Euler Sums and Multiple Zeta Values

J.A.M. Vermaseren

Nikhef and Humboldt foundation

- Introduction
- Notations
- MZVs and Euler sums.
- Relations
- Euler Sums
- Runs for MZVs
- Data mining
- A push down basis
- Conclusions

Work with J. Blümlein and D. Broadhurst.

Introduction

Lately harmonic sums have become popular in particle physics. Several categories of problems have found solutions by converting them from an integration problem to a summation problem. Two examples are:

- The QCD three loop anomalous dimensions in deep inelastic scattering.
- The master integrals for two loop Bhabha scattering were solved by a Mellin Barnes transformation, doing the then easy integrals and then solving the sums.

When one is confronted with such new techniques there is much to be discovered. And as it turns out, the mathematicians can help us only a little bit. This means that much of the discovering will have to be done by physicists.

In the framework of this talk we will concentrate on the results of sums to infinity. Many of these are the result of calculating dimensionless objects or basic Feynman diagrams, or a byproduct of (inverse) Mellin transforms. The idea here is that if we don't understand those objects sufficiently, then we will never understand the more complicated multi-parameter sums.

In other words: if you don't understand the solution space, don't start calculating.

Many insights find their origin in looking at data. Hence we will attack the problem with the application of brute force to generate as many results as technology will allow us.

To apply this brute force we have to derive some new equations and construct a relatively simple but yet amazingly powerful computer program.

The running of the programs gives us many Gigabytes of relations. We study the outputs and try to see some new patterns in them. We do find some.

This field is full of conjectures. We will add a few of our own.

For many results presented in this talk, we have made heavy use of symbolic computation. This has been done with the system FORM and its variety TFORM. TFORM is a parallel version which can use many threads simultaneously. The bigger runs were done on a computer with 8 Xeon cores, running about 7 times faster than on a single core. Also (T)FORM isn't much limited by the size of the CPU memory (which was 32 Gbytes) as it can use the disk rather efficiently (which was 4 Tbytes). Another advantage is that FORM has a very compact data representation and is (not only because of that) very fast. This comes in handy when you have to manipulate expressions of 10^9 terms.

Our worst example:

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

Notations

Physicists define harmonic sums by:

$$
\begin{aligned}
S_m(N) &= \sum_{i=1}^{N} \frac{1}{i^m} \\
S_{-m}(N) &= \sum_{i=1}^{N} \frac{(-1)^i}{i^m} \\
S_{m,m_2,\cdots,m_p}(N) &= \sum_{i=1}^{N} \frac{1}{i^m} S_{m_2,\cdots,m_p}(i) \\
S_{-m,m_2,\cdots,m_p}(N) &= \sum_{i=1}^{N} \frac{(-1)^m}{i^m} S_{m_2,\cdots,m_p}(i)
\end{aligned}
$$

Mathematician use mostly $i-1$ for the argument of the S in the recursive formula. Those sums we call Z-sums.

Sums that involve negative indices we call Euler sums and the ones that have only positive indices we call Multiple Zeta Values (MZVs).

The S-sums and the Z-sums can be converted into each other. In any decent symbolic system this is easily programmable.

Related functions are the harmonic polylogarithms (Hpl) which are defined by:

$$\begin{aligned} H(0;x) &= \ln x \\ H(1;x) &= \int_0^x \frac{dx'}{1-x'} = -\ln(1-x) \\ H(-1;x) &= \int_0^x \frac{dx'}{1+x'} = \ln(1+x) \end{aligned}$$

and the functions

$$f(0;x) = \tfrac{1}{x}, \qquad f(1;x) = \tfrac{1}{1-x}, \qquad f(-1;x) = \tfrac{1}{1+x}$$

If $\vec{a}_w$ is an array with W elements, all with value a, then:

$$\begin{aligned} H(\vec{0}_w;x) &= \frac{1}{W!}\ln^W x \\ H(a,\vec{m}_w;x) &= \int_0^x dx'\ f(a;x')\ H(\vec{m}_w;x') \end{aligned}$$

These functions are related by Mellin transforms:

$$M(f,N) = \int_0^1 dx\, x^N\, f(x)$$

This is one way by which the harmonic sums enter in field theory, but in the context of this talk, that is not what we are interested in. What we ARE interested in is that the Hpl's in one are related to the harmonic sums in infinity.

We can define a unified notation as in:

$$\begin{aligned} H_{0,0,1,0,-1} &= H_{3,-2} \\ S_{7,-2,1} &= S_{0,0,0,0,0,0,1,0,-1,1} \end{aligned}$$

The notation with the $0, 1, -1$ we call integral notation and the other notation we call sum notation. The number of indices in the integral notation is the weight, and the number of indices in the sum notation is the depth.

For MZVs we have $Z_{\vec{p}}(\infty) = H_{\vec{p}}(1)$ With Euler sums there can be signs.

Again: trivially programmable.

We will usually omit the argument of the S, Z and H functions. This means that they are taken in ∞, ∞ and one respectively.

Multiple Zeta Values and Euler sums

The sums in infinity (or the Hpl's in one) are called Multiple Zeta Values when all indices are positive. When there are negative indices the sums are called either alternating sums or Euler sums. There has been a sudden interest inthem by mathematicians in the 1990's, but then mainly the MZVs. We (=physicists) also need the Euler sums. Gastmans and Troost (1981) managed to give the relations for all weight 4 and a number of weight 5 sums. We need basically weight = 2× (number of loops). The number of Euler sums that exists is $2 \times 3^{W-1}$ and there are 2^{W-1} MZVs. If we remove the divergent sums these numbers become $4 \times 3^{W-2}$ and 2^{W-2} respectively. This means that for weight 6 there are 324 (16) constants that we have to determine. One of them:

