Functional equations for multiple polylogarithms

Herbert Gangl

Durham University

Amplitudes 2013, 30.4.13
Outline

• functional equations for classical polylogs
• relate multiple polylogs to classical ones, in low weight
• new phenomena in weight 4
• relating MPL's of different depths
• [if time] glimpse of a mathematical application
Outline

• functional equations for classical polylogs
Outline

• functional equations for classical polylogs
• relate multiple polylogs to classical ones, in low weight
Outline

• functional equations for classical polylogs
• relate multiple polylogs to classical ones, in low weight
• new phenomena in weight 4
Outline

• functional equations for classical polylogs
• relate multiple polylogs to classical ones, in low weight
• new phenomena in weight 4
• relating MPL’s of different depths

[if time] glimpse of a mathematical application
Outline

- functional equations for classical polylogs
- relate multiple polylogs to classical ones, in low weight
- new phenomena in weight 4
- relating MPL’s of different depths
- [if time] glimpse of a mathematical application
The classical case

For classical polylogarithms, (non-trivial) functional equations are known for small "weight" (= "transcendentality" for physicists):

- Weight 1: one has the 1-logarithm function
  \[
  \sum_{k \geq 1} z^k = -\log(1 - z) = \text{Li}_1(z).
  \]

The standard functional equation for \( \log(ab) = \log(a) + \log(b) \) (1) translates into an equation for \( \text{Li}_1(z) \):

\[
\text{Li}_1(x) + \text{Li}_1(y) - \text{Li}_1(x + y - xy) = 0.
\]

Typical here: arguments are rational functions in several variables, coefficients are integers. Hence for weight 1 can "combine" terms, by virtue of (1), thus successively reducing number of terms in a given expression.
The classical case

For **classical polylogarithms**, (non-trivial) functional equations are known for small “weight” (=“transcendentality” for physicists)

\[
\sum_{k \geq 1} z^k = -\log(1 - z) = \text{Li}_1(z).
\]
The classical case

For **classical polylogarithms**, (non-trivial) functional equations are known for small “weight” (="transcendental" for physicists)

- Weight 1: one has the *1-logarithm* function

\[
\sum_{k \geq 1} \frac{z^k}{k} = -\log(1-z) = \text{Li}_1(z).
\]

The standard functional equation for \( \log(ab) = \log(a) + \log(b) \) (1) translates into an equation for \( \text{Li}_1(z) \):

\[
\text{Li}_1(x) + \text{Li}_1(y) - \text{Li}_1(x+y-xy) = 0.
\]

Typical here: arguments are rational functions in several variables, coefficients are integers. Hence for weight 1 can “combine” terms, by virtue of (1), thus successively reducing number of terms in a given expression.
The classical case

For **classical polylogarithms**, (non-trivial) functional equations are known for small “weight” (=“transcendentality” for physicists)

- Weight 1: one has the *1-logarithm* function

\[
\sum_{k \geq 1} \frac{z^k}{k}
\]
The classical case

For **classical polylogarithms**, (non-trivial) functional equations are known for small “weight” (= “transcendentality” for physicists)

- **Weight 1**: one has the **1-logarithm** function

\[
\sum_{k \geq 1} \frac{z^k}{k} = -\log(1 - z)
\]
The classical case

For **classical polylogarithms**, (non-trivial) functional equations are known for small “weight” ("transcendentiality" for physicists)

- Weight 1: one has the 1-logarithm function

\[ \sum_{k \geq 1} \frac{z^k}{k} = -\log(1 - z) = Li_1(z). \]
The classical case

For classical polylogarithms, (non-trivial) functional equations are known for small “weight” (=“transcendentality” for physicists)

• Weight 1: one has the 1-logarithm function

\[ \sum_{k \geq 1} \frac{z^k}{k} = - \log(1 - z) = \text{Li}_1(z). \]

The standard functional equation for log
The classical case

For classical polylogarithms, (non-trivial) functional equations are known for small “weight” (= “transcendentality” for physicists)

- Weight 1: one has the 1-logarithm function

\[ \sum_{k \geq 1} \frac{z^k}{k} = -\log(1 - z) = Li_1(z). \]

The standard functional equation for log

\[ \log(ab) = \log(a) + \log(b) \quad (1) \]
The classical case

For classical polylogarithms, (non-trivial) functional equations are known for small “weight” (=“transcendentality” for physicists)

- Weight 1: one has the 1-logarithm function

\[ \sum_{k \geq 1} \frac{z^k}{k} = -\log(1 - z) = Li_1(z). \]

The standard functional equation for log

\[ \log(ab) = \log(a) + \log(b) \] (1)

distributes into an equation for \( Li_1(z) \):

\[ Li_1(x) + Li_1(y) - Li_1(x + y - xy) = 0. \]
The classical case

For classical polylogarithms, (non-trivial) functional equations are known for small “weight” (=“transcendentality” for physicists)

- Weight 1: one has the 1-logarithm function

\[
\sum_{k \geq 1} \frac{z^k}{k} = -\log(1 - z) = \text{Li}_1(z).
\]

The standard functional equation for log

\[
\log(ab) = \log(a) + \log(b)
\]

translates into an equation for \(\text{Li}_1(z)\):

\[
\text{Li}_1(x) + \text{Li}_1(y) - \text{Li}_1(x + y - xy) = 0.
\]

Typical here: arguments are rational functions in several variables, coefficients are integers.
The classical case

For classical polylogarithms, (non-trivial) functional equations are known for small “weight” (= “transcendentality” for physicists)

- Weight 1: one has the 1-logarithm function

\[
\sum_{k \geq 1} \frac{z^k}{k} = - \log(1 - z) = Li_1(z).
\]

The standard functional equation for log

\[
\log(ab) = \log(a) + \log(b)
\]

translates into an equation for \(Li_1(z)\):

\[
Li_1(x) + Li_1(y) - Li_1(x + y - xy) = 0.
\]

Typical here: arguments are rational functions in several variables, coefficients are integers.

Hence for weight 1 can “combine” terms, by virtue of (1), thus successively reducing number of terms in a given expression.
• Weight 2: the dilogarithm $Li_2(z)$,
• Weight 2: the *dilogarithm* $Li_2(z)$, or rather its single-valued variant $L_2(z) = (Li_2(z) - \log |z| Li_1(z))$, the *Bloch-Wigner dilog*,
• Weight 2: the dilogarithm $Li_2(z)$, or rather its single-valued variant $L_2(z) = \mathcal{S}(Li_2(z) - \log |z| Li_1(z))$, the Bloch-Wigner dilog, satisfies the famous **five term relation** $L_2(V(x, y)) = 0$, where

$$V(x, y) = [x] + [y] + \left[ \frac{1 - x}{1 - xy} \right] + [1 - xy] + \left[ \frac{1 - y}{1 - xy} \right], \quad (2)$$
• Weight 2: the *dilogarithm* $\text{Li}_2(z)$, or rather its single-valued variant $\mathcal{L}_2(z) = \mathcal{S}(\text{Li}_2(z) - \log |z| \text{Li}_1(z))$, the *Bloch-Wigner dilog*, satisfies the famous **five term relation** $\mathcal{L}_2(V(x, y)) = 0$, where

$$V(x, y) = [x] + [y] + \left[ \frac{1 - x}{1 - xy} \right] + [1 - xy] + \left[ \frac{1 - y}{1 - xy} \right], \quad (2)$$

better memorised by writing the arguments as suitable cross-ratios.
• Weight 2: the dilogarithm $Li_2(z)$, or rather its single-valued variant $L_2(z) = \Im (Li_2(z) - \log |z| Li_1(z))$, the Bloch-Wigner dilog, satisfies the famous five term relation $L_2(V(x, y)) = 0$, where

\[
V(x, y) = [x] + [y] + \left[ \frac{1 - x}{1 - xy} \right] + [1 - xy] + \left[ \frac{1 - y}{1 - xy} \right], \quad (2)
\]

better memorised by writing the arguments as suitable cross-ratios (take five points on projective line, drop one each and form the cross-ratio of the remaining four).
• Weight 2: the dilogarithm \( Li_2(z) \), or rather its single-valued variant \( \mathcal{L}_2(z) = \Re \left( Li_2(z) - \log |z| Li_1(z) \right) \), the Bloch-Wigner dilog, satisfies the famous \textbf{five term relation} \( \mathcal{L}_2(V(x, y)) = 0 \), where

\[
V(x, y) = [x] + [y] + \left[ \frac{1-x}{1-xy} \right] + [1-xy] + \left[ \frac{1-y}{1-xy} \right], \tag{2}
\]

better memorised by writing the arguments as suitable cross-ratios (take five points on projective line, drop one each and form the cross-ratio of the remaining four).

No obvious way of “combining” terms to a simpler expression as for weight 1 case.
• Weight 2: the dilogarithm $Li_2(z)$, or rather its single-valued variant $L_2(z) = \mathcal{S}(Li_2(z) - \log |z| Li_1(z))$, the Bloch-Wigner dilog, satisfies the famous **five term relation** $L_2(V(x, y)) = 0$, where

$$V(x, y) = [x] + [y] + \left[ \frac{1 - x}{1 - xy} \right] + [1 - xy] + \left[ \frac{1 - y}{1 - xy} \right], \quad (2)$$

better memorised by writing the arguments as suitable cross-ratios (take five points on projective line, drop one each and form the cross-ratio of the remaining four).

