DESY 09-003
SFB/CPP-09-65

The Multiple Zeta Value Data Mine

J. Blümlein[a], D.J. Broadhurst[b], J.A.M. Vermaseren[a,c] [1]

[a] *Deutsches Elektronen-Synchrotron, DESY, Platanenallee 6, D-15738 Zeuthen, Germany*

[b] *Physics and Astronomy Department, Open University, Milton Keynes MK7 6AA, UK*

[c] *Nikhef Theory Group Science Park 105, 1098 XG Amsterdam, The Netherlands*

Abstract

We provide a data mine of proven results for multiple zeta values (MZVs) of the form $\zeta(s_1, s_2, \ldots, s_k) = \sum_{n_1>n_2>\ldots>n_k>0}^{\infty} \left\{1/(n_1^{s_1} \ldots n_k^{s_k})\right\}$ with weight $w = \sum_{i=1}^{k} s_i$ and depth k and for Euler sums of the form $\sum_{n_1>n_2>\ldots>n_k>0}^{\infty} \left\{(\varepsilon_1^{n_1} \ldots \varepsilon_1^{n_k})/(n_1^{s_1} \ldots n_k^{s_k})\right\}$ with signs $\varepsilon_i = \pm 1$. Notably, we achieve explicit proven reductions of all MZVs with weights $w \le 22$, and all Euler sums with weights $w \le 12$, to bases whose dimensions, bigraded by weight and depth, have sizes in precise agreement with the Broadhurst–Kreimer and Broadhurst conjectures. Moreover, we lend further support to these conjectures by studying even greater weights ($w \le 30$), using modular arithmetic. To obtain these results we derive a new type of relation for Euler sums, the Generalized Doubling Relations. We elucidate the "pushdown" mechanism, whereby the ornate enumeration of primitive MZVs, by weight and depth, is reconciled with the far simpler enumeration of primitive Euler sums. There is some evidence that this pushdown mechanism finds its origin in doubling relations. We hope that our data mine, obtained by exploiting the unique power of the computer algebra language FORM, will enable the study of many more such consequences of the double-shuffle algebra of MZVs, and their Euler cousins, which are already the subject of keen interest, to practitioners of quantum field theory, and to mathematicians alike.

[1] Alexander-von-Humboldt Awardee.

1 Introduction

Multiple Zeta Values (MZVs) and Euler sums [1–3] have been of interest to mathematicians [1,4–7] and physicists [8] for a long time. One place in physics in which they are important is perturbative Quantum Field Theory. The interest became even larger when higher order calculations in Quantum Electrodynamics (QED) and Quantum Chromodynamics (QCD) started to need the multiple harmonic sums $S_{\vec{c}}(N)$ [9–11]. Euler sums are obtained as the limit $N \to \infty$ of the related multiple sums $Z_{\vec{c}}(N)$

$$\zeta_{\vec{c}} = \sum_{k=1}^{\infty} \frac{(\mathrm{sign}(b))^k}{k^{|b|}} \sum_{l=1}^{k-1} Z_{\vec{a}}(l) \; , \tag{1.1}$$

with $\vec{c} = (b, \vec{a}), \;\; b, a_i \in \mathbf{Z}$ and

$$Z_{b,\vec{a}}(N) = \sum_{k=1}^{N} \frac{(\sigma(b))^k}{k^{|b|}} Z_{\vec{a}}(k-1) \; , \quad Z_{\emptyset} = 1, \;\; Z_{\vec{a}}(0) = 0 \; , \tag{1.2}$$

with $\sigma(b) = \mathrm{sign(b)}$. Euler sums for which all indices are positive are called Multiple Zeta Values. Euler sums and MZVs with the first index $b = 1$ diverge, but will be included symbolically in the following, for convenience. Their degree of divergence can be uniquely traced back to a polynomial in the single harmonic sum

$$\sigma_0 = S_1(\infty) = \lim_{N\to\infty} \sum_{k=1}^{N} \frac{1}{k} \; . \tag{1.3}$$

We call the number of indices of the Euler sums and MZVs their depth d and

$$\mathsf{w} = \sum_{k=1}^{\mathsf{d}} |c_k| \tag{1.4}$$

their weight.

The number of Euler sums, resp. MZVs, up to a given weight w grows rapidly and amounts to $2 \cdot 3^{\mathsf{w}-1}$ and $2^{\mathsf{w}-1}$, respectively. A central question thus concerns to find all the relations between the Euler sums, resp. MZVs for *fixed weight* and *depth*, and even more importantly, *new relations* between MZVs at the one hand and Euler sums on the other hand, and the corresponding bases. Besides weight and depth, another degree of freedom, being discussed later, the *pushdown* p, quantifies the relation between MZVs and Euler sums. The way to view MZVs, embedded into Euler sums, dates back to Broadhurst [12], who conjectured the counting of basis elements at fixed $\{w, d\}$. The corresponding conjecture for the MZVs is due to Broadhurst and Kreimer [13][2]. For the number of basis elements for MZVs of a given weight, without regard to depth, an upper bound has been proven in [14]. This coincides with the result obtained by summing the numbers conjectured in [13] over all depths at a fixed weight.

[2]Conjectures for fixed weight are due to Zagier [2] and probably also independently due to Drinfel'd, Goncharov and Kontsevich.

The relations between MZVs and Euler sums in Ref. [12] are conjectured using algorithms for integer relations as PSLQ [15] and LLL [16] which use representations based on a large number of digits.

It is well-known that MZVs obey shuffle- and stuffle-relations. This is due to their representation in terms of Poincaré iterated integrals [17] at argument $x = 1$, which are harmonic polylogarithms [18] on the one hand, and harmonic sums [9–11] on the other hand. The former quantities obey a shuffle- the latter a quasi-shuffle algebra, i.e. shuffling with "stuff" from polynomials of harmonic sums of lower weight. Currently no other relation independent is known between MZVs. The Euler sums are also related by both the shuffle- and stuffle-relations, where now also negative indices occur to indicate alternating sums. However, these relations are not sufficient to obtain the minimal set of basis elements as being conjectured in [12]. Starting with $\mathsf{w} = 8$ it requires the doubling relation and with $\mathsf{w} = 11$ generalized doubling relations derived in the present paper. Beginning with $\mathsf{w} = 12$ relations occur, which allow to express MZVs of a given depth in terms of Euler sums of a lesser depth. Part of these relations have been conjectured in the past using integer relations [12, 19]. A main objective of the present paper is to prove these relations applying computer algebra methods and to find relations of this type in a more systematic way.

We investigate the Euler sums to $\mathsf{w} = 12$ completely, deriving basis-representations for all individual values in an explicit analytic calculation. For the MZVs the same analysis is being performed up to $\mathsf{w} = 22$. To $\mathsf{w} = 24$ we checked the conjectured size of the basis using modular arithmetic. Under the further conjecture that the basis elements can be chosen out of MZVs of depth $\mathsf{d} \leq \mathsf{w}/3$ we confirm the conjecture up to $\mathsf{w} = 26$. Furthermore, the following runs at limited depth, using modular arithmetic keeping the highest weight terms only, were performed: $\mathsf{d} = 7,\ \mathsf{w} = 27$; $\mathsf{d} = 6,\ \mathsf{w} = 28$; $\mathsf{d} = 7,\ \mathsf{w} = 29$; $\mathsf{d} = 6,\ \mathsf{w} = 30$. For the Euler sums complete results were obtained for $\mathsf{d} \leq 3,\ \mathsf{w} = 29$; $\mathsf{d} \leq 4,\ \mathsf{w} = 22$; $\mathsf{d} \leq 5,\ \mathsf{w} = 17$ and for $\mathsf{d} \leq 3,\ \mathsf{w} = 51$; $\mathsf{d} \leq 4,\ \mathsf{w} = 30$; $\mathsf{d} \leq 5,\ \mathsf{w} = 21$; $\mathsf{d} \leq 6,\ \mathsf{w} = 17$ using modular arithmetic neglecting products of lower weight. The conjectures on the number of basis elements w.r.t. $\{w, d\}$ were verified in all these cases. The results of our analysis are made available in the Multiple Zeta Data Mine [20], to allow users to search for yet un-discovered relations.

The paper is organized as follows. In Section 2 we summarize basic notations and the well known relations between Euler sums. A novel type of relations, the generalized doubling relations, is derived in Section 4. There we also discuss its impact in finding the basis elements at a given weight w and depth d. In Section 5 an outline is given on the details of the computer algebra code, which allowed to derive the basis-representations of the MZVs and Euler sums. Details on the running for the different cases are reported in Section 6. The results are stored in the Multiple Zeta Value Data Mine [3], which is described in Section 7. To establish the solution of the problems dealt with in the current project required some new features of `FORM` [21] and `TFORM` [22], which are described in Section 8. In Section 9 we briefly review the status achieved by other groups and present first results of the analysis. In particular a series of conjectures made in the mathematical literature are confirmed within the range explored in the present study. Here we discuss

[3] It goes without saying that also the Euler sums are covered here.

also particular choices for the respective bases. An interesting aspect representing MZVs by Euler sums concerns the so-called pushdowns, i.e. the representation of a MZV of a given depth d with Euler sums of depth d' with $\mathsf{d}' < \mathsf{d}$. These are studied in Section 10 in which we also introduce a new kind of object, the $A_{\vec{a}}$–functions. They play a key role in representing a class of Euler sums. Some more special Euler sums are studied in Section 11. Section 12 contains the conclusions and an outlook. In the Appendices we provide different basis representations and discuss the pushdowns in more detail.

2 Basic Formalism

In the following we work with three types of objects, the finite nested harmonic $S_{\vec{a}}$-sums, $Z_{\vec{a}}$-sums, both at argument $N \in \mathbf{N}$, and the harmonic polylogarithms $H_{\vec{a}}$ at argument $x, \quad 0 \leq x \leq 1$. They all can be used to define the MZVs and the Euler sums in the limit $N \to \infty$ and $x = 1$, respectively. We generally consider the case of colored objects corresponding to $n = 2$, i.e. numerator weights with $(\pm 1)^k$, i.e. polylogarithms of square root of unity.

The harmonic S-sums are defined by

$$
\begin{aligned}
S_{\vec{a}}(0) &= 0 \\
S_b(N) &= \sum_{k=1}^{N} \frac{(\sigma(b))^k}{k^{|b|}} \\
S_{b,\vec{a}}(N) &= \sum_{k=1}^{N} \frac{(\sigma(b))^k}{k^{|b|}} S_{\vec{a}}(k) \; .
\end{aligned}
\tag{2.1}
$$

In this form these sums are usually used by physicists. In particular results in QCD [23–26] are expressed in terms of these objects[4].

Next there are the Z-sums. They are defined in (1.2). These are of course very similar to the S-sums and it is straightforward to convert from one notation to the other. The Z-sums are mostly used by mathematicians. In the limit $N \to \infty$ and when $\sigma(b) = 1$ for all b they define the Multiple Zeta Values (MZVs):

$$
\zeta_{\vec{a}} = \lim_{N \to \infty} Z_{\vec{a}}(N) \; . \tag{2.2}
$$

When we allow $\sigma(b)$ to take the values $+1$ or -1 and we take the limit $N \to \infty$ we speak of Euler sums.

Finally, there are the harmonic polylogarithms, which we will also call H-functions. We consider the alphabets

$$
\begin{aligned}
\mathfrak{h} &= \{0, 1, -1\} \quad \text{and} \\
\mathfrak{H} &= \{1/x, 1/(1-x), 1/(1+x)\} \; ,
\end{aligned}
\tag{2.3}
$$

[4]The class of Euler sums is known to be too small in general to represent all Feynman diagrams for no-scale processes in scalar field theories, but have to be extended in higher orders [27–30]. This will apply also for field theories as QCD and QED. Feynman-integrals are periods [31] if all ratios of Lorenz invariants and masses have rational values [32].

which define the elements of the index set of the harmonic polylogarithms[5] and the functions in the iterated integrals, respectively. Let $\vec{a} = \{m_1, \ldots, m_k\}, \quad m_i, b \in \mathfrak{h}, \quad k \geq 1$, then

$$\begin{aligned} H_{b,\vec{a}}(x) &= \int_0^x dz f_b(z) H_{\vec{a}}(z) \\ H_0(x) &= \log(x) \\ H_1(x) &= -\log(1-x) \\ H_{-1}(x) &= \log(1+x) \ . \end{aligned} \tag{2.4}$$

The sums to infinity and the H-functions at unity are all related and can be readily transformed into each other. For some applications it is most convenient to work with one set of objects and for others other objects may be more useful. For reasons being explained later our computer programs work mostly with H-functions at unity.

A first aspect to note is that the index fields of the sums and the functions are of a different nature. This can be seen by introducing the notation in which the index n in the sums can alternatively be written as $n-1$ zeroes followed by a one and $-n$ is written as $n-1$ zeroes followed by a minus one. In the H-functions we can absorb alternatively the zeroes in the nonzero number to their right by raising its absolute value by one for each zero being absorbed. This leaves only the rightmost zeroes. Hence:

$$\begin{aligned} S_{-3,4}(N) &= S_{0,0,-1,0,0,0,1}(N) \\ Z_{2,-5}(N) &= Z_{0,1,0,0,0,0,-1}(N) \\ H_{0,1,-1,0,0,-1,0,0}(x) &= H_{2,-1,-3,0,0}(x) \ . \end{aligned} \tag{2.5}$$

The notation in terms of the $0, \pm 1$ we call the (iterated) integral notation. The natural notation of the sums we call the (nested) sum notation.

Reference to the alphabet $\mathfrak{h}$ allows us to count the number of objects and to classify them. The number of indices in this integral notation is called the weight of the sum or the function. For a given weight w there are $2 \cdot 3^{\mathrm{w}-1}$ sums and 3^{w} H-functions. When the sums are written in the original sum notation, the number of indices indicates the number of nested sums. This is also called the depth of the sum. When there are no trailing zeroes in the H-functions we can introduce the depth in the same way. Because of algebraic relations we can express the functions with trailing zeroes as products of powers of $\log(x)$ and H-functions with fewer indices [18, 37]. In that case the concept of depth can be used in a similar way as with the sums.

For any argument $x \neq 1$ the H-functions form a shuffle algebra:

$$H_{\vec{p}}(x) H_{\vec{q}}(x) = \sum_{\vec{r} \,\in\, \vec{p} \,\sqcup\!\sqcup\, \vec{q}} H_{\vec{r}}(x) \ , \tag{2.6}$$

where $\vec{p} \sqcup\!\sqcup \vec{q}$ denotes the shuffle product, cf. e.g. [37], and $p_i, q_i \in \mathfrak{h}$. When $x = 1$ H-functions for which the first index is one are divergent. It is however possible to express them in terms of a single divergent object and other finite terms in a consistent way.

[5]Special cases are the classical polylogarithms [33] and the Nielsen polylogarithms [34]. Generalizations of harmonic polylogarithms are found in [35, 36].

The only thing that breaks down is that there are correction terms to the shuffle relations when both objects in the left hand side are divergent, see also Ref. [18]. Because the number of non-zero indices remains the same during the shuffle operation, we call it depth preserving.

For general argument N the sums form a stuffle algebra, [37]. This is a general property of sums which we show here for a double sum:

$$\begin{aligned}\sum_{i=1}^{N}\sum_{j=1}^{N} &= \sum_{i=1}^{N}\sum_{j=1}^{i}+\sum_{j=1}^{N}\sum_{i=1}^{j}-\sum_{i=j=1}^{N} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{i-1}+\sum_{j=1}^{N}\sum_{i=1}^{j-1}+\sum_{i=j=1}^{N} \ .\end{aligned} \tag{2.7}$$

The diagonal terms give extra 'stuff' beyond the normal shuffling in the natural notation for the sums. Even though the diagonal terms add terms usually the stuffle relations have fewer terms because most of the time some of the indices will have an absolute value greater than one. We write in terms of S– or Z–notation :

$$S_m(N)S_n(N) = S_{m,n}(N)+S_{n,m}(N)-S_{m\&n}(N) \tag{2.8}$$

$$Z_m(N)Z_{n,k}(N) = Z_{m,n,k}(N)+Z_{n,m,k}(N)+Z_{n,k,m}(N)+Z_{m\&n,k}(N)+Z_{m,k\&n}(N) \ . \tag{2.9}$$

Here the operator & is defined by

$$\begin{aligned} m\&n &= \sigma(m)\sigma(n)(|m|+|n|) \\ &= \sigma_n m+\sigma_m n \ . \end{aligned} \tag{2.10}$$

The above algebraic relations can be used to bring an expression with many harmonic polylogarithms or harmonic sums into a standard form. For evaluation, however, it is often useful to work it the other way and reduce the number of objects at the highest weight in favor of products of objects with a lower weight which are easier to evaluate. For this the theory of Lyndon words [38] applies, but especially with the stuffles the extra terms which have the same weight but a lower depth have to be taken along and make things considerably more involved than pure shuffles.

A k-ary Lyndon word of length n is a n-letter concatenation product over an alphabet of size k, which, observing lexicographical ordering is smaller than all its suffixes. Equivalently, it is the unique minimal element in the lexicographical ordering of all its cyclic permutations. The uniqueness implies that a Lyndon word is aperiodic. So it differs from any of its non-trivial rotations. In our case we will usually replace minimal by maximal when we form Lyndon words of indices of MZVs or Euler sums. That is, we will put the larger indices to the left. One could also say that the concept of greater than is defined in a special way inside the alphabet. The practical advantage is that this guarantees that none of the MZVs of which the index string forms a Lyndon word is divergent.

When we use the stuffle relations to simplify the set of objects at a given weight, we can arrange that they are used in such a way that they never raise the value of the depth parameter. Some terms will have a lower value for the depth. Therefore we call the stuffles potentially depth lowering.

When we consider the sums to infinity there are two classes of extra relations worth mentioning. The first is the 'rule of the triangle' which is based on

$$\lim_{N\to\infty}\sum_{i=1}^{N}\sum_{j=1}^{N} = \lim_{N\to\infty}\sum_{i=1}^{N}\sum_{j=1}^{N-i}+\lim_{N\to\infty}\sum_{i=1}^{N}\sum_{j=N-i+1}^{N} . \tag{2.11}$$

For most sums the second term will give a limit that goes to zero with at least one power of $1/N$, possibly multiplied by powers of $\log(N)$. This system can be generalized to the product of any pair of sums and it can be proven that the limit of the second term vanishes when at least one of the sums in the left hand side is finite [10]. When both are divergent it is possible to work out which extra terms are needed. Because the sums of the first term in the right hand side can be worked out, even in the most general case, the above gives us an extra algebraic relation for the sums to infinity. These relations are depth preserving.

When we consider the H-functions at unity, it is easy to see that they can be written as nested sums to infinity of the same variety as the Z-sums or the S-sums. Hence they now obey also the stuffle algebra. And it can be shown that the 'rule of the triangle' is no more than the equivalent of the shuffle algebra for the H-functions, with the same restrictions for the double divergent terms.

The next set of relations is easy to see for finite sums:

$$\begin{aligned} S_m(N) &= \sum_{i=1}^{N}\frac{1}{i^m}=\sum_{i=1}^{N}2^m\frac{1}{(2i)^m}=\sum_{i=1}^{2N}2^{m-1}\frac{1+(-1)^i}{i^m} \\ &= 2^{m-1}\left[S_m(2N)+S_{-m}(2N)\right] , \end{aligned} \tag{2.12}$$

which generalizes into

$$S_{n_1,\cdots,n_p}(N) = 2^{n_1+\cdots+n_p-p}\sum_{\pm}S_{\pm n_1,\cdots,\pm n_p}(2N) . \tag{2.13}$$

Here the sum is over all 2^p plus/minus combinations. These relations are called the 'doubling relations'. For finite sums with $n_1 \neq 1$ these relations can be used directly. In the case that divergent sums are involved there are again correction terms.

The equivalent formula for the H-functions is obtained by looking at $H_{\vec{a}}(x^2)$ and noticing that at $x=1$ this is the same as $H_{\vec{a}}(x)$. In that case we have

$$H_{1,0,1}(x^2) = 2\left[H_{1,0,1}(x)-H_{-1,0,1}(x)-H_{1,0,-1}(x)+H_{-1,0,-1}(x)\right] , \tag{2.14}$$

which generalizes to any number of indices. The rule is that the factor is identical to 2^m in which m is the number of zeroes in the indices, and each one in the left hand side gives a doubling of terms in the right hand side: one term with a corresponding 1 and one with a corresponding -1 and an extra overall minus sign. In the left hand side one cannot have negative indices. Again one should be careful with the divergent functions.

Divergences are expressed in terms of the object $S_1(\infty)$, (1.3). In most cases one can use this as a regular symbol and take it along in the equations and expressions. Unless we mention the problems explicitly, one can exchange limits and sums when this object is combined with finite sums. The reason is that our finite sums converge faster than that this object diverges. A problem occurs when we use the doubling formula on it. We find:

$$\begin{aligned} S_1(\infty) &= S_1(2\infty) + S_{-1}(2\infty) \\ &= S_1(2\infty) - \log(2) \ , \end{aligned} \tag{2.15}$$

which just shows that the divergence of $S_1(\infty)$ is logarithmic, since

$$S_1(N) = \ln(N) + \gamma_E + \frac{1}{2N} + \frac{1}{12N^2} + O\left(\frac{1}{N^3}\right) \ , \tag{2.16}$$

cf. [25]. One can however use the stuffle relations on these objects. This allows one in principle to express the divergent sums in terms of products of $S_1(\infty)$ and finite sums as in

$$S_1(N)S_{m,n}(N) = S_{1,m,n}(N) + S_{m,1,n}(N) + S_{m,1,n}(N) - S_{m\&1,n}(N) - S_{m,n\&1}(N) \ . \tag{2.17}$$

If we assume $m \neq 1$ this allows us to express the divergent sum $S_{1,m,n}(\infty)$ the way we want it. Similarly one can now look at stuffles of $S_1 \cdot S_1$ to determine $S_{1,1}$ and then look at stuffles of $S_{1,1}(N)$ with finite sums. In the programs we give $S_1(\infty)$ the name `Sinf` which, due to the above, can be treated as a regular symbol.

Because we have two shuffle products - the stuffle-algebra is a quasi-shuffle algebra [39] - we can equate the result of the stuffle product of two objects with the result of the shuffle product of the same two objects. The resulting relation is called a double-shuffle relation and contains only objects of the same weight. These relations have been used in a number of calculations. For our type of calculations they are, however, not suitable. We will use the stuffle and the shuffle relations individually. This will allow a better optimization of the algorithms.

The concept of *duality* is very useful and allows us to roughly half the number of objects that need to be computed. The duality relation is defined in the integral notation using harmonic polylogarithms at one. It states that if we have a MZV and we reverse the order of its indices while at the same time transforming zeroes into ones and ones into zeroes the new object has the same value as the original. An example of this duality is the relation

$$H_{0,1,0,1,1,1,1,1} = H_{0,0,0,0,0,1,0,1} \tag{2.18}$$

In mathematics one traditionally considers this duality in sum notation. In that case, for a sequence

$$I = (p_1+1, \{1\}_{q_1-1}, p_2+1, \{1\}_{q_2-1}, \ldots, p_k+1, \{1\}_{q_k-1}) \tag{2.19}$$

there is a dual sequence

$$\tau(I) = (q_k+1, \{1\}_{p_k-1}, q_{k-1}+1, \{1\}_{p_{k-1}-1}, \ldots, q_1+1, \{1\}_{p_1-1}) \ . \tag{2.20}$$

The duality theorem [2] states

$$\zeta_I = \zeta_{\tau(I)} \ . \tag{2.21}$$

It was conjectured in [40] and is easily proven by the transformation $x \to 1-t$ of the corresponding iterated integrals.

Because for even weights there are some elements that are self-dual this does not divide the number of terms exactly by two. Considering that we do not have to consider the divergent objects we have $2^{\mathrm{w}-3}$ relevant objects when w is odd and $2^{\mathrm{w}-3}+2^{\mathrm{w}/2-2}$ relevant objects when w is even.

For Euler sums the equivalent transformation is more complicated due to the three letter alphabet. It is obtained by studying the transformation

$$x \to \frac{1-t}{1+t} \tag{2.22}$$

in the integral representation. Its effect is that given the alphabet

$$\begin{aligned} A &= 0 \quad \leftarrow \frac{1}{x} \\ B &= 1 \quad \leftarrow \frac{1}{1-x} \\ C &= -1 \quad \leftarrow \frac{1}{1+x} \end{aligned} \tag{2.23}$$

and a string of letters from this alphabet as indices of an Euler sum H, the ‘dual expression’ is obtained by reverting the string of letters and making the replacement

$$\begin{aligned} A &\to B \oplus C \\ B &\to A \ominus C \\ C &\to C\,. \end{aligned} \tag{2.24}$$

The addition and subtraction operators here mean that for each such transformation there will be a doubling of the number of terms, one with the first letter and the other with the other letter. The sign-operator $\oplus(\ominus)$ refers to the sign of the complete term. Because these relations can both raise and lower the depth of a term we call them depth mixing.

We have tested that this transformation does add something new beyond what the stuffles and the shuffles give us. In particular, when one derives equations for all sums at a given weight, they can be used to replace the doubling and the *Generalized Doubling Relations* (GDRs), see Section 4. We have tested this to weight $\mathrm{w}=12$. Unfortunately they cannot be used when the concept of depth of the sums is important and hence we have not used these equations in our programs.

A generalization of the Riemann ζ-function is Hurwitz’ ζ-function [6, 41] :

$$\zeta(n,a) = \sum_{k=1}^{\infty} \frac{(\mathrm{sign}(n))^k}{(k+a)^{|n|}}\,, \tag{2.25}$$

which can be extended to generalized Euler sums analogous to (1.1). Since a is a real parameter, one may differentiate $\zeta(\vec{c},a)$ w.r.t. a and seek for new relations. We investigated this possibility, but did not find new relations beyond those quoted above.

When we are discussing bases into which we write the MZVs and the Euler sums we recognize two types of basis :

Definition. A basis of a vector space of all Euler sums or MZVs at a given weight w is called a *Fibonacci basis*.

Definition. A basis of the ring of all Euler sums or MZVs at a given weight w is called a *Lyndon basis* if all its elements have an index field that forms a Lyndon word.

In a Fibonacci basis all basis elements are nested sums of the same weight. The name derives from the observation that the size of such a basis for the Euler sums seem to follow a Fibonacci rule [42]. Also the MZVs seem to follow the rule that the total number of their basis elements follow the Fibonacci-like Padovan numbers [43], see Appendix A.

In a Lyndon basis we write in the complete basis as many elements as possible as products of lower weight basis elements and what remains is the Lyndon basis. Simultaneously we require the index field to form a Lyndon word. Sometimes a Lyndon basis can be formed from a Fibonacci basis by just selecting the Lyndon words from it. The number of basis elements in the case of MZVs is counted by a Witt-type relation [44] based on the Perrin numbers [45]. In the case of the Euler sums the corresponding relation relies on the Lucas numbers [46], see Appendix A. Any other basis we will call a mixed basis.

We will usually try to arrange the Lyndon bases in such a way that they are 'minimal depth'. This means that if an element can be expressed in terms of objects with a lower depth, it cannot be a member of the basis. Details on a variety of bases are given in Appendix A. The complete basis we actually selected for the MZVs is presented in Appendix B.

3 Conjectures on Bases at Fixed Weight and Depth

Broadhurst [12] and Broadhurst and Kreimer [13] formulated conjectures on the size of the basis for Euler sums and MZVs, respectively, which we summarize in the following.