$$Z_{-4,-2} = -H_{-4,2} = \frac{97}{420}\zeta_2^3 - \frac{3}{4}\zeta_3^2$$

Example of an inverse Mellin transform of a weight 6 harmonic sum. We omit the relations that exist between the Euler sums.

```
#define SIZE "6"
#include- harmpol.h
Off statistics;
.global
Local F = S(R(-1,3,-2),N);
#call invmel(S,N,H,x)
Print +f +s;
.end
```

```
F =
    - sign_(N)*H(R(1,0,0),x)*Htab2(0,-1)*[1+x]^-1
    - sign_(N)*Htab5(0,-1,0,0,-1)*[1+x]^-1
    - sign_(N)*Htab5(0,-1,0,0,1)*[1+x]^-1
    + sign_(N)*Htab5(0,-1,1,0,0)*[1+x]^-1
    - 2*sign_(N)*Htab5(0,0,-1,0,1)*[1+x]^-1
    - 3*sign_(N)*Htab5(0,0,0,-1,1)*[1+x]^-1
    - 3*sign_(N)*Htab5(0,0,0,1,-1)*[1+x]^-1
```

```
- sign_(N)*Htab5(0,0,1,0,-1)*[1+x]^-1
+ sign_(N)*Htab5(0,1,-1,0,0)*[1+x]^-1
+ sign_(N)*Htab5(0,1,0,-1,0)*[1+x]^-1
+ sign_(N)*Htab5(0,1,0,0,-1)*[1+x]^-1
+ sign_(N)*Htab5(1,0,-1,0,0)*[1+x]^-1
+ 2*sign_(N)*Htab5(1,0,0,-1,0)*[1+x]^-1
+ 3*sign_(N)*Htab5(1,0,0,0,-1)*[1+x]^-1
- H(R(-1),x)*Htab4(0,0,-1,0)*[1-x]^-1
+ H(R(-1,-3,0),x)*[1-x]^-1
- H(R(-1,0),x)*Htab3(0,-1,0)*[1-x]^-1
- H(R(-1,0,0),x)*Htab2(-1,0)*[1-x]^-1
+ 6*Htab5(-1,-1,0,0,0)*[1-x]^-1
+ 5*Htab5(-1,0,-1,0,0)*[1-x]^-1
+ 3*Htab5(-1,0,0,-1,0)*[1-x]^-1
+ 4*Htab5(0,-1,-1,0,0)*[1-x]^-1
+ 3*Htab5(0,-1,0,-1,0)*[1-x]^-1
+ 2*Htab5(0,0,-1,-1,0)*[1-x]^-1
+ Htab5(0,0,-1,0,-1)*[1-x]^-1
+ Htab6(-1,0,-1,0,0,-1)
+ Htab6(-1,0,-1,0,0,1)
+ 2*Htab6(-1,0,0,-1,0,1)
+ 3*Htab6(-1,0,0,0,-1,1)
+ 3*Htab6(-1,0,0,0,1,-1)
```

```
 + Htab6(-1,0,0,1,0,-1)
 + 2*Htab6(0,-1,-1,0,0,-1)
 + 2*Htab6(0,-1,-1,0,0,1)
 + Htab6(0,-1,0,-1,0,-1)
 + 3*Htab6(0,-1,0,-1,0,1)
 + 2*Htab6(0,-1,0,0,-1,-1)
 + 5*Htab6(0,-1,0,0,-1,1)
 + 3*Htab6(0,-1,0,0,1,-1)
 + Htab6(0,-1,0,1,0,-1)
 + 4*Htab6(0,0,-1,-1,0,1)
 + 5*Htab6(0,0,-1,0,-1,1)
 + 3*Htab6(0,0,-1,0,1,-1)
 + Htab6(0,0,-1,1,0,-1)
 + 6*Htab6(0,0,0,-1,-1,1)
 + 3*Htab6(0,0,0,-1,1,-1)
;
```

The Htab objects are Hpl's in one in which for instance Htab6(0,0,0,-1,1,-1) stands for $H_{-4,1,-1}(1)$. These objects are related to the sums in infinity.

And now the same program, but this time the Euler sums are reduced to a set of independent objects:

```
F =
      - 51/32*[1-x]^-1*z5
      + 3/4*[1-x]^-1*z2*z3
      - 7/2*s6
      + 51/32*z5*ln2
      - 33/64*z3^2
      + 9/4*z2*z3*ln2
      + 121/840*z2^3
      - 51/32*sign_(N)*[1+x]^-1*z5
      + 3/4*sign_(N)*[1+x]^-1*z2*z3
      - 1/2*sign_(N)*H(R(1,0,0),x)*[1+x]^-1*z2
      + 21/20*H(R(-1),x)*[1-x]^-1*z2^2
      + H(R(-1,-3,0),x)*[1-x]^-1
      + 3/2*H(R(-1,0),x)*[1-x]^-1*z3
      + 1/2*H(R(-1,0,0),x)*[1-x]^-1*z2
     ;
```

It is clear that reducing the Euler sums to an independent set gives a much shorter answer.

Unfortunately there is no known constructive way to take one of these constants and express it into a basis. Already there are problems in determining what constitutes a good basis.
The only two ways to express them in an independent set that are currently known are:

- Write down all algebraic relations for these objects and solve the system of equations. Then tabulate all MZV/Euler sums and use table substitution afterwards.
- Guess a relation and fit the coefficients with a program like PSLQ or LLL after computing all objects in the relation numerically to a very large number of digits. Broadhurst has done much of this in the 1990's.

There are several conjectures about the size of a basis. The best known are the Zagier conjecture (MZVs), the Broadhurst conjecture (Euler sums) and the Broadhurst-Kreimer conjectures (MZVs).
There even exist conjectures about how to construct some specific bases.
Due to time restrictions we will not dwell on this. We will just say that currently there is no really good basis for the MZVs. The Broadhurst conjecture provides one for the Euler sums.

Relations

The harmonic sums obey a 'stuffle' algebra which is based on properties of sums:

$$\begin{aligned} S_{a,b}(N)S_{c,d}(N) \;=\; & S_{a,b,c,d}(N) + S_{a,c,b,d}(N) + S_{a,c,d,b}(N) \\ & + S_{c,a,b,d}(N) + S_{c,a,d,b}(N) + S_{c,d,a,b}(N) \\ & - S_{a+c,b,d}(N) - S_{a,c+b,d}(N) - S_{a,c,b+d}(N) \\ & - S_{c,a,b+d}(N) - S_{c,a+d,b}(N) + S_{a+c,b+d}(N) \end{aligned}$$

For the Z-sums the minus signs should be replaced by plus signs.
The harmonic polylogarithms obey a 'shuffle' algebra as in

$$\begin{aligned} H_{a,b}H_{c,d} \;=\; & H_{a,b,c,d} + H_{a,c,b,d} + H_{a,c,d,b} \\ & + H_{c,a,b,d} + H_{c,a,d,b} + H_{c,d,a,b} \end{aligned}$$

When we take the limit $N \to \infty$ or $x \to 1$ the sums and the Hpl's can be expressed into each other and we obtain MZV's. Hence the MZV's obey both relations. It should be noted that when there are negative indices the sum of the indices becomes a bit more complicated:

$$a + b \ \longrightarrow \ \sigma_a \sigma_b (|a| + |b|) = \sigma_a b + \sigma_b a$$

in which σ_a is the sign of a.