No obvious way of “combining” terms to a simpler expression as for weight 1 case.

There are also many **other** (less well-known) **equations**:
• Weight 2: the dilogarithm $Li_2(z)$, or rather its single-valued variant $L_2(z) = \Re(Li_2(z) - \log |z| Li_1(z))$, the Bloch-Wigner dilog, satisfies the famous **five term relation** $L_2(V(x, y)) = 0$, where

$$V(x, y) = [x] + [y] + \left[ \frac{1 - x}{1 - xy} \right] + [1 - xy] + \left[ \frac{1 - y}{1 - xy} \right], \quad (2)$$

better memorised by writing the arguments as suitable cross-ratios (take five points on projective line, drop one each and form the cross-ratio of the remaining four).

No obvious way of “combining” terms to a simpler expression as for weight 1 case.

There are also many **other** (less well-known) **equations**: e.g., given integers $a, b > 0$, take solutions $\{p_j^{a,b}(t)\}_j$ (algebraic fcts in $t$) of

$$x^a(1 - x)^b = t.$$
• Weight 2: the dilogarithm $Li_2(z)$, or rather its single-valued variant $\mathcal{L}_2(z) = \mathcal{S}(Li_2(z) - \log |z| Li_1(z))$, the Bloch-Wigner dilog, satisfies the famous **five term relation** $\mathcal{L}_2(V(x, y)) = 0$, where

$$V(x, y) = [x] + [y] + \left[ \frac{1 - x}{1 - xy} \right] + [1 - xy] + \left[ \frac{1 - y}{1 - xy} \right], \quad (2)$$

better memorised by writing the arguments as suitable cross-ratios (take five points on projective line, drop one each and form the cross-ratio of the remaining four).

No obvious way of “combining” terms to a simpler expression as for weight 1 case.

There are also many **other** (less well-known) **equations**: e.g., given integers $a, b > 0$, take solutions $\{p_j^{a,b}(t)\}_j$ (algebraic fcts in $t$) of

$$x^a(1 - x)^b = t.$$  

Then

$$\sum_j \mathcal{L}_2(p_j^{a,b}(t)) = 0.$$
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1-p_j) \right) = 0,
\]
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0,
\]

\( \mathcal{L}_n \) a single–valued version of \( \text{Li}_n \)

(\( \mathcal{L}_n \) a single–valued version of \( \text{Li}_n \))
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0 ,
\]

(\( \mathcal{L}_n \) a single–valued version of \( Li_n \)) and also for weight 4

\[
\sum_j \left( a(a+b) \mathcal{L}_4(p_j) - b(a+b) \mathcal{L}_4(1 - p_j) - ab \mathcal{L}_4(1 - \frac{1}{p_j}) \right) = 0 .
\]
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0,
\]

(\( \mathcal{L}_n \) a single–valued version of \( Li_n \)) and also for weight 4

\[
\sum_j \left( a(a + b) \mathcal{L}_4(p_j) - b(a + b) \mathcal{L}_4(1 - p_j) - ab \mathcal{L}_4(1 - \frac{1}{p_j}) \right) = 0.
\]

Our knowledge on fctl eqs is still quite limited, though: first non-trivial equations (mostly in two variables) were found for

• weight 3: Spence (1809), Kummer (1840);
• weight 4, 5: Kummer (1840);
• weight 6, 7: G. (1990, 1992);
• weight \( > 7 \): still unknown.

Significance: explicit presentation of algebraic K-groups. For \( n > 3 \), the currently known equations are considered insufficient for such a presentation. Motivates to look at...
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0,
\]

(\( \mathcal{L}_n \) a single–valued version of \( Li_n \)) and also for weight 4

\[
\sum_j \left( a(a + b) \mathcal{L}_4(p_j) - b(a + b) \mathcal{L}_4(1 - p_j) - ab \mathcal{L}_4(1 - \frac{1}{p_j}) \right) = 0.
\]

Our knowledge on fctl eqs is still quite limited, though: first non-trivial equations (mostly in two variables) were found for

- weight 3: Spence (1809), Kummer (1840);
- weight 4, 5: Kummer (1840)
- weight 6, 7: G. (1990, 1992)
- weight \( \geq 7 \): still unknown.

Significance: explicit presentation of algebraic K-groups. For \( n \geq 3 \), the currently known equations are considered insufficient for such a presentation. Motivates to look at...
In fact, these sets \( \{ p_j = p_{j}^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0,
\]

\( (\mathcal{L}_n \), a single–valued version of \( Li_n \)) and also for weight 4

\[
\sum_j \left( a(a + b) \mathcal{L}_4(p_j) - b(a + b) \mathcal{L}_4(1 - p_j) - ab \mathcal{L}_4(1 - \frac{1}{p_j}) \right) = 0.
\]

Our knowledge on fctl eqs is still quite limited, though: first non-trivial equations (mostly in two variables) were found for
- weight 3: Spence (1809), Kummer (1840);
- weight 4, 5: Kummer (1840)
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0,
\]

(\( \mathcal{L}_n \) a single–valued version of \( Li_n \)) and also for weight 4

\[
\sum_j \left( a(a + b) \mathcal{L}_4(p_j) - b(a + b) \mathcal{L}_4(1 - p_j) - ab \mathcal{L}_4(1 - \frac{1}{p_j}) \right) = 0.
\]

Our knowledge on fctl eqs is still quite limited, though: first non-trivial equations (mostly in two variables) were found for

- weight 3: Spence (1809), Kummer (1840);
- weight 4, 5: Kummer (1840)
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0,
\]

(\( \mathcal{L}_n \) a single–valued version of \( Li_n \)) and also for weight 4

\[
\sum_j \left( a(a + b) \mathcal{L}_4(p_j) - b(a + b) \mathcal{L}_4(1 - p_j) - ab \mathcal{L}_4(1 - \frac{1}{p_j}) \right) = 0.
\]

Our knowledge on fctl eqs is still quite limited, though: first non-trivial equations (mostly in two variables) were found for

- weight 3: Spence (1809), Kummer (1840);
- weight 4, 5: Kummer (1840)
- weight > 7: still unknown.
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0,
\]

(\( \mathcal{L}_n \) a single–valued version of \( Li_n \)) and also for weight 4

\[
\sum_j \left( a(a + b) \mathcal{L}_4(p_j) - b(a + b) \mathcal{L}_4(1 - p_j) - ab \mathcal{L}_4\left(1 - \frac{1}{p_j}\right) \right) = 0.
\]

Our knowledge on fctl eqs is still quite limited, though: first non-trivial equations (mostly in two variables) were found for

- weight 3: Spence (1809), Kummer (1840);
- weight 4, 5: Kummer (1840)
- weight \( \geq 7 \): still unknown.

Significance: explicit presentation of algebraic K-groups.
In fact, these sets \( \{ p_j = p_j^{a,b}(t) \} \) also suffice to produce (as \( a, b \) vary, infinitely many different) equations for weight 3, like

\[
\sum_j \left( a \mathcal{L}_3(p_j) - b \mathcal{L}_3(1 - p_j) \right) = 0,
\]

\( (\mathcal{L}_n \text{ a single–valued version of } Li_n) \) and also for weight 4

\[
\sum_j \left( a(a + b) \mathcal{L}_4(p_j) - b(a + b) \mathcal{L}_4(1 - p_j) - ab \mathcal{L}_4 \left( 1 - \frac{1}{p_j} \right) \right) = 0.
\]

Our knowledge on fctl eqs is still quite limited, though: first non-trivial equations (mostly in two variables) were found for

- weight 3: Spence (1809), Kummer (1840);
- weight 4, 5: Kummer (1840)
- weight > 7: still unknown.

Significance: explicit presentation of algebraic K-groups.

For \( n > 3 \), the currently known equations are considered insufficient for such a presentation. Motivates to look at...
The multiple case

Multiple polylogarithms (MPL’s, Goncharov polylogs) (Lappo-Danilevsky, Chen, Goncharov).
The multiple case

Multiple polylogarithms (MPL’s, Goncharov polylogs) (Lappo-Danilevsky, Chen, Goncharov).

Classical polylogs are somehow too special—it is important to consider multivariable variants.
The multiple case

**Multiple polylogarithms (MPL’s, Goncharov polylogs)** (Lappo-Danilevsky, Chen, Goncharov).

Classical polylogs are somehow too special—it is important to consider multivariable variants. Altogether they should be part of an overarching coLie algebra structure (Beilinson, Goncharov).

Examples: Non-classical polylogs in depth 2 (for $p, q > 0$) are $Li_{p, q}(x, y) = \sum_{0 < m < n} x^m y^n$.

Combined with the classical ones they also satisfy fctl eqns, e.g. since $\sum_{0 < m < n} x^m y^n + \sum_{0 < n < m} x^m y^n = \sum_{m, n}$ we get $Li_{p, q}(x, y) + Li_{q, p}(y, x) + Li_{p+q}(xy) = Li_p(x) Li_q(y)$.