Let $E_{w,d}$ be the number of independent Euler sums at weight $\mathsf{w} > 2$ and depth d that cannot be reduced to primitive Euler sums of lesser depth and their products. Thus we believe that $E_{3,1} = 1$, since there is no known relationship between ζ_3, π^2 and $\ln(2)$. It is rather natural to guess that $E_{w,d}$ is given by a filtration of the coefficients of powers of x and y in the expansion of $1/(1-xy-x^2)$, i.e. that

$$\prod_{w>2}\prod_{d>0}(1-x^w y^d)^{E_{w,d}} \stackrel{?}{=} \frac{1-xy-x^2}{(1-xy)(1-x^2)} = 1 - \frac{x^3 y}{(1-xy)(1-x^2)}. \tag{3.1}$$

It is then easy to obtain $E_{w,d}$ by Möbius transformation of the binomial coefficients in Pascal's triangle. Let

$$T(a,b) = \frac{1}{a+b}\sum_{d|a,b}\mu(d)\,\frac{(a/d+b/d)!}{(a/d)!(b/d)!} \tag{3.2}$$

where the sum is over all positive integers d that divide both a and b and the Möbius function is defined by

$$\mu(d) = \begin{cases} 1 & \text{when } d = 1 \\ 0 & \text{when } d \text{ is divisible by the square of a prime} \\ (-1)^k & \text{when } d \text{ is the product of } k \text{ distinct primes} \end{cases} \tag{3.3}$$

When w and d have the same parity, and $w > d$, one obtains from (3.1)

$$E_{w,d} = T\left(\frac{w-d}{2}, d\right). \tag{3.4}$$

With the exception of $\ln(2)$ and ζ_2, which act as the seeds xy and x^2, all elements of the basis are thereby conjecturally enumerated. In this paper we provide extensive evidence to support conjecture (3.1).

Now let $D_{w,d}$ be the number of independent MZVs at weight $w > 2$ and depth d that cannot be reduced to primitive MZVs of lesser depth and their products. Thus we believe that $D_{8,2} = 1$, since there is no known relationship between the double sum $Z_{5,3} = \sum_{m>n>0} 1/(m^5n^3)$ and single sums or their products. It is tempting to guess to that $D_{w,d}$ is generated by filtration of the expansion of $1/(1-x^2-x^3y)$, seeded by π^2 and ζ_3. But this is not the case, since the solution of the double-shuffle algebra at weight $w = 12$ leaves one quadruple sum undetermined, while the obvious guess would leave none. The conjecture [13] in this case is rather ornate, cf. Table 16.

$$\prod_{w>2}\prod_{d>0}(1-x^w y^d)^{D_{w,d}} \stackrel{?}{=} 1 - \frac{x^3y}{1-x^2} + \frac{x^{12}y^2(1-y^2)}{(1-x^4)(1-x^6)} \tag{3.5}$$

with a correction term whose numerator, $x^{12}y^2(1-y^2)$, ensures that $D_{12,4} = 1$ and $D_{12,2} = 1$, in agreement with the solution of the double-shuffle algebra. The denominator $(1-x^4)(1-x^6)$ is then chosen to give $D_{2m,2} = \lfloor (m-1)/3 \rfloor$ for the number of primitive double sums with weight $2m$. Conjecture (3.5) is impressively supported by the data mine.

Furthermore,

$$\prod_{w>2}\prod_{d>0}(1-x^w y^d)^{M_{w,d}} = \frac{1-x^2-x^3y}{1-x^2} \tag{3.6}$$

is the conjectured generating function of the basis elements $M_{w,d}$ of the MZVs when expressed as Euler sums in a minimal depth representation, see Table 17.

4 Generalized Doubling Relations

Up to $\mathrm{w} = 10$ the shuffle-, stuffle-, and doubling relations were sufficient to express the alternating Euler sums over a basis whose size is in accordance with the conjecture in Ref. [12]. This is not the case from $\mathrm{w} = 11$ onwards. Therefore one has to seek a new kind of relations, which we derive in the following. Of course, when we derive all relations at a given weight we could use the relations of (2.24). The fact that they are depth mixing makes them useless for calculations in which the concept of depth plays a role. Hence we need our new (depth lowering) relations anyway. We first present the derivation of this class of relations and discuss then their effect on the number of basis elements representing the Euler sums.

4.1 Derivation of the generalized doubling relations

The only relations we could find thus far adding something new to the system are the depth 2 relations of Ref. [12]. They are based on partial fractioning in two different ways. One way is:

$$\frac{1}{(2i+j)(j)} = \frac{1}{(2i+2j)(2i+j)} + \frac{1}{(2i+2j)(j)} \tag{4.1}$$

We can take out the factor two and in the first term the $2i$ is taken care of by changing the summation over i into a summation over the even numbers by including a factor $(1+(-1)^i)/2$, which introduces negative indices in some Euler sums. In the other way we use the more regular form

$$\frac{1}{(2i+j)(j)} = \frac{1}{(2i)(j)} - \frac{1}{(2i)(2i+j)}\ . \tag{4.2}$$

Together these partial fractions produce new types of relations.

Here we will give the new set of relations and their derivation. We will work with the Z-sums. The reason is a particularly handy representation of these sums to infinity [40, 47]:

$$\begin{aligned} Z_{m_1,\cdots,m_p}(\infty) &= \sum_{i_1>i_2>\cdots>i_p>0}^{\infty} \frac{\sigma_1^{i_1}\sigma_2^{i_2}\cdots\sigma_p^{i_p}}{i_1^{n_1}i_2^{n_2}\cdots i_p^{n_p}} \\ &= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\cdots\sum_{x_p=1}^{\infty} \frac{\sigma_1^{x_1+x_2+\cdots+x_p}\sigma_2^{x_2+\cdots+x_p}\cdots\sigma_p^{x_p}}{(x_1+x_2+\cdots+x_p)^{n_1}(x_2+\cdots+x_p)^{n_2}\cdots(x_p)^{n_p}}\ , \end{aligned} \tag{4.3}$$

in which we take $n_i = |m_i|$ and σ_i to be the sign of m_i.

Let us start with the re-derivation of the equation for depth $\mathrm{d} = 2$. Actually we do not reproduce it exactly, but we obtain a similar equation. Here we write for brevity $Z(a,b) = Z_{a,b}(\infty)$. Throughout this Section we assume that a,b,c and d are positive integers. We consider the following combination of Z-sums :

$$\begin{aligned} E(a,b) &= \frac{1}{2}(Z(a,b)+Z(-a,-b)) \\ &= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty} \frac{1}{(x_1+x_2)^a\, x_2^b}\frac{1+(-1)^{x_1}}{2} = \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty} \frac{1}{(2x_1+x_2)^a\, x_2^b} \\ &= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty} \left[\sum_{i=1}^{a} A_i^{(a,b)} \frac{1}{(2x_1+2x_2)^{a+b-i}\ (2x_1+x_2)^i} + \sum_{i=1}^{b} B_i^{(a,b)} \frac{1}{(2x_1+2x_2)^{a+b-i}\ x_2^i}\right] \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty} \frac{1}{(x_1+x_2)^{a+b-i}\,(2x_1+x_2)^i} \\
&+ \sum_{i=1}^{b} B_i^{(a,b)} 2^{i-a-b} Z(a+b-i,i) \\
&= \sum_{i=1}^{b} B_i^{(a,b)} 2^{i-a-b} Z(a+b-i,i) + \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} \sum_{x_1=1}^{\infty}\sum_{x_2=x_1+1}^{\infty} \frac{1}{(x_1+x_2)^i x_2^{a+b-i}} \\
&= \sum_{i=1}^{b} B_i^{(a,b)} 2^{i-a-b} Z(a+b-i,i) + \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty} \frac{1}{(x_1+x_2)^i x_2^{a+b-i}} \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{x_1} \frac{1}{(x_1+x_2)^i x_2^{a+b-i}} \\
&= \sum_{i=1}^{b} B_i^{(a,b)} 2^{i-a-b} Z(a+b-i,i) + \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} Z(i,a+b-i) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} \sum_{x_2=1}^{\infty}\sum_{x_1=x_2}^{\infty} \frac{1}{(x_1+x_2)^i x_2^{a+b-i}} \\
&= \sum_{i=1}^{b} B_i^{(a,b)} 2^{i-a-b} Z(a+b-i,i) + \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} Z(i,a+b-i) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} \sum_{x_2=1}^{\infty}\sum_{x_1=1}^{\infty} \frac{1}{(x_1+2x_2)^i x_2^{a+b-i}} - \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} \sum_{x_2=1}^{\infty} \frac{1}{(2x_2)^i x_2^{a+b-i}} \\
&= \sum_{i=1}^{b} B_i^{(a,b)} 2^{i-a-b} Z(a+b-i,i) + \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} Z(i,a+b-i) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} \sum_{x_2=1}^{\infty}\sum_{x_1=1}^{\infty} \frac{1+(-1)^{x_2}}{2} \frac{1}{(x_1+x_2)^i x_2^{a+b-i}} - \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b} \sum_{x_2=1}^{\infty} \frac{1}{x_2^{a+b}} \\
&= \sum_{i=1}^{b} B_i^{(a,b)} 2^{i-a-b} Z(a+b-i,i) + \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} Z(i,a+b-i) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} \frac{1}{2}\left(Z(i,a+b-i) + Z(i,-(a+b-i))\right) - \frac{(a+b-1)!}{(a-1)!\,b!} 2^{-a-b} Z(a+b)\ , \qquad (4.4)
\end{aligned}
$$

with

$$
A_i^{a,b} = \frac{(a+b-i-1)!}{(a-i)!(b-1)!} \qquad (4.5)
$$

$$
B_i^{a,b} = \frac{(a+b-i-1)!}{(b-i)!(a-1)!}\ . \qquad (4.6)
$$

Actually there is a slight problem with the above derivation. At two points we changed the summation range. Once from ∞ to $\infty/2$ and once from ∞ to 2∞. This causes no problems if the sum is finite, but for the divergent sums this needs a correction term. The second case is harmless as it concerns only an inner sum, the step in which $(-1)^{x_2}$ is introduced. But the first case, in the very first step of the derivation, needs a correction term. Hence the full formula becomes :

$$
\begin{aligned}
E(a,\sigma_b b) &= \frac{1}{2}(Z(a,\sigma_b b)+Z(-a,-\sigma_b b)) \\
&= \frac{1}{2}\delta(a-1)Z(-1)Z(\sigma_b b)-\frac{1}{2}\delta(a-1)\delta(\sigma_b b-1)Z(-2) \\
&+ \sum_{i=1}^{b} B_i^{(a,b)} 2^{i-a-b} Z(a+b-i,\sigma_b i) \\
&+ \sum_{i=1}^{a} A_i^{(a,b)} 2^{i-a-b} Z(\sigma_b i,a+b-i) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} \frac{1}{2}(Z(\sigma_b i,\sigma_b(a+b-i))+Z(\sigma_b i,-\sigma_b(a+b-i))) \\
&- \frac{(a+b-1)!}{(a-1)!\,b!} 2^{-a-b} Z(a+b)\ . \qquad (4.7)
\end{aligned}
$$

Here also the signs on the indices a and b are included which is only a very mild complication in the derivation. The function $\delta(m)$ is one when m is zero and zero otherwise. The σ-variables have a value that is either $+1$ or -1 and indicate non-alternating and alternating sums. Due to the symmetry of the starting formula a sign on the first variable is not necessary. If we put it anyway in the form of σ_a, σ_b will have to be replaced by $\sigma_a\sigma_b$ in the right hand side.

It is quite relevant to take these σ factors along. Although they are usually not needed to get a complete coverage of depth $\mathrm{d}=2$ sums, in the case of greater depth sums they are necessary.

The above derivation shows basically all techniques we need for the derivation of the greater depth formulas. In the sequel we will only carry the σ factors that survive conditions posed during the derivation.

The derivation of the depth 3 formula follows a similar but slightly more complicated path. Again, we first omit the signs of the indices and the correction terms for divergent integrals when we double or half the summation range. Then we present the complete formula. In the derivation we will be a bit shorter this time as the techniques are all similar to what we have shown above.

$$
\begin{aligned}
E(a,b,c) &= \frac{1}{2}(Z(a,b,c)+Z(-a,-b,c)) \\
&= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\sum_{x_3=1}^{\infty} \frac{1}{(x_1+x_2+x_3)^a\ (x_2+x_3)^b\ x_3^c} \frac{1+(-1)^{x_1}}{2} \\
&= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\sum_{x_3=1}^{\infty} \frac{1}{(2x_1+x_2+x_3)^a\ (x_2+x_3)^b\ x_3^c}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\sum_{x_3=1}^{\infty}\sum_{i=1}^{a} A_i^{(a,b)} \frac{1}{(2x_1+2x_2+2x_3)^{a+b-i}(2x_1+x_2+x_3)^i\, x_3^c} \\
&+ \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\sum_{x_3=1}^{\infty}\sum_{i=1}^{b} B_i^{(a,b)} \frac{1}{(2x_1+2x_2+2x_3)^{a+b-i}(x_2+x_3)^i\, x_3^c} \\
&= \sum_{i=1}^{b} B_i^{(a,b)} 2^{-a-b+i} Z(a+b-i,i,c) \\
&+ \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} Z(i,a+b-i,c) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} K_1^{(1)}(a+b-i,i,c) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} K_2^{(1)}(i,a+b-i,c)\ ,
\end{aligned} \tag{4.8}
$$

with the K functions given below. The full formula becomes

$$
\begin{aligned}
E(a,\sigma_b\, b,\sigma_c\, c) &= \frac{1}{2}(Z(a,\sigma_b\, b,\sigma_c\, c)+Z(-a,-\sigma_b\, b,\sigma_c\, c)) \\
&= \frac{1}{2}Z(-1)Z(\sigma_b b,\sigma_c c)\delta(a-1) \\
&- \frac{1}{2}Z(-2)Z(\sigma_c c)\delta_(a-1)\delta(\sigma_b b-1) \\
&+ \frac{1}{2}Z(-3)\delta(a-1)\delta(\sigma_b b-1)\delta(\sigma_c c-1) \\
&+ \sum_{i=1}^{b} B_i^{(a,b)} 2^{-a-b+i} Z(a+b-i,\sigma_b i,\sigma_c c) \\
&+ \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} Z(\sigma_b i,a+b-i,\sigma_c c) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} K_1^{(1)}(a+b-i,\sigma_b i,\sigma_c c) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} K_2^{(1)}(\sigma_b i,a+b-i,\sigma_c c)\ .
\end{aligned} \tag{4.9}
$$

The correction terms with the δ–functions are due to the halving of the summation range in the first step. The K–functions are given by

$$
\begin{aligned}
K_1^{(1)}(a,\sigma_b b,\sigma_c c) &= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty} \frac{\sigma_b^{2x_1+x_2}\sigma_c^{x_2}}{(x_1+x_2)^a\ (2x_1+x_2)^b\ x_2^c} \\
&= (-1)^b \sum_{i=1}^{a} A_i^{(a,b)} 2^{a-i} Z(i,\sigma_b\sigma_c(a+b+c-i))
\end{aligned}
$$

$$
\begin{aligned}
&+\quad (-1)^b \sum_{i=1}^{b} B_i^{(a,b)} 2^{a-1} (Z(i,\sigma_b\sigma_c(a+b+c-i)) \\
&\qquad\qquad +Z(-i,-\sigma_b\sigma_c(a+b+c-i))) \\
&+\quad (-1)^b B_1^{(a,b)} 2^{a-1} Z(-1) Z(\sigma_b\sigma_c(a+b+c-1))
\end{aligned}
\tag{4.10}
$$

$$
\begin{aligned}
K_2^{(1)}(\sigma_a a, b, \sigma_c c) &= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\sum_{x_3=1}^{\infty} \frac{\sigma_a^{x_1+2x_2+x_3}\sigma_c^{x_3}}{(x_1+2x_2+x_3)^a\ (x_2+x_3)^b\ x_3^c} \\
&= (-1)^c 2^{b-1} \sum_{i=1}^{c} B_i^{(b,c)} (-1)^i (Z(\sigma_a a,(b+c-i),\sigma_c i) \\
&\qquad +Z(\sigma_a a,-(b+c-i),-\sigma_c i)) \\
&- (-1)^c 2^{b-1} \sum_{i=1}^{b} A_i^{(b,c)} (Z(\sigma_a a,\sigma_c(b+c-i),i) \\
&\qquad +Z(\sigma_a a,\sigma_c(b+c-i),-i)) \\
&- (-1)^c 2^{b-1} \frac{(b+c-1)!}{(b-1)!\ c!} (Z(\sigma_a a,(b+c)) + Z(\sigma_a a,-(b+c)))
\end{aligned}
\tag{4.11}
$$

The last term in the function $K_1^{(1)}$ is also a correction term because we have to double the summation range on the Z-function of which the first index is one. Because the second index cannot be one in that case, we only need one correction term.

At depth 4 the relation becomes yet a bit more complicated but the derivation follows exactly the same path. We start with applying the non-trivial partial fractioning and then we have to try to rewrite the results in terms of Z–functions by percolating the factors two to the right. As there is one more sum this takes another step and we get two layers of K–functions:

$$
\begin{aligned}
E(a,\sigma_b\, b,\sigma_c\, c,\sigma_d\, d) &= \frac{1}{2}(Z(a,\sigma_b\, b,\sigma_c\, c,\sigma_d\, d) + Z(-a,-\sigma_b\, b,\sigma_c\, c,\sigma_d\, d)) \\
&= \frac{1}{2} Z(-1) Z(\sigma_b b,\sigma_c c,\sigma_d\, d))\delta(a-1) \\
&- \frac{1}{2} Z(-2) Z(\sigma_c c,\sigma_d\, d))\delta_(a-1)\delta(\sigma_b b-1) \\
&+ \frac{1}{2} Z(-3) Z(\sigma_d\, d))\delta_(a-1)\delta(\sigma_b b-1)\delta(\sigma_c c-1) \\
&- \frac{1}{2} Z(-4)\delta(a-1)\delta(\sigma_b b-1)\delta(\sigma_c c-1)\delta(\sigma_d d-1) \\
&+ \sum_{i=1}^{b} B_i^{(a,b)} 2^{-a-b+i} Z(a+b-i,\sigma_b i,\sigma_c c,\sigma_d\, d)) \\
&+ \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} Z(\sigma_b i,a+b-i,\sigma_c c,\sigma_d\, d)) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} K_1^{(1)}(a+b-i,\sigma_b i,\sigma_c c,\sigma_d\, d)) \\
&- \sum_{i=1}^{a} A_i^{(a,b)} 2^{-a-b+i} K_2^{(1)}(\sigma_b i,a+b-i,\sigma_c c,\sigma_d\, d))
\end{aligned}
\tag{4.12}
$$

$$
\begin{aligned}
K_1^{(1)}(a,\sigma_b b,\sigma_c c,\sigma_d d) &= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\sum_{x_3=1}^{\infty}\frac{\sigma_b^{2x_1+x_2+x_3}\sigma_c^{x_2+x_3}\sigma_d^{x_3}}{(x_1+x_2+x_3)^a\,(2x_1+x_2+x_3)^b\,(x_2+x_3)^c\,x_3^d}\\
&= (-1)^b\sum_{i=1}^{a}A_i^{(a,b)}2^{a-i}Z(i,\sigma_b\sigma_c(a+b+c-i),\sigma_d d)\\
&+ (-1)^b\sum_{i=1}^{b}B_i^{(a,b)}2^{a-1}(Z(i,\sigma_b\sigma_c(a+b+c-i),\sigma_d d)\\
&\qquad +Z(-i,-\sigma_b\sigma_c(a+b+c-i),\sigma_d d))\\
&+ (-1)^b B_1^{(a,b)}2^{a-1}Z(-1)Z(\sigma_b\sigma_c(a+b+c-1),\sigma_d d) \qquad (4.13)\\
K_2^{(1)}(\sigma_a a,b,\sigma_c c,\sigma_d d) &= \sum_{x_1=1}^{\infty}\cdots\sum_{x_4=1}^{\infty}\frac{\sigma_a^{x_1+2x_2+x_3+x_4}\sigma_c^{x_3+x_4}\sigma_d^{x_4}}{(x_1+2x_2+x_3+x_4)^a\,(x_2+x_3+x_4)^b\,(x_3+x_4)^c\,x_4^d}\\
&= (-1)^c2^{b-1}\sum_{i=1}^{c}B_i^{(b,c)}(-1)^i(Z(\sigma_a a,(b+c-i),\sigma_c i,\sigma_d d)\\
&\qquad +Z(\sigma_a a,-(b+c-i),-\sigma_c i,\sigma_d d))\\
&+ (-1)^c\sum_{i=1}^{b}A_i^{(b,c)}2^{b-i}Z(\sigma_a a,\sigma_c(b+c-i),i,\sigma_d d)\\
&- (-1)^c\sum_{i=1}^{b}A_i^{(b,c)}2^{b-i}K_1^{(2)}(\sigma_a a,\sigma_c\,(b+c-i),i,\sigma_d\,d))\\
&- (-1)^c\sum_{i=1}^{b}A_i^{(b,c)}2^{b-i}K_2^{(2)}(\sigma_a a,\sigma_c\,(b+c-i),i,\sigma_d\,d)) \qquad (4.14)\\
K_1^{(2)}(\sigma_a\,a,\sigma_b b,c,\sigma_d d) &= \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\sum_{x_3=1}^{\infty}\frac{\sigma_a^{x_1+2x_2+x_3}\sigma_b^{2x_2+x_3}\sigma_d^{x_3}}{(x_1+2x_2+x_3)^a\,(2x_2+x_3)^b\,(x_2+x_3)^c\,x_3^d}\\
&= (-1)^d\sum_{i=1}^{c}A_i^{(c,d)}2^{c-i}Z(\sigma_a\,a,\sigma_b\sigma_d\,(b+c+d-i),i)\\
&+ (-1)^d2^{c-1}\sum_{i=1}^{d}B_i^{(c,d)}(-1)^i(Z(\sigma_a\,a,(b+c+d-i),\sigma_b\sigma_d\,i)\\
&\qquad +Z(\sigma_a\,a,-(b+c+d-i),-\sigma_b\sigma_d\,i))\\
&- (-1)^d2^{c-1}\sum_{i=1}^{c}A_i^{(c,d)}(Z(\sigma_a\,a,\sigma_b\sigma_d\,(b+c+d-i),i)\\
&\qquad +Z(\sigma_a\,a,\sigma_b\sigma_d\,(b+c+d-i),-i))\\
&- (-1)^d2^{c-1}\frac{(c+d-1)!}{(c-1)!\,d!}(Z(\sigma_a\,a,(b+c+d))\\
&\qquad +Z(\sigma_a\,a,-(b+c+d))) \qquad (4.15)\\
K_2^{(2)}(\sigma_a a,\sigma_b b,c,\sigma_d d) &= \sum_{x_1=1}^{\infty}\cdots\sum_{x_4=1}^{\infty}\frac{\sigma_a^{x_1+x_2+2x_3+x_4}\sigma_b^{x_2+2x_3+x_4}\sigma_d^{x_4}}{(x_1+x_2+2x_3+x_4)^a\,(x_2+2x_3+x_4)^b\,(x_3+x_4)^c\,x_4^d}
\end{aligned}
$$

$$
\begin{aligned}
= \;& (-1)^d 2^{c-1}\sum_{i=1}^{d} B_i^{(c,d)}(-1)^i(Z(\sigma_a a,\sigma_b b,(c+d-i),\sigma_d i)\\
&\qquad +Z(\sigma_a a,\sigma_b b,-(c+d-i),-\sigma_d i))\\
+ \;& (-1)^d \sum_{i=1}^{c} A_i^{(c,d)} 2^{c-i} Z(\sigma_a a,\sigma_b b,\sigma_d(c+d-i),i)\\
- \;& (-1)^d 2^{c-1}\sum_{i=1}^{c} A_i^{(c,d)}(Z(\sigma_a a,\sigma_b b,\sigma_d(c+d-i),i)\\
&\qquad +Z(\sigma_a a,\sigma_b b,\sigma_d(c+d-i),-i)\\
- \;& (-1)^d 2^{c-1}\frac{(c+d-1)!}{(c-1)!\,d!}(Z(\sigma_a a,\sigma_b b,(b+c))\\
&\qquad +Z(\sigma_a a,\sigma_b b,-(b+c)))\ .
\end{aligned}
\tag{4.16}
$$

When we do depth 5 we see that, like $K_2^{(1)}$, also the $K_1^{(1)}$ splits off two new functions. Hence to produce a generic routine for any depth we have to look at a few very general steps.

In the general case the equations (4.12, 4.13) and (4.15) stay more or less the same. They just get more indices to the right. The difference comes with the equations for $K^{(2)}$. We have to make a distinction whether there are still many indices to the right or whether we are terminating. The terminating equations are also more or less the same as the equations for $K^{(2)}$ above, but now with more indices to the left. This leaves the 'intermediary' objects:

$$
K_1^{(i)}(M,\sigma_a a,b,\sigma_c c,N) = \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\frac{\sigma_a^{x_M+2x_1+x_2+x_N}\sigma_c^{x_2+x_N}}{(x_M+2x_1+x_2+x_N)^a(x_1+x_2+x_N)^b(x_2+x_N)^c}
\tag{4.17}
$$

$$
K_2^{(i)}(M,\sigma_a a,\sigma_b b,c,\sigma_d d,N) = \sum_{x_1=1}^{\infty}\sum_{x_2=1}^{\infty}\frac{\sigma_a^{x_M+2x_1+x_2+x_N}\sigma_b^{2x_1+x_2+x_N}\sigma_d^{x_2+x_N}}{(x_M+2x_1+x_2+x_N)^a(2x_1+x_2+x_N)^b(x_1+x_2+x_N)^c(x_2+x_N)^d}\ .
\tag{4.18}
$$

In these formulas M and N indicate a range of indices. There are more sums and factors in the numerator and denominator, but we just omit them as they do not take part in the 'action'. We use the same techniques applied before to move the factor 2 that multiplies x_1 to x_2, to the right. When N is empty we run into a termination condition and switch to the equations for $K^{(2)}$,

$$
\begin{aligned}
&K_1^{(i)}((M),\sigma_a a,b,\sigma_c c,(n_1,N)) =\\
&\quad (-1)^c\sum_{i=1}^{b} A_i^{(b,c)} 2^{b-i} Z(M,\sigma_a a,\sigma_c(b+c-i),i,n_1,N)\\
&\quad +(-1)^c\sum_{i=1}^{c} B_i^{(b,c)} 2^{b-1}(-1)^i(Z(M,\sigma_a a,(b+c-i),\sigma_c i,n_1,N)\\
&\qquad +Z(M,\sigma_a a,-(b+c-i),-\sigma_c i,n_1,N))
\end{aligned}
$$

$$
\begin{aligned}
&-(-1)^c \sum_{i=1}^{b} A_i^{(b,c)} 2^{b-i} K_1^{(i)}((M,\sigma_a a),\sigma_c(b+c-i),i,n_1,(N))\\
&-(-1)^c \sum_{i=1}^{b} A_i^{(b,c)} 2^{b-i} K_2^{(i)}((M),\sigma_a a,\sigma_c(b+c-i),i,n_1,(N)) \qquad (4.19)
\end{aligned}
$$

$$
\begin{aligned}
K_2^{(i)}&((M),\sigma_a a,\sigma_b b,c,\sigma_d d,(n_1,N)) =\\
&(-1)^d \sum_{i=1}^{c} A_i^{(c,d)} 2^{c-i} Z(M,\sigma_a a,\sigma_b\sigma_d(c+d-i),i,n_1,N)\\
&+(-1)^d \sum_{i=1}^{d} B_i^{(c,d)} 2^{c-1}(-1)^i (Z(M,\sigma_a a,\sigma_b(b+c+d-i),\sigma_d i,n_1,N)\\
&\quad +Z(M,\sigma_a a,-\sigma_b(b+c+d-i),-\sigma_d i,n_1,N))\\
&-(-1)^d \sum_{i=1}^{c} A_i^{(c,d)} 2^{c-i} K_1^{(i)}((M,\sigma_a a),\sigma_b\sigma_d(b+c+d-i),i,n_1,(N))\\
&-(-1)^d \sum_{i=1}^{c} A_i^{(c,d)} 2^{c-i} K_2^{(i)}((M,\sigma_a a,(b+c+d-i),\sigma_b\sigma_d i,n_1,(N)) \ . \qquad (4.20)
\end{aligned}
$$

As one can see, each step of the iteration diminishes N by one unit (n_1 is an index with its sign) and M may or may not get one more index.