For the Euler sums there are more relations.

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

They are called the doubling relations. When $n \to \infty$ and the sums are finite, this gives useful relations.

For the Euler sums there is yet another category of relations which we call the generalized doubling relations (GDR's). They are based on similar principles but we have only a computer algorithm to generate them. No closed formula.

The construction and its derivation are described in a (forthcoming) paper. These equations can be lengthy.

It will be necessary to take divergent sums into account. The divergences are rather mild and hence not difficult to regularize. They pose no special problems.

Equipped with the above relations we want to construct a computer program that generates all possible equations and then solves for the MZV's, leaving us in the end with a minimal set as remaining unknowns.

What are we up against?

For the MZVs sums there are 2^{W-3} objects to be determined (there is a duality relation that cuts the number down by (roughly) a factor 2).

We would like to go beyond what M. Kaneko, M. Noro and K. Tsurumaki managed. They treated this as a matrix problem (with a size of $2^{W-3} \times 2^{W-4}$) and went to W=20. Using calculus modulus a 15 bits prime they needed about 18 Gbytes of memory and could not go beyond this.

W	size	time
16	72M	150
17	288M	880
18	1.2G	5000
19	4.6G	33000
20	18G	245000

Parameters of the Kaneko et al program on an 8 core computer.

All the program managed to determine was the size of a basis. The size was according to the Zagier conjecture.
It should be noticed that the matrix is sparse. In our program the weight 20 expression has at its worst 4158478 terms (100 Mbytes) which means that only one in 2000 entries of the matrix would not be zero.

For the Euler sums one needs to calculate $4 \times 3^{W-2}$ objects. Results have been reported in the past for $W = 8$ by the Lille group and $W = 8, 9, 10$ by JV. The results up to $W = 9$ have been available in the FORM distribution.

To $W = 7$ the stuffles and the shuffles suffice. At $W = 8, 9, 10$ it is sufficient to add the doubling relations. Starting at $W = 11$ the generalized doubling relations are needed to obtain a minimal basis that is in accordance with the Broadhurst conjecture.

For $W = 12$ there will be 236196 Euler sums to be determined.

We will run three types of programs.

1. A full expression of all MZV's in a minimal basis.
2. An expression of all MZV's in a minimal basis modulus a prime number. We drop all terms that are products of lower weight objects.
3. An expression of all MZV's in a minimal basis modulus a prime number. We drop all terms that are products of lower weight objects. We consider only elements up to a given depth D.

Euler Sums

The Euler sums need the doubling ($W \geq 8$) and the generalized doubling ($W \geq 11$) formulas. They are also needed if we want to obtain results up to a given depth. Details are in the paper.

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

The size of the outputs becomes a bigger problem than the running time.

We have also runs with restricted depth. The most important ones are where we limit the depth to 6 or less. In this case we have used modular arithmetic and dropped all terms that are products of lower weight objects in an all out attempt to obtain $W = 18, D \leq 6$.

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

The last run was rather impressive. It took one month on an 8 core Xeon machine, working its way through a combined total of more than 7×10^{12} terms or 7 TeraTerms!

Runs to depth 5 are to weight 21 and runs to depth 4 are to weight 30.

Runs for MZVs

In the first sequence of programs we try to see how far we can get. We use a 31 bits prime (2147479273) and try to determine a minimal basis. We drop all terms that are products of lower weight objects. We want expressions for all MZV's of the given weights in terms of the basis.

W	Group	size	output	CPU	time	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

At this point we noticed that all basis elements had a depth that fulfilled $D \leq W/3$. Hence assuming that this will be always the case we made a few more runs. And in addition we made some 'incomplete' runs.

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

It shouldn't come as a great surprise that all the results of the above runs are in agreement with the Zagier and Broadhurst-Kreimer conjectures.

More later.....

We also made complete runs. That is: over the rationals and including products of lower weight objects. This gave the following:

W	size	output	num	CPU	real	Eff.	Rat.
16	10.9M	10.6M	21	254	59	4.29	1.05
17	30M	29M	19	690	149	4.62	1.11
18	86M	77M	25	3491	700	4.98	1.51
19	218M	205M	27	9460	1855	5.10	1.54
20	756M	552M	31	65640	11086	5.92	2.30
21	1.63G	1.55G	39	165561	27771	5.96	2.24
22	8.05G	4.00G	36	2276418	326489	6.97	4.97