• Nothing new for weights 2 and 3: MPL’s of weight $\leq 3$ are well-known to be expressible in terms of the classical ones, e.g. $Li_{1, 1}(x, y) = Li_2(1-x-1-y-1) - Li_2(1-y-1) - Li_2(xy)$.
The multiple case

**Multiple polylogarithms (MPL’s, Goncharov polylogs)** (Lappo-Danilevsky, Chen, Goncharov).

Classical polylogs are somehow too special—it is important to consider multivariable variants. Altogether they should be part of an overarching coLie algebra structure (Beilinson, Goncharov).

*Examples:* Non-classical polylogs in *depth* 2 (for $p, q > 0$) are
The multiple case

**Multiple polylogarithms (MPL’s, Goncharov polylogs)**
(Lappo-Danilevsky, Chen, Goncharov).
Classical polylogs are somehow too special—it is important to consider multivariable variants. Altogether they should be part of an overarching coLie algebra structure (Beilinson, Goncharov).

*Examples:* Non-classical polylogs in *depth* 2 (for \( p, q > 0 \)) are

\[
Li_{p,q}(x, y) = \sum_{0<m<n} \frac{x^m y^n}{m^p n^q},
\]

Combined with the classical ones they also satisfy fctl eqns, e.g.

since \( \sum_{0<m<n} + \sum_{0<n<m} + \sum_{n=m} = \sum_{n,m} \) we get

\[
Li_{p,q}(x, y) + Li_{q,p}(y, x) + Li_{p+q}(xy) = Li_{p}(x)Li_{q}(y).
\]
The multiple case

Multiple polylogarithms (MPL’s, Goncharov polylogs) (Lappo-Danilevsky, Chen, Goncharov).

Classical polylogs are somehow too special—it is important to consider multivariable variants. Altogether they should be part of an overarching coLie algebra structure (Beilinson, Goncharov).

**Examples:** Non-classical polylogs in depth 2 (for \( p, q > 0 \)) are

\[
Li_{p,q}(x,y) = \sum_{0 < m < n} \frac{x^m y^n}{m^p n^q},
\]

Combined with the classical ones they also satisfy fctl eqns, e.g. since
\[
\sum_{0 < m < n} + \sum_{0 < n < m} + \sum_{n = m} = \sum_{n,m}
\]
we get

\[
Li_{p,q}(x,y) + Li_{q,p}(y,x) + Li_{p+q}(xy) = Li_{p}(x)Li_{q}(y).
\]

• **Nothing new for weights 2 and 3:** MPL’s of weight \( \leq 3 \) are well-known to be **expressible** in terms of the classical ones, e.g.
The multiple case

Multiple polylogarithms (MPL’s, Goncharov polylogs) (Lappo-Danilevsky, Chen, Goncharov).

Classical polylogs are somehow too special—it is important to consider multivariable variants. Altogether they should be part of an overarching coLie algebra structure (Beilinson, Goncharov).

Examples: Non-classical polylogs in depth 2 (for \( p, q > 0 \)) are

\[
Li_{p,q}(x, y) = \sum_{0 < m < n} \frac{x^m y^n}{m^p n^q},
\]

Combined with the classical ones they also satisfy fctl eqns, e.g. since \( \sum_{0 < m < n} + \sum_{0 < n < m} + \sum_{n = m} = \sum_{n, m} \) we get

\[
Li_{p,q}(x, y) + Li_{q,p}(y, x) + Li_{p+q}(xy) = Li_p(x) Li_q(y).
\]

• Nothing new for weights 2 and 3: MPL’s of weight \( \leq 3 \) are well-known to be expressible in terms of the classical ones, e.g.

\[
Li_{1,1}(x, y) = Li_2\left(\frac{1 - x}{1 - y^{-1}}\right) - Li_2\left(\frac{1}{1 - y^{-1}}\right) - Li_2(xy),
\]
and there are similar, more complicated expressions of $Li_{2,1}(x,y)$ and $Li_{1,1,1}(x,y,z)$ in terms of $Li_3(z)$ (and products of lower weight polylogarithms).
and there are similar, more complicated expressions of $Li_{2,1}(x, y)$ and $Li_{1,1,1}(x, y, z)$ in terms of $Li_3(z)$ (and products of lower weight polylogarithms).

From **weight 4 on**, things are **different**: $Li_4$ not sufficient...
and there are similar, more complicated expressions of $Li_{2,1}(x, y)$
and $Li_{1,1,1}(x, y, z)$ in terms of $Li_{3}(z)$ (and products of lower weight
polylogarithms).

From **weight 4** on, things are **different**: $Li_{4}$ not sufficient... we need a function of at least two variables if we want to express all weight 4 MPL's:
and there are similar, more complicated expressions of $Li_{2,1}(x, y)$ and $Li_{1,1,1}(x, y, z)$ in terms of $Li_3(z)$ (and products of lower weight polylogarithms).

From **weight 4 on**, things are **different**: $Li_4$ not sufficient... we need a function of at least *two* variables if we want to express all weight 4 MPL’s:

**e.g.** $Li_{2,2}(x, y)$, $Li_{3,1}(x, y)$ or even $Li_{1,1,1,1}(x, y, z, w)$
and there are similar, more complicated expressions of $Li_{2,1}(x, y)$ and $Li_{1,1,1}(x, y, z)$ in terms of $Li_3(z)$ (and products of lower weight polylogarithms).

From **weight 4 on**, things are **different**: $Li_4$ not sufficient... we need a function of at least *two* variables if we want to express all weight 4 MPL's:

- e.g. $Li_{2,2}(x, y)$, $Li_{3,1}(x, y)$ or even $Li_{1,1,1,1}(x, y, z, w)$ ($Li_4$ being a kind of limiting case of these, modulo products).
and there are similar, more complicated expressions of $Li_{2,1}(x, y)$ and $Li_{1,1,1}(x, y, z)$ in terms of $Li_3(z)$ (and products of lower weight polylogarithms).

From **weight 4 on**, things are **different**: $Li_4$ not sufficient... we need a function of at least *two* variables if we want to express all weight 4 MPL’s:

- e.g. $Li_{2,2}(x, y)$, $Li_{3,1}(x, y)$ or even $Li_{1,1,1,1}(x, y, z, w)$ ($Li_4$ being a kind of limiting case of these, modulo products).

It is often convenient to work with a notation corresponding to **iterated integrals** (similar to physicists’ $G$-fct)
and there are similar, more complicated expressions of \( Li_{2,1}(x, y) \) and \( Li_{1,1,1}(x, y, z) \) in terms of \( Li_3(z) \) (and products of lower weight polylogarithms).

From **weight 4 on**, things are **different**: \( Li_4 \) not sufficient... we need a function of at least **two** variables if we want to express all weight 4 MPL's:

- e.g. \( Li_{2,2}(x, y) \), \( Li_{3,1}(x, y) \) or even \( Li_{1,1,1,1}(x, y, z, w) \) (\( Li_4 \) being a kind of limiting case of these, modulo products).

It is often convenient to work with a notation corresponding to **iterated integrals** (similar to physicists’ \( G \)-fct)

\[
l_{m_1,\ldots,m_k}(x_1, \ldots, x_k) = (-1)^k Li_{m_1,\ldots,m_k}(z_1, \ldots, z_k)
\]
and there are similar, more complicated expressions of $Li_{2,1}(x, y)$ and $Li_{1,1,1}(x, y, z)$ in terms of $Li_3(z)$ (and products of lower weight polylogarithms).

From **weight 4 on**, things are different: $Li_4$ not sufficient. . . we need a function of at least two variables if we want to express all weight 4 MPL’s:

- e.g. $Li_{2,2}(x, y)$, $Li_{3,1}(x, y)$ or even $Li_{1,1,1,1}(x, y, z, w)$ ($Li_4$ being a kind of limiting case of these, modulo products).

It is often convenient to work with a notation corresponding to **iterated integrals** (similar to physicists’ $G$-fct)

$$I_{m_1,\ldots,m_k}(x_1, \ldots, x_k) = (-1)^k Li_{m_1,\ldots,m_k}(z_1, \ldots, z_k)$$

where $x_1 = (z_1 \ldots z_k)^{-1}$, $x_2 = (z_2 \ldots z_k)^{-1}$, $\ldots$, $x_k = z_k^{-1}$,

$$= G(\underbrace{0, \ldots, 0}_{m_k-1}, x_k, \ldots, 0, \ldots, 0, x_1; 1).$$
Functional equations in weight 4; case of depth 2 (write $= \mod$ for “mod products”):

$$I_{3,1}(x, y) + I_{3,1}(y, x) = 0,$$

furthermore, if we simultaneously transform both variables under the usual $S_3$-action, we get that modulo $L_i 4$-terms (denoted by $\equiv$)

$$I_{3,1}(x, y) \equiv I_{3,1}(1-x, 1-y) \equiv I_{3,1}(x, 1-y) \equiv \ldots$$

One can be more precise and give the $L_i 4$-expressions explicitly:

for the first congruence, there is actually no such term needed, but for the second one we find

$$I_{3,1}(x, y) - I_{3,1}(1-x, 1-y) = L_i 4 \left( [x] - [y] + 3 [xy] \right),$$

and similar for the remaining ones.