The above formulas can be programmed rather easily and compactly in a language like `FORM`. We have first programmed and tested the cases 2, 3, 4, 5 and after that we have made a generic routine that can handle any depth. Also this routine has been tested exhaustively. It can be found in the library.

4.2 The Role of the Generalized Doubling Relations

Let us start with a modification of the program for expressing Euler sums into a minimal set that was used for testing `TFORM` [22]. It was modified, so as to allow running only with sums/functions up to a given depth. We use the same relations, up to that depth, as in the complete program, i.e. we use the stuffles, the shuffles and the doubling relations, but not the GDRs. This should generate new information because one is often interested in sums of limited depth but large weight.

When we compare the number of remaining variables with the conjectures [12, 13], we note that in many cases we have more variables left. However, if we increase the depth these remaining variables are eliminated after all. We set up the program in such a way that these objects may be recognized easily. In Table 1 we present how many of these constants are left and at which depth.

Table 1 indicates that there must be a significant ‘leaking’ of relations at greater depths that create nontrivial results at lower depth. As an example we derived the $\mathsf{d} = 2$ relation at weight 6 without substituting the lower weight constants and keeping track of all products of lower weight objects that combined in shuffles and in stuffles. The relation we refer to is given as Eq. (27) in Ref. [12] :

$$
Z_{-4,-2}(\infty) \quad = \quad -H_{-4,2}(1) = \frac{97}{420}\zeta_2^3 - \frac{3}{4}\zeta_3^2 \ . \qquad (4.1)
$$

weight	depth	number	type
6	2	1	$d=2$
6	3	1	$d=2$
6	4	0	
6	5	0	
6	6	0	
7	3	1	$d=3$
7	5	0	
8	2	1	$d=2$
8	4	2	$d=2$,$d=4$
8	6	0	
9	3	3	$3\times(d=3)$
9	5	2	$d=3$,$d=5$
9	7	0	
10	2	2	$2\times(d=2)$
10	3	2	$2\times(d=2)$
10	4	6	$2\times(d=2)$,$4\times(d=4)$
10	5	6	$2\times(d=2)$,$4\times(d=4)$
10	6	3	$d=2$,$d=4$,$d=6$
10	8	0	

Table 1: Number of constants remaining when running at fixed depth for a given weight. With fixed depth we mean all depths up to the given value.

depth	shuffles	stuffles
2	11	8
3	52	19
4	72	41

Table 2: Number of shuffles and stuffles separated by depth contributing to equation (4.1).

The results are shown in Table 2.

We see that a total of 203 equations make contributions to the final result. Considering this, it should not come as a great surprise that attempts to derive this equation by hand using shuffle and stuffle relations have failed thus far.

It is of course possible to obtain this result by different means as was shown in ref [26] where the finite harmonic sum $S_{-4,-2}(N)$ was calculated in terms of the following one-dimensional integral representation:

$$\begin{aligned} S_{-4,-2}(N) \quad = \quad & -\mathbf{M}\left[\left(\frac{4\mathrm{Li}_5(-\mathrm{x})-\ln(\mathrm{x})\mathrm{Li}_4(-\mathrm{x})}{x-1}\right)_+\right](N) \\ & +\frac{1}{2}\zeta_2\left[S_4(N)-S_{-4}(N)\right]-\frac{3}{2}\zeta_3 S_3(N)+\frac{21}{8}\zeta_4 S_2(N)-\frac{15}{4}\zeta_5 S_1(N)\ , \end{aligned} \tag{4.2}$$

where

$$\mathbf{M}[f(x)](N)=\int_0^1 dx\, x^N\, f(x)\ . \tag{4.3}$$

Since

$$\int_0^1 dx \frac{4\left[\mathrm{Li}_5(-\mathrm{x})+(15/16)\zeta_5\right]-\ln(x)\mathrm{Li}_4(-\mathrm{x})}{x-1} = -\frac{811}{840}\zeta_2^3+\frac{3}{4}\zeta_3^2 \tag{4.4}$$

one obtains with

$$Z_{-4,-2} \quad = \quad \lim_{N\to\infty} S_{-4,-2}(N)-\zeta_6 \tag{4.5}$$

the above result. It should, however, be clear that if such methods are needed to replace the phenomenon of leakage, it will be a near impossibility to go to much greater values of the weight parameter.

Using the GDRs at depth $\mathsf{d}=2$ resolves the problem completely. Only the depth $\mathsf{d}=2$ shuffles and stuffles in combination with these GDRs give already the desired formula.

To study the problem at depth $\mathsf{d}=3$, we recreated an old program by one of us[6] that only determines relations at leading depth for objects of which the index field is a Lyndon word. The `FORM` version of the program is rather fast when applied at depth `d = 3`, see Table 3.

We see a steady increase in the number of undetermined constants. In Tables 3, 4 we list under 'expected' the number of undetermined constants according to conjecture [12]. The results for the weights 7 and 9 are in agreement with the numbers in Table 1.

To see whether we could improve the situation, we tried programming generalizations of the formulas D_0 and D_1 of Ref. [48]. They made no difference. Close inspection reveals that the formula D_0 is another form of the shuffle formulas with the combinatorics included properly. The formula D_1, or Markett formula [49], also does not add anything new. It seems to be a combination of shuffles and stuffles. Next we applied the GDRs at depth $\mathsf{d}=3$ and these reduce the number of undetermined constants to their expected value. This means that if we include the GDRs we can run the program at maximum depth $\mathsf{d}=3$ and get a complete set of expressions for all depth $d=1,2$ and 3 objects. At the moment we have verified this for all weights up to $\mathsf{w}=51$. The run for the highest weight took about 20 hours of CPU time on a single Xeon processor at 2.33 GHz.

We have made a similar program for depth $\mathsf{d}=4$. This is of course much slower and hence we cannot go to such large values for the weight. The results are given in Table 4. Again we see an increase in the number of extra undetermined objects and again application of the GDRs resolved the issue.

The phenomenon of leakage is rather messy. Basically equations that are in nature of a greater depth have to combine first to eliminate most objects of this depth. After this a few equations remain between lower depth objects. Such leakage is impossible without the stuffle relations. The shuffle relations by themselves do not give terms with a lower depth and neither do the relations based on the doubling formula. But whether these extra relations come from the stuffles alone or materialize only after combining stuffles and shuffles, and maybe doublings, is currently not clear. What is clear is that they involve a very large number of equations. In all cases which we studied the leakage goes over at

[6]The program had an error and hence gave rise to a wrong conjecture.

weight	constants	expected
5	1	1
7	3	2
9	6	3
11	11	5
13	17	7
15	23	9
17	32	12
19	41	15
21	51	18
23	63	22
25	76	26
27	89	30
29	105	35
31	121	40
33	138	45
35	157	51
37	177	57
39	197	63
41	220	70
43	243	77
45	267	84
47	293	92
49	320	100
51	347	108

Table 3: Remaining constants at depth $\mathtt{d} = 3$ compared to the number of expected constants.

weight	constants	expected
6	1	1
8	3	2
10	9	5
12	21	8
14	39	14
16	66	20
18	102	30
20	149	40
22	209	55

Table 4: Remaining constants at depth $\mathtt{d} = 4$ compared to the number of expected constants.

weight	no doubling	no GDRs
8	1	0
10	1	0
11	2	1
12	3	1

Table 5: Number of excess elements when no doubling relations (also no GDRs) are used, and when only no GDRs are used.

least two units of depth. This makes it very difficult to investigate. Fortunately the GDRs seem to resolve these problems. We formulate

Conjecture 1: The stuffle, shuffle, doubling and Generalized Doubling Relations are sufficient to reduce the Euler sums of a given weight and depth to a minimal set that is in agreement with the conjecture [12], both in weight and in depth. □

Even if we could dispense with the GDRs up to weight $\mathsf{w} = 10$, the whole situation changes at weight $\mathsf{w} = 11$, see Table 5. Running only stuffles, shuffles and doubling relations leaves one variable in excess of the conjecture [12]. The GDRs provide the missing equation by which this variable is expressed in terms of the other remaining variables and agreement with conjecture [12] is reached. The same effect occurs at weight $\mathsf{w} = 12$. Again there is one variable too many if the GDRs are not used. We cannot check this beyond weight $\mathsf{w} = 12$, because leakage forces us to run all depths for a given weight if we exclude the GDRs. This becomes excessive in terms of current computer resources. Alternatively one could have used the relations of equation (2.24) to resolve this issue, but these relations do not help with the problem of running at a limited depth. Hence we have to add the GDRs anyway.

5 The Computer Program

We have combined the above relations into a new computer program to resolve all relations between MZVs and reduce them to a minimal set. In principle this is done by writing down all equations for the MZVs of a given weight and then solving the system. A few variables at the given weight may remain and there will be products of objects of lower weight.

Considering the size of the problem and its sparsity it did not look to us like a typical problem to solve by matrix techniques even though other people have done so [50, 51]. Typically there would be many thousands of zeroes for each non-zero element. The advantage of computer algebra is that in a sparse polynomial representation those zeroes will not be present and need no attention. Hence we have selected a rather special method the essence of which has already been used in references [10, 18, 22], although not described there in detail. We select the `FORM` system, because it is by far the best suited for this kind of problems. Since we go to much greater weights than previously investigated, we take the opportunity to give here a better description of the completely renewed version of the program.

We start generating a master expression which contains one term for each sum that we want to compute. For the MZVs of weight $w = 4$ this expression looks in computer terms like

```
FF =
   +E(0,0,0,1)*(H(0,0,0,1))
   +E(0,0,1,1)*(H(0,0,1,1))
   +E(0,1,0,1)*(H(0,1,0,1));
```

We have used already that we will only compute the finite elements and that there is a duality that allows us to eliminate all elements with a depth greater than half the weight. When the depth is exactly half the weight we choose from a sum and its dual the element that comes first lexicographically. We work in terms of the H-functions because for the Euler sums the basis of reference [12] turns out to be ideal. This basis consists of all Lyndon words of negative odd integers that add up in absolute value to the weight. For the MZVs these H-functions and the Z-functions are identical anyway and hence we could keep a single program for most procedures.

We pull the function E outside brackets. The contents of a bracket is what we know about the object indicated by the indices of the function E. In the beginning this is all trivial knowledge.

Assume now that we generate the stuffle relation

$$H_{0,1}H_{0,1} = H_{0,0,0,1} + 2H_{0,1,0,1} \tag{5.1}$$

The left hand side can be substituted from the tables for the lower weight MZVs. Hence it becomes ζ_2^2. The right hand side objects are replaced by the contents of the corresponding E brackets in the master expression. These are for now trivial substitutions. From the result we generate the substitution

```
id H(0,1,0,1) = z2^2/2-H(0,0,0,1)/2;
```

which we apply to the master expression. Hence the master expression becomes

```
FF =
   +E(0,0,0,1)*(H(0,0,0,1))
   +E(0,0,1,1)*(H(0,0,1,1))
   +E(0,1,0,1)*(z2^2/2-H(0,0,0,1)/2);
```

Let us now generate the corresponding shuffle relation:

$$H_{0,1}H_{0,1} = 4H_{0,0,1,1} + 2H_{0,1,0,1} \tag{5.2}$$

and replace the right hand side objects by the contents of the corresponding E brackets in the master expression. This gives

$$\zeta_2^2 = 4H_{0,0,1,1} + \zeta_2^2 - H_{0,0,0,1} \tag{5.3}$$

which leads to the substitution

```
id H(0,0,1,1) = H(0,0,0,1)/4;
```

and we obtain

```
FF =
   +E(0,0,0,1)*(H(0,0,0,1))
   +E(0,0,1,1)*(H(0,0,0,1)/4)
   +E(0,1,0,1)*(z2^2/2-H(0,0,0,1)/2);
```

We also need the divergent shuffles and stuffles. This is done by including the shuffles involving the basic divergent object and breaking down the multiple divergent sums with the stuffle relations as in:

$$\begin{aligned} H_1 H_{0,0,1} &= 2H_{0,0,1,1} + H_{0,1,0,1} + H_{1,0,0,1} \\ &= -H_{0,0,0,1} + H_{0,0,1,1} + H_{0,1,0,1} + H_1 H_{0,0,1}; \end{aligned} \tag{5.4}$$

In the case we use H_1 as the only divergent object, this is equivalent to using Hoffmann's [52] relation. We can use any combination involving divergent objects, provided not both are divergent simultaneously. Substituting from the master expression we get the relation

$$0 = -\frac{5}{4} H_{0,0,0,1} + \frac{1}{2}\zeta_2^2 \tag{5.5}$$

and hence the substitution

```
id H(0,0,0,1) = z2^2*2/5;
```

and finally the master expression becomes

```
FF =
   +E(0,0,0,1)*(z2^2*2/5)
   +E(0,0,1,1)*(z2^2/10)
   +E(0,1,0,1)*(z2^2*3/10);
```

Now we can read off the values of all MZVs of weight 4 that we set out to compute. All other elements can be obtained from these by trivial operations that involve the use of one or two relations only.

The method should be clear now: we generate the master expression that contains all nontrivial objects that we need to compute. Then we generate all known equations one by one, putting in the knowledge that is contained in the master expression. After that we incorporate the new knowledge in the master expression (provided the equation does not become trivial which will happen frequently, because we have more equations than variables).

With this method we do not need all equations to be in memory simultaneously. But there is a very important observation: the order in which the equations are generated will determine the size the master expression can have during the calculation. This intermediate expression swell should be controlled as much as possible, because it can make many orders of magnitude difference in the execution time and the space needed. And there

w/g	64	128	256	512	1024	2048	4096
9	62	56	61				
10		477	406	442			
11		5826	4651	3799	3623	5157	
12					65591	50926	62867

Table 6: Execution times in seconds for Euler sums at any depth as a function of weight and the size of the groups in the Gaussian elimination scheme. All runs were with TFORM on an 8 Xeon-cores machine at 3 GHz.

is another problem: substituting a new equation in the master expression can be rather costly when this expression becomes rather big. To have to do this each time is wasteful because the master expression will have to be brought to normal order again. Therefore we have adopted a scheme in which we generate the equations in groups. Then we apply first a Gaussian elimination scheme among the equations in the group, eliminating both above and below the diagonal. If we have G equations left we can substitute G variables in the master expression simultaneously. Again, this is not optimal yet as that would give G substitution statements and hence each term needs G pattern matchings. To improve upon this we enter these G objects in a temporary table and the substitution in the master expression is by a single table lookup. This is a binary search inside FORM and hence when we have grouped for instance 512 equations, the lookup takes only 9 compares, each of which is anyway much faster than a full pattern matching. The difference shows in a run we made on a machine with a single Opteron processor. When running the equations for MZVs one by one at weight 18, the run took 26761 sec, while with groups of 256 equations the same program ran in 2974 sec. Over the range in weights that we experimented with, the optimal group size we found for the MZVs was close to $2^{(w-1)/2}$. This is the value we use in the program. For the Euler sums the best value obeys a more involved relation because the number of variables goes with a power of three. We have measured the effect and it is shown in Table 6. From this Table it looks like a decent value for the size of the groups is $2^{3w/2-7}$ in which the exponent is rounded down to the nearest integer. We see, however, that the exact value is not very critical.

If it would be of great importance to improve over this scheme, one could set up a tree structure in the Gaussian scheme. This would change its quadratic (in the size of the groups) nature to a $G\log(G)$ behaviour. It would, however, make the code much more complicated and anyway, this is not where currently most computer time is used. As a consequence we decided to stay with the simple grouping.

This leaves determining a good order in which to generate the equations. It requires much trial and error and we are not claiming that we have the best scheme possible. The scheme for the stuffles is rather good, but for the shuffles it could probably be better. Once we could run what we wanted to run, we have stopped searching intensively. Anyway, the intermediate expression swell is rather moderate as is shown in the Tables containing the results below.

Before we discuss the order of the equations, we make several observations:

- Shuffles preserve depth.

- Stuffles either preserve depth or lower it.
- The number of indices that are one in sum notation is either preserved or lowered by stuffles.
- The shuffle relations can contain many more terms than the stuffle relations.
- The shuffles (which are executed in integral notation) can contain large combinatoric factors when there are long sequences of zeroes or ones. This lowers the number of terms in the equation.

Based on the above observations we start with the equations with the lowest depth, and then do the ones with the next depth, etc. In the case that we only look at the MZVs, we only need to go up to half the weight (rounded down), because the duality relation takes care of the other sums. In the case of the Euler sums we have to go ‘all the way’.

For each depth we do first the stuffles and then the shuffles. There are actually conjectures about that one does not need all stuffles but only a limited subset. We do not use these conjectures because they would make it necessary to apply more shuffle relations and those are more complicated than the stuffles that we would omit. We have verified experimentally that this would make the program significantly slower.

In the case of Euler sums we have two more categories of equations: the equations due to the doubling relation and the equations due to the GDRs. It looks like we do not need all equations from the latter category, but because they are not extremely costly, we have not been motivated enough to run many programs testing what can be done here. We just run them all and this way there is no risk that we omit something essential. They are, however, more costly than the shuffle equations and hence we put them after the shuffles. But more ordering within the group of (generalized) doubling equations is not relevant as there are only comparatively few substitutions generated by them.

To deal with the stuffles at a given weight and depth we generate an expression that contains one term for each stuffle relation that we will use. Then we apply several operations that multiply each term with a function with arguments based on the equation to be generated. The effect of this is that at the next sorting the equations will be ordered according to these arguments. This can be done in a rather flexible way. The ordering is in sum notation according to:

- The number of indices that are one.
- Next comes the number of indices that are two, then three etc.
- The number of indices in the sum with the smallest depth.
- The largest first index in either of the two sums.

This relatively simple ordering is amazingly effective. When we compute the size of the basis, using arithmetic over a 31-bit prime number, it gives a nearly monotonically increasing size in the master expression, indicating that it will be very hard to improve upon it. Once we have this expression we use a feature of `FORM` that allows one to define a loop in which the loop variable takes a value which is (sequentially) each time a term from a given expression. This way we can now create expressions for each equation and each time we have enough equations to fill a group we call the routine that will expand the equations and process them. We do not consider stuffle equations that contain a divergent sum. Those are taken into consideration anyway when we have to extract the divergences in the shuffle equations, and for the Euler sums the GDRs.

For the shuffles things are more complicated. Again we generate an expression for all shuffles for the given depth. In this case we generate however only those objects that correspond to shuffles in which one of the objects is only of depth one. This seems to be sufficient. We have never run across a case where the other shuffles had any additional effect. It is actually possible to restrict the number of shuffle equations even more, although this is only based on conjectures and experimentation. A formal proof is missing. The ordering is now done according to

- The weight of the object of depth one.
- The number of indices that are one in sum notation.
- For each sum we compute the sum of the squares of the indices in sum notation. We order by the maximum of either of the two. The biggest comes first.
- We select which of the two sums has the smallest first index. The larger values for this number come first.
- We add the first indices of the two sums. The larger values come first.

The complicating factor here is that we have to keep divergent sums. We only keep those equations in which at most one object is divergent, and there is only a single divergence. Hence sums that have the first two indices equal to one are not considered.

According to observation the shuffle equations that fulfill all following requirements always reduce to trivial ($0 = 0$) equations:

- The combined depth is at least three.
- There is at least one index that is equal to one.
- The depth one object has at least weight two.
- If the depth one object has weight $w = 2$, there are at least two indices equal to one in the other object.

Harmonic sums with all the same index decompose algebraically into a polynomial of single harmonic sums. It is easily shown that the algebraic relations [37] always allow to write any harmonic sum in terms of polynomials of $S_1(N)$ and sums, which converge in the limit $N \to \infty$. All the above greatly reduces the number of shuffle equations that have to be evaluated. Because this evaluation is one of the expensive steps, it speeds up the program significantly. On the other hand, it is only an observation made in runs that do

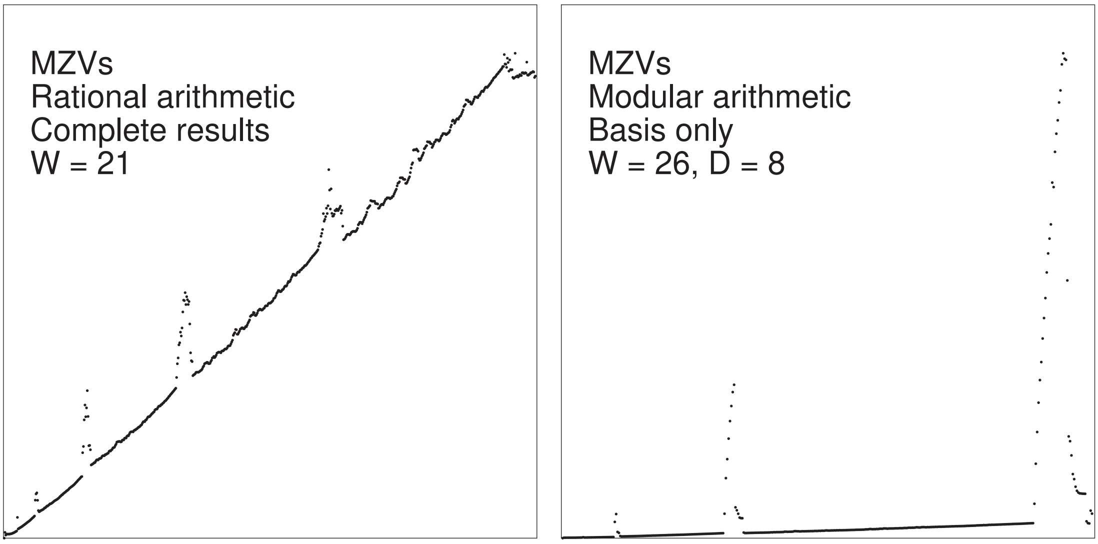

Figure 1: Performance of the program. On the *x*-axis we have the number of the module in which one group of equations is substituted and on the *y*-axis the size of the expression at the end of the module (arbitrary units). The spikes are due to the shuffles.

not involve the greatest weights. For the more critical runs[7] we have left these equations active and spent the extra computer time.

The above describes the basic program. At this point we split it in several varieties. To first determine whether shuffles and stuffles are sufficient to reduce all MZVs to a basis of the conjectured size, we have made the simplifications:

- All products of lower weight objects are set to zero. This means we will only determine whether reduction to a Lyndon basis takes place.

- We work modulus a 31-bit prime.

We have also made runs over the rational numbers. This becomes only problematic for the very highest values of the weight.

For constructing tables of all sums at a given weight we run the full program. The performance of the program is shown in Figure 1 for a complete run at weight $\mathsf{w} = 21$ and a run to depth $\mathsf{d} = 8$ at weight $\mathsf{w} = 26$. We see that the stuffles give a steady growth of the master expression but that the shuffles cause intermediate expression swell which is worse when the depth is much less than half the weight. The result is that when we run the complete system most time is spent with the stuffle relations while for the limited depth runs by far most time is spent with the shuffle relations.

In the case of Euler sums the master expression is created with a three letter alphabet $(-1,0,1)$ rather than the two letter alphabet (0,1) for the MZVs. In addition there are

[7]With this we mean the programs that determine the size of the basis when using arithmetic over a prime number. Once we have established this, any further runs to for instance determine all values over the rational numbers, we can safely drop these equations.

many more equations to consider because the number of lower weight objects that we can multiply either by shuffles or stuffles is correspondingly greater. Of course also for the Euler sums it is possible to just study the basis.

In addition it is possible to study sums to a limited depth. This way we can go to much greater values of the weight. This is of course only possible if we use a basis in which the concept of depth is relevant, like the basis of the odd negative indices that form a Lyndon word. Without such a basis the calculations become much harder.

When we are constructing tables we cannot go quite as far in weight as when we are determining rank deficiency. When we use a Lyndon basis, the majority of terms consists of products of basis elements of lower weights. This means that we have many more terms to carry around. We observe, in addition, that the coefficients containing the most digits are in the terms with powers of ζ_2. This is to be expected since ζ_2^N is our repository for all terms of the form $\zeta_{2a}^m\zeta_{2b}^n$ with $N = ma+nb$.

The representation we have selected, together with the modular arithmetic, makes for a very fast treatment of the terms. This is reflected in the number of terms that can be processed. In one run, which took more than 30 days the program generated a total of more than $7 \cdot 10^{12}$ terms. This seems to be a new record.

6 The Running of the Programs

We have used the programs of the previous Section to obtain results to as high a weight and depth as possible, both for MZVs and Euler sums. Before we start discussing these results we show the parameters of these runs to give the reader an impression of what is available and why there are limitations to obtain more.

We start with the Euler sums. We have first run the complete system for the given weights, see Table 7. This means that for $\mathsf{w} = 12$ there are expressions for all 236196 Euler sums with that weight, all expressed in terms of the basis of Lyndon words of the negative odd integers, see Appendix A, which is the basis we use for all Euler sums, unless mentioned differently.

The columns marked ‘variables’ mentions how many variables there are at the start of the program. ‘Remaining’ tells how many basis elements remain in the end. Under ‘output’ we give the size of the output expression in text format. The column ‘size’ refers to the largest size of the master expression during the calculation. Time refers to real time to run the program. If the column ‘CPU time’ is present it refers to the total CPU time by all processors. We notice that computer time is not the issue here, see Table 7[8]. The size of the results becomes the major problem. This is one of the reasons why we stopped at $\mathsf{w} = 12$. Technically the run at $\mathsf{w} = 13$ is feasible as it should take of the order of 10 days. The output is, however, projected at almost 8 Gbytes which we considered excessive.

We have also run programs that go to a maximum value of the depth. This involves only a subset of the Euler sums of that weight and hence such programs are much faster. As a consequence we can go to much greater values of the weight.

[8]The first time we ran the $\mathsf{w} = 12$ case on an 8-core Xeon machine at 2.33 GHz the run took two full weeks. It just shows how good a test case this problem is. Both `(T)FORM` and the MZV program have been improved greatly during this project.

w	variables	eqns	remaining	size	output	time
4	36	57	1	4.3K	2.0K	0.06
5	108	192	2	21K	8.9K	0.12
6	324	665	2	98K	42K	0.37
7	972	2205	4	472K	219K	1.71
8	2916	7313	5	2.25M	1.15M	7.78
9	8748	23909	8	11M	6.3M	50
10	26244	77853	11	58M	36M	353
11	78732	251565	18	360M	213M	3266
12	236196	809177	25	3.1G	1.29G	47311

Table 7: Runs on an 8-core Xeon computer at 3 GHz and with 32 Gbytes of memory. The column ‘eqns’ gives the number of equations that was considered.

weight	constants	running time [sec]	output [Mbyte]
9	956	7	0.26
10	1412	13	0.64
11	1996	24	1.25
12	2724	39	3.18
13	3612	68	5.04
14	4676	108	17.1
15	5932	199	17.1
16	7396	436	71.1
17	9084	602	54.9
18	11012	1323	275.9
19	13196	2761	157.1
20	15652	5424	877
21	18396	14090	395
22	21444	21875	2559

Table 8: Summary of the runs at $\mathsf{d} = 4$. The runs were performed on a computer with 8 Xeons at 3 GHz, using TFORM.