It should become clear by now that the size of the output becomes a major obstacle. To store millions of expressions, each of them with quite a number of terms, will take Gigabytes.

```
Fill htable22(0,0,1,0,1,0,1,0,1,0,1,0,0,0,0,0,0,1,1,0,1,1)=229121/1728*
        z14z3z1z1z2z1+173609/576*z14z3z1z2z1z1+15692195/31104*
        z14z3z2z1z1z1+3726961/31104*z14z4z1z1z1z1-56339/1152*z14z5z1z2
        -3378973/13824*z14z5z2z1+1007419717/2488320*z14z6z1z1-3423/16*
        z15z2z1z2z1z1+2073365/1296*z15z3z1z1z1z1-307559/216*z15z4z1z2-
        666657535/165888*z15z4z2z1+2485272541/1658880*z15z5z1z1-
        502565387/31104*z16z2z1z1z1z1-8240323/1728*z16z3z1z2-
        50468588359/3317760*z16z3z2z1-4457267917/829440*z16z4z1z1-
        188177646093889/8599633920*z16z6+193151925403/19906560*
        z17z3z1z1+6998148491689/13271040*z18z2z1z1+5830492751924959/
        6879707136*z18z4-64399622164350811/1911029760*z20z2-1415173/
        43200*z5z3z3^2-141/4*z7z3*z8z2z1z1+15765715/62208*z7z3*z9z3+
        108/25*z5z3*z5z3z3z3+2332219/48600*z5z3*z9z5+654535363/5702400
        *z5z3*z11z3-30606548603/921600*z11*z5z3z3+4674331597474072633/
        57330892800*z11^2+3646960903267/217728000*z9*z5z5z3-
        26283756319/1451520*z9*z7z3z3+227618177777097021133/
        1504935936000*z9*z13-54161081/10368*z7*z10z2z1z1z1-14895806515/
        4644864*z7*z7z5z3+1810659173/497664*z7*z9z3z3+
        1444920424682016255/120394874880*z7*z15+7516571189/1126400*
        z7^2*z5z3-4571/5*z5*z5z3z3z3z3-27702313/5184*z5*z12z2z1z1z1+
        1897913010697639/388949299200*z5*z7z5z5-737558452534697/
        155579719680*z5*z7z7z3-8678023289443/13891046400*z5*z9z5z3+
        65728422985853/11112837120*z5*z11z3z3+185458251647/136857600*
        z5*z9*z5z3+655173768451/34836480*z5*z7*z7z3+
        8980494081229019842921/134420877803520*z5*z17-3819/4*z5^2*
        z8z2z1z1-271512575762737/20836569600*z5^2*z9z3-
        1538354691225468l/55564185600*z5^3*z7-2969/8*z3*z12z4z1z1z1+
        126/25*z3*z5z3z5z3z3-163253/400*z3*z5z5z3z3z3+5677/16*z3*
        z7z3z3z3z3-69740687/10368*z3*z14z2z1z1z1-374706432302269505/
        41015642443776*z3*z7z7z5+559257961960828567/109863327974400*z3
        *z9z5z5-675929428026804667/219726655948800*z3*z9z7z3+
        472645097440330207/97656291532800*z3*z11z5z3+17405218743810383/
        2048733388800*z3*z13z3z3+186/25*z3*z5z3*z5z3z3+560126822557/
        8294400*z3*z11*z5z3+241944929861/4976640*z3*z9*z7z3-48533/32*
        z3*z7*z8z2z1z1+3258424132907/44789760*z3*z7*z9z3+32205/16*z3*
        z5*z5z3z3z3+62730931353098707/4069012147200*z3*z5*z9z5-
        211693794294616819/4882814576640*z3*z5*z11z3-117303745103293/
        164229120*z3*z5*z7^2-3785404660891098517/439453311897 6*z3*z5^2
        *z9-979481944666231474286437 1/109375046516736000*z3*z19-150567/
        1120*z3^2*z5z5z3z3+37369/224*z3^2*z7z3z3z3-76731/64*z3^2*
        z12z2z1z1+5836777489/4257792*z3^2*z11z5-631656298061/56609280*
        z3^2*z13z3-24/5*z3^2*z5z3^2-1476536914610227/4269957120*z3^2*
        z7*z9-940205/1728*z3^2*z5*z5z3z3-63798454917713/181149696*z3^2
        *z5*z11+89314457/907200*z3^3*z5z5z3-4391335/36288*z3^3*z7z3z3-
        102881298198157/1045094400*z3^3*z13+2015873/25920*z3^3*z5*z5z3
        +4771/112*z3^4*z7z3+178901285/1306368*z3^4*z5^2+129247787/
        466560*z3^5*z7-188/5*z2*z5z3z3z3z3z3-838*z2*z14z2z1z1z1z1-
        400090555909/130636800*z2*z7z3z5z5-860982225443/104509440*z2*
        z7z7z3z3-410971121201/87091200*z2*z7z5z5z3+432991955441/
        55987200*z2*z9z3z5z3-5561422085/1119744*z2*z9z5z3z3+
        30038614163/2488320*z2*z11z3z3z3+12317476820806379/11287019520
        *z2*z13z7-4814984387/46656*z2*z16z2z1z1-26973572103166541417/
        3386105856000*z2*z15z5+650628965993715945353/11512759910400*z2
        *z17z3+2703067/16128*z2*z7z3^2+15297217/51840*z2*z5z3*z9z3-
        1967338523/116640*z2*z9*z5z3z3+443971105937439694528 9/
        3837586636800*z2*z9*z11+203331234901/16329600*z2*z7*z5z5z3-
        2245163981/163296*z2*z7*z7z3z3+172861806934439936513/
        213199257600*z2*z7*z13-2530*z2*z5*z10z2z1z1z1+221934828641/
        37324800*z2*z5*z7z5z3-185137871143/18662400*z2*z5*z9z3z3+
        2356857770584504644547037/6120950685696000*z2*z5*z15-
        8784777689/466560*z2*z5*z7*z5z3-29339484871/12441600*z2*z5^2*
        z7z3-946617250799/97977600*z2*z5^4+4388/5*z2*z3*z5z3z3z3z3-
        2050*z2*z3*z12z2z1z1z1-2515919247697/1620622080*z2*z3*z7z5z5-
        5508608353973/1620622080*z2*z3*z7z7z3-65616653437/19293120*z2*
        z3*z9z5z3+4317757951/602910*z2*z3*z11z3z3+2459401/2880*z2*z3*
        z9*z5z3-5826659/2268*z2*z3*z7*z7z3+1685897928474783669523733/
        19824227181158400*z2*z3*z17-3112*z2*z3*z5*z8z2z1z1-
        1913867931511/347276160*z2*z3*z5*z9z3-12126144556601/
        2083656960*z2*z3*z5^2*z7-1086/5*z2*z3^2*z5z3z3z3-4867384441/
        1088640*z2*z3^2*z9z5+71577340969/3991680*z2*z3^2*z11z3+
        11050634658317/143700480*z2*z3^2*z7^2+449759798507/4490640*z2*
        z3^2*z5*z9+128*z2*z3^3*z5z3z3-5793264895/139968*z2*z3^3*z11-
        207/5*z2*z3^4*z5z3-162/5*z2*z3^5*z5+27/5*z2^2*z12z2z1z1z1z1-
        984359/75600*z2^2*z7z5z5z1+2137981343/2721600*z2^2*z5z5z5z3-
        11370756889/1814400*z2^2*z7z5z3z3+1301016437/233280*z2^2*
        z9z3z3z3-7911180517/155520*z2^2*z14z2z1z1+336721679218271/
        4528742400*z2^2*z13z5-630621466648781 29/62705664000*z2^2*z15z3
        +6644509/43200*z2^2*z5z3*z7z3-38514635023952878361/
        1630347264000*z2^2*z9^2-15429815879/1944000*z2^2*z7*z5z3z3-
        8274399031910863279/271724544000*z2^2*z7*z11+4208229059/544320
        *z2^2*z5*z5z5z3-658253387/77760*z2^2*z5*z7z3z3-
        8720289305450158267/952528896000*z2^2*z5*z13-50810851429/
        5443200*z2^2*z5^2*z5z3+999/5*z2^2*z3*z10z2z1z1z1-45306816419/
        2268000*z2^2*z3*z7z5z3+571783303/30375*z2^2*z3*z9z3z3+
        98747576355234045376281 7/127441460450304000*z2^2*z3*z15-
        670666193/72000*z2^2*z3*z7*z5z3-131835349/25920*z2^2*z3*z5*
        z7z3+73744749319/6531840*z2^2*z3*z5^3+1593/10*z2^2*z3^2*
        z8z2z1z1-5617847/40320*z2^2*z3^2*z9z3+113181386863/2177280*
        z2^2*z3^2*z5*z7+186543726721/6531840*z2^2*z3^3*z9-951/100*z2^2
        *z3^6+24711581/15120*z2^3*z5z5z3z3-234965329/136080*z2^3*
        z7z3z3z3-146515315/6048*z2^3*z12z2z1z1-435261786095987/
        7185024000*z2^3*z11z5+2456425078110467/7547904000*z2^3*z13z3+
        12415031/252000*z2^3*z5z3^2+117865176559161139/1046139494400*
        z2^3*z7*z9-226177577/45360*z2^3*z5*z5z3z3+
        5393961686980635863 69/212366317363200*z2^3*z5*z11+1568719081/
        661500*z2^3*z3*z5z5z3-811187497/317520*z2^3*z3*z7z3z3-
        26840936328970507766 81/953087845248000*z2^3*z3*z13-6731243/
        2800*z2^3*z3*z5*z5z3+1080509/15120*z2^3*z3^2*z7z3+2009725/168*
        z2^3*z3^2*z5^2+570093989/52920*z2^3*z3^3*z7+428519309/105000*
        z2^4*z5z3z3z3+2054864774262694 7/411505920000*z2^4*z9z5-
        910144972791054017/6035420160000*z2^4*z11z3-
        1373575155838415614 9/12070840320000*z2^4*z7^2-
        9468869571342609912 7/58342394880000*z2^4*z5*z9-141084539/78750
        *z2^4*z3*z5z3z3+140544106016863793716739/2601929817527040000*
        z2^4*z3*z11+17966741/252000*z2^4*z3^2*z5z3+5233954847/13608000
        *z2^4*z3^3*z5-89747783/12474*z2^5*z8z2z1z1+42587330003873/
        2235340800*z2^5*z9z3+1974614546123368323 7/53480528640000*z2^5*
        z5*z7+1287323935999686801847583/30665601420854400 00*z2^5*z3*z9
        +1323224553841/1571724000*z2^5*z3^4+196664555715971051/
        22884301440000*z2^6*z7z3+689800062898138493 23/
        11355698914560000*z2^6*z5^2+9497144071306335619298287 3/
        104646364848665640000 0*z2^6*z3*z7+313619248788976309/
        44951306400000*z2^7*z5z3+909871564554223072 79/1064596773240000
        *z2^7*z3*z5+216415730248739246 87/3863315055600000*z2^8*z3^2-
        28899425519949609920538362700642 7/162732217997457108000000 00*
        z2^11;
```