Rewrite equations. It is convenient to introduce cross-ratios $\left( \begin{array}{cccc} a & b & c & d \end{array} \right) = cr(a, b, c, d) = \frac{a-c}{a-d} \cdot \frac{b-d}{b-c}$ in the arguments for the $I_{\cdots}$—to make symmetries more apparent.
Functional equations in weight 4; case of depth 2 (write $= \text{for "mod products"}$):

\[ l_{3,1}(x, y) + l_{3,1}(y, x) = 0, \]

Furthermore, if we simultaneously transform both variables under the usual $S_3$-action, we get that modulo $L_4$-terms (denoted by $\equiv$)

\[ l_{3,1}(x, y) \equiv l_{3,1}(1 - x, 1 - y) \equiv l_{3,1}(1x, 1y) \equiv \ldots \]

One can be more precise and give the $L_4$-expressions explicitly:

for the first congruence, there is actually no such term needed, but for the second one we find

\[ l_{3,1}(x, y) - l_{3,1}(1x, 1y) = L_4\left(\left[ x \right] - \left[ y \right] + 3\left[ xy \right]\right), \]

and similar for the remaining ones.

Rewrite equations. It is convenient to introduce cross-ratios $(abcd) = cr(a, b, c, d) = a - c/a - d \cdot b - d/b - c$ in the arguments for the $l_{\cdot \cdot \cdot}$—to make symmetries more apparent.
Functional equations in weight 4; case of depth 2 (write $=$ for “mod products”):

$$I_{3,1}(x, y) + I_{3,1}(y, x) = 0,$$

furthermore, if we simultaneously transform both variables under the usual $S_3$-action, we get that $modulo$ $Li_4$-terms (denoted by $\equiv$)
Functional equations in weight 4; case of depth 2 (write = for “mod products”):

\[ l_{3,1}(x, y) + l_{3,1}(y, x) = 0, \]

furthermore, if we simultaneously transform both variables under the usual \( S_3 \)-action, we get that \textit{modulo} \( Li_4 \)-terms (denoted by \( \equiv \))

\[ l_{3,1}(x, y) \equiv l_{3,1}(1 - x, 1 - y) \equiv l_{3,1}\left(\frac{1}{x}, \frac{1}{y}\right) \equiv \ldots \]
Functional equations in weight 4; case of depth 2 (write = for “mod products”):

\[ I_{3,1}(x, y) + I_{3,1}(y, x) = 0, \]

furthermore, if we simultaneously transform both variables under the usual \( S_3 \)-action, we get that \textit{modulo} \( Li_4 \)-terms (denoted by \( \equiv \))

\[ I_{3,1}(x, y) \equiv I_{3,1}(1 - x, 1 - y) \equiv I_{3,1}\left(\frac{1}{x}, \frac{1}{y}\right) \equiv \ldots \]

One can be more precise and give the \( Li_4 \)-expressions explicitly:
Functional equations in weight 4; case of depth 2 (write \(=\) for “mod products”):

\[I_{3,1}(x, y) + I_{3,1}(y, x) = 0,\]

furthermore, if we simultaneously transform both variables under the usual \(S_3\)-action, we get that \textit{modulo} \(Li_4\)-terms (denoted by \(\equiv\))

\[I_{3,1}(x, y) \equiv I_{3,1}(1-x, 1-y) \equiv I_{3,1}\left(\frac{1}{x}, \frac{1}{y}\right) \equiv \ldots\]

One can be more precise and give the \(Li_4\)-expressions explicitly: for the first congruence, there is actually no such term needed,
Functional equations in weight 4; case of depth 2 (write = for “mod products”):

\[ I_{3,1}(x, y) + I_{3,1}(y, x) = 0, \]

furthermore, if we simultaneously transform both variables under the usual \( S_3 \)-action, we get that \textit{modulo} \( Li_4 \)-terms (denoted by \( \equiv \))

\[ I_{3,1}(x, y) \equiv I_{3,1}(1 - x, 1 - y) \equiv I_{3,1} \left( \frac{1}{x}, \frac{1}{y} \right) \equiv \ldots \]

One can be more precise and give the \( Li_4 \)-expressions explicitly: for the first congruence, there is actually no such term needed, but for the second one we find

\[ I_{3,1}(x, y) - I_{3,1} \left( \frac{1}{x}, \frac{1}{y} \right) = Li_4 \left( [x] - [y] + 3 \left[ \frac{x}{y} \right] \right), \]

and similar for the remaining ones.
Functional equations in weight 4; case of depth 2 (write $\equiv$ for “mod products”):

$$I_{3,1}(x, y) + I_{3,1}(y, x) = 0,$$

furthermore, if we simultaneously transform both variables under the usual $S_3$-action, we get that modulo $Li_4$-terms (denoted by $\equiv$)

$$I_{3,1}(x, y) \equiv I_{3,1}(1-x, 1-y) \equiv I_{3,1}\left(\frac{1}{x}, \frac{1}{y}\right) \equiv \ldots$$

One can be more precise and give the $Li_4$-expressions explicitly: for the first congruence, there is actually no such term needed, but for the second one we find

$$I_{3,1}(x, y) - I_{3,1}\left(\frac{1}{x}, \frac{1}{y}\right) = Li_4\left([x] - [y] + 3\left[\frac{x}{y}\right]\right),$$

and similar for the remaining ones.

**Rewrite equations.** It is convenient to introduce cross-ratios

$$(abcd) = cr(a, b, c, d) = \frac{a - c}{a - d} \cdot \frac{b - d}{b - c}$$
Functional equations in weight 4; case of depth 2 (write = for “mod products”):

\[ l_{3,1}(x, y) + l_{3,1}(y, x) = 0 , \]

furthermore, if we simultaneously transform both variables under the usual \( S_3 \)-action, we get that modulo \( Li_4 \)-terms (denoted by \( \equiv \))

\[ l_{3,1}(x, y) \equiv l_{3,1}(1 - x, 1 - y) \equiv l_{3,1}\left(\frac{1}{x}, \frac{1}{y}\right) \equiv \ldots \]

One can be more precise and give the \( Li_4 \)-expressions explicitly: for the first congruence, there is actually no such term needed, but for the second one we find

\[ l_{3,1}(x, y) - l_{3,1}\left(\frac{1}{x}, \frac{1}{y}\right) = Li_4([x] - [y] + 3 \left[\frac{x}{y}\right]) , \]

and similar for the remaining ones.

**Rewrite equations.** It is convenient to introduce cross-ratios

\[ (abcd) = cr(a, b, c, d) = \frac{a - c}{a - d} \cdot \frac{b - d}{b - c} \]

in the arguments for the \( l \ldots \)—to make symmetries more apparent.
Abbreviate \((ABCDE)_{31} = l_{31}((abcd), (abce))\), then above reads
• Abbreviate \((abcede)_{31} = l_{31}((abcd), (abce))\), then above reads

\[
(abcede)_{31} \equiv (acbde)_{31} \equiv (bacde)_{31} \equiv -(abced)_{31},
\]
• Abbreviate \((abcde)_{31} = l_{31}((abcd), (abce))\), then above reads

\[
(abcde)_{31} \equiv (acbde)_{31} \equiv (bacde)_{31} \equiv -(abced)_{31},
\]

(symmetric in first three slots, antisymmetric in last two).
Abbreviate \((abcde)_{31} = l_{31}((abcd), (abce))\), then above reads

\[
(abcde)_{31} \equiv (acbde)_{31} \equiv (bacde)_{31} \equiv -(abcded)_{31},
\]
(symmetric in first three slots, antisymmetric in last two).

Similarly, for \((abcde)_{22} = l_{22}(abcd, abce)\) find, mod \(L_i\)-terms,

\[
(abcde)_{22} \equiv (bacde)_{22} \equiv -(abdce)_{22} \equiv -(badce)_{22},
\]
(symmetric in first two slots, antisymmetric in slots 3 and 4).
• Abbreviate \((abcde)_{31} = l_{31}((abcd), (abce))\), then above reads
\[
(abcde)_{31} \equiv (acbde)_{31} \equiv (bacde)_{31} \equiv -(abcd)_{31},
\]
(symmetric in first three slots, antisymmetric in last two).

• Similarly, for \((abcde)_{22} = l_{22}(abcd, abce)\) find, mod \(Li_4\)-terms,
\[
(abcde)_{22} \equiv (bacde)_{22} \equiv -(abdce)_{22} \equiv -(badce)_{22},
\]
(symmetric in first two slots, antisymmetric in slots 3 and 4).

• Particularly nice here (swapping both slots simultaneously):
\[
(abcde)_{22} + (badce)_{22} = (abcd)_{4}.
\]
• Abbreviate $(abcde)_{31} = I_{31}((abcd), (abce))$, then above reads

$$(abcde)_{31} \equiv (acbde)_{31} \equiv (bacde)_{31} \equiv -(abced)_{31},$$

(symmetric in first three slots, antisymmetric in last two).