In Table 8 we show the statistics of the runs up to depth $\mathsf{d} = 4$. These are full runs in the sense that they are over the rational numbers and we have kept all terms, including the products of lower weight objects.

The dependence on the parity of the weight for the higher values is due to the fact that we run up to an even depth and the independent variables we use have an even depth for even weights and an odd depth for odd weights. This means for instance that the depth 4 objects for weight $\mathsf{w} = 17$ can all be expressed in terms of depth $\mathsf{d} = 3$ objects. The results for the depth 5 runs are summarized in Table 9.

We have a nice example here of what happens if we change the order in which we deal with the shuffles and the stuffles. We reran the program of Table 9 for the weights $\mathsf{w} = 14$ and $\mathsf{w} = 15$ under these conditions, obtaining running times of 100973 and 493489 sec respectively. This is more than an order of magnitude slower than the order we select in

weight	constants	remaining	running time [sec]	output [Mbyte]
9	3394	7	27	1.15
10	5702	7	72	3.11
11	9042	13	172	8.5
12	13686	11	478	20.9
13	19938	22	1330	68.9
14	28134	17	4306	133
15	38642	35	27607	473
16	51862	24	110336	688
17	68226	55	450462	2767

Table 9: Summary of the runs at $\mathtt{d} = 5$. Same computer as used in Table 8.

weight	constants	running time [sec]	output [Mbyte]
14	4676	35	1.3
16	7396	105	2.9
18	11012	323	6.0
20	15652	939	11.3
22	21444	2211	20.5
24	28516	5335	35
26	36996	13127	57
28	47012	47056	89
30	58692	100813	137

Table 10: Summary of the runs at $\mathtt{d} = 4$ in modular arithmetic, dropping all terms that are products of lower weight objects.

the regular programs.

Because we like to compare results of the MZV runs with those of the Euler runs to as high a weight as possible we made also runs in which we do all calculus modulus a 31-bit prime number. The number we selected is 2147479273. We never ran into a case in which this seemed to cause problems. In the programs in which we used this modulus we also dropped all terms that are products of lower weight objects. This means that in the end all sums are expressed into elements from the same-weight Lyndon part of the basis only. Such programs are much faster. This can be seen in Tables 10, 11 and 12 which are for depth $\mathtt{d} \leq 4$, depth $\mathtt{d} \leq 5$ and depth $\mathtt{d} \leq 6$, respectively.

The run at $\mathtt{w} = 18, \mathtt{d} = 6$ deserves some special attention. It was our most costly run and during the running `TFORM` processed more than $7 \cdot 10^{12}$ terms.

We come now to our runs for the Multiple Zeta Values. Those runs look more spectacular because there is much more literature on them. First we present the 'complete' runs in which all calculus is over the rational numbers and all terms are kept, cf. Table 13.

'Rat' is the real time of this run divided by the real time of a run with a 31-bit prime number dropping also products of lower weight objects. Together with the numbers in the 'num' column it shows that making several runs modulus a 31-bit prime and then using the Chinese remainder theorem [53], will not be efficient. We would need at least 12

weight	constants	running time [sec]	output [Mbyte]
13	16812	388	5.5
15	33388	2932	18
17	60044	18836	53
19	100236	118874	131
21	157932	554870	299

Table 11: Summary of the runs at $d = 5$ in modular arithmetic, dropping all terms that are products of lower weight objects.

weight	constants	remaining	running time [sec]	output [Mbyte]
13	56940	22	2611	
14	90564	37	12716	51
15	138636	35	55204	87
16	205412	66	206951	214
17	295916	55	789540	288
18	416004	109	2622157	711

Table 12: Summary of the runs at $d = 6$ in modular arithmetic, dropping all terms that are products of lower weight objects. Times refer to an 8 Xeon core machine at 3 GHz and 32 GBytes of memory.

w	d	G	size	output	num	CPU	real	Eff.	Rat.
16	8	128	11M	7M	22	289	56	5.16	0.99
17	8	256	30M	21M	19	677	129	5.25	0.96
18	9	256	88M	64M	29	3071	517	5.94	1.11
19	9	512	224M	182M	28	6848	1206	5.68	1.00
20	10	512	790M	558M	36	44883	6834	6.57	1.42
21	10	1024	1766M	1821M	40	86318	13851	6.23	1.12
22	11	1024	8856M	5927M	46	1572605	208972	7.53	3.18

Table 13: Runs on an 8-core Xeon computer at 3 GHz and with 32 Gbytes of memory. 'Num' indicates, for the final expressions, the maximum number of decimal digits in either a numerator or a denominator. 'Eff.' is the ratio of CPU time versus real time indicating how well the processors are used. The meaning of the column labeled 'Rat.' is explained in the text. The anomaly between size and output for $w = 21$ is due to the fact that the output is in text and size is in `FORM` binary notation.

w	G	size	output	CPU	real	Eff.
16	128	1.7M	1.2M	300	57	5.25
17	256	5.6M	3.2M	713	134	5.32
18	256	14.4M	7.2M	2706	465	5.82
19	512	39M	19M	6901	1206	5.72
20	512	104M	45M	30097	4819	6.25
21	1024	239M	114M	75302	12379	6.08
22	1024	767M	280M	449202	65644	6.84
23	2048	2.17G	734M	992431	151337	6.56
24	2048	8.04G	1.77G	9251325	1268247	7.29

Table 14: Runs on an 8-core Xeon computer at 3 GHz and with 32 Gbytes of memory. G is the size of the group used in the Gaussian elimination, 'size' is the maximum size of the master expression during the run, 'output' is the size of the master expression in the end, CPU is the total CPU time of all processors together in seconds, 'real' denotes the elapsed time in seconds and 'Eff.' is the pseudo efficiency, defined by the CPU time divided by the real time.

runs for the $\mathsf{w} = 22$ case and even then we have to account for dropping the lower weight terms.

We indicate the maximum value of the depth which, due to the duality relation for MZVs, is sufficient to obtain all MZVs at the given weight.

The basis in which these results are presented is described in Appendix B. If we let the program select the basis, the outputs are shorter but from the viewpoint of basis elements selected there is less structure.

The next sequence of runs is performed using in modular arithmetic in which we refer to the same 31-bit prime number as before. Again we run the full range of depths needed to obtain all sums. As usual in modular runs, we drop the products of lower weight objects. The results are given in Table 14.

The output of the run at $\mathsf{w} = 23$ gives the results for 2^{20} MZVs expressed in terms of the 28 same-weight elements of a Lyndon basis selected by the program.

In Table 15 we give the statistics of runs to a more restricted depth. If the conjecture [13] is correct the runs at $\mathsf{w} = 25, 26$ should still give us a complete basis. In the higher runs some elements will be missing.

We would have liked to have a run for depth $\mathsf{d} \leq 9$ at $\mathsf{w} = 27$, but it would probably take more than a year with current technology. A run for depth $\mathsf{d} \leq 8$ at $\mathsf{w} = 28$ will require a smaller CPU time. The reason why these runs are interesting is explained in Section 10 on pushdowns. They may give us a new type of basis element that would indicate a double pushdown.

The outputs of all of the above runs are collected in the data mine, together with some files in which the results have been processed to make them more accessible.

At the end of this Section we would like to discuss the status of the general investigation of MZVs and Euler sums in the foregoing literature. The relations between MZVs were studied both by mathematicians and physicists. An early study is due to Gastmans and Troost [54], which gave a nearly complete list for the Euler sums of $\mathsf{w} = 4$ and many

w	D	G	size	output	CPU	real	Eff.
23	7	2048	1.55G	89M	61447	9579	6.41
24	8	2048	673M	380M	536921	72991	7.36
25	7	4096	6.37G	244M	369961	50197	7.37
26	8	4096	38.3G	1160M	4786841	651539	7.35
27	7	6144	12.7G	914M	2152321	277135	7.77
28	6	6144	2.88G	314M	235972	30960	7.62
29	7	6144	41.0G	3007M	8580364	1112836	7.71
30	6	6144	6.27G	658M	829701	106353	7.80

Table 15: Runs on an 8-core Xeon computer at 3 GHz and with 32 Gbytes of memory. D indicates the maximum depth (see text). We reran at $\mathsf{w} = 23$ and $\mathsf{w} = 24$ to have information for extrapolation purposes.

relations for $\mathsf{w} = 5$, supplemented in [11] later. Various authors, among them D. Broadhurst, to $\mathsf{w} = 9$, and D. Zagier, performed precision numerical studies [55] using PARI [56] during the 1990's for MZVs, which were not published. A very far-reaching investigation concerned the study of some of the MZVs at $\mathsf{w} = 23$ and depth $\mathsf{d} = 7$ by Broadhurst by numerical techniques (PSLQ). Double sums were studied in [57] using the PSLQ method [15]. Vermaseren both studied the MZVs and the Euler sums to $\mathsf{w} = 9$ [10] using a FORM program [21]. This was the situation around the year 2000, when the Lille group presented their $\mathsf{w} = 12$ results for the MZVs and $\mathsf{w} = 7$ results for the Euler sums [58]. In Ref. [59] the solution of $\mathsf{w} = 8$ for the Euler sums is mentioned by the Lille–group. However, the data-tables made available [58] only contain the relations to $\mathsf{w} = 7$. Moreover, the relations used in [59] do not cover the doubling relation, which is needed to reduce to the conjectured basis at this weight, as will be shown later. For the MZVs $\mathsf{w} = 10$ had been solved in [60] and $\mathsf{w} = 13$ in [61], cf. [62]. Vermaseren could extend the MZVs to $\mathsf{w} = 16$ [63]. Studies for $\mathsf{w} = 16$ were also performed at Lille [64] without making the results public. In the studies by Vermaseren also the divergent harmonic sums $\zeta_{1,\vec{a}}$ were included, as this is sometimes necessary for physics applications, cf. also [11].

The primary goal in this paper is to derive explicit representations of the MZVs over several bases suitable to the respective questions investigated. If one only wants to determine the size of the basis one may proceed differently, cf. [50]. Here for $\mathsf{w} = 19$ in the MZV case it was shown, that the basis has the expected length, but modulo powers of π^2 at even weights. In [51] the case $\mathsf{w} = 20$ was studied determining the size of the basis calculating the rank of the associated matrix modulo a 15-bit prime. Although the computation times are not excessive, higher weights could not be investigated yet because of memory limitations. Since these methods are based on the respective algebra only they can be extended to colored multiple zeta values by extending the underlying alphabet.

7 The Data Mine

The results of our runs, together with a number of FORM programs to manipulate them and clarifying text, are available on the internet in pages that we call the MZV data mine. It can be located as a link in the `FORM` home page [65]. Here we will describe the notations and how to use the programs.

The notations we use in the data mine are that the MZVs are represented either by a function `Z` of which the variables are its indices or by a single symbol that consists of a string of objects of which the first character is the letter `z` and the remaining characters are decimal digits. Each of these strings refers to an index of the MZV. Let us give an example :

```
z11z3z3 = Z(11,3,3)
```

For the Euler sums we use mostly the function `H`. It can have positive and negative indices, the negative ones indicating alternating or Euler sums. When we use basis elements a compact notation is the letter `h` followed by a number of alphabetic characters or digits. Each character stands for a negative index. The digits $1,\cdots,9$ stand for the indices $-1,\cdots,-9$ and the upper case characters $A,\cdots,Z$ stand for the indices $-10,\cdots,-35$. We had no need to go further in this notation. The next example should illustrate this:

```
hL33 = H(-21,-3,-3).
```

If there is ever any doubt about which variable indicates which object one can look in the corresponding library file (always included as a file with the extension `.h` in the directory in which the integrals reside) in the procedure ‘frombasis’.

For reasons of economy[9] the `H`-functions with a single negative index have a different notation. They are related to the constants η_k defined by

$$\eta_k = \left(1 - \frac{1}{2^{k-1}}\right)\zeta_k\ . \tag{7.1}$$

In the program we call these constants `e3,e5,....`

In some cases we use a variable with a notation similar to the notation for the MZVs, except for that the character `z` is replaced by the character `a`.

```
aiajak = A(i,j,k)
       = Z(i,j,k)+Z(-i,j,-k)+Z(i,-j,-k)+Z(-i,-j,k)
       = H(i,j,k)-H(-i,j, k)-H(i,-j, k)+H(-i,-j,k)
```

Here A is the function defined in (10.3).

In exceptional cases we refer to Z-functions with negative indices. The most common notation for this in the literature is to put a bar over the number. This is however a notation that cannot be used in programs like `FORM`. Hence we use negative indices for the alternating sums there. For the symbolic variables we use the notation for the MZVs but with the character `m` between `z` and the number :

```
zm11zm3z3 = Z(-11,-3,3) = -H(-11,3,3)
```

[9] It turns out that the number of digits in the fractions is somewhat smaller in η-notation than in ζ-notation.

The programs run in what we call integral notation. This means that the master expression has the index fields of the functions `E`, `H` and `HH`[10] in terms of the three letter alphabet $\{0,1,-1\}$ for Euler sums and the two letter alphabet $\{0,1\}$ for MZVs. This is then the way the outputs are presented. Actually, internally the whole string of indices is put together as one large ternary number for Euler sums and one large binary number for MZVs. This speeds up the calculation, but makes it virtually impossible to interpret intermediate results.

The outputs are presented in a method that one may consider unusual. In `FORM` it is often more efficient to have one big expression, rather than 2^{20} expressions as would be the case for the MZVs at $\mathrm{w}=23$. Hence the output contains functions `H` with the indices of the corresponding MZV and each `H` is multiplied by what this MZV is equal to. In the case that we fixed a basis this can be an expression that consists of symbols like we defined above. In the case that we did the calculus modulus a prime and only wanted to determine a basis, it will be an expression that consists of terms that each contain a single function `HH` with its indices in integral notation. These `HH` functions form the basis. Often at the end of the program there is a list of the `HH` functions used. Because `FORM` will print the output in such a way that the functions `H` are taken outside brackets, the contents of each bracket are what each `H` function is equal to. With a decent editor it takes very few (≤ 4) edit commands to convert such output into the definition of 2^{20} table elements.

If this output should be used as input for other systems, this can be done, provided that the expressions do not cause memory problems. The format is in principle compatible with `Pari/GP`, `Reduce` and `Maple`. There may be a problem with large coefficients. `FORM` does not like to make output lines that are longer than a typical screen width. Hence they are usually broken up after some 75 characters. This holds also for long numbers. These are broken off by a backslash character and continued on the next line. The problem is usually that `FORM` places some white space at the beginning of the line and some programs may have problems with that. Hence one can use an editor to remove all white space (blanks and tabs) at the beginning of the lines.

The data mine consists of several parts. The main part is formed by the different data sets. The remainder files give information about how to use the data mine and links to other useful information and/or programs. The data are divided over a number of directories, each containing the results of one type of runs for a range of values of the weight. In each directory there are several types of files again. The log–files of the runs are stored. These contain the run time statistics and the output of the runs in text format. Then there are the table files. They are in text format and contain table definitions for FORM programs. Their extension is `.prc` as in `mzv21.prc`. Some of these files have been split into several files because they become much to big to be handled conveniently. These tables can be read and compiled. Yet the case of the MZVs at $\mathrm{w}=22$ with its nearly 6 Gbytes can be too large for a system with 'only' 16 Gbytes. If one does not have a bigger machine to ones disposal, one should use either the binary `.sav` file or the `.tbl` file defined below.

The third type of files are the binary `.sav` files. They can be used to read in the complete tables without having to go through the compiler and without having to load

[10]The function `HH` is the same as the function `H`. We need two different names because when we present the results the function `H` marks the brackets and the function `HH` marks the remaining basis elements.

the complete table as table elements (which needs also big compiler buffers). Finally we have created so-called tablebases which allow very fast access to individual elements. A tablebase is a type of database for large tables. They are particular to FORM and have been used with great success in a number of very large calculations. Their working is explained in Ref. [66] and the FORM manual. The tablebase files have traditionally the `.tbl` extension.

In each directory we have also the programs that were used to create the various files and in some cases some example programs.

There is another section in the data mine that contains pages in which it is explained how to manipulate the information in the files. Although many files are in text format it is not easy to manipulate a 4 Gbyte text file and hence it might become necessary to either use FORM and one of the binary files, or to use the STedi editor which has been used to manipulate these files on a computer with 16 Gbytes of main memory. Links are set to these programs. FORM programs are provided for the most common manipulations of the data. They contain much commentary. This should make it easy for the user to customize the programs should the need arise. The data mine is located at `http://www.nikhef.nl/~form/datamine/datamine.html`. Its structure is given in Figure 2 :

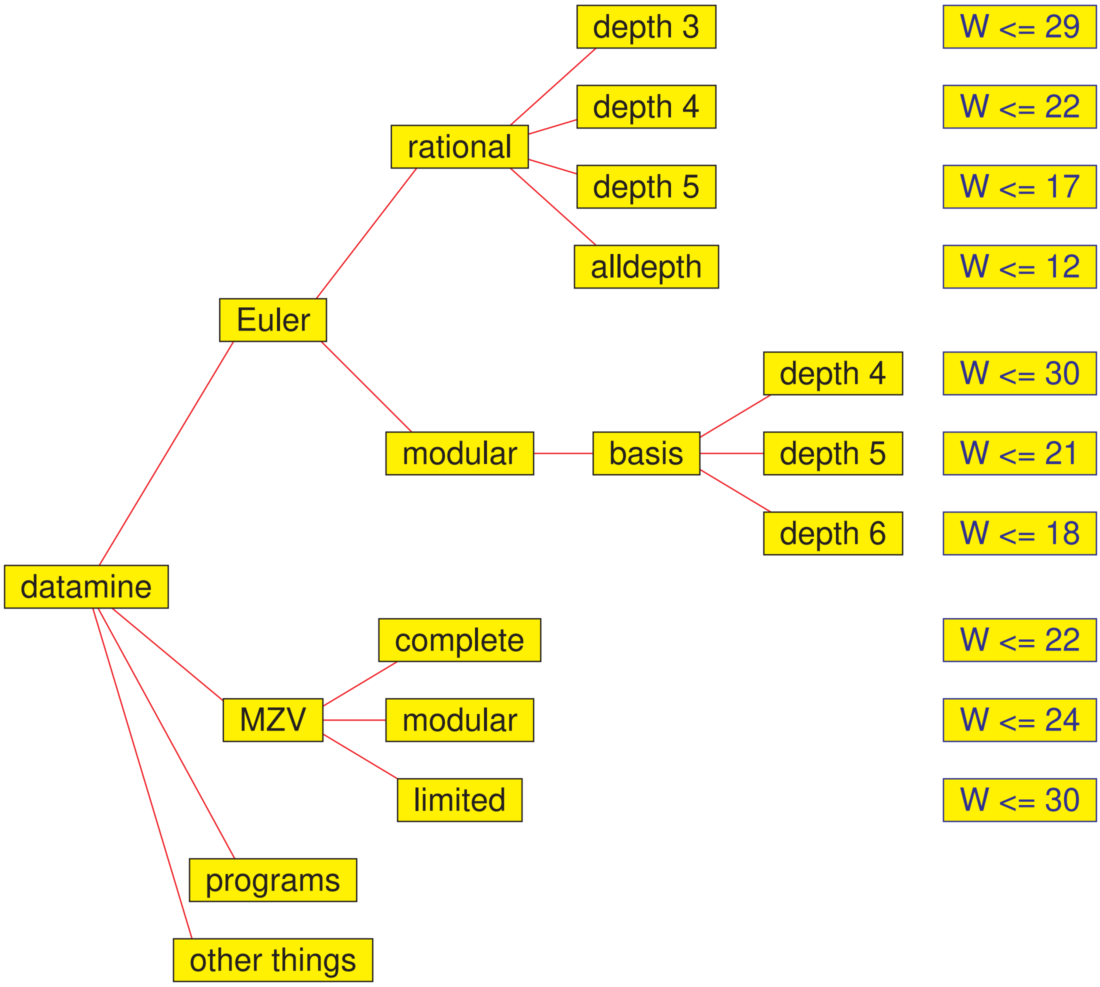

Figure 2: Layout of the data part of the data mine.

In this figure we use the following names:

complete	Complete expressions over the rational numbers.
modular	Products of lower weight terms are dropped and the computation is performed modulus a large prime.
limited	As modular but incomplete bases.
rational	Complete expressions over the rational numbers.
other things	Conventions, publications, help, links, etc.

The main problem with the data mine is its size. Many files are several Gbytes long. We have used `bzip2` on most files, because it gives a better compression ratio than `gzip`, even though it is much slower, both in compressing and decompressing. But even with `bzip2` the combined files are larger than 30 Gbytes.

All programs are `FORM` (or `TFORM`) codes. They will run with the latest versions of `FORM` (or `TFORM`). The executables of `FORM` can be obtained from the `FORM` web site: `http://www.nikhef.nl/∼form`. Please remember the license condition: if you use `FORM` (or `TFORM`) for a publication, you should refer to Ref. [21].

8 FORM Aspects

As mentioned the running of the programs used posed great challenges for `FORM` and `TFORM`. This is not simply a matter of whether the system contains errors. It is much more a matter of whether the system deals with the problem in a sensible and efficient way. Where are the bottlenecks? What is inefficient? A clear example is the conversion between sum notation and integral notation. This can be programmed in one line:

```
repeat id H(?a,n?!{-1,0,1},?b) = H(?a,0,n-sig_(n),?b);
```

for going to integral notation and

```
repeat id H(?a,0,n?!{0,0},?b) = H(?a,n+sig_(n),?b);
```

for going to sum notation. It turns out that when one goes to large weights (for instance more than 20), this becomes very slow because it involves very much pattern matching. Considering also that the use of harmonic sums is becoming more and more popular it was decided to built two new commands in `FORM` for this transformation:

```
ArgImplode,H;
ArgExplode,H;
```

The first one converts `H` to sum notation and the second one to integral notation. This made the program noticeably faster and easier to read.

Another addition to `FORM` concerns built-in shuffle and stuffle commands. One of the problems with shuffles is that the simple programming of it usually gives many identical terms. This means that the shuffle product of two MZVs can become very slow, which is illustrated by the following little program:

```
S   n1,n2;
CF  H,HH;
L   F = H(3,5,3)*H(6,2,5);
ArgExplode,H;
Multiply HH;
repeat;
  id  HH(?a)*H(n1?,?b)*H(n2?,?c) =
            +HH(?a,n1)*H(?b)*H(n2,?c)
            +HH(?a,n2)*H(n1,?b)*H(?c);
endrepeat;
id  HH(?a)*H(?b)*H(?c) = H(?a,?b,?c);
.end
```

```
Time =      37.38 sec    Generated terms =    2496144
               F         Terms in output =       2146
                         Bytes used      =      63176
```

By putting much combinatorics in the built-in shuffle statement we could solve most of these problems (although not all as the combinatorics can become very complicated). With the shuffle command the program becomes:

```
S   n1,n2;
CF  H,HH;
L   F = H(3,5,3)*H(6,2,5);
ArgExplode,H;
Shuffle,H;
.end
```

```
Time =       0.01 sec    Generated terms =       5163
               F         Terms in output =       2146
                         Bytes used      =      63176
```

This is a great improvement of course.

For the stuffle product things are much easier. There we have the complication that there are two definitions. One is the product used for the Z-sums and the other is the product used for the S-sums. We have resolved that by appending a + for the Z-notation and a - for the S-notation:

```
stuffle,Z+;
stuffle,S-;
```

Not only did this make the program significantly faster, it also made it more readable.

This way the stuffle product of two Euler sums in integral notation becomes in principle (assuming that we are in integral notation):

```
ArgImplode,H;
#call convertHtoZ(H,Z)
```

```
Stuffle,Z+;
#call convertZtoH(Z,H)
ArgExplode,H;
```

except for that in the actual program we substituted the contents of the two conversion procedures. Of course for MZVs the conversions are not needed and we can use just:

```
ArgImplode,H;
Stuffle,H+;
ArgExplode,H;
```

A third improvement concerns the parallelization. The original parallelization of TFORM [22] assumed the treatment of a single large expression of which the terms are distributed over the workers and later gathered in by the master. During the phase in which we execute a Gaussian elimination inside a group of identities, this is very inefficient, because we deal with many small expressions, each giving a certain amount of overhead when they are distributed over the worker threads. Hence it was decided to create a new form of parallelization in which the user tells the program that there are many small expressions coming. The reaction of the master thread is now to divide the expressions over the workers. It only has to tell each worker which expression to do next, after which the worker is responsible for obtaining its input and writing its output. The only remaining inefficiencies are that the writing of the output causes a traffic jam because that has to be done sequentially. The final results are kept in principle in a single file or its cached version. Additionally, there may be some load balancing problem in the end. This load balancing becomes rapidly less when the size of the groups of equations that is treated becomes bigger. The running of this phase of the program can give nearly ideal efficiencies.

A fourth improvement concerns the fact that very lengthy programs run a risk of discontinuity. This could be a power failure or a sudden urge of the service department to 'update' the system, etc. For this a facility has been implemented inside FORM that allows one to make 'snapshots' of the current internal state, cf. [67]. At a later moment one can then restart from the point of the snapshot. The completion of this facility came however too late to have a practical impact for this paper.

The possibility to perform the calculus modulus a prime number has existed in FORM since its first version. Much of it remained untested because these facilities had not been used extensively. It turned out to be necessary to redesign parts of it and add a few new features.

Other aspects of TFORM performed amazingly well. We have seen the program running with eight workers who all eight had to enter the fourth stage of the sorting simultaneously. This is rather rare even for single threads and only happens for very large expressions. It gives a bit of a slow down due to the great amount of disk accesses, but it all worked without any problems. The most impressive single module result observed was

```
Time =   15720.03 sec    Generated terms =1202653196013
             FF          Terms in output =   1508447974
 substitution(7-sh)-7621 Bytes used      =  36215474400
```

The execution time is that of the master. Actually the master spent 1000 CPU sec on this step and the eight workers each almost 200000 CPU sec.

One may wonder about the probability that calculations, done with a system under development, give correct answers. We have several remarks concerning this topic:

- Whenever `FORM` failed, it was always in a very obvious way, like crashing because it couldn't interpret something.
- The full all-depth outputs from the MZVs up to $\mathrm{w} = 22$ and the Euler sums up to $\mathrm{w} = 12$ have been tested numerically by completely independent programs, run under `PARI-GP` [56].
- Because of both `TFORM` and the MZV programs being under development many programs have been run at least several times with different configurations and/or different orders of solving the equations.
- `TFORM` operates in a rather non-deterministic fashion. Terms are rarely distributed twice in the same way over the workers because the master serves the workers when they have finished a task and this is usually not in the same order. In the case of errors this would lead to different results in different runs.
- There are effects that are expected on the basis of extrapolation, like the pushdowns and the construction of a basis. If anything goes wrong, such effects are absent.
- If for instance a term gets lost in a calculation over the rational numbers, usually the output would have terms with fractions that are abnormally much more complicated than the others. This is due to the fact that in intermediate stages the coefficients are usually much more complicated than at the end. Such terms are spotted relatively easily.

