs, have been reexpressed in terms of this minimal set using relatively recently developed methods with names such as "symbols" and co-products [3, 7, 12]. Yet explicit expressions for a general GPL in terms of this minimal basis, valid everywhere in complex phase space, was not known (or at least not written down) and one of the goals of this project was to remedy this, up to weight four.

## 2. Reduction

At weight one the expression for a general GPL is easily seen from eq. (1.1) to be

$$G(a,x) = \log(1-x/a). \tag{2.1}$$

We will here show the derivation in full detail for the corresponding expression at weight two, where we assume that $a$, $b$, and $x$ are all different and non-zero. (If that is not the case, the derivation and the result are similar but simpler).

$$G(a,b,x) = G(a/b,1,x/b) \equiv G(1-\alpha,1,1-\chi) = \int_0^{1-\chi} \frac{G(1,z)}{z-(1-\alpha)} dz \tag{2.2}$$

$$= \int_0^{1-\chi} \frac{G(0,1-z)}{z-(1-\alpha)} dz = \int_1^{\chi} \frac{G(0,y)}{y-\alpha} dy \tag{2.3}$$

$$= G(\alpha,0;\chi) - G(\alpha,0;1) + 2\pi i G(0,\alpha)\,\text{sgn}(\alpha)T(1,\chi,\alpha) \tag{2.4}$$

$$= G(\alpha,\chi)G(0;\chi) - G(0,\alpha;\chi) - G(\alpha;1)G(0;1)$$
$$+ G(0,\alpha;1) + 2\pi i G(0;\alpha)\,\text{sgn}(\alpha)T(1,\chi,\alpha) \tag{2.5}$$

$$= \log(1-\chi/\alpha)\log(\chi) + \text{Li}_2(\chi/\alpha) - \text{Li}_2(1/\alpha) + 2\pi i \log(\alpha)\text{sgn}(\alpha)T(1,\chi,\alpha) \tag{2.6}$$

where the last step used the relation $G(0,a,x) = -\text{Li}_2(x/a)$. The function $T(x,y,z)$ is defined to be one if the complex $z$ is inside the triangle in the complex plane formed by $0$, $x$, and $y$, and zero otherwise, see fig. 1.

---

[1]The weight of a GPL is defined as the number of iterated integrals, so e.g. the GPL given by eq. (1.1) has weight $n$.
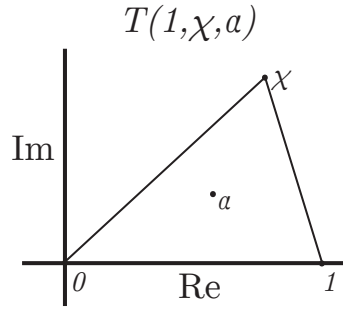
**Figure 1:** This figure shows the triangle referred to by the function $T(1,\chi,\alpha)$. The function evaluates to one when $\alpha$ is inside the triangle, and to zero otherwise.

At weights three and four the derivation is similar. The first step, i.e. the mapping of e.g. $G(a,b,c,d;x)$ to $G(\alpha,\beta,\gamma,0;\chi)$ can be done recursively as above, see refs. [10, 1]. The remaining step unto the minimal basis gets progressively more difficult for each weight, requiring the systematic use of all shuffle and stuffle relations, in addition to a number of non-trivial integral relations. The result at weight three, for the general $G(a,b,c;x)$, takes up about one page, and the corresponding expression for $G(a,b,c,d;x)$ at weight four, would take up about 20 pages. For the specific expressions and more details of the derivations, see ref. [1] and its ancillary files.

The mentioned reductions up to weight four, are implemented as a Mathematica replacement rule called `gtolrules`, which is included as an ancillary file to ref. [1].

## 3. Evaluation

Having reduced the GPLs to $\text{Li}_n$ and $\text{Li}_{2,2}$, it becomes desirable to have a quick algorithm for the evaluation of those functions. The implementation of such an algorithm was the second goal of this project. The classical polylogarithm $\text{Li}_n$ is fairly well studied, and fast methods for its evaluation are described in e.g. refs. [13, 10], and in ref. [1] we described a combined and optimized version of those methods. Yet here we will describe a more primitive evaluation method, as that is the one that generalizes to the case of $\text{Li}_{2,2}$.