We have of course more results when we restrict the depth. They are less interesting from the viewpoint of this talk.

Data mining

The results of all the runs we made have been put together in a place that will be publicly accessible. We call it the MZV datamine.
The format of the files is text (but in a notation that is most suited for FORM). In some cases there may be binary FORM files for faster access.
There are also FORM programs that help to read the files. And there are example files that show how one can manipulate the data. In particular there are some programs that show how to change bases.
One should keep in mind that one needs more than the average laptop to manipulate some of these files. Putting a 4 Gbyte file in an editor is rather stressful for a computer.
The FORM binary files are easier to manipulate. Even laptops may do in many cases.
For the bigger tables 32-bits processors may not work. FORM has some restrictions there.
Of course, FORM and TFORM are freely available.

The first things we look up in the datamine are some relations that Broadhurst discovered in the 1990's with the use of PSLQ. Now we can obtain 'formal' proof of them. They are so-called push down relations in which an object that has at least depth D as a MZV, can be expressed in terms of depth $D-2$ Euler sums.

The simplest example of such a push down relation is the following:

$$\begin{aligned} H_{8,2,1,1} \;=\; & -\frac{1593344}{47475}H_{-11,-1}+\frac{10624}{28485}H_{-9,-3}+\frac{56896}{712125}H_{-7,-5} \\ & +\frac{64}{243}H_{-3}^{4}+\frac{194772992}{2421225}H_{-9}H_{-3}+\frac{56203264}{712125}H_{-7}H_{-5} \\ & +\frac{21504}{1583}\zeta_2H_{-9,-1}-\frac{768}{1583}\zeta_2H_{-7,-3}-\frac{8660992}{299187}\zeta_2H_{-7}H_{-3} \\ & -\frac{529216}{39575}\zeta_2H_{-5}^{2}+\frac{512}{171}\zeta_2^2H_{-7,-1}-\frac{512}{2565}\zeta_2^2H_{-5,-3} \\ & -\frac{98624}{12825}\zeta_2^2H_{-5}H_{-3}-\frac{352}{315}\zeta_2^3H_{-3}^{2}-\frac{5975591045926624 6}{1876000193254687 5}\zeta_2^6 \end{aligned}$$

The next one at $W = 15$ becomes already rather bad.

$$
\begin{aligned}
H_{6,2,5,1,1} \;=\; & -\frac{28009182704961773376996398903118174942184754265798529122596}{305651913521473711081726272715815595332022071566091290625}\zeta_2^6 H_{-3} \\
& -\frac{68687238807894361714855018645761222083481069778506279 44}{38707190153725780323875000478239018538890298220085625}\zeta_2^5 H_{-5} \\
& -\frac{35262089944835923595670805062878298367884474534 2656}{101363801241020822533090221202991974121254097 4465}\zeta_2^4 H_{-7} \\
& -\frac{4503461895027462759479496241136800293638799661608 32}{1079689612216387207432665263440390701792806802375}\zeta_2^3 H_{-9} \\
& +\frac{2176}{945}\zeta_2^3 H_{-3}^3 - \frac{2037950288768}{2234346324525}\zeta_2^2 H_{-3}^2 H_{-5} + \frac{176193784832}{29791284327}\zeta_2^2 H_{-3} H_{-7,-1} \\
& -\frac{19599298746371297483193212289321032985913744503680}{472522983228818871958766444705676871843440153 51}\zeta_2^2 H_{-11} \\
& -\frac{172882684928}{446869264905}\zeta_2^2 H_{-3} H_{-5,-3} - \frac{25300992}{8296097}\zeta_2^2 H_{-9,-1,-1} + \frac{74885120}{174218037}\zeta_2^2 H_{-7,-3,-1} \\
& +\frac{18508800}{58072679}\zeta_2^2 H_{-7,-1,-3} - \frac{111818752}{871090185}\zeta_2^2 H_{-5,-5,-1} - \frac{224668672}{7839811665}\zeta_2^2 H_{-5,-3,-3} \\
& -\frac{2212690676795201726 6176}{612211434481649101 05}\zeta_2 H_{-3}^2 H_{-7} - \frac{30664508461328784676096}{4372938817726065 0075}\zeta_2 H_{-3} H_{-5}^2 \\
& +\frac{3632939862491022991 36}{323921393905634445}\zeta_2 H_{-3} H_{-9,-1} - \frac{4369910014768059392}{107973797968544815}\zeta_2 H_{-3} H_{-7,-3}
\end{aligned}
$$