9 Results

Armed with the vast amount of information contained in the data mine we start with having a look at a number of conjectures in this this field. They concern the number of basis elements, either just as a function of the weight or as a function of weight and depth. We first check some conjectures made in the literature using the data mine and then describe the selection of the basis to represent the Euler sums and MZVs in the data mine.

9.1 Checking some Conjectures with the Data Mine

Zagier conjecture [2]:
The number of elements in a Lyndon-basis for the MZVs at weight w is given by Eq. (A.13). □
As far as we can check, the Zagier conjecture holds to weight 22. Assuming that in the modular calculus no terms were lost due to spurious zeroes, we can say that it holds to

weight 24. With the additional assumption that all (Lyndon) basis elements have a depth of at most one third of the weight we can even say that it holds to weight 26. If we combine the findings in the thesis of Racinet [68] that there may be 2 basis elements of depth 9 for weight 27 with our runs to depth 7, the Zagier conjecture holds also at weight 27. This conjecture is in accordance with the upper bound for the size of the basis being derived in Refs. [14].

Hoffman conjecture [69]:
A Fibonacci-basis for the MZVs at a given weight w is formed out of MZVs the index set of which is formed out of all words over the alphabet $\{2,3\}$. □
We could test the basis conjectured by Hoffman up to weight $\mathsf{w}=22$. If we take the subvariety in which we only look at the Lyndon words made from the indices 2 and 3, we can even verify this Lyndon basis to weight 24. Because this basis is not centered around the concept of depth, we cannot use the partial runs at larger weights and limited depths for further validation.

Broadhurst conjecture [12]:
The number of basis elements of the Euler sums at fixed weight w and depth d is given by Eq. (3.4). □
All our runs for Euler sums are in complete agreement with the Broadhurst conjecture about the size and the form of a basis for these sums. This means complete verification up to weight 12, for depth 6 verification (in modular arithmetic) to weight 18, for depth 5 complete verification to weight 17 and modular verification to weight 21. For depth 4 these numbers are weight 22 and weight 30 respectively.

Broadhurst-Kreimer conjectures [13]:
The number of basis elements of the MZVs at fixed weight w and depth d is given by Eq. (3.5). The number of basis elements for MZVs when expressed in terms of Euler sums in a minimal depth representation is given by Eq. (3.6) □
The runs for the MZVs confirm this conjecture over a large range, cf. Tables 16, 17. The second part of the conjecture is harder to check than the first part, because for this we need the results for the corresponding Euler sums.

Another conjecture by Hoffman [3]:

$$H_{2,1,2,3}-H_{2,2,2,2}-2H_{2,3,3} = 0 \tag{9.1}$$
$$H_{2,1,2,2,3}-H_{2,2,2,2,2}-2H_{2,2,3,3} = 0 \tag{9.2}$$
$$H_{2,1,2,2,2,3}-H_{2,2,2,2,2,2}-2H_{2,2,2,3,3} = 0 \tag{9.3}$$
$$H_{2,1,2,2,2,2,3}-H_{2,2,2,2,2,2,2}-2H_{2,2,2,2,3,3} = 0 \tag{9.4}$$
$$H_{2,1,\{2\}_k,3}-H_{\{2\}_{k+3}}-2H_{\{2\}_k,3,3} = 0 \quad \Box \tag{9.5}$$

We verified these relations up to weight $\mathsf{w}=22$. At $\mathsf{w}=24$ we checked the weight-24 part, since we have only the modular representation at this level.

There are identities for special patterns of indices as

$$2\zeta_{m,1}=m\zeta_{m+1}-\sum_{k=1}^{m-2}\zeta_{m-k}\zeta_{k+1},\ \ 2\leq m\ \in\ \mathbf{Z}\ , \tag{9.6}$$

w /d	1	2	3	4	5	6	7	8	9	10
1										
2	$\underline{1}$									
3	$\underline{1}$									
4										
5	$\underline{1}$									
6		$\underline{0}$								
7	$\underline{1}$									
8		$\underline{1}$								
9	$\underline{1}$		$\underline{0}$							
10		$\underline{1}$								
11	$\underline{1}$		$\underline{1}$							
12		$\underline{1}$		$\underline{1}$						
13	$\underline{1}$		$\underline{2}$							
14		$\underline{2}$		$\underline{1}$						
15	$\underline{1}$		$\underline{2}$		$\underline{1}$					
16		$\underline{2}$		$\underline{3}$						
17	$\underline{1}$		$\underline{4}$		$\underline{2}$					
18		$\underline{2}$		$\underline{5}$		$\underline{1}$				
19	$\underline{1}$		$\underline{5}$		$\underline{5}$					
20		$\underline{3}$		$\underline{7}$		$\underline{3}$				
21	$\underline{1}$		$\underline{6}$		$\underline{9}$		$\underline{1}$			
22		$\underline{3}$		$\underline{11}$		$\underline{7}$				
23	$\underline{1}$		$\underline{8}$		$\underline{15}$		$\underline{4}$			
24		$\underline{3}$		$\underline{16}$		$\underline{14}$		$\underline{1}$		
25	$\underline{1}$		$\underline{10}$		$\underline{23}$		$\underline{11}$			
26		$\underline{4}$		$\underline{20}$		$\underline{27}$		$\underline{5}$		
27	$\underline{1}$		$\underline{11}$		$\underline{36}$		$\underline{23}$		2	
28		$\underline{4}$		$\underline{27}$		$\underline{45}$		16		
29	$\underline{1}$		$\underline{14}$		$\underline{50}$		$\underline{48}$		7	
30		$\underline{4}$		$\underline{35}$		$\underline{73}$		37		2

Table 16: Number of basis elements for MZVs as a function of weight and depth in a minimal depth representation. Underlined are the values we have verified with our programs.

w/d	1	2	3	4	5	6	7	8	9	10
1										
2	$\underline{1}$									
3	$\underline{1}$									
4										
5	$\underline{1}$									
6										
7	$\underline{1}$									
8		$\underline{1}$								
9	$\underline{1}$									
10		$\underline{1}$								
11	$\underline{1}$		$\underline{1}$							
12		$\underline{2}$								
13	$\underline{1}$		$\underline{2}$							
14		$\underline{2}$		$\underline{1}$						
15	$\underline{1}$		$\underline{3}$							
16		$\underline{3}$		$\underline{2}$						
17	$\underline{1}$		$\underline{5}$		$\underline{1}$					
18		$\underline{3}$		$\underline{5}$						
19	$\underline{1}$		$\underline{7}$		3					
20		$\underline{4}$		8		1				
21	$\underline{1}$		$\underline{9}$		7					
22		$\underline{4}$		14		3				
23	$\underline{1}$		12		14		1			
24		$\underline{5}$		20		9				
25	$\underline{1}$		15		25		4			
26		$\underline{5}$		30		20		1		
27	$\underline{1}$		18		42		12			
28		$\underline{6}$		40		42		4		
29	$\underline{1}$		22		66		30		1	
30		6		55		75		15		

Table 17: Number of basis elements for MZVs as a function of weight and depth when expressed as Euler sums in a minimal depth representation. Underlined are the values we have verified with our programs.

cf. [1, 4] or

$$\zeta_{\{3,1\}_n} = \frac{1}{2n+1}\zeta_{\{2\}_n} = \frac{1}{4^n}\zeta_{\{4\}_n} = \frac{2\pi^{4n}}{(4n+2)!}\ , \tag{9.7}$$

conjectured in [2] and proven in [47]. Another relation is

$$\zeta_{2,\{1,3\}_n} = \frac{1}{4^n}\sum_{k=0}^{n}(-1)^k\zeta_{\{4\}_{n-k}}\left\{(4k+1)\zeta_{4k+2} - 4\sum_{j=1}^{k}\zeta_{4j-1}\zeta_{4k-4j+3}\right\} \tag{9.8}$$

conjectured in [19] and proven in [70]. For the Euler sums one finds, [71],

$$\zeta_{\{3\}_n} = 8^n\zeta_{\{-2,1\}_n}\ . \tag{9.9}$$

In Ref. [19] conjectures were given for special cases based on PSLQ,

$$\zeta_{\{4,1,1\}_2} = \frac{3\pi^4}{16}\left[\zeta_{6,2} - 4\zeta_5\zeta_3\right] - \frac{41\pi^6}{5040}\left[\zeta_3^2 - \frac{77023\pi^6}{14414400}\right] + \frac{397}{8}\zeta_9\zeta_3 + \zeta_3^4 \tag{9.10}$$

$$\zeta_{2,2,1,2,3,2} = \frac{75\pi^2}{32}\left[\zeta_{8,2} - 2\zeta_7\zeta_3 + \frac{34}{225}\zeta_5^2 + \frac{4528801\pi^{10}}{61297236000}\right] - \frac{825}{8}\zeta_7\zeta_5\ , \tag{9.11}$$

which we verified. A series of special relations for the Euler sums were conjectured in [19] based on PSLQ :

$$\zeta_{2,1,-2,-2} = \frac{39}{128}\zeta_4\zeta_3 - \frac{193}{64}\zeta_5\zeta_2 + \frac{593}{128}\zeta_7 \tag{9.12}$$

$$\zeta_{-2,-2,1,2} = \frac{9}{128}\zeta_4\zeta_3 + \frac{447}{128}\zeta_5\zeta_2 - \frac{1537}{256}\zeta_7 \tag{9.13}$$

$$\begin{aligned}\zeta_{\{-3,1\}_2} = & -7\left[\alpha_5 - \frac{39}{64}\zeta_5 + \frac{1}{8}\zeta_4\ln(2)\right]\zeta_3 + \left[2\alpha_4 - \frac{1}{4}\zeta_4\right]^2 \\ & +2\left[\alpha_4 - \frac{15}{16}\zeta_4 + \frac{7}{8}\zeta_3\ln(2)\right]^2 - \frac{1}{32}\zeta_8\ .\end{aligned} \tag{9.14}$$

Here

$$\alpha_n = \mathrm{Li_n}(1/2) + (-1)^{\mathrm{n}}\left[\frac{\ln^{\mathrm{n}}(2)}{\mathrm{n}!} - \frac{\zeta_2}{2}\frac{\ln^{(\mathrm{n}-2)}(2)}{(\mathrm{n}-2)!}\right]\ . \tag{9.15}$$

These relations are verified analytically as well by our data base. Relations (9.10–9.13) were also obtained in [58].

In Ref. [12] a series of relations was conjectured for weight $\mathsf{w} = 8\ ...12$ and $\mathsf{d} = 3{,}4$ for Euler sums being related to values $\zeta_{-|a_1|,-|a_2|}$.

$$\zeta_{3,-3,-3} = 6\zeta_{5,-1,-3} + 6\zeta_{3,-1,-5} - \frac{315}{32}\ln(2)\zeta_3\zeta_5 + 6\zeta_{-5,-1}\zeta_3$$

$$
\begin{aligned}
&+ \quad \frac{40005}{128}\zeta_2\zeta_7 - \frac{39}{64}\zeta_3^3 + \frac{1993}{256}\zeta_3\zeta_6 + \frac{8295}{128}\zeta_4\zeta_5 - \frac{226369}{384}\zeta_9,
\end{aligned} \tag{9.16}
$$

$$
\begin{aligned}
\zeta_{3,-5,-3} &= \frac{1059}{80}\zeta_{5,3,3} + 15\zeta_{7,-1,-3} + 15\zeta_{3,-1,-7} + \frac{701}{69}\zeta_{-5,-3}\zeta_3 \\
&+ 15\zeta_{-7,-1}\zeta_3 - \frac{6615}{256}\ln(2)\zeta_3\zeta_7 - \frac{11852967}{2560}\zeta_{11} + \frac{301599}{128}\zeta_2\zeta_9 \\
&- \frac{124943}{5888}\zeta_3^2\zeta_5 + \frac{1753577}{35328}\zeta_3\zeta_8 + \frac{2960103}{5120}\zeta_4\zeta_7 + \frac{3405}{32}\zeta_5\zeta_6,
\end{aligned} \tag{9.17}
$$

$$
\begin{aligned}
\zeta_{3,-1,3,-1} &= \frac{61}{27}\zeta_{-3,-3,-1,-1} - \frac{14}{3}\zeta_{-5,-1,-1,-1} - \frac{185}{27}\zeta_{-5,-1}\zeta_2 \\
&- \frac{163499}{22356}\zeta_{-5,-3} + \frac{2051}{54}\zeta_{-7,-1} + \frac{28}{9}\ln^2(2)\zeta_{-5,-1} + \frac{35}{96}\ln^2(2)\zeta_3^2 \\
&- \frac{581}{64}\ln^2(2)\zeta_6 - \frac{8735}{576}\ln(2)\zeta_2\zeta_5 - \frac{903}{64}\ln(2)\zeta_3\zeta_4 \\
&- \frac{1441}{288}\zeta_2\zeta_3^2 + \frac{10365875}{476928}\zeta_3\zeta_5 + \frac{36916435}{1907712}\zeta_8.
\end{aligned} \tag{9.18}
$$

$$
\begin{aligned}
2^5\cdot 3^3\zeta_{4,4,2,2} &= 2^5\cdot 3^2\zeta_3^4 + 2^6\cdot 3^3\cdot 5\cdot 13\zeta_9\zeta_3 + 2^6\cdot 3^3\cdot 7\cdot 13\zeta_7\zeta_5 \\
&+2^7\cdot 3^5\zeta_7\zeta_3\zeta_2 + 2^6\cdot 3^5\zeta_5^2\zeta_2 - 2^6\cdot 3^3\cdot 5\cdot 7\zeta_5\zeta_4\zeta_3 \\
&-2^8\cdot 3^2\zeta_6\zeta_3^2 - \frac{13177\cdot 15991}{691}\zeta_{12} \\
&+2^4\cdot 3^3\cdot 5\cdot 7\zeta_{6,2}\zeta_4 - 2^7\cdot 3^3\zeta_{8,2}\zeta_2 - 2^6\cdot 3^2\cdot 11^2\zeta_{10,2} \\
&+2^{14}\zeta_{-9,-3}\ .
\end{aligned} \tag{9.19}
$$

These relations were verified using the current data base. Eq. (9.19) is particularly interesting since it implies a relation between MZVs mediated by one term of the kind $\zeta_{-|a_1|,-|a_2|}$.

There is a series of Theorems proven on the MZVs, which can be verified using the data base. We used already the duality theorem [2]. For the MZVs a large variety of relations has been proven, which can be verified for specific examples using the data mine.

The first of these general relations is the **Sum Theorem**, Ref. [1, 72],

$$
\sum_{i_1+\ldots+i_k=n,\, i_1>1} \zeta_{i_1,\ldots,i_k} = \zeta_n\ . \tag{9.20}
$$

The sum-theorem was conjectured in [40], cf. [39]. For its derivation using the Euler connection formula for polylogarithms, cf. [73].

Further identities are given by the **Derivation Theorem**, [40, 52] Let $I = (i_1,\ldots,i_k)$ any sequence of positive integers with $i_1 > 1$. Its derivation $D(I)$ is given by

$$
\begin{aligned}
D(I) &= (i_1+1, i_2,\ldots,i_k) + (i_1, i_2+1,\ldots,i_k) + \ldots (i_1,i_2,\ldots,i_k+1) \\
\zeta_{D(I)} &= \zeta_{(i_1+1,i_2,\ldots,i_k)} + \ldots + \zeta_{(i_1,i_2,\ldots,i_k+1)}\ .
\end{aligned} \tag{9.21}
$$

The Derivation Theorem states

$$
\zeta_{D(I)} = \zeta_{\tau(D(\tau(I)))}\ . \tag{9.22}
$$

Here τ denotes the duality-operation (2.21). We call an index-word w admissible, if its first letter is not 1. The words form the set $\mathfrak{H}^0$. $|w|$ = w is the weight and $d(w)$ the depth of w. For the MZVs the words w are build in terms of concatenation products $x_0^{i_1-1}x_1x_0^{i_2-1}x_0...x_0^{i_k-1}x_1$. The height of a word, $\mathrm{ht}(w)$, counts the number of (non-commutative) factors $x_0^a x_1^b$ of w. The operator D and its dual $\overline{D}$ act as follows [7],

$$Dx_0 = 0, \quad Dx_1 = x_0x_1, \quad \overline{D}x_0 = x_0x_1, \quad \overline{D}x_1 = 0\ .$$

Define an anti-symmetric derivation

$$\partial_n x_0 = x_0(x_0+x_1)^{n-1}x_1\ .$$

A generalization of the Derivation Theorem was given in [52, 74] :
The identity

$$\zeta(\partial_n w) = 0 \tag{9.23}$$

holds for any $n \geq 1$ and any word $w \in \mathfrak{H}^0$. Further theorems are the **Le–Murakami Theorem**, [75], the **Ohno Theorem**, [76], which generalizes the sum- and duality theorem, the **Ohno–Zagier Theorem**, [77], which covers the Le–Murakami theorem and the sum theorem, and generalizes a theorem by Hoffman [39, 40], and the **cyclic sum theorem**, [78].

Finally, we mention a main conjecture for the MZVs. Consider tuples $\mathbf{k} = (k_1,\ldots,k_r) \in \mathbb{N}^r, k_1 \geq 1$. One defines

$$\begin{aligned} \mathcal{Z}_0 &:= \mathbb{Q} \\ \mathcal{Z}_1 &:= \{0\} \\ \mathcal{Z}_w &:= \sum_{|\mathbf{k}|=w} \mathbb{Q}\cdot\zeta(\mathbf{k}) \subset \mathbb{R}\ . \end{aligned} \tag{9.24}$$

If further

$$\mathcal{Z}^{\mathrm{Go}} := \sum_{w=0}^{\infty} \mathcal{Z}_w \subset \mathbb{R} \qquad (\text{Goncharov}) \tag{9.25}$$

$$\mathcal{Z}^{\mathrm{Ca}} := \bigoplus_{w=0}^{\infty} \mathcal{Z}_w \qquad (\text{Cartier}) \tag{9.26}$$

the conjecture states
(a) $\mathcal{Z}^{\mathrm{Go}} \cong \mathcal{Z}^{\mathrm{Ca}}$. There are no relations over $\mathbb{Q}$ between the MZVs of different weight w.
(b) $\dim \mathcal{Z}_w = d_w$, with $d_0 = 1, d_1 = 0, d_2 = 1, d_w = d_{w-2} + d_{w-3}$.
(c) All relations between MZVs are given by the extended double-shuffle relations [79], cf. also [80]. If this conjecture turns out to be true all MZVs are irrational numbers.

9.2 Selection of a Basis

Thus far we have not specified which basis we have been using for the MZVs. In first instance, we actually let the program select the basis. The result was the collection of remaining elements after elimination of as many elements as possible. The ordering in the elimination process was such that the remaining elements would be minimal in depth and maximal in their sum notation. Hence $Z_{20,2,1,1}$ would be preferred over $Z_{18,4,1,1}$. As it turned out, all remaining elements had an index field which formed a Lyndon word. This is not really surprising due to the ordering. Unfortunately there was not much systematics found in these elements.

Next came the idea that if the Euler sums have a basis made out of Lyndon words of only negative odd indices, maybe one should investigate to which extent one can write a basis for the MZVs in terms of Lyndon words with positive odd indices only. It turns out that a number of elements can be selected with odd-only indices, but it is not possible for the whole basis. A number of basis elements needs at least two even indices.

Definition.
L_w is the set of Lyndon words made out of positive odd-integer indices, with no index $i = 1$ at given weight w. $\Box$

We observed that Table 17 can be reproduced by basis elements with indices in L_w. As mentioned, this is not a basis for the MZVs, but if we write as many elements of the basis as possible as elements of the set L_w, the remaining elements of the basis have a depth that is at least two greater than the elements that are remaining in the L_w set and need at least two even indices. Additionally, it looks like that they can be written as an extended version of these remaining elements by adding two indices 1 at the end and subtracting one from the first two indices as in

$$Z_{7,5,3} \quad \to \quad Z_{6,4,3,1,1} \ . \tag{9.27}$$

We have been able to construct bases with these properties all the way up to weight $\mathrm{w} = 26$. The complete (non-unique) recipe for such bases is:

1. Construct the set L_w of all Lyndon words of positive odd integers excluding one that add up to w.
2. Starting at lowest depth, write as many basis elements of the basis as possible in terms of elements of L_w. Call the remaining elements in L_w at this depth $R_W^{(D)}$.
3. At the next depth, two units larger than the previous one, write again as many basis elements of the basis as possible in terms of elements of L_w and construct $R_W^{(D+2)}$.
4. Write the elements of the basis with depth $D+2$ that could not be written as elements of L_w as 1-fold extended elements of $R_W^{(D)}$.
5. Write the elements of the basis with depth $D+2$ that could not be written as elements of L_w or 1-fold extended elements of $R_W^{(D)}$ as 2-fold extended elements of what remains of $R_W^{(D-2)}$, etc.
6. If we are not done yet, raise D by two and go back to step 3.

The concept of n-fold extension is defined by subtracting one from the first $2n$ indices and adding $2n$ indices with the value one at the end of the index set.

To illustrate this we give two examples. First the basis at weight $\mathrm{w} = 12$:

$$
\begin{array}{lll}
L_{12}: & H_{9,3} & H_{7,5} \\
P_{12}: & H_{9,3} & H_{6,4,1,1}
\end{array}
$$

and next the basis at weight $\mathrm{w} = 18$:

$$
\begin{array}{lllllllll}
L_{18}: & H_{15,3} & H_{13,5} & H_{11,7} & H_{9,3,3,3} & H_{7,5,3,3} & H_{7,3,5,3} & H_{7,3,3,5} & H_{5,5,5,3} \\
P_{18}: & H_{15,3} & H_{13,5} & H_{10,6,1,1} & H_{9,3,3,3} & H_{6,4,3,3,1,1} & H_{7,3,5,3} & H_{7,3,3,5} & H_{5,5,5,3}
\end{array}
$$

From the basis at weight 18 it should be clear why we put so much effort in obtaining the results for the Euler sums at weight 18, depth 6.

Because the construction does not tell which elements of L_w to select the results are not unique. In fact quite a few selections are not possible because of dependencies between the elements of L_w. Hence the whole procedure requires a certain amount of experimenting before a good basis is found. In Appendix B we have tried to find a basis in which the elements that are taken from L_w have the highest values when their index set is seen as a multi-digit number. Because of reasons being explained in the next Section we call these bases ‘pushdown bases’.

We do not have complete runs for the weights $\mathrm{w} = 27$ and $\mathrm{w} = 28$. In these cases the elements with the greatest depth are missing. But we can go through the construction as far as possible and make predictions about the missing elements. It turns out that for both these weights a 2-fold extension is needed. For weight $\mathrm{w} = 27$ this would be for depth 5 to depth 9 and for weight $\mathrm{w} = 28$ for depth 4 to depth 8. This concept was not taken into account in the conjectures in Ref. [13]. Hence we formulate a new conjecture that not only specifies the number of elements for each weight and depth but also how many elements need how many extensions.

Conjecture 2.
The number of basis elements $D(w,d,p)$ of MZVs with weight w, depth d, and pushdown p is obtained from the generating function

$$
\prod_{w=3}^{\infty}\prod_{d=1}^{\infty}\prod_{p=0}^{\infty}(1-x^w y^d z^p)^{D(w,d,p)} \quad = \quad 1-\frac{x^3 y}{1-x^2}+\frac{x^{12}y^2(1-y^2 z)}{(1-x^4)(1-x^6)} \tag{9.28}
$$

solving for the coefficients of the monomials $x^w y^d z^p$. □

This formula predicts the first n-fold extension ($n > 1$) at weight $w = 12n+3$ and it will be to depth $d = 4n+1$. The exception is the first extension at weight 12. We show this in Table 18.

It is a great pity that with the resources that were at our disposal we just could not get direct access to a double extension or pushdown. Extrapolating from the numbers in Table 15 indicates computer times of the order of half a year (for weight 28, depth 8) to more than a year (for weight 27, depth 9).

10 Pushdowns

As mentioned in the previous Section, there are elements that as MZVs can only be written with a certain depth, while, when written in terms of Euler sums, can be written with a

w/d	1	2	3	4	5	6	7	8	9	10
1										
2	1									
3	1									
4										
5	1									
6		0								
7	1									
8		1								
9	1		0							
10		1								
11	1		1							
12		1		0,1						
13	1		2							
14		2		1						
15	1		2		0,1					
16		2		2,1						
17	1		4		1,1					
18		2		4,1		0,1				
19	1		5		3,2					
20		3		6,1		1,2				
21	1		6		6,3		0,1			
22		3		10,1		3,4				
23	1		8		11,4		1,3			
24		3		14,2		8,6		0,1		
25	1		10		18,5		4,7			
26		4		19,1		16,11		1,4		
27	1		11		29,7		11,12		0,1,1	
28		4		25,2		31,14		4,11,1		
29	1		14		42,8		25,23		1,5,1	
30		4		33,2		52,21		14,22,1		0,1,1

Table 18: Number of basis elements for MZVs as a function of weight, depth and extension(or pushdown). If there are several numbers, separated by commas, the first indicates the number of elements that came from L_w, the second the number of 1-fold extensions from depth $d-2$, the third the number of 2-fold extensions from depth $d-4$, etc. A single number refers to the elements of L_w.

smaller depth. This phenomenon is called pushdown. The simplest example occurs at weight $\mathrm{w} = 12$ and can be looked up in the Tables for the Euler sums. It is

$$\begin{aligned}
Z_{6,4,1,1} \;=\; & -\frac{2107648}{15825}H_{-11,-1} + \frac{50048}{9495}H_{-9,-3} - \frac{117568}{237375}H_{-7,-5} + \frac{100352}{1583}\zeta_2 H_{-9,-1} \\
& - \frac{3584}{1583}\zeta_2 H_{-7,-3} + \frac{320}{57}\zeta_2^2 H_{-7,-1} - \frac{64}{171}\zeta_2^2 H_{-5,-3} - \frac{2535128220786914}{481025690578125}\zeta_2^6 \\
& + \frac{69528448}{427275}\eta_3\eta_9 - \frac{32}{35}\eta_3^2\zeta_2^3 + \frac{64}{243}\eta_3^4 - \frac{21236224}{299187}\eta_7\eta_3\zeta_2 \\
& - \frac{11072}{1425}\eta_5\eta_3\zeta_2^2 + \frac{696654848}{4984875}\eta_5\eta_7 - \frac{11690624}{356175}\eta_5^2\zeta_2 \,, \qquad (10.1)
\end{aligned}$$

in which we remind the reader that $\eta_n = H_{-n}$. The next equation is at weight $\mathrm{w} = 15$ and is already considerably lengthier. The rhs of $Z_{6,4,3,1,1}$ contains 49 terms when written in this form and some of the fractions consist of more than 100 decimal digits. The phenomenon of these pushdowns seems to be intimately connected with the doubling and generalized doubling relations. We have investigated this at the weight $\mathrm{w} = 12$ system. This is the only system over which we have complete control, because we have the full results for all depths for all Euler sums up to this weight. If we run this system without the use of the doubling and generalized doubling relations there are three more elements left in the 'basis', see Table 5. Two are of depth 4 and one is of depth 2. And additionally there is no pushdown. The element $Z_{6,4,1,1} = H_{6,4,1,1}$ needs one of these extra elements at depth 4. If we use the doubling relations, but we do not use the GDRs, there is only one extra element of depth 4, but the pushdown does take place. If we use only the GDRs, there are no remaining elements beyond the regular basis and the pushdown takes place.