$\text{Li}_n$ may be expressed by the sum

$$\text{Li}_n(x) = \sum_{i=1}^{\infty} \frac{x^i}{i^n} \tag{3.1}$$

which converges whenever $|x| \leq 1$. In the opposite case one might use the inversion relation for $\text{Li}_n$

$$\text{Li}_n(x) = (-1)^{n-1}\text{Li}_n\left(\tfrac{1}{x}\right) - \tfrac{1}{n!}\log^n(-x) + 2\sum_{r=1}^{\lfloor \frac{n}{2} \rfloor} \frac{\log^{n-2r}(-x)}{(n-2r)!}\left(2^{1-2r}-1\right)\zeta(2r) \tag{3.2}$$

to map to an $\text{Li}_n$ function with the argument inside the convergent region. An thus we are in principle able to evaluate $\text{Li}_n(x)$ for any argument.

$\text{Li}_{2,2}$ is given as a similar sum

$$\text{Li}_{2,2}(x,y) = \sum_{i>j>0}^{\infty} \frac{x^i y^j}{i^2 j^2} \tag{3.3}$$

which converges whenever $|x| \leq 1$ and $|xy| \leq 1$. For cases where $|xy| > 1$ we may use the inversion relation

$$\mathrm{Li}_{2,2}(x,y) = \mathrm{Li}_{2,2}\left(\tfrac{1}{x},\tfrac{1}{y}\right) - \mathrm{Li}_4(xy) + 3\left(\mathrm{Li}_4\left(\tfrac{1}{x}\right) + \mathrm{Li}_4(y)\right) + 2\left(\mathrm{Li}_3\left(\tfrac{1}{x}\right) - \mathrm{Li}_3(y)\right)\log(-xy)$$

$$+ \mathrm{Li}_2\left(\tfrac{1}{x}\right)\left(\frac{\pi^2}{6} + \frac{\log^2(-xy)}{2}\right) + \frac{1}{2}\mathrm{Li}_2(y)\left(\log^2(-xy) - \log^2(-x)\right), \tag{3.4}$$

and when $|x| > 1$ we may use the stuffle-relation

$$\mathrm{Li}_{2,2}(x,y) = -\mathrm{Li}_{2,2}(y,x) - \mathrm{Li}_4(xy) + \mathrm{Li}_2(x)\mathrm{Li}_2(y), \tag{3.5}$$

and together these relations allow for the mapping of $\mathrm{Li}_{2,2}(x,y)$ of any set of arguments into the convergent region, see fig. 2.
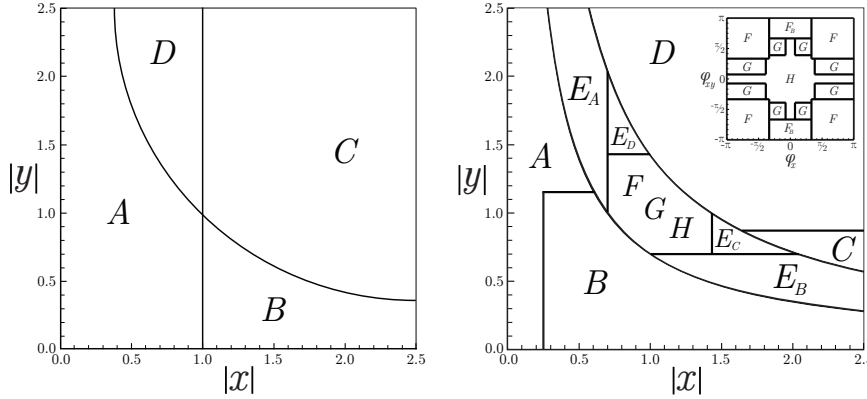


**Figure 2:** This figure describes the regions into which the evaluation of $\mathrm{Li}_{2,2}(x,y)$ is split. The left figure shows the regions as they are described in the main text, while the right figure shows the regions used by the actual implementation in ref. [1]. 'A' denotes the region of convergence of the sum of eq. (3.3), 'B' the region requiring the use of the stuffle-relation (3.5), 'C' the region of the inversion relation (3.4), and 'D' a region requiring the use of both stuffle and inversion. 'E', 'F', 'G', and 'H', refer to regions where different algorithms were used, as described in ref. [1].