• Similarly, for $(abcde)_{22} = I_{22}(abcd, abce)$ find, mod $L_i^4$-terms,

$$(abcde)_{22} \equiv (bacde)_{22} \equiv -(abdce)_{22} \equiv -(badce)_{22},$$

(symmetric in first two slots, antisymmetric in slots 3 and 4).

• Particularly nice here (swapping both slots simultaneously):

$$(abcde)_{22} + (badce)_{22} = (abcd)_{4}.$$

• Moreover, get 4-term equations
• Abbreviate \((abcde)_{31} = l_{31}((abcd), (abce))\), then above reads

\[(abcde)_{31} \equiv (acbde)_{31} \equiv (bacde)_{31} \equiv -(abced)_{31},\]

(symmetric in first three slots, antisymmetric in last two).

• Similarly, for \((abcde)_{22} = l_{22}(abcd, abce)\) find, mod \(Li_4\)-terms,

\[(abcde)_{22} \equiv (bacde)_{22} \equiv -(abdce)_{22} \equiv -(badce)_{22},\]

(symmetric in first two slots, antisymmetric in slots 3 and 4).

• Particularly nice here (swapping both slots simultaneously):

\[(abcde)_{22} + (badce)_{22} = (abcd)_4.\]

• Moreover, get 4-term equations (e.g. cyclic symmetry in last four)

\[(e(abcd)^{cyc})_{22} \equiv 0, \quad ((abcd)^{cyc}e)_{31} \equiv 0.\]
• Abbreviate \((abcde)_{31} = I_{31}((abcd), (abce))\), then above reads
\[
(abcde)_{31} \equiv (acbde)_{31} \equiv (bacde)_{31} \equiv -(abced)_{31},
\]
(symmetric in first three slots, antisymmetric in last two).
• Similarly, for \((abcde)_{22} = I_{22}(abcd, abce)\) find, mod \(Li_4\)-terms,
\[
(abcde)_{22} \equiv (bacde)_{22} \equiv -(abdce)_{22} \equiv -(badce)_{22},
\]
(symmetric in first two slots, antisymmetric in slots 3 and 4).
• Particularly nice here (swapping both slots simultaneously):
\[
(abcde)_{22} + (badce)_{22} = (abcd)_{4}.
\]
• Moreover, get 4-term equations (e.g. cyclic symmetry in last four)
\[
(e(abcde)^{cyc})_{22} \equiv 0, \quad ((abcde)^{cyc} e)_{31} \equiv 0.
\]
• Among the 120 terms \((a_{\sigma(1)} \cdots a_{\sigma(5)})_{31}, \sigma \in S_5\), there are only 6
linearly independent ones mod \(Li_4\).
• Abbreviate \((abcde)_{31} = l_{31}(abcd, abce)\), then above reads

\[
(abcde)_{31} \equiv (acbde)_{31} \equiv (bacde)_{31} \equiv -(abcded)_{31},
\]
(symmetric in first three slots, antisymmetric in last two).

• Similarly, for \((abcde)_{22} = l_{22}(abcd, abce)\) find, mod \(Li_{4}\)-terms,

\[
(abcde)_{22} \equiv (bacde)_{22} \equiv -(abdce)_{22} \equiv -(badce)_{22},
\]
(symmetric in first two slots, antisymmetric in slots 3 and 4).

• Particularly nice here (swapping both slots simultaneously):

\[
(abcde)_{22} + (badce)_{22} = (abcd)_{4}.
\]

• Moreover, get 4-term equations (e.g. cyclic symmetry in last four)

\[
(e(abcd)^{cyc})_{22} \equiv 0, \quad ((abcd)^{cyc}e)_{31} \equiv 0.
\]

• Among the 120 terms \((a_{\sigma(1)} \ldots a_{\sigma(5)})_{31}, \sigma \in S_{5}\), there are only 6 linearly independent ones mod \(Li_{4}\). Similarly for \(l_{22}\) instead of \(l_{31}\).
Functional equations in depth 3:
Functional equations in depth 3: Similar convention:

\[(abcdef)_{211} = l_{211}(cr(a, b, c, d), cr(a, b, c, e), cr(a, b, c, f)) \text{ etc.}\]
Functional equations in depth 3: Similar convention: 
\[(abcdef)_{211} = l_{211}(cr(a, b, c, d), cr(a, b, c, e), cr(a, b, c, f))\] etc.

- For \(l_{211}\), get that the alternating sum of six terms

\[
\sum_{\sigma \in S_3} (-1)^{|\sigma|} ((abc)^\sigma def)_{211}
\]

is symmetric, mod \(Li_4\), in last three slots,
Functional equations in depth 3: Similar convention:

\[(abcdef)_{211} = l_{211}(cr(a, b, c, d), cr(a, b, c, e), cr(a, b, c, f))\] etc.

• For \(l_{211}\), get that the alternating sum of six terms

\[
\sum_{\sigma \in S_3} (-1)^{|\sigma|} ((abc)^{\sigma} def)_{211}
\]

is symmetric, mod \(Li_4\), in last three slots, its difference to “fde”, say, being expressed as a sum of six \(Li_4\)–terms (coefficients \(\pm 4\)).
**Functional equations in depth 3:** Similar convention: 

\[ (abcdef)_{211} = l_{211}(cr(a, b, c, d), cr(a, b, c, e), cr(a, b, c, f)) \text{ etc.} \]

- For \( l_{211} \), get that the *alternating* sum of six terms

\[
\sum_{\sigma \in S_3} (-1)^{|\sigma|} ((abc)^\sigma def)_{211}
\]

is symmetric, mod \( Li_4 \), in last three slots, its difference to "fde", say, being expressed as a sum of six \( Li_4 \)-terms (coefficients \( \pm 4 \)).

- Similar expressions for \( l_{121} \) (but not for \( l_{112} \)).
Functional equations in depth 3: Similar convention: 
\[(abcdef)_{211} = l_{211}(cr(a, b, c, d), cr(a, b, c, e), cr(a, b, c, f))\] etc.

- For \(l_{211}\), get that the alternating sum of six terms

\[
\sum_{\sigma \in S_3} (-1)^{|\sigma|} \left( (abc)^\sigma def \right)_{211}
\]

is symmetric, mod \(Li_4\), in last three slots, its difference to “fde”, say, being expressed as a sum of six \(Li_4\)–terms (coefficients \(\pm 4\)).

- Similar expressions for \(l_{121}\) (but not for \(l_{112}\)).

- Can represent \(l_{211}\) in terms of \(l_{31}\):

\[2 (a_1 a_2 a_3 a_4 a_5 a_6)_{211} \equiv \text{sum of 36 terms of form } \pm (a_{i_1} \ldots a_{i_5})_{31}\]

with \(i_k \in \{1, \ldots, 6\}\). (Note coefficients all \(\pm 1\).)
**Functional equations in depth 3:** Similar convention:

\[(abcdef)_{211} = l_{211}(cr(a, b, c, d), cr(a, b, c, e), cr(a, b, c, f))\] etc.

- For \(l_{211}\), get that the *alternating* sum of six terms

  \[
  \sum_{\sigma \in S_3} (-1)^{|\sigma|} ((abc)^\sigma def)_{211}
  \]

  is symmetric, mod \(Li_4\), in last three slots, its difference to “fde”, say, being expressed as a sum of six \(Li_4\)–terms (coefficients \(\pm 4\)).

- Similar expressions for \(l_{121}\) (but not for \(l_{112}\)).

- Can represent \(l_{211}\) in terms of \(l_{31}\):

  \[
  2 (a_1a_2a_3a_4a_5a_6)_{211} \equiv \text{sum of 36 terms of form } \pm (a_{i_1} \ldots a_{i_5})_{31}
  \]

  with \(i_k \in \{1, \ldots, 6\}\). (Note coefficients all \(\pm 1\).)

- Can also represent \(l_{211}\) via 36 \(l_{22}\)-terms, coefficients \(\pm \frac{1}{6}, \pm \frac{1}{2}\).
Functional equations in depth 4: (convention now obvious)

For \( I_{1111} \), the quadruple logarithm, find:

- simple 2-fold symmetries
  \[
  (abcdefg)_{1111} = - (abcgfed)_{1111}
  \]
  \[
  (abcdefg)_{1111} = - (acbdefg)_{1111}
  \]

- functional equations with four terms \((\bigast\bigast)\) (shuffle)
  \[
  (abc)(defg)_{1111} = 0 = (abc)(defg + cyc)_{1111}
  \]

- and several ones with 6 terms, e.g.
  \[
  (a(c(bcd)cyc)efg)_{1111}
  \]
  is symmetric in \( e \) and \( g \).