$$+\frac{1644070289092638208}{1841625107486235}\zeta_2 H_{-5} H_{-7,-1} - \frac{3361783780033637888}{5524875322458705}\zeta_2 H_{-5} H_{-5,-3}$$
$$+\frac{853627469707858391615100678967489449812221696}{597132601687688036631228981023886538887725}\zeta_2 H_{-13}$$
$$-\frac{58973326655000576}{40925002388583}\zeta_2 H_{-11,-1,-1} + \frac{117774300067486720}{122775007165749}\zeta_2 H_{-9,-3,-1}$$
$$+\frac{204058184143646672}{613875035828745}\zeta_2 H_{-9,-1,-3} - \frac{840629459695016}{613875035828745}\zeta_2 H_{-7,-5,-1}$$
$$+\frac{1152979070087168}{368325021497247}\zeta_2 H_{-7,-3,-3} + \frac{122738670251837444}{613875035828745}\zeta_2 H_{-7,-1,-5}$$
$$-\frac{2873606698310656}{1841625107486235}\zeta_2 H_{-5,-5,-3} - \frac{1792}{3645} H^5_{-3}$$
$$-\frac{4256896288848871864427599757056}{34508279292586490964865596165} H^2_{-3} H_{-9}$$
$$+\frac{390750819618975077368265702232899584}{712288099409986982475670367743425} H_{-3} H_{-5} H_{-7}$$
$$-\frac{1208984451017729087145407744}{375907181836454149944069675} H_{-3} H_{-7,-5}$$
$$+\frac{3840626217263581248362959360}{135326585461123493979865083} H_{-3} H_{-9,-3}$$
$$-\frac{409378446382355312335204364288}{676632927305617469899325415} H_{-3} H_{-11,-1}$$
$$+\frac{224360652920825136173473713980416}{1142178828829754987399963173875} H^3_{-5}$$

$$
\begin{aligned}
&-\frac{66613761278338041301228507648 0}{101527007007089332213330059 9}H_{-5}H_{-9,-1}\\
&+\frac{879380015176193352870400256}{376025951878108637827148 37}H_{-5}H_{-7,-3}\\
&-\frac{2844342500576392653874336768 0}{853006439162531976271187 49}H_{-7}H_{-7,-1}\\
&+\frac{5688685001152785307748673536}{255901931748759592881356247}H_{-7}H_{-5,-3}\\
&-\frac{211253345951081514775291987695078 4}{15761057698646347406673907498 5}H_{-15}\\
&+\frac{85294165615990794439499776}{7126202499269272984721 7}H_{-13,-1,-1}\\
&-\frac{17490794990045584642269184}{21378607497807818954165 1}H_{-11,-3,-1}-\frac{125855319359428327200389 12}{21378607497807818954165 1}H_{-11,-1,-3}\\
&+\frac{46718277100014917876531 20}{21378607497807818954165 1}H_{-9,-5,-1}+\frac{48724244806847137202155 52}{19240746748027037058748 59}H_{-9,-3,-3}\\
&+\frac{86271225757794923471052 8}{7126202499269272984721 7}H_{-9,-1,-5}-\frac{51011715149917107929907 2}{7126202499269272984721 7}H_{-7,-7,-1}\\
&-\frac{47446498099966692848998 4}{19240746748027037058748 59}H_{-7,-5,-3}-\frac{24737704682643273406412 8}{64135822493423456862495 3}H_{-7,-3,-5}
\end{aligned}
$$

It just gives some more respect for Broadhurst who located these relations with the help of PSLQ in the 90's.

Verifying push downs isn't necessarily a trivial lookup in the tables. For example there are two MZVs at weight 17 and depth 5. There should be one push down. It is however a linear combination of the two that obtains the push down as in

$$H_{6,4,5,1,1} + \frac{72}{5} H_{5,3,3,3,3} \;\rightarrow\; (D \leq 3)$$

We do not show the right hand side as it involves 99 terms. Just one:

$$-\frac{391637561921020510388495527693101233498239312730472192870557734150375516560722487938377037680200651787224018403770884822305866731511488}{12096842033646879193852836812120840799898022503305835922565953025114624797521762549901601984894859006780341916995765114718351715625}\zeta_2^7 H_{-3}$$

This means that checks of the more complicated push downs require quite an amount of algebra first to get the 'non-push downs' out of the way.

These push downs seem to exist because of the doubling and the generalized doubling relations.

We checked this for the only system that we have complete control over: $W = 12$. Here we have the object $H_{8,2,1,1}$.

If we omit the doubling and the generalized doubling relations, there are three extra undetermined objects. Two of depth 4 and one of depth 2. The push down doesn't take place.

If we use the doubling relations and we omit the generalized doubling relations there is only one extra undetermined object of depth 4. And the push down does take place.

Unfortunately we cannot test other push downs. The next one is at $W = 15$ and if we omit the GDR's we have to run nearly all depths.

Without the GDR's many relations at a given depth are only obtained by combining many relations at a greater depth!

A push down basis

Broadhurst and Kreimer gave a conjecture for the number of basis elements for each weight and depth for MZVs. They also gave a conjecture for each weight and depth when the MZVs are expressed in terms of Euler sums. These conjectures are given on the next page. In red are the numbers we explicitly verified.

From them one can see that there should be MZV basis elements that have fewer indices when expressed in terms of Euler basis elements as we have seen before. The push downs.

From the tables one can derive how many there should be, under the assumption that a push down is only from D to $D-2$.