Unfortunately we cannot run this test for other weights. Not using the GDRs means that we cannot run at restricted depth, due to the phenomenon of leakage. Of course it is rather adventurous to make the statement that doubling is at the origin of the pushdowns, when we have only a single case, but there is more supporting evidence as we will see below.

The way we have presented the pushdown in (10.1), although correct, is not its most transparent form. One can rewrite it to as many MZVs as possible and obtain a much simpler representation. One can, for instance, write

$$\begin{aligned}
H_{-9,-3} \;=\; & \frac{1055}{1024}\Bigg[-Z_{9,3} - \frac{185874}{5275}\zeta_7\zeta_5 - \frac{37332}{1055}\zeta_9\zeta_3 \\
& + \frac{1024}{26375}H_{-7,-5} + \frac{187392}{5275}H_{-11,-1} + \frac{92649159488}{23101203125}\zeta_2^6\Bigg] \,. \qquad (10.2)
\end{aligned}$$

Additionally, we introduce a new function A as

$$A_{n_1,n_2,\cdots,n_{p-1},n_p} \;=\; \sum_{\pm} s H_{\pm n_1,\pm n_2,\cdots,\pm n_{p-1},n_p} \qquad (10.3)$$

in which the sum is over the 2^{p-1} possible sign combinations and $s = -1$ if the number of minus signs inside H is odd and $s = +1$ if this number is even as in

$$A_{7,5,3} \;=\; H_{7,5,3} - H_{-7,5,3} - H_{7,-5,3} + H_{-7,-5,3} \,. \qquad (10.4)$$

Notice that the last index is always positive. In terms of the Z-notation the function A is the sum over all Z-sums with an even number of negative indices, but the absolute values of the indices are identical to the indices of the A-function. We rewrite then

$$\begin{aligned} H_{-7,-5} &= -\frac{25}{3}\Bigg[-A_{7,5}+\frac{1295}{2304}Z_{9,3}+\frac{461399}{15360}\zeta_7\zeta_5+\frac{3213}{128}\zeta_9\zeta_3 \\ &\quad -\frac{126}{5}H_{-11,-1}-\frac{39238805939}{12612600000}\zeta_2^6\Bigg], \end{aligned} \tag{10.5}$$

and finally the result for the pushdown becomes:

$$\begin{aligned} Z_{6,4,1,1} &= -\frac{64}{27}A_{7,5}-\frac{7967}{1944}Z_{9,3}+\frac{1}{12}\zeta_3^4+\frac{11431}{1296}\zeta_7\zeta_5 \\ &\quad -\frac{799}{72}\zeta_9\zeta_3+3\zeta_2Z_{7,3}+\frac{7}{2}\zeta_2\zeta_5^2+10\zeta_2\zeta_7\zeta_3 \\ &\quad +\frac{3}{5}\zeta_2^2Z_{5,3}-\frac{1}{5}\zeta_2^2\zeta_5\zeta_3-\frac{18}{35}\zeta_2^3\zeta_3^2-\frac{5607853}{6081075}\zeta_2^6, \end{aligned} \tag{10.6}$$

which is much simpler than equation (10.1). We see the same happening in the expression for $Z_{6,4,3,1,1}$,

$$\begin{aligned} Z_{6,4,3,1,1} &= +\frac{1408}{81}A_{7,5,3}+\frac{16663}{11664}Z_{9,3,3}+\frac{150481}{68040}Z_{7,3,5}+10\zeta_3Z_{6,4,1,1} \\ &\quad +\frac{162823}{3888}\zeta_3Z_{9,3}-\frac{17}{20}\zeta_3^5-\frac{101437}{38880}\zeta_5Z_{7,3}-\frac{1520827}{38880}\zeta_5^3 \\ &\quad +\frac{1903}{120}\zeta_7Z_{5,3}-\frac{93619}{1296}\zeta_7\zeta_5\zeta_3+\frac{3601}{48}\zeta_9\zeta_3^2-\frac{20651486329}{4082400}\zeta_{15} \\ &\quad +\frac{14}{5}\zeta_2Z_{5,5,3}-2\zeta_2Z_{7,3,3}-27\zeta_2\zeta_3Z_{7,3}-\frac{21}{2}\zeta_2\zeta_5Z_{5,3}-\frac{61}{2}\zeta_2\zeta_5^2\zeta_3 \\ &\quad -84\zeta_2\zeta_7\zeta_3^2+\frac{31753363}{12960}\zeta_2\zeta_{13}-4\zeta_2^2Z_{5,3,3}-5\zeta_2^2\zeta_3Z_{5,3} \\ &\quad +\frac{9}{2}\zeta_2^2\zeta_5\zeta_3^2+\frac{979621}{1701}\zeta_2^2\zeta_{11}+\frac{186}{35}\zeta_2^3\zeta_3^3-\frac{490670609}{3572100}\zeta_2^3\zeta_9 \\ &\quad -\frac{1455253}{283500}\zeta_2^4\zeta_7+\frac{4049341}{311850}\zeta_2^5\zeta_5+\frac{12073102}{1488375}\zeta_2^6\zeta_3. \end{aligned} \tag{10.7}$$

In both relations there is only a single object in the equation that is not an MZV: the function A. This means that we can write this A-function alternatively as a combination of MZVs of which one has a depth $\mathsf{d}'=\mathsf{d}+2$. We have done that with $A_{7,5}$ to obtain (10.7), see the fourth term in the right hand side. The intriguing part about it all is that this function A contains half of the terms on the right hand side of the doubling relation in equation (2.14). In terms of H-functions it are the terms in which the last index is positive and in terms of Z-functions it are all terms with an even number of negative indices.

We have been able to construct pushdown relations for all extended basis elements up to weight $\mathsf{w}=21$ and one for weight $\mathsf{w}=22$. Some of these could be constructed directly from the data mine. The more difficult ones are, however, outside the range of the files in the data mine. There we could use the data mine as an aid in limiting the search with numerical algorithms like `LLL` or `PSLQ`. More details are given in Appendix C. This search for pushdowns is not always as simple as the two examples we gave above. Sometimes there is more than one pushdown at a given depth, and sometimes there are elements at a given depth that should be pushed down, but there are also elements that remain at that depth. In the last case it is usually a linear combination of the extended element(s) and the remaining element(s) that get(s) pushed down. But for all cases that we could check there is a single function A associated with each pushed down element. If there are several pushdowns at a given weight and depth the right hand side may contain linear combinations of the corresponding A-functions. In all cases we could select the bases such that the index fields of the A-functions corresponded to the index fields of the elements of the set L_w that had to be extended.

The above indicates that these A-functions have a special status within the Euler sums. They are quite similar to the MZVs.

It should be noted that not all A-functions can be written in terms of MZVs only. This holds only for a limited subset as we will see in the next Section. Additionally, not all A-functions that can be rewritten in terms of MZVs can be used for pushdowns, because a number of them can be rewritten in terms of MZVs that have at most the same depth as the A-function itself.

The above observations lead to the following conjecture:

Conjecture 3.
At each weight w, there exists a set of Lyndon words L_w from which one may construct a basis for MZVs as follows. For each Lyndon word one chooses either the associated Z value or the associated A value, with the number of A values chosen to agree with the Broadhurst-Kreimer conjectures. Linear combinations of these A values then provide the pushdowns for the extensions of Z values by a pair unit indices, as exemplified in Appendix C. □

What the above says is that we can find a good basis for the MZVs using the set L_w, provided we borrow some elements from the Euler sums. In such terms the basis for weight w = 18 would look like

$$\begin{array}{lllllllll} L_{18}: & Z_{15,3} & Z_{13,5} & Z_{11,7} & Z_{9,3,3,3} & Z_{7,5,3,3} & Z_{7,3,5,3} & Z_{7,3,3,5} & Z_{5,5,5,3} \\ P_{18}: & Z_{15,3} & Z_{13,5} & A_{11,7} & Z_{9,3,3,3} & A_{7,5,3,3} & Z_{7,3,5,3} & Z_{7,3,3,5} & Z_{5,5,5,3} \end{array}$$

11 Special Euler Sums

The discovery of the A-functions brings up a new point. Which Euler sums can be written as a linear combination of MZVs only? This is of course a perfect question for a system like the data mine in which exhaustive searches are relatively cheap. At the same time we ask of course the question which A-functions can be written in terms of MZVs only. We should distinguish two cases :

- The object can be written in terms of MZVs that have at most the same depth as the object.

w/d	2	3	4	5
7	13	9	2	0
8	5	10	8	2
9	19	26	2	0
10	7	22	17	7
11	25	38	6	0
12	9	40	43	13
13	31	62	4	1
14	11	62	77	23
15	37	90	6	3
16	13	90	137	34
17	43	121	6	3

Table 19: Number of Euler sums with at least one negative index that can be rewritten in terms of MZVs only as a function of weight (w) and depth (d).

w/d	2	3	4	5
7	4	5	2	0
8	5	8	4	0
9	6	13	9	3
10	7	18	17	7
11	8	25	31	17
12	9	32	49	34
13	10	41	74	67
14	11	50	106	116
15	12	61	148	192
16	13	72	198	298
17	14	85	259	449

Table 20: Number of A-functions that can be rewritten in terms of MZVs only as a function of weight (w) and depth (d).

- The object needs MZVs of a higher depth. This occurs when there is already an A-function that is used in a pushdown. In that case many other A-functions may be rewritten in terms of this A-function and MZVs of the same depth or lower depth.

We find that whenever the second case can occur, it will for a large fraction of the A-functions of that depth. The number of H-functions with at least one negative index that can be rewritten completely in terms of MZVs is given in Table 19. In Table 20 we show the same for the A-functions. Here there are clearly many more. Actually a sizable fraction of the A-functions can be rewritten like this. For example, there are 1365 finite A-functions of $\mathsf{w}=17, \mathsf{d}=5$ of which 449 can be rewritten in terms of MZVs only.

Considering that a number of the Euler sums can be rewritten in terms of MZVs only, one may raise the question whether the pushdowns can be rewritten in such a way that they do not have the A-functions, but rather have a single Euler sum in their right hand side. This turned out to be a difficult question to answer, because the pushdown at $\mathsf{w}=21, \mathsf{d}=7$

was very time consuming and took several days for each trial. At first the number of candidates was rather large. We could make a list of candidates in a way, similar to that of Table 19 for $\mathrm{w} = 21, \mathrm{d} = 5$ and see which Euler sums could be expressed in terms of MZVs and $A_{7,5,3,3,3}$ which is the object that was used in the pushdown[11]. Unfortunately the results for $\mathrm{w} = 21, \mathrm{d} = 5$ are in modular arithmetic and without the products of lower weight objects. Trying several elements of the list gave negative results indicating that many objects that give only MZVs for the terms with the same weight may have terms that are products of Euler sums of a lower weight. Then, after constructing Table 19 we looked for patterns and we noticed that the only eligible elements for $\mathrm{w} = 13, \mathrm{w} = 15, \mathrm{w} = 17$ are

$$\begin{aligned}
Z_{3,-2,3,-2,3} &= H_{3,-2,-3,2,3} \\
Z_{3,-4,3,-2,3} &= H_{3,-4,-3,2,3} \\
Z_{3,-2,3,-4,3} &= H_{3,-2,-3,4,3} \\
Z_{3,-6,3,-2,3} &= H_{3,-6,-3,2,3} \\
Z_{3,-4,3,-4,3} &= H_{3,-4,-3,4,3} \\
Z_{3,-2,3,-6,3} &= H_{3,-2,-3,6,3} \,.
\end{aligned} \tag{11.1}$$

Trying to rewrite $Z_{3,-6,3,-6,3}$ in terms of $A_{7,5,3,3,3}$ by means of `LLL` (a 130 elements search) gave the desired result. Hence by now all pushdowns have been obtained as well in terms of MZVs as in terms of one single Euler sum only. Unfortunately the index field of these Euler sums seems to be completely unrelated to the index fields of our basis elements.

12 Outlook

The data mine has given us already much information and it may yield more yet. But the current results leave also many new questions. To name a few:

- Can the GDRs be derived and/or written in a simpler way?
- Why can the GDRs resolve the problem of 'leakage'?
- Why do we need the doubling relations at all?
- What is the relation between the doubling formula and the pushdowns?
- Is it possible to see which A-functions can be used for pushdowns without needing the Euler sums of the data mine?
- Can a pushdown basis be constructed without needing the MZVs of the data mine?

In addition there is some 'unfinished business'. We did not get more than partial evidence for the double pushdowns at weight 27 and weight 28. Although we can guess the basis at weight 27, an `LLL` search for the complete formula would involve more than 800 elements and probably more than 10 times the number of digits than what our current

[11] Originally we worked with $A_{9,3,3,3,3}$ and it was only at a very late stage that we converted to $A_{7,5,3,3,3}$. Hence a number of the 'raw' results still refer to $A_{9,3,3,3,3}$.

searches needed. Considering the asymptotic behaviour of the `LLL` algorithm, this would mean at least 10^7 times the computer time we needed for the current determinations. The data mine approach is also not very attractive. There we would need the Euler sums to weight 27, depth 9. This might need even more extra orders of magnitude in resources than for the `LLL` algorithm. What would be very welcome is an algorithm by which we can determine a (small) subset of the Euler sums that includes the *A*-functions and combine this subset with the MZVs. For the MZV part of these double pushdowns things look much brighter. In modular arithmetic the continuously improving hardware and software technology should place those runs within reach soon. With a better ordering of the processing of the equations, which unfortunately we do not have, the runs could already be attempted. Again, finding non-trivial subsets to which one might limit oneself, would immediately lead to great progress as well. We hope, that the empirical discoveries we made in this paper for harmonic sums up to $\mathrm{w} = 30$ will stimulate mathematical research and eventually lead to proofs of more far reaching theorems in the future. Here we regard the consideration of the embedding of the MZVs into the Euler sums of importance. Likewise one may consider colored 'MZVs' with even higher roots of unity [81] in the future, which have not been the objective of this paper.

The data mine will be extended whenever new and relevant results are obtained. there is a history page that shows additions and corrections. If others have interesting contributions, they should contact one of the authors.

Acknowledgments.
The work has been supported in part by the research program of the Dutch Foundation for Fundamental Research of Matter (FOM), by DFG Sonderforschungsbereich Transregio 9, Computergestützte Theoretische Teilchenphysik, the European Commission MRTN HEPTOOLS under Contract No. MRTN-CT-2006-035505, J.V. would also like to thank the Humboldt foundation for its generous support, and DESY, Zeuthen and the University of Karlsruhe for its hospitality during this work. The runs for creating the data mine were done on the computer system of the Theoretical Particle Physics group (TTP) at the University of Karlsruhe and computers at DESY and Nikhef.

A Fibonacci and Lyndon Bases at Fixed Weight

In the past several bases have been considered for both the MZVs and the Euler sums. In some of these the concept of depth is not relevant and hence for the counting rules we should sum over the depth. We will discuss those bases in this Appendix. For a number of these bases conjectures are formulated in the literature, which cannot be broken down fixing the depth. The counting relation for the MZVs was conjectured in [2, 13] and [12], respectively.

The vector space of MZVs can be constructed allowing basis elements, which contain besides the ζ–values the index of which is a Lyndon word products of this type of ζ-values of lower weight. One basis of this kind is

$$\mathsf{w} = 2 \qquad \zeta_2 \tag{A.1}$$
$$\mathsf{w} = 3 \qquad \zeta_3 \tag{A.2}$$
$$\mathsf{w} = 4 \qquad \zeta_2^2 \tag{A.3}$$
$$\mathsf{w} = 5 \qquad \zeta_5, \zeta_2\zeta_3 \tag{A.4}$$
$$\mathsf{w} = 6 \qquad \zeta_3^2, \zeta_2^3 \tag{A.5}$$
$$\mathsf{w} = 7 \qquad \zeta_7, \zeta_5\zeta_2, \zeta_3\zeta_2^2 \tag{A.6}$$
$$\mathsf{w} = 8 \qquad \zeta_{5,3}, \zeta_5\zeta_3, \zeta_3^2\zeta_2, \zeta_2^4 \tag{A.7}$$
$$\mathsf{w} = 9 \qquad \zeta_9, \zeta_7\zeta_2, \zeta_5\zeta_2^2, \zeta_3^3, \zeta_3\zeta_2^3 \tag{A.8}$$
$$\mathsf{w} = 10 \qquad \zeta_{7,3}, \zeta_{5,3}\zeta_2, \zeta_7\zeta_3, \zeta_5^2, \zeta_5\zeta_3\zeta_2, \zeta_3^2\zeta_2^2, \zeta_2^5, \text{ etc.} \tag{A.9}$$

The number of these basis elements is counted by the Padovan numbers, $\hat{P}_k$, [43], which have the same recursion as the Perrin numbers, but start from the initial values $\hat{P}_1 = \hat{P}_2 = \hat{P}_3 = 1$. Their generating function is

$$G(\hat{P}_k, x) = \frac{1+x}{1-x^2-x^3} = \sum_{k=0}^{\infty} x^k \hat{P}_k \; . \tag{A.10}$$

They also obey a Binet-like formula. The first values are given in Table 21.

w	1	2	3	4	5	6	7	8	9	10
$\hat{P}_w$	1	1	1	2	2	3	4	5	7	9
w	11	12	13	14	15	16	17	18	19	20
$\hat{P}_w$	12	16	21	28	37	49	65	86	114	151
w	21	22	23	24	25	26	27	28	29	30
$\hat{P}_w$	200	265	351	465	616	816	1081	1432	1897	2513

Table 21: The first 30 Padovan numbers.

The above basis is of the Fibonacci type. Another basis of the Fibonacci type is the Hoffman basis [69] which consists of all elements of which the index field is made up

w	1	2	3	4	5	6	7	8	9	10
P_w	0	2	3	2	5	5	7	10	12	17
w	11	12	13	14	15	16	17	18	19	20
P_w	22	29	39	51	68	90	119	158	209	277
w	21	22	23	24	25	26	27	28	29	30
P_w	367	486	644	853	1130	1497	1983	2627	3480	4610

Table 22: The first 30 Perrin numbers.

from 2's and 3's only. If one uses the following construction it is easy to see that the number of basis elements follows the Padovan sequence.

$$\begin{aligned} \mathsf{w} &= 1 \qquad \emptyset \\ \mathsf{w} &= 2 \qquad (2) \\ \mathsf{w} &= 3 \qquad (3)\ . \end{aligned} \tag{A.11}$$

The index words at weight w are given by

$$I_w = \bigcup_{|a|=(w-2)} (2, I_a) \quad \cup \quad \bigcup_{|b|=(w-3)} (3, I_b)\ . \tag{A.12}$$

Let us now turn to Lyndon bases for the MZVs. Using a Witt-type relation [44] the size of the basis is conjectured to be given by

$$\begin{aligned} l(w) &= \frac{1}{w}\sum_{d|w}\mu\left(\frac{w}{d}\right)P_d, \\ & \quad P_1 = 0, P_2 = 2, P_3 = 3, P_d = P_{d-2} + P_{d-3}, \ \ d \geq 3\ . \end{aligned} \tag{A.13}$$

Here the sum runs over the divisors d of the weight w and P_d denotes the Perrin-numbers [45, 46]. They are given by the Binet-like formula

$$\begin{aligned} P_n &= \alpha^n + \beta^n + \gamma^n, \quad \text{with } \alpha,\ \beta,\ \gamma \text{ the roots of} \\ x^3 - x - 1 &= 0 \end{aligned} \tag{A.14}$$

and can be derived from the generating function

$$G(P_k, x) = \frac{3 - x^2}{1 - x^2 - x^3} = \sum_{k=0}^{\infty} x^k P_k\ . \tag{A.15}$$

The first values are given in Table 22.

For the basis different choices are possible, which yield equivalent representations. Here we choose the basis in terms of ζ–values, with an index field which forms a Lyndon word. Our first choice consists of indices, which contain as widely as possible odd integers. In case of even weights in a series of cases also indices with only even numbers occur from $\mathsf{w} = 12$ onwards, as e.g. for

$$\mathsf{w} = 18\ :\quad \zeta_{15,3},\ \zeta_{13,5},\ \zeta_{9,3,3,3},\ \zeta_{7,5,3,3},\ \zeta_{5,5,5,3},\ \zeta_{7,5,5,1},\ \zeta_{8,2,2,2,2,2},\ \zeta_{12,2,2,2}\ . \tag{A.16}$$

w	1	2	3	4	5	6	7	8	9	10
l_w	0	1	1	0	1	0	1	1	1	1
w	11	12	13	14	15	16	17	18	19	20
l_w	2	2	3	3	4	5	7	8	11	13
w	21	22	23	24	25	26	27	28	29	30
l_w	17	21	28	34	45	56	73	92	120	151

Table 23: Number of basis elements of the Lyndon basis for the MZVs for fixed weight w.

A second natural choice is to take the afore mentioned Hoffman basis and select from it only those elements of which the index field forms a Lyndon word. Because the algebraic relations for the product of basis elements of lower weight do not give objects that are closely related to the basis elements at the higher weight, this basis is not used very frequently.

As an example we consider the case w = 30 and calculate the size of the bases using the Witt formula (A.13) resp. the number of Lyndon words made up by the letters 2 and 3 only with $2 < 3$. 30 has the following decomposition

$$30 \equiv k_i * 3 \; + \; l_i * 2 = 2 * 3 \; + \; 12 * 2 = 4 * 3 \; + \; 9 * 2 = 6 * 3 \; + \; 6 * 2 = 8 * 3 \; + 3 * 2 \; . \tag{A.17}$$

We now calculate the number of Lyndon words for each of these contributions, with $m_i = k_i + l_i$,

$$n_i = \frac{1}{m_i} \sum_{d|m_i} \mu(d) \frac{(m_i/d)!}{(k_i/d)!(l_i/d)!} \; . \tag{A.18}$$

One obtains

$$\begin{aligned} L_{\{2,3\}}(30) &= \frac{1}{14}\left[\frac{14!}{12!2!} - \frac{7!}{6!}\right] + \frac{1}{13}\frac{13!}{9!4!} + \frac{1}{12}\left[\frac{12!}{6!^2} - \frac{6!}{3!^2} - \frac{4!}{2!^2} + \frac{2!}{1!^2}\right] + \frac{1}{11}\frac{11!}{8!3!} \\ &= 151 \; . \end{aligned} \tag{A.19}$$

Using (A.13) the result is

$$\begin{aligned} l(30) &= \frac{1}{30}\left[P_{30} - P_{15} - P_{10} - P_6 + P_5 + P_3 + P_2 - P_0\right] \\ &= \frac{1}{30}\left[4610 - 68 - 17 - 5 + 5 + 3 + 2 - 0\right] = 151 \; . \end{aligned} \tag{A.20}$$

A basis up to weight $\mathsf{w} = 17$ for the MZVs was also constructed in [82].

For the Euler sums the Fibonacci basis is counted by the Fibonacci numbers. When we consider also all divergent multiple zeta values the Fibonacci sequence is merely shifted. It is easily shown that the divergent Euler sums can be represented by the convergent sums and the element σ_0. As in the MZV case we may span the vector space of the Euler sums

w	1	2	3	4	5	6	7	8	9	10
f_w	1	1	2	3	5	8	13	21	34	55
w	11	12	13	14	15	16	17	18	19	20
f_w	89	144	233	377	610	987	1597	2584	4181	6765

Table 24: The first 20 Fibonacci numbers.

by forming a basis, which includes products of lower weight basis elements contained in a Lyndon-basis. One basis of this type, used in the `summer` program [10] reads

$$\mathsf{w} = 1 \qquad \ln(2) \tag{A.21}$$
$$\mathsf{w} = 2 \qquad \zeta_2, \ln^2(2) \tag{A.22}$$
$$\mathsf{w} = 3 \qquad \zeta_3, \zeta_2 \ln(2), \ln^3(2) \tag{A.23}$$
$$\mathsf{w} = 4 \qquad \mathrm{Li}_4(1/2), \zeta_3 \ln(2), \zeta_2^2, \zeta_2 \ln^2(2), \ln^4(2) \tag{A.24}$$
$$\mathsf{w} = 5 \qquad \mathrm{Li}_5(1/2), \zeta_5, \mathrm{Li}_4(1/2)\ln(2), \zeta_3\zeta_2, \zeta_3 \ln^2(2), \zeta_2 \ln^3(2), \zeta_2^2 \ln(2), \ln^5(2) \tag{A.25}$$
$$\begin{aligned}\mathsf{w} = 6 \qquad & \mathrm{Li}_6(1/2), \zeta_{-5,-1}, \mathrm{Li}_5(1/2)\ln(2), \zeta_5 \ln(2), \mathrm{Li}_4(1/2)\zeta_2, \\ & \mathrm{Li}_4(1/2)\ln^2(2), \zeta_3^2, \zeta_3\zeta_2 \ln(2), \zeta_3 \ln^3(2), \zeta_2^3, \zeta_2^2 \ln^2(2), \zeta_2 \ln^4(2), \\ & \ln^6(2), \text{ etc.}\end{aligned} \tag{A.26}$$

These bases are counted by the Fibonacci-numbers [42, 83], f_{w+1}, which obey the same recursion relation as the Lucas numbers, but with the initial conditions $f_0 = 0, f_1 = 1$. They are represented by the formula given by J.P.M. Binet (1843)[12]

$$f_d = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^d - \left(\frac{1-\sqrt{5}}{2} \right)^d \right], \tag{A.27}$$

and result from the generating function

$$G(f_k, x) = \frac{x}{1-x-x^2} = \sum_{k=0}^{\infty} x^k f_k \,. \tag{A.28}$$

The first values are given in Table 24.

Another Fibonacci basis can be constructed as

$$\begin{aligned} \mathsf{w} = 0 \qquad & \emptyset \\ \mathsf{w} = 1 \qquad & (-1) \\ \mathsf{w} = 2 \qquad & (0,-1)\,. \end{aligned} \tag{A.29}$$

$H_{-1}(1)$ and $H_{0,-1}(1) = H_{-2}(1)$ are chosen as basis elements.

Conjecture 4.
With the above starting conditions, consider the index words at weight w to be

$$I_w = \bigcup_{|a|=(w-1)} (-1, I_a) \quad \cup \quad \bigcup_{|b|=(w-2)} (-2, I_b)\,. \tag{A.30}$$

[12] The relation was known to Euler and Moivre.

The basis elements for the Euler sums are then given by the ζ-values with indices out of I_w. The elements of which the index sets are a Lyndon word form a Lyndon basis. □

The Fibonacci version of this basis seems to have been discovered independently by S. Zlobin, see Ref. [71].

This construction is analogous to that by Hoffman in the case of MZVs. It also uses a 2-letter alphabet. The different decomposition of the weight w, however, leads to a basis of different length. Again we may derive the length of the basis using the Witt-formula (A.52) or counting the basis elements as Lyndon words of the index set (A.30). Let us give an example for w = 20.

$$\begin{aligned} 20 &= k_i*1+l_i*2 = 18*1\ +\ 1*2 = 16*1\ +\ 2*2 = 14*1\ +\ 3*2 \\ &= 12*1\ +\ 4*2 = 10*1\ +\ 5*2 = 8*1\ +\ 6*2 = 6*1\ +\ 7*2 \\ &= 4*1\ +\ 8*2 = 2*1\ +\ 9*2 \end{aligned} \tag{A.31}$$

Similar to the non-alternating case one obtains

$$\begin{aligned} L_{\{-1,-2\}}(20) &= \frac{1}{19}\frac{19!}{18!1!}+\frac{1}{18}\left[\frac{18!}{16!2!}-\frac{9!}{8!1!}\right]+\frac{1}{17}\frac{17!}{14!3!}+\frac{1}{16}\left[\frac{16!}{12!4!}-\frac{9!}{8!1!}\right] \\ &+ \frac{1}{15}\left[\frac{15!}{10!5!}-\frac{3!}{2!1!}\right]+\frac{1}{14}\left[\frac{14!}{8!6!}-\frac{7!}{4!3!}\right]+\frac{1}{13}\frac{13!}{7!6!} \\ &+ \frac{1}{12}\left[\frac{12!}{8!4!}-\frac{6!}{4!2!}\right]+\frac{1}{11}\frac{11!}{9!2!} = 750\ . \end{aligned} \tag{A.32}$$

Likewise the Witt-formula (A.52) yields

$$\begin{aligned} l(20) &= \frac{1}{20}\left[l_{20}-l_{10}-l_4+l_2\right] \\ &= \frac{1}{20}\left[15127-123-7+3\right] = 750\ . \end{aligned} \tag{A.33}$$

The above basis suffers from the same shortcoming as the Hoffman basis in that the concept of depth lacks relevance. Hence we did not use it.