But all the above are 'too simple' to combine two different depths: they do not involve \( I_{31} \)–terms or \( \text{Li}_4 \)–terms.
Functional equations in depth 4: (convention now obvious)
For $l_{1111}$, the quadruple logarithm, find
• simple 2-fold symmetries

$$(abcdefg)_{1111} = -(abcgfed)_{1111}$$
Functional equations in depth 4: (convention now obvious)

For \( l_{1111} \), the *quadruple logarithm*, find

- simple 2-fold symmetries

\[
(abc\text{defg})_{1111} = -(abc\text{gfde})_{1111}
\]

and

\[
(abc\text{defg})_{1111} = -(acb\text{defg})_{1111},
\]
Functional equations in depth 4: (convention now obvious)
For \( l_{1111} \), the quadruple logarithm, find
• simple 2-fold symmetries

\[
(abcdefg)_{1111} = -(abcgfed)_{1111}
\]
and

\[
(abcdefg)_{1111} = -(acbdefg)_{1111},
\]
• functional equations with four terms (III = shuffle)
Functional equations in depth 4: (convention now obvious)

For $I_{1111}$, the quadruple logarithm, find

- simple 2-fold symmetries

\[
(abcdefg)_{1111} = -(abcgfed)_{1111}
\]

and

\[
(abcdefg)_{1111} = -(acbdefg)_{1111},
\]

- functional equations with four terms ($\bigvee\bigvee = \text{shuffle}$)

\[
\left(abc((def)\bigvee g)\right)_{1111} = 0 = \left(abc(defg)^\text{cyc}\right)_{1111},
\]
Functional equations in depth 4: (convention now obvious)
For $l_{1111}$, the quadruple logarithm, find

- simple 2-fold symmetries
  \[(abcdefg)_{1111} = -(abcgfed)_{1111}\]
  and
  \[(abcdefg)_{1111} = -(acbdefg)_{1111},\]

- functional equations with four terms (III = shuffle)
  \[\left(abc((def)_{III} g)\right)_{1111} = 0 = \left(abc(defg)^{\text{cyc}}\right)_{1111},\]

- and several ones with 6 terms,
Functional equations in depth 4: (convention now obvious)
For \( l_{1111} \), the quadruple logarithm, find

- simple 2-fold symmetries

\[
(abcdefg)_{1111} = -(abcfged)_{1111}
\]

and

\[
(abcdefg)_{1111} = -(acbdefg)_{1111},
\]

- functional equations with four terms (\( \text{III} =\text{shuffle} \))

\[
\left( abc((def)\text{III}g) \right)_{1111} = 0 = \left( abc(defg)_{\text{cyc}} \right)_{1111},
\]

- and several ones with 6 terms, e.g.

\[
(a(bcd)_{\text{cyc}} efg)_{1111}
\]

is symmetric in \( e \) and \( g \).
Functional equations in depth 4: (convention now obvious)
For $l_{1111}$, the quadruple logarithm, find

- simple 2-fold symmetries

$$(abc\, defg)_{1111} = -(abcfged)_{1111}$$

and

$$(abc\, defg)_{1111} = -(acb\, defg)_{1111},$$

- functional equations with four terms (III = shuffle)

$$\left(abc\,(def)\, I\!I\!I\, g)\right)_{1111} = 0 = \left(abc\,(defg)^{\text{cyc}}\right)_{1111},$$

- and several ones with 6 terms, e.g.

$$\left(a(bcd)^{\text{cyc}}\, efg\right)_{1111}$$

is symmetric in $e$ and $g$.

But all the above are ‘too simple’ to combine two different depths: they do not involve $l_{31}$–terms or $Li_4$–terms.
More interesting: eqs with 18 $l_{1111}$–terms, adding up to a combination of $Li_4$–terms; arguments of 3 types, indices mod 3:

$$\left( a_i b_j b_{j+1} b_{j+2} a_{i+1} c a_{i+2} \right),$$

$$\left( b_j b_{j+1} b_{j+2} a_1 c a_2 a_3 \right),$$

$$\left( c b_j b_{j+1} a_3 b_{j+2} a_1 a_2 \right).$$

Most interesting perhaps is the following one:

**Theorem.** The alternating sum over 1, 2, 3, 4 of

$$\left( a_1 a_2 a_3 a_4 b c d \right)$$

is antisymmetric, mod $Li_4$–terms, under exchanging the first entry with the sixth.

Moreover, its antisymmetrisation equals the same alternating sum (over 1, 2, 3, 4) of

$$\left( a_1 a_2 a_3 b \right) d_{i+4} + \left( a_1 a_2 a_3 d \right) a_{i+4}.$$
More interesting: eqs with 18 $l_{1111}$–terms, adding up to a combination of $Li_4$–terms; arguments of 3 types, indices mod 3: $(a_i b_j b_{j+1} b_{j+2} a_{i+1} c a_{i+2})$, $(b_j b_{j+1} b_{j+2} a_1 c a_2 a_3)$, $(c b_j b_{j+1} a_3 b_{j+2} a_1 a_2)$.
More interesting: eqs with 18 $I_{1111}$–terms, adding up to a combination of $Li_4$–terms; arguments of 3 types, indices mod 3: 

\[(a; b; b_{j+1}b_{j+2}a_{i+1}c a_{i+2}), (b; b_{j+1}b_{j+2}a_1ca_2a_3), (c b; b_{j+1}a_3b_{j+2}a_1a_2)\].

Most interesting perhaps is the following one:
More interesting: eqs with 18 \( l_{1111} \)-terms, adding up to a combination of \( Li_4 \)-terms; arguments of 3 types, indices mod 3: 
\[
(a; b_j b_{j+1} b_{j+2} a_{i+1} c a_{i+2}), (b_j b_{j+1} b_{j+2} a_1 c a_2 a_3), (c b_j b_{j+1} a_3 b_{j+2} a_1 a_2).
\]

**Most interesting** perhaps is the following one:

**Theorem.** The alternating sum over 1,2,3,4 of \( (a_1 a_2 a_3 a_4 b c d)_{1111} \) is antisymmetric, mod \( Li_4 \)-terms, under exchanging the first entry with the sixth.
More interesting: eqs with 18 $I_{1111}$–terms, adding up to a combination of $Li_4$–terms; arguments of 3 types, indices mod 3:
\[(a_i b_j b_{j+1} b_{j+2} a_{i+1} c a_{i+2}), (b_j b_{j+1} b_{j+2} a_1 c a_2 a_3), (c b_j b_{j+1} a_3 b_{j+2} a_1 a_2)\].

**Most interesting** perhaps is the following one:

**Theorem.** The alternating sum over 1,2,3,4 of $(a_1 a_2 a_3 a_4 b c d)_{1111}$ is antisymmetric, mod $Li_4$-terms, under exchanging the first entry with the sixth.

Moreover, its antisymmetrisation equals the same alternating sum (over 1,2,3,4) of $(a_1 a_2 a_3 b)_4 + (a_1 a_2 a_3 d)_4$. 
More interesting: eqs with 18 $I_{1111}$–terms, adding up to a combination of $Li_4$–terms; arguments of 3 types, indices mod 3:

\[(a; b_j b_{j+1} b_{j+2} a_{i+1} c a_{i+2}), (b; b_{j+1} b_{j+2} a_1 c a_2 a_3), (c b_j b_{j+1} a_3 b_{j+2} a_1 a_2).\]

**Most interesting** perhaps is the following one:

**Theorem.** *The alternating sum over 1,2,3,4 of $(a_1 a_2 a_3 a_4 b c d)_{1111}$ is antisymmetric, mod $Li_4$–terms, under exchanging the first entry with the sixth.*

Moreover, *its antisymmetrisation equals the same alternating sum (over 1,2,3,4) of $(a_1 a_2 a_3 b)_4 + (a_1 a_2 a_3 d)_4.*

We believe that all these functional equations are new.
More interesting: eqs with 18 $I_{1111}$–terms, adding up to a combination of $Li_4$–terms; arguments of 3 types, indices mod 3: $(a_i b_j b_{j+1} b_{j+2} a_{i+1} c a_{i+2}), (b_j b_{j+1} b_{j+2} a_1 c a_2 a_3), (c b_j b_{j+1} a_3 b_{j+2} a_1 a_2)$.

**Most interesting** perhaps is the following one:

**Theorem.** The alternating sum over 1,2,3,4 of $(a_1 a_2 a_3 a_4 b c d)_{1111}$ is antisymmetric, mod $Li_4$–terms, under exchanging the first entry with the sixth.

Moreover, its antisymmetrisation equals the same alternating sum (over 1,2,3,4) of $(a_1 a_2 a_3 b)_4 + (a_1 a_2 a_3 d)_4$.

We believe that all these functional equations are new. Earlier results by N.Dan (explicitly relating $I_{1111}, I_{31}$ and $Li_4$),
More interesting: eqs with 18 $l_{1111}$–terms, adding up to a combination of $Li_4$–terms; arguments of 3 types, indices mod 3: $(a_i b_j b_{j+1} b_{j+2} a_{i+1} c a_{i+2})$, $(b_j b_{j+1} b_{j+2} a_1 c a_2 a_3)$, $(c b_j b_{j+1} a_3 b_{j+2} a_1 a_2)$.

**Most interesting** perhaps is the following one:

**Theorem.** The alternating sum over 1,2,3,4 of $(a_1 a_2 a_3 a_4 b c d)_{1111}$ is antisymmetric, mod $Li_4$–terms, under exchanging the first entry with the sixth.

Moreover, its antisymmetrisation equals the same alternating sum (over 1,2,3,4) of $(a_1 a_2 a_3 b)_4 + (a_1 a_2 a_3 d)_4$.

We believe that all these functional equations are new.