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		1						
13	1		2							
14		2		1						
15	1		2		1					
16		2		3						
17	1		4		2					
18		2		5		1				
19	1		5		5					
20		3		7		3				
21	1		6		9		1			
22		3		11		7				
23	1		8		15		4			
24		3		16		14		1		
25	1		10		23		11			
26		4		20		27		5		
27	1		11		36		23		2	
28		4		27		45		16		
29	1		14		50		48		7	
30		4		35		73		37		2

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

In determining a nice basis for the MZVs we noticed that the number of elements for each weight followed a prescription. They were equal to the number of elements one obtains when making all Lyndon words out of odd integers ≥ 3 in which the integers add up to the weight. Let us call this set L_W. The number of elements of a given weight and given depth in this construction follows exactly the second Broadhurst-Kreimer table! Next we tried to write as many basis elements as possible in terms of elements of this set.

This would not cover the whole set. The remaining elements could be obtained by allowing two even integers (say the first two indices) and making the last two indices equal to one. These elements would match the missing elements of our set if one would take away the ones and add them to the even integers. We call such a basis P_W.

Example: $W = 12$.

$$L_{12}: \; H_{9,3} \quad H_{7,5}$$
$$P_{12}: \; H_{9,3} \quad H_{6,4,1,1}$$

Example: $W = 18$.

$$L_{18}: \; H_{15,3} \; H_{13,5} \; H_{11,7} \; H_{9,3,3,3} \; H_{7,5,3,3} \; H_{7,3,5,3} \; H_{7,3,3,5} \; H_{5,5,5,3}$$
$$P_{18}: \; H_{15,3} \; H_{13,5} \; H_{10,6,1,1} \; H_{9,3,3,3} \; H_{7,5,3,3} \; H_{7,3,5,3} \; H_{6,2,3,5,1,1} \; H_{5,5,5,3}$$

The interesting thing is that each of these special elements seems to be connected to a push down relation.

This is why we needed the run for Euler sums at $W = 18, D = 6$.

$$H_{10,6,1,1} + 46630979 \; H_{5,5,5,3} + 122713096 \; H_{7,5,3,3} + 1002156999 \; H_{9,3,3,3}$$
$$\rightarrow 672686306 \; H_{-17,-1} + 72010179 \; H_{-15,-3} - 705663559 \; H_{-13,-5}$$
$$+817296192 \; H_{-11,-7} + \cdots$$

The complete recipe is:

1. Write basis elements always with the lowest depth possible.
2. Generate the set L_W of all Lyndon words of odd-only ≥ 3 indices.
3. Starting at the lowest depth D, write as many elements of the basis as elements of L_W. Keep the remaining elements.
4. At the next depth D+2 write as many elements of the basis as elements of L_W. Extend the elements of L_W that remained at D according to prescription A_1 and write as many basis elements as possible as these 'extended' elements.
5. Do the same at $D+4$, fill with elements at $D+2$, extended with A_1 and possibly with elements still remaining from D, extended according to prescription A_2
6. Keep raising the depth till there is no more and a complete basis has been obtained.

Prescription A_n: Of a list of indices, subtract one from the first $2n$ elements and add $2n$ ones to the end of the list.

Note 1: it may be necessary to backtrack. The selections in the steps 3-5 are not unique and one may have to alter the selection when at a later stage things don't work out.
Note 2: the result of prescription A_n should be a Lyndon word. If not, this element is not eligible for extension and note 1 applies.

Conjecture: It is always possible, with a suitable choice of steps 3 and following, to obtain a basis.

Conjecture: The elements with added pairs of ones correspond to push downs and the number of ones indicate the units in depth that the push down corresponds to.

Example, $W = 26$:

The basis, as determined by the computer program has a depth distribution of (4,20,27,5) for D=(2,4,6,8). The depth distribution of set L_{26} is (5,30,20,1).

We start with $D = 2$ and see that we have one element left in L_{26}.

Next at $D = 4$ we can write 19 basis elements as elements of L_{26}. This means that there is one element still to be determined. We take the element that remained at $D = 2$ and extend it with A_1 to depth 4. This gives us for instance the element $H_{14,10,1,1}$. If the 19 other elements have been selected properly this one completes the $D = 4$ part of P_{26}. There are 11 elements of L_{26} remaining at $D = 4$.

Next we try the same at $D = 6$. 16 elements can be written as elements of L_{26}. For the remaining 11 we can take the A_1-extended elements we had left at $D = 4$. It is very unlikely that this 'fits' immediately and one may have to go back to the previous step to make a different selection for the 19 elements of L_{26} at the onset of that step. Eventually it fits. There are 4 elements left at $D = 6$ in L_{26}.

Finally at depth $D = 8$ there is one element in L_{26} and the A_1-extension of the 4 elements that were left in the previous step complete the 5 basis elements that we need.

$$
\begin{aligned}
P_{26} = \; & H_{17,9}, H_{19,7}, H_{21,5}, H_{23,3}, H_{7,7,7,5}, H_{9,5,9,3}, H_{11,3,9,3}, H_{11,5,3,7}, \\
& H_{11,5,5,5}, H_{11,5,7,3}, H_{11,7,3,5}, H_{11,7,5,3}, H_{11,9,3,3}, H_{13,3,3,7}, H_{13,3,5,5}, \\
& H_{13,3,7,3}, H_{13,5,3,5}, H_{13,5,5,3}, H_{13,7,3,3}, H_{15,3,3,5}, H_{15,3,5,3}, H_{15,5,3,3}, \\
& H_{17,3,3,3}, H_{14,10,1,1}, H_{5,5,5,3,5,3}, H_{5,5,5,5,3,3}, H_{7,3,3,5,5,3}, H_{7,3,5,3,5,3}, \\
& H_{7,3,5,5,3,3}, H_{7,3,7,3,3,3}, H_{7,5,3,3,5,3}, H_{7,5,3,5,3,3}, H_{7,5,5,3,3,3}, \\
& H_{7,7,3,3,3,3}, H_{9,3,3,3,3,5}, H_{9,3,3,3,5,3}, H_{9,3,3,5,3,3}, H_{9,3,5,3,3,3}, \\
& H_{9,5,3,3,3,3}, H_{11,3,3,3,3,3}, H_{8,2,7,7,1,1}, H_{8,4,5,7,1,1}, H_{8,4,7,5,1,1}, \\
& H_{8,6,3,7,1,1}, H_{8,6,5,5,1,1}, H_{8,6,7,3,1,1}, H_{8,8,3,5,1,1}, H_{8,8,5,3,1,1}, \\
& H_{10,2,3,9,1,1}, H_{10,2,5,7,1,1}, H_{10,2,7,5,1,1}, H_{5,3,3,3,3,3,3,3}, \\
& H_{6,2,3,3,5,5,1,1}, H_{6,2,3,5,3,5,1,1}, H_{6,2,5,3,3,5,1,1}, H_{6,4,3,3,3,5,1,1}
\end{aligned}
$$