In a similar way we can construct yet another Fibonacci basis:

Conjecture 5.
With the starting conditions of (A.29), consider the index words at weight w to be

$$I_w = \bigcup_{|a|=(w-1)}(-1,I_a) \quad \cup \quad \bigcup_{|b|=(w-2)}(0,0,I_b)\ . \tag{A.34}$$

The basis elements for the Euler sums are then given by the ζ-values of indices I_w. The elements of which the index fields are a Lyndon word and all indices are odd valued if $w>2$ form a Lyndon basis. □

The Lyndon basis of this construction happens to be the basis proposed in ref [12]. We can divide I_w

$$I_w = I_w^{\mathsf{odd}} \oplus I_w^{\neg\mathsf{odd}}\ , \tag{A.35}$$

with the indices in I_w^{odd} are all odd and the last index of $I_w^{\neg\mathsf{odd}}$ even, all others odd. The Lyndon words of I_w^{odd}, $\mathrm{Ly}[\mathrm{I}_{\mathrm{w}}^{\mathsf{odd}}]$, form the basis elements at weight w and they are counted by (A.52). Note, that the basis element at $\mathsf{w} = 2$ is not odd, which is an exception.

As an illustration we consider the case $\mathsf{w} = 6$. The following words are generated, where we assume the ordering $0 < 1$ and let the digit 1 play the role of -1.

$$\begin{aligned} &\{000001, 000011, 001001, 001101, 001111\}; \\ &\{100001, 100101, 100111, 110001, 110011, 111001, 111101, 111111\}\,. \end{aligned} \tag{A.36}$$

The Lyndon words are

$$\begin{aligned} &(000011) \equiv (-5,-1); \qquad &&(001111) \equiv (-3,-1,-1,-1); \\ &(000001) \equiv (-6); &&(001101) \equiv (-3,-1,-2)\,. \end{aligned} \tag{A.37}$$

The Lyndon words with odd indices taken as index of an Euler sum are basis elements, which we express through the harmonic polylogarithms at argument $x = 1$, $H_{-5,-1}(1)$ and $H_{-3,-1,-1,-1}(1)$. On the other hand,

$$H_{-6} = \frac{62}{35} H_{-2}^3 \tag{A.38}$$

$$H_{-3,-1,-2} = H_{-5,-1} + H_{-2} H_{-3,-1} + \frac{452}{105} H_{-2}^3 - \frac{55}{18} H_{-3}^2 \tag{A.39}$$

do not belong to the basis.

The last Lyndon basis is the one we actually use in the programs. It is depth oriented and no element can be written as a linear combination of elements of lower depth or products of elements with lower weight. To weight $\mathsf{w} = 12$ the complete basis for the finite elements is given by

$$\mathsf{w} = 1 \qquad H_{-1}; \tag{A.40}$$

$$\mathsf{w} = 2 \qquad H_{-2}; \tag{A.41}$$

$$\mathsf{w} = 3 \qquad H_{-3}; \tag{A.42}$$

$$\mathsf{w} = 4 \qquad H_{-3,-1}; \tag{A.43}$$

$$\mathsf{w} = 5 \qquad H_{-5},\ H_{-3,-1,-1}; \tag{A.44}$$

$$\mathsf{w} = 6 \qquad H_{-5,-1},\ H_{-3,-1,-1,-1}; \tag{A.45}$$

$$\mathsf{w} = 7 \qquad H_{-7},\ H_{-5,-1,-1},\ H_{-3,-3,-1},\ H_{-3,-1,-1,-1,-1}; \tag{A.46}$$

$$\begin{aligned} \mathsf{w} = 8 \qquad & H_{-7,-1},\ H_{-5,-3},\ H_{-5,-1,-1,-1},\ H_{-3,-3,-1,-1}, \\ & H_{-3,-1,-1,-1,-1,-1}; \end{aligned} \tag{A.47}$$

$$\begin{aligned} \mathsf{w} = 9 \qquad & H_{-9},\ H_{-7,-1,-1},\ H_{-5,-3,-1},\ H_{-5,-1,-3},\ H_{-5,-1,-1,-1,-1}, \\ & H_{-3,-3,-1,-1,-1},\ H_{-3,-1,-3,-1,-1},\ H_{-3,-1,-1,-1,-1,-1,-1}; \end{aligned} \tag{A.48}$$

$$\begin{aligned} \mathsf{w} = 10 \qquad & H_{-9,-1},\ H_{-7,-3},\ H_{-7,-1,-1,-1},\ H_{-5,-3,-1,-1},\ H_{-5,-1,-3,-1}, \\ & H_{-5,-1,-1,-3},\ H_{-3,-3,-1,-1},\ H_{-5,-1,-1,-1,-1,-1},\ H_{-3,-3,-1,-1,-1,-1}, \\ & H_{-3,-1,-3,-1,-1,-1},\ H_{-3,-1,-1,-1,-1,-1,-1,-1}; \end{aligned} \tag{A.49}$$

w	1	2	3	4	5	6	7	8	9	10
l_w	1	3	4	7	11	18	29	47	76	123
w	11	12	13	14	15	16	17	18	19	20
l_w	199	322	521	843	1364	2207	3571	5778	9349	15127

Table 25: The first 20 Lucas numbers.

$$\begin{aligned}
\mathsf{w}=11 \quad & H_{-11},\ H_{-9,-1,-1},\ H_{-7,-3,-1},\ H_{-7,-1,-3},\ H_{-5,-5,-1},\\
& H_{-5,-3,-3},\ H_{-3,-3,-1,-3,-1},\ H_{-3,-3,-3,-1,-1},\ H_{-5,-1,-1,-1,-3},\\
& H_{-5,-1,-1,-3,-1},\ H_{-5,-1,-3,-1,-1},\ H_{-5,-3,-1,-1,-1},\\
& H_{-7,-1,-1,-1,-1},\ H_{-3,-1,-1,-3,-1,-1,-1},\ H_{-3,-1,-3,-1,-1,-1,-1},\\
& H_{-3,-3,-1,-1,-1,-1,-1},\ H_{-5,-1,-1,-1,-1,-1,-1},\\
& H_{-3,-1,-1,-1,-1,-1,-1,-1,-1}; \qquad (A.50)\\
\mathsf{w}=12 \quad & H_{-7,-5},\ H_{-9,-3},\ H_{-11,-1},\ H_{-5,-1,-3,-3},\ H_{-5,-3,-1,-3},\\
& H_{-5,-3,-3,-1},\ H_{-5,-5,-1,-1},\ H_{-7,-1,-1,-3},\ H_{-7,-1,-3,-1},\\
& H_{-7,-3,-1,-1},\ H_{-9,-1,-1,-1},\ H_{-3,-3,-1,-1,-3,-1},\\
& H_{-3,-3,-1,-3,-1,-1},\ H_{-3,-3,-3,-1,-1,-1},\ H_{-5,-1,-1,-1,-1,-3},\\
& H_{-5,-1,-1,-1,-3,-1},\ H_{-5,-1,-1,-3,-1,-1},\ H_{-5,-1,-3,-1,-1,-1},\\
& H_{-5,-3,-1,-1,-1,-1},\ H_{-7,-1,-1,-1,-1,-1},\ H_{-3,-1,-1,-3,-1,-1,-1,-1},\\
& H_{-3,-1,-3,-1,-1,-1,-1,-1},\ H_{-3,-3,-1,-1,-1,-1,-1,-1},\\
& H_{-5,-1,-1,-1,-1,-1,-1,-1},\ H_{-3,-1,-1,-1,-1,-1,-1,-1,-1,-1}; \qquad (A.51)
\end{aligned}$$

For the Lyndon basis the conjectured length is [12]

$$\begin{aligned}
l(w) &= \frac{1}{w}\sum_{d|w}\mu\left(\frac{w}{d}\right) l_d, \quad w \geq 2\\
& \quad l_1 = 1, l_2 = 3, l_3 = 4, l_d = l_{d-1} + l_{d-2}, \quad d \geq 4\ .\\
l(1) &= 2 \qquad (A.52)
\end{aligned}$$

l_d denote the Lucas-numbers [46, 83]. They are represented by

$$l_d = \left(\frac{1+\sqrt{5}}{2}\right)^d + \left(\frac{1-\sqrt{5}}{2}\right)^d\ , \qquad (A.53)$$

and derive from the generating function

$$G(l_k, x) = \frac{2-x}{1-x-x^2} = \sum_{k=0}^{\infty} x^k l_k\ . \qquad (A.54)$$

The first values are given in Table 25. The case $w = 1$ is special as two elements contribute.

w	1	2	3	4	5	6	7	8	9	10
l_w	1	1	1	1	2	2	4	5	8	11
w	11	12	13	14	15	16	17	18	19	20
l_w	18	25	40	58	90	135	210	316	492	750

Table 26: Number of basis elements of the Lyndon basis for the Euler sums for fixed weight w.

B Pushdown Bases

We have tried to select a basis in which the elements of the set L_w are maximal and the extended elements are minimal. At the same time the extended elements should be Lyndon words. This means for instance that an element like $H_{5,5,5,3}$ cannot be extended and hence has to be part of the basis, even though it is the minimal element at weight $\mathrm{w} = 18$. One could of course reverse the criteria. For the construction of the bases this does not really diminish the amount of work. In both cases there are elements that should be skipped because of linear dependencies. We call the basis below the 'minimal pushdown basis'. In addition we have used the requirement that for the extended elements the corresponding A-function should be usable for a pushdown. This requirement we could enforce up to weight $\mathrm{w} = 22$. For higher weights we do not have the information in the data mine, and hence we do not know whether this requirement can be achieved.

$$
\begin{aligned}
P_2 &= H_2 && \text{(B.1)}\\
P_3 &= H_3 && \text{(B.2)}\\
P_5 &= H_5 && \text{(B.3)}\\
P_7 &= H_7 && \text{(B.4)}\\
P_8 &= H_{5,3} && \text{(B.5)}\\
P_9 &= H_9 && \text{(B.6)}\\
P_{10} &= H_{7,3} && \text{(B.7)}\\
P_{11} &= H_{11}, H_{5,3,3} && \text{(B.8)}\\
P_{12} &= H_{9,3}, H_{6,4,1,1} && \text{(B.9)}\\
P_{13} &= H_{13}, H_{7,3,3}, H_{5,5,3} && \text{(B.10)}\\
P_{14} &= H_{11,3}, H_{9,5}, H_{5,3,3,3} && \text{(B.11)}\\
P_{15} &= H_{15}, H_{7,3,5}, H_{9,3,3}, H_{6,4,3,1,1} && \text{(B.12)}\\
P_{16} &= H_{11,5}, H_{13,3}, H_{5,5,3,3}, H_{7,3,3,3}, H_{8,6,1,1} && \text{(B.13)}\\
P_{17} &= H_{17}, H_{7,5,5}, H_{9,3,5}, H_{9,5,3}, H_{11,3,3}, H_{5,3,3,3,3}, H_{6,6,3,1,1} && \text{(B.14)}\\
P_{18} &= H_{13,5}, H_{15,3}, H_{5,5,5,3}, H_{7,3,3,5}, , H_{7,3,5,3}, H_{9,3,3,3}, H_{10,6,1,1}, H_{6,4,3,3,1,1} && \text{(B.15)}\\
P_{19} &= H_{19}, H_{9,3,7}, H_{9,5,5}, H_{11,3,5}, H_{11,5,3}, H_{13,3,3},\\
&\quad H_{5,3,5,3,3}, H_{5,5,3,3,3}, H_{7,3,3,3,3}, H_{6,6,5,1,1}, H_{8,6,3,1,1} && \text{(B.16)}
\end{aligned}
$$

$$
\begin{aligned}
P_{20} &= H_{13,7}, H_{15,5}, H_{17,3}, H_{7,3,5,5}, H_{7,5,5,3}, H_{7,7,3,3}, H_{9,3,3,5}, \\
&\quad H_{9,3,5,3}, H_{11,3,3,3}, H_{10,8,1,1}, H_{5,3,3,3,3,3}, H_{6,4,3,5,1,1}, H_{8,4,3,3,1,1} && \text{(B.17)} \\
P_{21} &= H_{21}, H_{9,5,7}, H_{9,9,3}, H_{11,3,7}, H_{13,3,5}, H_{13,5,3}, H_{15,3,3}, \\
&\quad H_{5,5,3,5,3}, H_{5,5,5,3,3}, H_{7,3,3,3,5}, H_{7,3,3,5,3}, H_{7,3,5,3,3}, H_{9,3,3,3,3}, \\
&\quad H_{8,6,5,1,1}, H_{10,4,5,1,1}, H_{10,6,3,1,1}, H_{6,4,3,3,3,1,1} && \text{(B.18)} \\
P_{22} &= H_{15,7}, H_{17,5}, H_{19,3}, H_{7,5,7,3}, H_{7,7,3,5}, H_{9,3,5,5}, H_{9,3,7,3}, \\
&\quad H_{9,5,3,5}, H_{9,5,5,3}, H_{11,3,3,5}, H_{11,3,5,3}, H_{11,5,3,3}, H_{13,3,3,3}, H_{12,8,1,1}, \\
&\quad H_{5,3,5,3,3,3}, H_{5,5,3,3,3,3}, H_{7,3,3,3,3,3} \\
&\quad H_{6,4,5,5,1,1}, H_{6,6,5,3,1,1}, H_{8,2,3,7,1,1}, H_{8,6,3,3,1,1} && \text{(B.19)} \\
P_{23} &= H_{23}, H_{11,7,5}, H_{11,9,3}, H_{13,3,7}, H_{13,5,5}, H_{13,7,3}, H_{15,3,5}, H_{15,5,3}, \\
&\quad H_{17,3,3}, H_{5,5,5,5,3}, H_{7,3,7,3,3}, H_{7,3,5,5,3}, H_{7,5,3,5,3}, H_{7,5,5,3,3}, \\
&\quad H_{7,7,3,3,3}, H_{9,3,3,3,5}, H_{9,3,3,5,3}, H_{9,3,5,3,3}, H_{9,5,3,3,3}, H_{11,3,3,3,3}, \\
&\quad H_{8,6,7,1,1}, H_{8,8,5,1,1}, H_{10,2,9,1,1}, H_{10,4,7,1,1}, \\
&\quad H_{5,3,3,3,3,3,3} H_{6,2,3,5,5,1,1}, H_{6,2,5,3,5,1,1}, H_{6,4,3,3,5,1,1} && \text{(B.20)} \\
P_{24} &= H_{17,7}, H_{19,5}, H_{21,3}, H_{7,7,7,3}, H_{9,7,3,5}, H_{9,7,5,3}, H_{9,9,3,3}, \\
&\quad H_{11,3,3,7}, H_{11,3,5,5}, H_{11,3,7,3}, H_{11,5,3,5}, H_{11,5,5,3}, H_{11,7,3,3}, H_{13,3,3,5}, \\
&\quad H_{13,3,5,3}, H_{13,5,3,3}, H_{15,3,3,3}, H_{12,10,1,1}, H_{14,8,1,1}, H_{5,5,3,3,5,3}, \\
&\quad H_{5,5,3,5,3,3}, H_{5,5,5,3,3,3}, H_{7,3,3,3,5,3}, H_{7,3,3,5,3,3}, H_{7,3,5,3,3,3}, H_{7,5,3,3,3,3}, \\
&\quad H_{9,3,3,3,3,3}, H_{6,6,5,5,1,1}, H_{8,2,5,7,1,1}, H_{8,2,7,5,1,1}, H_{8,4,3,7,1,1}, H_{8,4,5,5,1,1}, \\
&\quad H_{8,4,7,3,1,1}, H_{6,2,3,3,3,5,1,1} && \text{(B.21)} \\
P_{25} &= H_{25}, H_{11,11,3}, H_{13,5,7}, H_{13,7,5}, H_{13,9,3}, H_{15,3,7}, H_{15,5,5}, H_{15,7,3}, \\
&\quad H_{17,3,5}, H_{17,5,3}, H_{19,3,3}, H_{7,3,7,3,5}, H_{7,5,3,7,3}, H_{7,5,7,3,3}, \\
&\quad H_{9,3,3,3,7}, H_{9,3,3,5,5}, H_{9,3,3,7,3}, H_{9,3,5,3,5}, H_{9,3,5,5,3}, H_{9,3,7,3,3}, \\
&\quad H_{9,5,3,3,5}, H_{9,5,3,5,3}, H_{9,5,5,3,3}, H_{9,7,3,3,3}, H_{11,3,3,3,5}, H_{11,3,3,5,3}, \\
&\quad H_{11,3,5,3,3}, H_{11,5,3,3,3}, H_{13,3,3,3,3}, H_{8,8,7,1,1}, H_{10,4,9,1,1}, \\
&\quad H_{10,6,7,1,1}, H_{10,8,5,1,1}, H_{12,2,9,1,1}, H_{5,3,3,5,3,3,3}, H_{5,3,5,3,3,3,3}, \\
&\quad H_{5,5,3,3,3,3,3}, H_{7,3,3,3,3,3,3}, H_{6,2,5,5,5,1,1}, H_{6,4,3,5,5,1,1}, H_{6,4,5,3,5,1,1}, \\
&\quad H_{6,4,5,5,3,1,1}, H_{6,6,3,3,5,1,1}, H_{6,6,3,5,3,1,1}, H_{6,6,5,3,3,1,1} && \text{(B.22)} \\
P_{26} &= H_{17,9}, H_{19,7}, H_{21,5}, H_{23,3}, H_{7,7,7,5}, H_{9,5,9,3}, H_{11,3,9,3}, H_{11,5,3,7}, \\
&\quad H_{11,5,5,5}, H_{11,5,7,3}, H_{11,7,3,5}, H_{11,7,5,3}, H_{11,9,3,3}, H_{13,3,3,7}, H_{13,3,5,5}, \\
&\quad H_{13,3,7,3}, H_{13,5,3,5}, H_{13,5,5,3}, H_{13,7,3,3}, H_{15,3,3,5}, H_{15,3,5,3}, H_{15,5,3,3}, \\
&\quad H_{17,3,3,3}, H_{14,10,1,1}, H_{5,5,5,3,5,3}, H_{5,5,5,5,3,3}, H_{7,3,3,5,5,3}, H_{7,3,5,3,5,3}, \\
&\quad H_{7,3,5,5,3,3}, H_{7,3,7,3,3,3}, H_{7,5,3,3,5,3}, H_{7,5,3,5,3,3}, H_{7,5,5,3,3,3}, \\
&\quad H_{7,7,3,3,3,3}, H_{9,3,3,3,3,5}, H_{9,3,3,3,5,3}, H_{9,3,3,5,3,3}, H_{9,3,5,3,3,3}, \\
&\quad H_{9,5,3,3,3,3}, H_{11,3,3,3,3,3}, H_{8,2,7,7,1,1}, H_{8,4,5,7,1,1}, H_{8,4,7,5,1,1}, \\
&\quad H_{8,6,3,7,1,1}, H_{8,6,5,5,1,1}, H_{8,6,7,3,1,1}, H_{8,8,3,5,1,1}, H_{8,8,5,3,1,1}, \\
&\quad H_{10,2,3,9,1,1}, H_{10,2,5,7,1,1}, H_{10,2,7,5,1,1}, H_{5,3,3,3,3,3,3,3}, \\
&\quad H_{6,2,3,3,5,5,1,1}, H_{6,2,3,5,3,5,1,1}, H_{6,2,5,3,3,5,1,1}, H_{6,4,3,3,3,5,1,1} && \text{(B.23)}
\end{aligned}
$$

The above bases are complete. For the following basis we miss the two elements at depth 9 due to limited computer resources. Yet the construction based on L_{27} allows us to predict the last two elements:

$$
\begin{aligned}
P_{27} \;=\; & H_{27}, H_{11,7,9}, H_{13,11,3}, H_{15,3,9}, H_{15,5,7}, H_{15,7,5}, H_{15,9,3}, H_{17,5,5}, H_{17,7,3}, \\
& H_{19,3,5}, H_{19,5,3}, H_{21,3,3}, H_{7,5,5,7,3}, H_{7,5,7,3,5}, H_{7,7,3,7,3}, H_{7,7,7,3,3}, \\
& H_{9,3,9,3,3}, H_{9,5,3,5,5}, H_{9,5,3,7,3}, H_{9,5,5,3,5}, H_{9,5,5,5,3}, H_{9,5,7,3,3}, H_{9,7,3,3,5}, \\
& H_{9,7,3,5,3}, H_{9,7,5,3,3}, H_{9,9,3,3,3}, H_{11,3,3,3,7}, H_{11,3,3,5,5}, H_{11,3,3,7,3}, H_{11,3,5,3,5}, \\
& H_{11,3,5,5,3}, H_{11,3,7,3,3}, H_{11,5,3,3,5}, H_{11,5,3,5,3}, H_{11,5,5,3,3}, H_{11,7,3,3,3}, H_{13,3,3,3,5}, \\
& H_{13,3,3,5,3}, H_{13,3,5,3,3}, H_{13,5,3,3,3}, H_{15,3,3,3,3}, H_{10,8,7,1,1}, H_{10,10,5,1,1}, \\
& H_{12,2,11,1,1}, H_{12,4,9,1,1}, H_{12,6,7,1,1}, H_{12,8,5,1,1}, H_{16,2,7,1,1}, \\
& H_{5,3,5,3,5,3,3}, H_{5,5,3,3,3,5,3}, H_{5,5,3,3,5,3,3}, H_{5,5,3,5,3,3,3}, H_{5,5,5,3,3,3,3}, \\
& H_{7,3,3,3,3,3,5}, H_{7,3,3,3,3,5,3}, H_{7,3,3,3,5,3,3}, H_{7,3,3,5,3,3,3}, H_{7,3,5,3,3,3,3}, \\
& H_{9,3,3,3,3,3,3}, H_{6,4,5,5,5,1,1}, H_{6,6,3,5,5,1,1}, H_{6,6,5,3,5,1,1}, H_{6,6,5,5,3,1,1}, \\
& H_{8,2,3,5,7,1,1}, H_{8,2,3,7,5,1,1}, H_{8,2,5,3,7,1,1}, H_{8,2,5,5,5,1,1}, H_{8,2,5,7,3,1,1}, \\
& H_{8,2,7,3,5,1,1}, H_{8,2,7,5,3,1,1}, H_{8,4,3,3,7,1,1}, \\
& H_{7,5,7,5,3} \rightarrow ?H_{6,4,6,4,3,1,1,1,1}, H_{7,5,3,3,3,3,3} \rightarrow ?H_{6,4,3,3,3,3,3,1,1}
\end{aligned}
$$

We have selected the last two elements for the necessary extension on the basis of the Appendix in the thesis by Racinet [68] in which for these two elements the numbers 6 and 4 seem to play a special role.

Although we have also results for P_{28} in which the leading depth is missing, there are too many elements missing to give a reliable list of the basis elements. It should be remarked though that also for P_{28} we expect a 2-fold pushdown from depth 8 to depth 4.

C Explicit pushdowns

Below we list all pushdowns up to w = 21 and one at w = 22 with the mixing with terms of equal weight and depth in the left hand side and all remaining Euler sums in the right hand side. The function A is defined in (10.3).

We only list that part of the pushdowns that we consider particularly interesting. The complete formulas can be found in the data mine in the programs part. The name of the file is `pushdowns.h`.

$$Z_{6,4,1,1} = -\frac{64}{27}A_{7,5}+\cdots \tag{C.1}$$

$$Z_{6,4,3,1,1} = \frac{1408}{81}A_{7,5,3}+\cdots \tag{C.2}$$

$$Z_{8,6,1,1}+\frac{542}{175}Z_{5,5,3,3}-\frac{19}{7}Z_{7,3,3,3} = -\frac{1024}{405}A_{9,7}+\cdots \tag{C.3}$$

$$Z_{6,6,3,1,1} - \frac{14}{5}Z_{5,3,3,3,3} = \frac{5120}{243}A_{7,7,3} + \cdots \tag{C.4}$$

$$\begin{aligned} Z_{10,6,1,1} - \frac{10}{3}Z_{9,3,3,3} & \\ -\frac{124}{35}Z_{7,3,5,3} - \frac{124}{35}Z_{7,3,3,5} & \\ -\frac{3282}{875}Z_{5,5,5,3} &= -\frac{8192}{3375}A_{11,7} + \cdots \end{aligned} \tag{C.5}$$

$$Z_{6,4,3,3,1,1} = -\frac{392}{27}A_{7,5,3,3} + \cdots \tag{C.6}$$

$$\begin{aligned} Z_{8,6,3,1,1} - \frac{61}{7}Z_{7,3,3,3,3} & \\ +\frac{1774}{175}Z_{5,5,3,3,3} + \frac{2}{5}Z_{5,3,5,3,3} &= \frac{647168}{34263}A_{7,7,5} + \frac{45056}{1215}A_{9,7,3} + \cdots \end{aligned} \tag{C.7}$$

$$\begin{aligned} Z_{6,6,5,1,1} + 13Z_{7,3,3,3,3} & \\ -\frac{268}{25}Z_{5,5,3,3,3} + \frac{6}{5}Z_{5,3,5,3,3} &= -\frac{3598336}{125631}A_{7,7,5} - \frac{759808}{4455}A_{9,7,3} + \cdots \end{aligned} \tag{C.8}$$

$$\begin{aligned} Z_{10,8,1,1} - \frac{13}{2}Z_{11,3,3,3} - \frac{304}{45}Z_{9,3,3,5} & \\ -\frac{3601}{525}Z_{9,3,5,3} - \frac{3799}{525}Z_{7,3,5,5} & \\ +\frac{1371}{196}Z_{7,7,3,3} + \frac{163}{2450}Z_{7,5,5,3} &= -\frac{16384}{6615}A_{11,9} + \cdots \end{aligned} \tag{C.9}$$

$$Z_{6,4,3,5,1,1} - \frac{68}{5}Z_{5,3,3,3,3,3} = -\frac{118784}{243}A_{9,5,3,3} - \frac{2560}{243}A_{7,5,3,5} + \cdots \tag{C.10}$$

$$Z_{8,4,3,3,1,1} - \frac{28}{5}Z_{5,3,3,3,3,3} = \frac{32768}{81}A_{9,5,3,3} - \frac{10240}{2187}A_{7,5,3,5} + \cdots \tag{C.11}$$

$$\begin{aligned} Z_{8,6,5,1,1} - \frac{68}{9}Z_{9,3,3,3,3} - \frac{832}{105}Z_{7,3,5,3,3} & \\ -\frac{967}{105}Z_{7,3,3,5,3} - \frac{1042}{105}Z_{7,3,3,3,5} & \\ -\frac{13182}{875}Z_{5,5,5,3,3} - \frac{6}{7}Z_{5,5,3,5,3} &= -\frac{194240512}{9628875}A_{9,7,5} \\ &\quad -\frac{229376}{1125}A_{11,7,3} \\ &\quad -\frac{80972546048}{337010625}A_{11,5,5} + \cdots \end{aligned} \tag{C.12}$$

$$\begin{aligned} Z_{10,4,5,1,1} - \frac{46}{9}Z_{9,3,3,3,3} - \frac{67}{21}Z_{7,3,5,3,3} & \\ -\frac{73}{21}Z_{7,3,3,5,3} - \frac{79}{21}Z_{7,3,3,3,5} & \\ -\frac{482}{175}Z_{5,5,5,3,3} - \frac{46}{175}Z_{5,5,3,5,3} &= +\frac{15966208}{641925}A_{9,7,5} + \frac{32768}{2025}A_{11,7,3} \\ &\quad -\frac{1691951104}{67402125}A_{11,5,5} + \cdots \end{aligned} \tag{C.13}$$

$$
\begin{aligned}
Z_{10,6,3,1,1} - \frac{46}{9}Z_{9,3,3,3,3} - \frac{632}{105}Z_{7,3,5,3,3}& \\
-\frac{86}{15}Z_{7,3,3,5,3} - \frac{572}{105}Z_{7,3,3,3,5}& \\
-\frac{4792}{875}Z_{5,5,5,3,3} + \frac{46}{175}Z_{5,5,3,5,3} &= +\frac{124608512}{9628875}A_{9,7,5} + \frac{16384}{10125}A_{11,7,3} \\
&\quad -\frac{758235136}{48144375}A_{11,5,5} + \cdots \qquad \text{(C.14)} \\
Z_{6,4,3,3,3,1,1} &= -\frac{5120}{81}A_{7,5,3,3,3} + \cdots \qquad \text{(C.15)} \\
Z_{12,8,1,1} + \frac{13598459235}{18816311591}Z_{7,5,7,3}& \\
-\frac{9790486696}{6109192075}Z_{9,3,5,5} - \frac{3021879830}{2688044513}Z_{9,3,7,3}& \\
-\frac{560739181022}{201603338475}Z_{9,5,3,5} - \frac{19968330538}{13440222565}Z_{9,5,5,3}& \\
-\frac{66543918797}{40320667695}Z_{9,7,3,3} + \frac{2598592817}{707380135}Z_{11,3,3,5}& \\
-\frac{3186058443}{2688044513}Z_{11,3,5,3} - \frac{20352278271}{13440222565}Z_{11,5,3,3}& \\
+\frac{7925677546}{1221838415}Z_{13,3,3,3} &= -\frac{524288}{212625}A_{13,9} + \cdots \qquad \text{(C.16)}
\end{aligned}
$$

The $+\cdots$ indicates terms that are purely MZVs of lower depth or products of lower weight MZVs. The complete relations can have up to about 150 terms. Hence we give them in a file in the data mine. The first 15 of these relations were derived with the help of `PSLQ` and/or the `LLL` algorithm. Seven of them could be derived with the data mine. Unfortunately for depth $\mathsf{d} = 5$ objects we have only exact results up to weight $\mathsf{w} = 17$ and for depth $\mathsf{d} = 4$ we have only exact results up to weight $\mathsf{w} = 22$.