Earlier results by N.Dan (explicitly relating $l_{1111}$, $l_{31}$ and $Li_4$), by F.Brown (developed rep theory of polylogs, implicit equations)
More interesting: eqs with 18 $I_{1111}$-terms, adding up to a combination of $Li_4$-terms; arguments of 3 types, indices mod 3: 

$\left( a_i b_j b_{j+1} b_{j+2} a_{i+1} c a_{i+2} \right), \left( b_j b_{j+1} b_{j+2} a_1 c a_2 a_3 \right), \left( c b_j b_{j+1} a_3 b_{j+2} a_1 a_2 \right)$.

**Most interesting** perhaps is the following one:

**Theorem.** The alternating sum over 1,2,3,4 of $(a_1 a_2 a_3 a_4 b c d)_{1111}$ is antisymmetric, mod $Li_4$-terms, under exchanging the first entry with the sixth.

Moreover, its antisymmetrisation equals the same alternating sum (over 1,2,3,4) of $(a_1 a_2 a_3 b)_4 + (a_1 a_2 a_3 d)_4$.

We believe that all these functional equations are new.

Earlier results by N.Dan (explicitly relating $I_{1111}$, $I_{31}$ and $Li_4$), by F.Brown (developed rep theory of polylogs, implicit equations) and in Rhodes’s thesis (weight $\leq 5$, not neglecting products).
Check time!
How can one find these equations?

In one word: compute the *symbols* of potential candidate terms and use (multi-)linear algebra;
How can one find these equations?

In one word: compute the *symbols* of potential candidate terms and use (multi-)linear algebra; convenient Mathematica implementation by C. Duhr, established for our previous joint work (JHEP 2012, Duhr, G., Rhodes).
Mathematical Impact

A Conjecture of Goncharov

Method successfully used also to tackle an old (∼1992) question of Goncharov, who reduced Zagier's Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems. The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of $\text{Li}_4$. By a remark of N.Dan, one can take this expression to be $I_{31}(V(x,y), z)$, i.e. a sum of five $I_{31}$-terms, where $V(x,y)$ denotes any variant of the 5-term equation for the dilogarithm.

**Theorem.** There is an $\text{Li}_4$–expression in three variables $x, y, z$, whose "symbol" agrees with the one for $I_{31}(V(x,y), z)$. Moreover, there is an explicit solution consisting of 122 terms.

Goncharov outlined how one should get from such an expression to a functional equation for $\text{Li}_4$. Indeed:

**Theorem.** There is a resulting functional equation for $\text{Li}_4$ in four variables consisting of . . . 931 terms.
Mathematical Impact

A Conjecture of Goncharov

Method successfully used also to tackle an old (∼1992) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

Theorem. There is an $\operatorname{Li}_4$–expression in three variables $x, y, z$, whose “symbol” agrees with the one for $I_{31}(V(x, y), z)$. Moreover, there is an explicit solution consisting of 122 terms. Goncharov outlined how one should get from such an expression to a functional equation for $\operatorname{Li}_4$. Indeed:

Theorem. There is a resulting functional equation for $\operatorname{Li}_4$ in four variables consisting of . . . $931$ terms.
A Conjecture of Goncharov

Method successfully used also to tackle an old (≈ 1992) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of $Li_4$.

Theorem.

There is a $Li_4$–expression in three variables $x, y, z$, whose “symbol” agrees with the one for $I_{31}(V(x, y), z)$, i.e. a sum of five $I_{31}$-terms, where $V(x, y)$ denotes any variant of the 5-term equation for the dilogarithm.

Moreover, there is an explicit solution consisting of 122 terms.

Goncharov outlined how one should get from such an expression to a functional equation for $Li_4$. Indeed:

Theorem.

There is a resulting functional equation for $Li_4$ in four variables consisting of . . . 931 terms.
Mathematical Impact

A Conjecture of Goncharov

Method successfully used also to tackle an old (∼ 1992) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of $Li_4$. By a remark of N.Dan, one can take this expression to be $l_{31}(V(x, y), z)$, i.e. a sum of five $l_{31}$-terms,
Mathematical Impact

A Conjecture of Goncharov

Method successfully used also to tackle an old (\(\sim 1992\)) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of \(\text{Li}_4\). By a remark of N.Dan, one can take this expression to be \(I_{31}(V(x, y), z)\), i.e. a sum of five \(I_{31}\)-terms, where \(V(x, y)\) denotes any variant of the 5-term equation for the dilogarithm.
Mathematical Impact

A Conjecture of Goncharov

Method successfully used also to tackle an old (∼1992) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of $\text{Li}_4$. By a remark of N.Dan, one can take this expression to be $l_{31}(V(x, y), z)$, i.e. a sum of five $l_{31}$-terms, where $V(x, y)$ denotes any variant of the 5-term equation for the dilogarithm.

**Theorem.** *There is an $\text{Li}_4$–expression in three variables $x, y, z$, whose “symbol” agrees with the one for $l_{31}(V(x, y), z)$.*
A Conjecture of Goncharov

Method successfully used also to tackle an old (∼ 1992) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of $\text{Li}_4$. By a remark of N.Dan, one can take this expression to be $I_{31}(V(x, y), z)$, i.e. a sum of five $I_{31}$-terms, where $V(x, y)$ denotes any variant of the 5-term equation for the dilogarithm.

**Theorem.** There is an $\text{Li}_4$–expression in three variables $x, y, z$, whose “symbol” agrees with the one for $I_{31}(V(x, y), z)$. Moreover, there is an explicit solution consisting of 122 terms.
Mathematical Impact

A Conjecture of Goncharov

Method successfully used also to tackle an old (∼ 1992) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of $Li_4$. By a remark of N.Dan, one can take this expression to be $I_{31}(V(x, y), z)$, i.e. a sum of five $I_{31}$-terms, where $V(x, y)$ denotes any variant of the 5-term equation for the dilogarithm.

**Theorem.** There is an $Li_4$–expression in three variables $x, y, z$, whose “symbol” agrees with the one for $I_{31}(V(x, y), z)$. Moreover, there is an explicit solution consisting of 122 terms.

Goncharov outlined how one should get from such an expression to a functional equation for $Li_4$. Indeed:
Mathematical Impact

A Conjecture of Goncharov

Method successfully used also to tackle an old (∼ 1992) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of $Li_4$. By a remark of N.Dan, one can take this expression to be $l_{31}(V(x, y), z)$, i.e. a sum of five $l_{31}$-terms, where $V(x, y)$ denotes any variant of the 5-term equation for the dilogarithm.

**Theorem.** There is an $Li_4$-expression in three variables $x, y, z$, whose “symbol” agrees with the one for $l_{31}(V(x, y), z)$. Moreover, there is an explicit solution consisting of 122 terms.

Goncharov outlined how one should get from such an expression to a functional equation for $Li_4$. Indeed:

**Theorem.** There is a resulting functional equation for $Li_4$ in four variables consisting of . . .
Mathematical Impact

A Conjecture of Goncharov

Method successfully used also to tackle an old (∼ 1992) question of Goncharov, who reduced Zagier’s Conjecture on polylogarithms for weight 4, say, to two highly delicate combinatorial problems.

The first of these two problems boils down to the conjecture that a certain expression can be expressed in terms of $Li_4$. By a remark of N.Dan, one can take this expression to be $l_{31}(V(x, y), z)$, i.e. a sum of five $l_{31}$-terms, where $V(x, y)$ denotes any variant of the 5-term equation for the dilogarithm.

**Theorem.** There is an $Li_4$–expression in three variables $x, y, z$, whose “symbol” agrees with the one for $l_{31}(V(x, y), z)$. Moreover, there is an explicit solution consisting of 122 terms.

Goncharov outlined how one should get from such an expression to a functional equation for $Li_4$. Indeed:

**Theorem.** There is a resulting functional equation for $Li_4$ in four variables consisting of . . . 931 terms.
A typical more complicated term has the form

\[
\frac{(-1 + w)(-1 + xy)(1 - y - z + xyz)}{w(-1 + x)(-1 + y)y(-1 + wz)}
\]
A typical more complicated term has the form

\[
\frac{(-1 + w)(-1 + xy)(1 - y - z + xyz)}{w(-1 + x)(-1 + y)y(-1 + wz)}
\]

After symmetrisation can reduce to 9 terms under a big symmetry group \((S_5 \times S_5 \times S_2)\):
A typical more complicated term has the form

\[
\frac{(-1 + w)(-1 + xy)(1 - y - z + xyz)}{w(-1 + x)(-1 + y)y(-1 + wz)}
\]

After symmetrisation can reduce to 9 terms under a big symmetry group \((S_5 \times S_5 \times S_2)\):

**Theorem.** Let \(\{c_j\}_j = (-1, 2, 2, 4, 4, 8, 2, 3, -6)\); let \(A_i, B_j \in C\) \((i, j = 1, \ldots, 5)\). Then

\[
\sum_{j=1}^{9} \sum_{\sigma, \tau \in S_5} \pm \mathcal{L}_4 \left( f_j(A_{\sigma(1)}, \ldots, A_{\sigma(5)}, cr(B_{\tau(1)}, \ldots, B_{\tau(4)})) \right)
\]

is antisymmetric under \(A \leftrightarrow B\), where \(\{f_j\}_{j=1}^{9}\) are given by
A typical more complicated term has the form

\[
(−1 + w)(−1 + xy)(1 − y − z + xyz) \\
w(−1 + x)(−1 + y)y(−1 + wz)
\]