It is easily realized that the sum of eq. (3.3) is extremely slowly converging close to $|xy| = 1$. For that reason the authors developed a number of alternative algorithms for use in that region. A description of the exact nature of these algorithms is beyond the scope of these proceedings, see ref. [1] for a more thorough explanation.

A C++ implementation denoted `lievaluate` of $\mathrm{Li}_n$ and $\mathrm{Li}_{2,2}$ using the algorithms mentioned above, was included as ancillary files to ref. [1] alongside code enabling the linking of `lievaluate` from `Mathematica`. The authors note that since the publication of ref. [1], an alternative C++ implementation of these algorithms have been published as ref. [14].

## 4. Discussion

The reductions described in section 2 made no use of the previously mentioned "symbols" or coproducts which have been driving a lot of the recent developments in the study and use of GPLs.

This was a conscious decision on the side of the authors, as the symbols catch the algebraic but not the analytic parts of the relations they describe. As the goal was to derive relations valid everywhere in the complex phase space, irrespectively of the branch-cuts of the GPL and Li functions, a symbol-based approach was considered unsuited. Yet there are developments of the use of the coproduct which takes steps towards capturing also the analytic properties of the functions [15], and if one wanted to find the explicit reductions of GPLs at higher weights (such as the 5 and 6 which are needed for three-loop computations), perhaps such an approach would be desirable or even necessary.

## Acknowledgments

## References

[1] H. Frellesvig, D. Tommasini, and C. Wever, "On the reduction of generalized polylogarithms to $\mathrm{Li}_n$ and $\mathrm{Li}_{2,2}$ and on the evaluation thereof," *JHEP*, vol. 03, p. 189, 2016.

[2] A. B. Goncharov, "Multiple polylogarithms, cyclotomy and modular complexes," *Math Res. Letters*, vol. 5, pp. 497–516, July 1998.

[3] A. B. Goncharov, M. Spradlin, C. Vergu, and A. Volovich, "Classical Polylogarithms for Amplitudes and Wilson Loops," *Phys. Rev. Lett.*, vol. 105, p. 151605, 2010.

[4] E. Remiddi and J. A. M. Vermaseren, "Harmonic polylogarithms," *Int. J. Mod. Phys.*, vol. A15, pp. 725–754, 2000.

[5] C. Duhr, "Mathematical aspects of scattering amplitudes," in *Theoretical Advanced Study Institute in Elementary Particle Physics: Journeys Through the Precision Frontier: Amplitudes for Colliders (TASI 2014) Boulder, Colorado, June 2-27, 2014*, 2014.

[6] J. M. Henn, "Lectures on differential equations for Feynman integrals," *J. Phys.*, vol. A48, p. 153001, 2015.

[7] C. Duhr, H. Gangl, and J. R. Rhodes, "From polygons and symbols to polylogarithmic functions," *Journal of High Energy Physics*, vol. 10, p. 75, Oct. 2012.

[8] J. Ablinger, J. Blümlein, and C. Schneider, "Analytic and Algorithmic Aspects of Generalized Harmonic Sums and Polylogarithms," *J. Math. Phys.*, vol. 54, p. 082301, 2013.

[9] D. J. B. Jonathan M. Borwein, David M. Bradley and P. Lisonek, "Special Values of Multiple Polylogarithms," *Trans. of the Am. Math. Soc.*, vol. 353, n. 3, pp. 907–941, 2001.

[10] J. Vollinga and S. Weinzierl, "Numerical evaluation of multiple polylogarithms," *Comput. Phys. Commun.*, vol. 167, p. 177, 2005.

[11] E. Panzer, "Algorithms for the symbolic integration of hyperlogarithms with applications to Feynman integrals," *Comput. Phys. Commun.*, vol. 188, pp. 148–166, 2014.

[12] C. Duhr, "Hopf algebras, coproducts and symbols: an application to Higgs boson amplitudes," *JHEP*, vol. 08, p. 043, 2012.

[13] R. E. Crandall, "Note on fast polylogarithm computation," 2006.

[14] S. Kirchner, "LiSK - A C++ Library for Evaluating Classical Polylogarithms and $Li_{22}$," 2016.

[15] S. Abreu, R. Britto, C. Duhr, and E. Gardi, "From multiple unitarity cuts to the coproduct of Feynman integrals," *JHEP*, vol. 10, p. 125, 2014.

PoS(LL2016)040