The above results used the available resources to their limit. The formula in (C.15) needed 45 hours of running time using the `LLL` algorithms as implemented in `PARI` in a 152 parameter search at 8000 digits and was checked afterwards at 10000 digits.

We have expressed the pushdowns in terms of the A-function that has the same indices as the element of L_w that was extended. It is not clear whether this scheme can be maintained for pushdowns beyond the ones we present. Some A-functions cannot be used because they express directly in terms of equal or lower depth MZVs. This then has again influence on the selection of the basis. In the end it may be that we have to drop one or more requirements for the basis. A simple example of such an A-function exists already at weight $\mathsf{w} = 15$:

$$
\begin{aligned}
A_{7,3,5} &= +\frac{7649}{143360}Z_{7,3,5} - \frac{7089}{143360}\zeta_5 Z_{7,3} - \frac{2097}{71680}\zeta_5^3 - \frac{3429}{5120}\zeta_7 Z_{5,3} \\
&\quad -\frac{116396017}{2867200}\zeta_{15} + \frac{1083797}{40960}\zeta_2\zeta_{13} + \frac{81059}{71680}\zeta_2^2\zeta_{11} \\
&\quad -\frac{110993}{627200}\zeta_2^3\zeta_9 - \frac{43311}{448000}\zeta_2^4\zeta_7 - \frac{27831}{78400}\zeta_2^5\zeta_5 \,. \qquad \text{(C.17)}
\end{aligned}
$$

It is also possible to express each pushdown in terms of a single Euler sum rather than an A-function. In a sense this is less telling. After all the A-function contains half of the terms of the doubling relation and the doubling relations seem to be at the origin of the pushdowns. Also we could not find much structure concerning which Euler sum(s) to select. There are often many possibilities. In the case of the A-functions one can make a unique selection: the A-function should have the same index field as the element of the set L_w that represents the pushdown. Anyway, for completeness we give here a single Euler sum for each of the pushdowns. We have dropped all factors and terms which have MZVs of the same weight or products of MZVs with lower weight.

		H-representation	Z-representation
$A_{7,5}$	$\rightarrow$	$H_{-9,3}$	$Z_{-9,-3}$
$A_{7,5,3}$	$\rightarrow$	$H_{-6,-3,6}$	$Z_{-6,3,-6}$
$A_{9,7}$	$\rightarrow$	$H_{-13,3}$	$Z_{-13,-3}$
$A_{7,7,3}$	$\rightarrow$	$H_{-6,-5,6}$	$Z_{-6,5,-6}$
$A_{11,7}$	$\rightarrow$	$H_{-15,3}$	$Z_{-15,-3}$
$A_{7,5,3,3}$	$\rightarrow$	$H_{6,-5,4,3}$	$Z_{6,-5,-4,3}$
$A_{9,7,3}$	$\rightarrow$	$H_{-8,-3,8}, H_{-6,-7,6}$	$Z_{-8,3,-8}, Z_{-6,7,-6}$
$A_{7,7,5}$	$\rightarrow$	$H_{-8,-3,8}, H_{-6,-7,6}$	$Z_{-8,3,-8}, Z_{-6,7,-6}$
$A_{11,9}$	$\rightarrow$	$H_{-17,3}$	$Z_{-17,-3}$
$A_{7,5,3,5}$	$\rightarrow$	$H_{8,-5,4,3}, H_{6,-5,6,3}$	$Z_{8,-5,-4,3}, Z_{6,-5,-6,3}$
$A_{9,5,3,3}$	$\rightarrow$	$H_{8,-5,4,3}, H_{6,-5,6,3}$	$Z_{8,-5,-4,3}, Z_{6,-5,-6,3}$
$A_{9,7,5}$	$\rightarrow$	$H_{-8,-5,8}, H_{-6,-9,6}, H_{-8,-3,10}$	$Z_{-8,5,-8}, Z_{-6,9,-6}, Z_{-8,3,-10}$
$A_{11,5,5}$	$\rightarrow$	$H_{-8,-5,8}, H_{-6,-9,6}, H_{-8,-3,10}$	$Z_{-8,5,-8}, Z_{-6,9,-6}, Z_{-8,3,-10}$
$A_{11,7,3}$	$\rightarrow$	$H_{-8,-5,8}, H_{-6,-9,6}, H_{-8,-3,10}$	$Z_{-8,5,-8}, Z_{-6,9,-6}, Z_{-8,3,-10}$
$A_{7,5,3,3,3}$	$\rightarrow$	$H_{3,-6,-3,6,3}$	$Z_{3,-6,3,-6,3}$
$A_{13,9}$	$\rightarrow$	$H_{-19,3}$	$Z_{-19,-3}$

Of course more complete results can be found in the data mine.

References

[1] L. Euler, *Meditationes circa singulare serium genus*, Novi Comm. Acad. Sci. Petropol. **20** (1775) 140–186, reprinted in Opera Omnia ser I vol. 15, (B.G. Teubner, Berlin, 1927), 217–267.

[2] D. Zagier, *Values of zeta functions and their applications*, in : First European Congress of Mathematics, Vol. II, (Paris, 1992), Progr. Math., **120**, (Birkhäuser, Basel–Boston, 1994), pp. 497–512.

[3] For an extended list of references, see : M.E. Hoffman's page `http://www.usna.edu/Users/math/∼meh/biblio.html`.

[4] P. H. Fuss (ed.), *Correspondance Mathématique et Physique de quelques célèbres Géomètres* (Tome 1), St. Petersburg, 1843;
N. Nielsen, *Die Gammafunktion* (Chelsea, New York, 1965), Reprint of *Handbuch der Theorie der Gammafunktion* (Teubner, Leipzig, 1906).

[5] S. Fischler, *Irrationalité de valeurs de zéta*, Sém. Bourbaki, Novembre 2002, exp. no. **910**, Asterisque **294** (2004) 27–62, `http://www.math.u-psud.fr/∼fischler/publi.html`;
P. Colmez, *Arithmetique de la fonction zêta*, in: Journées X-UPS 2002. La fontion zêta. Editions de l'Ecole polytechnique, Paris, 2002, 37–164, `http://www.math.polytechnique.fr/xups/vol02.html`;
M. Waldschmidt, *Multiple Polylogarithms: an Introduction*, Number Theory and Discrete Mathematics, Editors: A.K. Agarwal, B.C. Berndt, C.F. Krattenthaler, G.L. Mullen, K. Ramachandra and M. Waldschmidt, (Hindustan Book Agency, 2002), 1–12;
M. Waldschmidt, *Valeurs zêta multiples. Une introduction*, Journal de théorie des nombres de Bordeaux, **12** (2) (2000) 581–595;
M. Huttner and M. Petitot, *Arithmeétique des fonctions d'zetas et Associateur de Drinfel'd*, (UFR de Mathématiques, Lille, 2005);
C. Hertling, AG Mannheim-Heidelberg, SS2007.

[6] P. Cartier, *Fonctions polylogarithmes, nombres polyzêtas et groupes pro-unipotents*, Sém. Bourbaki, Mars 2001, 53*e* année, exp. no. **885**, Asterisque **282** (2002) 137–173.

[7] V.V. Zudilin, *Algebraic relations for multiple zeta values*, Uspekhi Mat. Nauk **58** (1) 3–22.

[8] R. Barbieri, J. A. Mignaco and E. Remiddi, *Electron form-factors up to fourth order. 1*, Nuovo Cim. A **11** (1972) 824–864;
M. J. Levine, E. Remiddi and R. Roskies, *Analytic Contributions To The G Factor Of The Electron In Sixth Order*, Phys. Rev. D **20** (1979) 2068–2076;
A. Devoto and D. W. Duke, *Table Of Integrals And Formulae For Feynman Diagram Calculations*, Riv. Nuovo Cim. **7N6** (1984) 1–39.

[9] A. Gonzalez-Arroyo, C. Lopez and F. J. Yndurain, *Second Order Contributions To The Structure Functions In Deep Inelastic Scattering. 1. Theoretical Calculations*, Nucl. Phys. B **153** (1979) 161–186.

[10] J. A. M. Vermaseren, *Harmonic sums, Mellin transforms and integrals*, Int. J. Mod. Phys. A **14** (1999) 2037–2076, [arXiv:hep-ph/9806280].

[11] J. Blümlein and S. Kurth, *Harmonic sums and Mellin transforms up to two-loop order*, Phys. Rev. D **60** (1999) 014018, 31 p., [arXiv:hep-ph/9810241].

[12] D. J. Broadhurst, *On the enumeration of irreducible k-fold Euler sums and their roles in knot theory and field theory*, arXiv:hep-th/9604128.

[13] D. J. Broadhurst and D. Kreimer, *Association of multiple zeta values with positive knots via Feynman diagrams up to 9 loops*, Phys. Lett. B **393** (1997) 403–412, [arXiv:hep-th/9609128].

[14] A.B. Goncharov, *Multiple polylogarithms and mixed Tate motives*, arxiv:math.AG/0103059;
T. Terasoma, *Mixed Tate Motives and Multiple Zeta Values*, Invent. Math. **149** (2) (2002) 339–369, arxiv:math.AG/010423;
P. Deligne and A.B. Goncharov, *Groupes fondamentaux motiviques de Tate mixtes*, Ann. Sci. Ecole Norm. Sup., Série IV **38** (1) (2005) 1–56.

[15] H.R.P. Ferguson and D.H. Bailey, D. H. *A Polynomial Time, Numerically Stable Integer Relation Algorithm*, RNR Techn. Rept. RNR-91-032, Jul. 14, 199;
D. H. Bailey and D.J. Broadhurst, *Parallel integer relation detection: techniques and applications*, Math. Comp. 70 (2001), no. 236, 1719–1736 (electronic), [arxiv: math.NA/9905048].

[16] A. K. Lenstra, H. W. Lenstra, and L. Lovasz, *Factoring Polynomials with Rational Coefficients*, Math. Ann. **261** (1982) 515-534.

[17] H. Poincaré, *Sur les groupes d'équations linéaires*, Acta Math. **4** (1984) 201–311;
J.A. Lappo-Danielevsky, *Mémoirs sur la Théorie des Systèmes Différentiels Linéaires*, (Chelsea, New York, 1953);
K.T. Chen, *Formal differential equations*, Ann. of Math. (2) **73** (1961) 110–133; *Iterated integrals of differential forms and loop space homology*, Ann. Math. **97** (1973) 217–246.

[18] E. Remiddi and J. A. M. Vermaseren, *Harmonic polylogarithms*, Int. J. Mod. Phys. A **15** (2000) 725–754, [arXiv:hep-ph/9905237].

[19] J.M. Borwein, D.M. Bradley, and D.J. Broadhurst, *Evaluation of k-fold Euler/Zagier sums: a compendium of results for arbitrary k*, Elec. J. Combin. **4** (1997) no.2 #R5, hep-th/9611004.

[20] http://www.nikhef.nl/∼form/datamine/datamine.html

[21] J. A. M. Vermaseren, *New features of FORM*, arXiv:math-ph/0010025.

[22] M. Tentyukov and J. A. M. Vermaseren, *The multithreaded version of FORM*, arXiv:hep-ph/0702279.

[23] S. Moch, J. A. M. Vermaseren and A. Vogt, *The three-loop splitting functions in QCD: The non-singlet case*, Nucl. Phys. B **688** (2004) 101–134, [arXiv:hep-ph/0403192];
A. Vogt, S. Moch and J. A. M. Vermaseren, *The three-loop splitting functions in QCD: The singlet case*, Nucl. Phys. B **691** (2004) 129–181, [arXiv:hep-ph/0404111];
J. A. M. Vermaseren, A. Vogt and S. Moch, *The third-order QCD corrections to deep-inelastic scattering by photon exchange*, Nucl. Phys. B **724** (2005) 3–182, [arXiv:hep-ph/0504242].

[24] J. Blümlein, A. De Freitas, W. L. van Neerven and S. Klein, *The longitudinal heavy quark structure function $F_L^{Q\overline{Q}}$ in the region $Q^2 \gg m^2$ at $O(\alpha_s^3)$*, Nucl. Phys. B **755** (2006) 272 [arXiv:hep-ph/0608024];
I. Bierenbaum, J. Blümlein and S. Klein, *Mellin Moments of the $O(\alpha_s^3)$ Heavy Flavor Contributions to unpolarized Deep-Inelastic Scattering at $Q^2 \gg m^2$ and Anomalous Dimensions*, arXiv:0904.3563 [hep-ph], Nucl. Phys. **B820** (2009) 417;
J. Blümlein, M. Kauers, S. Klein and C. Schneider, *Determining the closed forms of the $O(a_s^3)$ anomalous dimensions and Wilson coefficients from Mellin moments by means of computer algebra*, arXiv:0902.4091 [hep-ph]; Comp. Phys. Commun. (2009) in print;

[25] J. Blümlein, *Structural Relations of Harmonic Sums and Mellin Transforms up to Weight w = 5*, arXiv:0901.3106 [hep-ph]; Comp. Phys. Commun. (2009) in print;
J. Blümlein and S. Klein, *Structural Relations between Harmonic Sums up to w=6*, arXiv:0706.2426 [hep-ph], 5 p.

[26] J. Blümlein, *Structural Relations of Harmonic Sums and Mellin Transforms at Weight w=6*, arXiv:0901.0837 [math-ph], Clay Mathematical Institute Proceedings, in print.

[27] D. J. Broadhurst, J. A. Gracey and D. Kreimer, *Beyond the triangle and uniqueness relations: Non-zeta counterterms at large N from positive knots*, Z. Phys. C **75** (1997) 559–574, [arXiv:hep-th/9607174].

[28] D. J. Broadhurst, *Massive 3-loop Feynman diagrams reducible to SC* primitives of algebras of the sixth root of unity,* Eur. Phys. J. C **8** (1999) 311–333, [arXiv:hep-th/9803091].

[29] Y. Andre, *Ambiguity Theory, Old and New*, to appear in: Bollettino U.M.I. (8) **I** (2008) and talk at the Motives, Quantum Field Theory, and Pseudodifferential Operators Boston University, June 2–13, 2008.

[30] F. Brown, *The massless higher-loop two-point function*, Commun. Math. Phys. **287** (2009) 925-958, [arXiv:0804.1660 [math.AG]].

[31] M. Kontsevich and D. Zagier, *Periods*, Mathematics Unlimited-2001 and Beyond, (Springer, Berlin, 2001), 771–808.

[32] C. Bogner and S. Weinzierl, J. Math. Phys. **50** (2009) 042302 [arXiv:0711.4863 [hep-th]].

[33] L. Lewin, *Polylogarithms and Associated Functions*, (North Holland, New York, 1981).

[34] N. Nielsen, *Der Eulersche Dilogarithmus und seine Verallgemeinerungen*, Nova Acta Leopoldina, **XC** (3)(1909) 125–211;
K. S. Kölbig, *Nielsen's Generalized Polylogarithms*, SIAM J. Math. Anal. **17** (1986) 1232–1258.

[35] S. Moch, P. Uwer and S. Weinzierl, *Nested sums, expansion of transcendental functions and multi-scale multi-loop integrals*, J. Math. Phys. **43** (2002) 3363–3386, [arXiv:hep-ph/0110083].

[36] A.B. Goncharov, *Multiple polylogarithms, cyclotomy and modular complexes*, Math. Res. Lett. **5** (1998) 497–516.

[37] J. Blümlein, *Algebraic relations between harmonic sums and associated quantities*, Comput. Phys. Commun. **159** (2004) 19–54, [arXiv:hep-ph/0311046].

[38] R.C. Lyndon, *On Burnsides problem*, Trans. Amer. Math. Soc. **77** (1954) 202–215; *On Burnsides problem II*, Trans. Amer. Math. Soc. **78** (1955) 329–332;
C. Reutenauer, *Free Algebras*, (Calendron Press, Oxford, 1993).

[39] M.E. Hoffman, *Algebraic Aspects of Multiple Zeta Values*, in: Zeta Functions, Topology and Quantum Physics, Developments in Mathematics, **14**, T. Aoki et. al. (eds.), Springer, New York, 2005, pp. 51–74, arXiv:math/0309452 [math.QA].

[40] M.E. Hoffman, *Multiple harmonic series*, Pacific J. Math. **152** (1992) 275–290.

[41] A. Hurwitz, *Einige Eigenschaften der Dirichlet'schen Funktionen $F(s) = \sum(D/n) \cdot 1/(n^s)$, die bei der Bestimmung der Klassenanzahlen Binärer quadratischer Formen auftreten*, Z. für Math. und Physik **27** (1882) 86–101;
H. Hasse, *Ein Summierungsverfahren für die Riemannsche ζ–Reihe*, Math. Z. **32** (1930) 458–464.

[42] Leonardus Pisanus de filiis Bonaccij, *Liber abaci*, Cap. 12.7, (Pisa, 1202);
L.E. Sigler, *Fibonacci's Liber Abaci*, (Springer, Berlin, 2002).

[43] see http://en.wikipedia.org/wiki/Padovan_sequence;
http://www.emis.de/journals/NNJ/conferences/N2002-Padovan.html

[44] E. Witt, *Treue Darstellung Liescher Ringe*, Journ. Reine Angew. Mathematik **177** (1937) 152–160. *Die Unterring der freien Lieschen Ringe*, Math. Zeitschr. **64** (1956) 195–216.

[45] R. Perrin, *Item 1484*, L'Intermédiare des Math. **6**, (1899) 76–77;
A. Williams and D. Shanks, *Strong primality tests that are not sufficient*, Mathematics of Computation **39** (159)(1982) 255–300.

[46] E. Lucas, *Théorie des fonctions numériques simplement périodiques*, American Journal of Mathematics **1** (1878) 197–240.

[47] J. M. Borwein, D. M. Bradley, D. J. Broadhurst and P. Lisonek, *Special Values of Multiple Polylogarithms*, Trans. Am. Math. Soc. **353** (2001) 907–941, [arXiv:math/9910045]

[48] J.M. Borwein and R. Girgensohn, *Evaluation of Triple Euler Sums*, The electronic journal of combinatorics **3** (1996), #R23.

[49] C. Markett, *Triple sums and the Riemann zeta function*. J. Number Theory 48 (1994), 113–132.

[50] M. Espie, J.-C. Novelli, and G. Racinet, *Formal Computations about Multiple Zeta Values*, Preprint SFB-478 Univ. Münster Heft 260 (2003), in : From Combinatorics to Dynamical Systems, Strasbourg, 2002, IMRA Lect. Math. Theor. Phys. **3**, F. Fauvet and C. Mitschi (eds.), (de Gryter, Berlin, 2003), 1–16.

[51] M. Kaneko, M. Noro and K. Tsurumaki, *On a conjecture for the dimension of the space of the multiple zeta values*, Software for Algebraic Geometry, IMA 148 (2008), 47–58.

[52] M.E. Hoffmann and Y. Ohno, *Relations of multiple zeta values and their algebraic expressions*, J. Algebra **262** (2003) 332–347.

[53] Sun Tzu, *Sun Zi suanjing*, 3rd century AD (between 280 and 473 AD);
re-publshed by Qin Jiushao in: *Shushu Jiuzhang*, 1247.

[54] R. Gastmans and W. Troost, *On the evaluation of polylogarithmic integrals*, Simon Stevin, **55** (1981), 205–219, KUL-TF-80/10, MR 0647134 (83c:65028).

[55] D.J. Broadhurst, Data Base to $w = 9$, privately distributed;
D. Zagier, private communication.

[56] The PARI/GP page: http://pari.math.u-bordeaux.fr/

[57] D.H. Bailey, J.M. Borwein, and R. Girgensohn, *Experimental Evaluation of Euler Sums*, Experiment. Math. **3** (1994), 17–30.

[58] M. Bigotte, G. Jacob, N.E. Oussous, and M. Petitot, *Tables des relations de la fonction zéta colorée avec 1 racine*, LIFL USTL Lille preprint IT–332, Nov. (1998). Non-alternating part: pp. 1–227; alternating part: pp. 1–274. Only the finite values are tabulated.

[59] M. Bigotte, G. Jacob, N.E. Oussous, and M. Petitot, *Lyndon words and shuffle algenras for generating the coloured multiple zeta values relation tables,* Theor. Comp. Sci. **273** (2002) 271–282.

[60] H.N. Minh and M. Petitot, *Lyndon words, polylogarithms and the Riemann ζ function,* Discrete Maths. **217** (2000) 273–292.

[61] El Wardi - mémoire DEA, Lille 1, Juillet, 1999.

[62] H.N. Minh, G. Jacob, N.E. Oussuous and M. Petitot, *Aspects combinatoires des polylogarithmes et des sommes d'Euler-Zagier,* J. Électr. Sém. Lothar. Combin. **43** (2000), Art. B43e, 29 pp.

[63] J.A.M. Vermaseren, `http://www.nikhef.nl/~t68/FORMapplication`, Nov. 2003.

[64] H.N. Minh, private communication, June 2008.

[65] J.A.M. Vermaseren, `http://www.nikhef.nl/~form`.

[66] J.A.M. Vermaseren, Tuning form with large calculations. Nucl. Phys. Proc. Suppl. 116: 343-347,2003.

[67] J.A.M.Vermaseren and J.Vollinga, in prepartion.

[68] G. Racinet, *Séries génératrices non-commutatives de polyzêtas et associateurs de Drinfel'd*, Ph.D. Thesis, Amiens, France, 2000.

[69] M.E. Hoffman, *The Algebra of Multiple Harmonic Series*, Journ. of Algebra **194** (1997) 477–495.

[70] D. Bowman and D. M. Bradley, *The algebra and combinatorics of shuffles and multiple zeta values*, J. Combin. Theory Ser. **A97** (2002) 43=-61.

[71] J. Zhao, *Double Shuffle Relations of Euler Sums*, `arXiv:0705.2267 [math.NT]`.

[72] M. E. Hoffman and C. Moen, *Sums of triple harmonic series*, J. Number Theory 60 (1996), 329–331;
A. Granville, *A decomposition of Riemann's zeta-function*, in Analytic Number Theory, London Mathematical Society Lecture Note Series 247, Y. Motohashi (ed.), (Cambridge University Press, Cambridge, 1997) pp. 95–101;
D. Zagier, Multiple zeta values, preprint.

[73] J. Okuda and K. Ueno, *The Sum Formula of Multiple Zeta Values and Connction Problem of the Formal Knizhnik–Zamolodchikov Equation*, in: Zeta Functions, Topology and Quantum Physics, Developments in Mathematics, **14**, T. Aoki et. al. (eds.), Springer, New York, 2005, pp. 145–170, `arXiv:math/0310259 [math.NT]`.

[74] K. Ihara and M. Kaneko, *A note on relations of multiple zeta values*, preprint.

[75] T.Q.T. Le and J. Murakami, *Kontsevich's integral for the Homfly polynomial and relations between values of multiple zeta functions*, Topology Appl. **62** (1995) 193–206.

[76] Y. Ohno, *A generalization of the duality and sum formulas on the multiple zeta values*, J. Number Theory, **74** (1999) 189–209.

[77] Y. Ohno and D. Zagier, *Multiple zeta values of fixed weight, depth, and height*, Indag. Math. (N.S.) **12** (2001) 483–487.

[78] Y. Ohno and N. Wakabayashi, *Cyclic sum of multiple zeta values,'* Acta Arithmetica **123** (2006), 289–295.

[79] K. Ihara, M. Kaneko, and D.Zagier, *Derivation and double shuffle relations for multiple zeta values*, Compositio Math. **142** (2006) 307–338; preprint MPIM2004-100.

[80] J. Écalle, *Théorie des moules*. 3 vol, prépublications mathématiques d'Orsay, 1981, 1982, 1985; *La libre génération des multicêtas et leur d'ecomposition canonico-explicite en irréductibles*, (automne 1999); *Ari/gari et la décomposition des multizêtas en irréductibles*, Prépublication, avril 2000.

[81] J. Zhao, *Linear Relations of Special Values of Multiple Polylogarithms at Roots of Unity*, `arXiv:0707.1459 [math.NT]`.

[82] H. Tsunogai, *On ranks of the stable derivation algebra and Delinge's problem*, Proc. Japan Acad. Ser. A Math. Sci. **73** (1997) 29–31;
H. Furusho, *The Multiple Zeta Value Algebra and the Stable Derivation Algebra*, Publ. RIMS, Kyoto Univ. **39** (2003) 695–720.

[83] G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers*, (Calendron Press, Oxford, 2002).