After symmetrisation can reduce to 9 terms under a big symmetry group \((S_5 \times S_5 \times S_2)\):

**Theorem.** Let \(\{c_j\}_{j=1} = (-1, 2, 2, 4, 4, 8, 2, 3, -6)\); let \(A_i, B_j \in \mathbb{C}\) \((i, j = 1, \ldots, 5)\). Then

\[
\sum_{j=1}^{9} \sum_{\sigma, \tau \in S_5} \pm \mathcal{L}_4 \left( f_j(A_{\sigma(1)}, \ldots, A_{\sigma(5)}, cr(B_{\tau(1)}, \ldots, B_{\tau(4)})) \right)
\]

is antisymmetric under \(A \leftrightarrow B\), where \(\{f_j\}_{j=1}^{9}\) are given by

\[
f_1(a, b, c, d, e, g) = -\frac{g(eabc - g)ebd \cdot eab}{(eabd - g)^2},
\]
A typical more complicated term has the form

\[
\frac{(-1 + w)(-1 + xy)(1 - y - z + xyz)}{w(-1 + x)(-1 + y)y(-1 + wz)}
\]

After symmetrisation can reduce to 9 terms under a big symmetry group \((S_5 \times S_5 \times S_2)\):

**Theorem.** Let \(\{c_j\}_j = (-1, 2, 2, 4, 4, 8, 2, 3, -6)\); let \(A_i, B_j \in \mathbb{C}\) \((i, j = 1, \ldots, 5)\). Then

\[
\sum_{j=1}^{9} \sum_{\sigma, \tau \in S_5} \pm \mathcal{L}_4 \left( f_j(A_{\sigma(1)}, \ldots, A_{\sigma(5)}, cr(B_{\tau(1)}, \ldots, B_{\tau(4)})) \right)
\]

is antisymmetric under \(A \leftrightarrow B\), where \(\{f_j\}_{j=1}^{9}\) are given by

\[
f_1(a, b, c, d, e, g) = -\frac{g(eabcd - g)ecbd \cdot eabd}{(eabd - g)^2},
\]

\[
f_2(a, b, c, d, e, g) = \frac{g^2 cade \cdot cabe}{1 - g eabd - g}, \ldots
\]
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object... $K_n^2(F)$ in terms of generators and relations as follows: generators (roughly): elements in kernel of symbol map for $Li_n$; relations: "universal" such elements, i.e. functional eqns for $Li_n$.

Such a presentation, denoted the $n$-th higher Bloch group of $F$, is known for $n = 2$ by combining work of Bloch, Borel and Suslin, and for $n = 3$ an explicit candidate was given by Goncharov in his proof of an important corollary about the Dedekind zeta value $\zeta_F(3)$ (often also called Zagier's Conjecture). For higher $n$, no candidate has been given.

Punchline: We can now give an explicit candidate for $K_7(F)$, using the functional equation in 931 terms mentioned above.
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object. . .

\[
K_n - 1(F)
\]
in terms of generators and relations as follows:

- **Generators** (roughly): elements in kernel of symbol map for \( L_i \);
- **Relations**: "universal" such elements, i.e. functional eqns for \( L_i \).

Such a presentation, denoted the \( n \)-th higher Bloch group of \( F \), is known for \( n = 2 \) by combining work of Bloch, Borel and Suslin, and for \( n = 3 \) an explicit candidate was given by Goncharov in his proof of an important corollary about the Dedekind zeta value \( \zeta_F(3) \) (often also called Zagier's Conjecture).

For higher \( n \), no candidate has been given.

Punchline: We can now give an explicit candidate for \( K_7(F) \), using the functional equation in 931 terms mentioned above.
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object. . . $K_{2n-1}(F)$ in terms of generators and relations as follows:
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object... $K_{2n-1}(F)$ in terms of generators and relations as follows:

- **generators** (roughly): elements in kernel of symbol map for $Li_n$;
- **relations**: “universal” such elements, i.e. functional eqns for $Li_n$.

Such a presentation, denoted the $n$-th higher Bloch group of $F$, is known for $n = 2$ by combining work of Bloch, Borel and Suslin, and for $n = 3$ an explicit candidate was given by Goncharov in his proof of an important corollary about the Dedekind zeta value $\zeta_F(3)$ (often also called Zagier’s Conjecture). For higher $n$, no candidate has been given.

Punchline: We can now give an explicit candidate for $K_7(F)$, using the functional equation in 931 terms mentioned above.
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object... $K_{2n-1}(F)$ in terms of generators and relations as follows:

- **generators** (roughly): elements in kernel of symbol map for $Li_n$;
- **relations**: “universal” such elements, i.e. functional eqns for $Li_n$. 

Punchline: We can now give an explicit candidate for $K_7(F)$, using the functional equation in 931 terms mentioned above.
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object... $K_{2n-1}(F)$ in terms of generators and relations as follows:

* **generators** (roughly): elements in kernel of symbol map for $Li_n$;
* **relations**: “universal” such elements, i.e. functional eqns for $Li_n$.

Such a presentation, denoted the *$n$-th higher Bloch group* of $F$, is known for $n = 2$ by combining work of Bloch, Borel and Suslin,
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object... $K_{2n-1}(F)$ in terms of generators and relations as follows:

**generators** (roughly): elements in kernel of symbol map for $Li_n$;

**relations**: “universal” such elements, i.e. functional eqns for $Li_n$.

Such a presentation, denoted the *$n$-th higher Bloch group* of $F$, is known for $n = 2$ by combining work of Bloch, Borel and Suslin, and for $n = 3$ an explicit candidate was given by Goncharov in his proof of an important corollary about the Dedekind zeta value $\zeta_F(3)$ (often also called Zagier’s Conjecture).
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object... $K_{2n-1}(F)$ in terms of generators and relations as follows:

- **generators** (roughly): elements in kernel of symbol map for $Li_n$;
- **relations**: “universal” such elements, i.e. functional eqns for $Li_n$.

Such a presentation, denoted the *n-th higher Bloch group* of $F$, is known for $n = 2$ by combining work of Bloch, Borel and Suslin, and for $n = 3$ an explicit candidate was given by Goncharov in his proof of an important corollary about the Dedekind zeta value $\zeta_F(3)$ (often also called Zagier’s Conjecture).

For higher $n$, no candidate has been given.

Punchline: We can now give an explicit candidate for $K_7(F)$, using the functional equation in 931 terms mentioned above.
A higher Bloch group candidate in weight 4.

Zagier conjectures that, for $F$ a number field and $n \geq 2$, there is a presentation of some mysterious mathematical object... $K_{2n-1}(F)$ in terms of generators and relations as follows:

*generators* (roughly): elements in kernel of symbol map for $Li_n$;
*relations*: “universal” such elements, i.e. functional eqns for $Li_n$.

Such a presentation, denoted the $n$-th higher Bloch group of $F$, is known for $n = 2$ by combining work of Bloch, Borel and Suslin, and for $n = 3$ an explicit candidate was given by Goncharov in his proof of an important corollary about the Dedekind zeta value $\zeta_F(3)$ (often also called Zagier’s Conjecture).

For higher $n$, no candidate has been given.

**Punchline:** We can now give an explicit candidate for $K_7(F)$, using the functional equation in 931 terms mentioned above.
Summary

- Non-trivial functional equations are not easy to come by, even for classical polylogs.
- In weights $\leq 3$, classical polylogs cover all MPL's.
- For weight $\geq 4$, one needs to grasp new functions, such as $I_{31}$ and $I_{1111}$.
- Found new functional equations for MPL's in weight 4 (different depths).
- Solved an old question of Goncharov, amounting to express $I_{31}(V(x, y), z)$ via $Li_4$. 
Summary

- Non-trivial fctl eqs not easy to come by, even for classical polylogs.
Summary

- non-trivial fctl eqs not easy to come by, even for classical polylogs
- in weights \( \leq 3 \), classical polylogs cover all MPL’s
Summary

• non-trivial fctl eqs not easy to come by, even for classical polylogs
• in weights $\leq 3$, classical polylogs cover all MPL’s
• for weight $\geq 4$, need a grip on new functions, e.g. $I_{31}$, $I_{1111}$
Summary

• non-trivial fctl eqs not easy to come by, even for classical polylogs
• in weights $\leq 3$, classical polylogs cover all MPL’s
• for weight $\geq 4$, need a grip on new functions, e.g. $I_{31}$, $I_{1111}$
• found new fctl eqs for MPL’s in weight 4 (different depths)
Summary

- non-trivial fctl eqs not easy to come by, even for classical polylogs
- in weights \( \leq 3 \), classical polylogs cover all MPL’s
- for weight \( \geq 4 \), need a grip on new functions, e.g. \( I_{31}, I_{1111} \)
- found new fctl eqs for MPL’s in weight 4 (different depths)
- solved an old question of Goncharov, amounting to express \( I_{31}(V(x,y), z) \) via \( Li_4 \).