Similarly one obtaines for P_{27}:

$$H_{27}, H_{11,7,9}, H_{13,11,3}, H_{15,3,9}, H_{15,5,7}, H_{15,7,5}, H_{15,9,3}, H_{17,5,5}, H_{17,7,3},$$
$$H_{19,3,5}, H_{19,5,3}, H_{21,3,3}, H_{7,5,5,7,3}, H_{7,5,7,3,5}, H_{7,7,3,7,3}, H_{7,7,7,3,3},$$
$$H_{9,3,9,3,3}, H_{9,5,3,5,5}, H_{9,5,3,7,3}, H_{9,5,5,3,5}, H_{9,5,5,5,3}, H_{9,5,7,3,3}, H_{9,7,3,3,5},$$
$$H_{9,7,3,5,3}, H_{9,7,5,3,3}, H_{9,9,3,3,3}, H_{11,3,3,3,7}, H_{11,3,3,5,5}, H_{11,3,3,7,3}, H_{11,3,5,3,5},$$
$$H_{11,3,5,5,3}, H_{11,3,7,3,3}, H_{11,5,3,3,5}, H_{11,5,3,5,3}, H_{11,5,5,3,3}, H_{11,7,3,3,3}, H_{13,3,3,3,5},$$
$$H_{13,3,3,5,3}, H_{13,3,5,3,3}, H_{13,5,3,3,3}, H_{15,3,3,3,3}, H_{10,8,7,1,1}, H_{10,10,5,1,1},$$
$$H_{12,2,11,1,1}, H_{12,4,9,1,1}, H_{12,6,7,1,1}, H_{12,8,5,1,1}, H_{16,2,7,1,1},$$
$$H_{5,3,5,3,5,3,3}, H_{5,5,3,3,3,5,3}, H_{5,5,3,3,5,3,3}, H_{5,5,3,5,3,3,3}, H_{5,5,5,3,3,3,3},$$
$$H_{7,3,3,3,3,3,5}, H_{7,3,3,3,3,5,3}, H_{7,3,3,3,5,3,3}, H_{7,3,3,5,3,3,3}, H_{7,3,5,3,3,3,3},$$
$$H_{9,3,3,3,3,3,3}, H_{6,4,5,5,5,1,1}, H_{6,6,3,5,5,1,1}, H_{6,6,5,3,5,1,1}, H_{6,6,5,5,3,1,1},$$
$$H_{8,2,3,5,7,1,1}, H_{8,2,3,7,5,1,1}, H_{8,2,5,3,7,1,1}, H_{8,2,5,5,5,1,1}, H_{8,2,5,7,3,1,1},$$
$$H_{8,2,7,3,5,1,1}, H_{8,2,7,5,3,1,1}, H_{8,4,3,3,7,1,1},$$
$$H_{7,5,7,5,3} \longrightarrow ?H_{6,4,6,4,3,1,1,1,1}, H_{7,5,3,3,3,3,3} \longrightarrow ?H_{6,4,3,3,3,3,3,1,1}$$

The last two elements are guessed from the remaining odds-only elements.

One seems to indicate a double push down!

$$
\begin{aligned}
P_7 &= H_7 \\
P_8 &= H_{5,3} \\
P_9 &= H_9 \\
P_{10} &= H_{7,3} \\
P_{11} &= H_{11}, H_{5,3,3} \\
P_{12} &= H_{9,3}, H_{6,4,1,1} \\
P_{13} &= H_{13}, H_{7,3,3}, H_{5,5,3} \\
P_{14} &= H_{11,3}, H_{9,5}, H_{5,3,3,3} \\
P_{15} &= H_{15}, H_{7,5,3}, H_{9,3,3}, H_{6,2,5,1,1} \\
P_{16} &= H_{11,5}, H_{13,3}, H_{5,5,3,3}, H_{7,3,3,3}, H_{8,6,1,1} \\
P_{17} &= H_{17}, H_{7,7,3}, H_{9,3,5}, H_{9,5,3}, H_{11,3,3}, H_{5,3,3,3,3}, H_{6,4,5,1,1} \\
P_{18} &= H_{13,5}, H_{15,3}, H_{5,5,5,3}, H_{7,3,5,3}, H_{7,5,3,3}, H_{9,3,3,3}, H_{10,6,1,1}, H_{6,2,3,5,1,1} \\
P_{19} &= H_{19}, H_{9,5,5}, H_{9,7,3}, H_{11,3,5}, H_{11,5,3}, H_{13,3,3}, \\
&\quad H_{5,3,5,3,3}, H_{5,5,3,3,3}, H_{7,3,3,3,3}, H_{6,6,5,1,1}, H_{8,2,7,1,1}
\end{aligned}
$$

$$
\begin{aligned}
P_{20} &= H_{13,7}, H_{15,5}, H_{17,3}, H_{7,5,5,3}, H_{7,7,3,3}, H_{9,3,3,5}, H_{9,3,5,3}, \\
&\quad H_{9,5,3,3}, H_{11,3,3,3}, H_{10,8,1,1}, H_{5,3,3,3,3,3}, H_{6,2,5,5,1,1}, H_{6,4,3,5,1,1} \\
P_{21} &= H_{21}, H_{9,9,3}, H_{11,3,7}, H_{11,7,3}, H_{13,3,5}, H_{13,5,3}, H_{15,3,3}, \\
&\quad H_{5,5,3,5,3}, H_{5,5,5,3,3}, H_{7,3,3,5,3}, H_{7,3,5,3,3}, H_{7,5,3,3,3}, H_{9,3,3,3,3}, \\
&\quad H_{8,4,7,1,1}, H_{8,6,5,1,1}, H_{10,4,5,1,1}, H_{6,2,3,3,5,1,1} \\
P_{22} &= H_{15,7}, H_{17,5}, H_{19,3}, H_{7,5,7,3}, H_{9,3,5,5}, H_{9,3,7,3}, H_{9,5,3,5}, \\
&\quad H_{9,5,5,3}, H_{9,7,3,3}, H_{11,3,3,5}, H_{11,3,5,3}, H_{11,5,3,3}, H_{13,3,3,3}, H_{12,8,1,1}, \\
&\quad H_{5,3,5,3,3,3}, H_{5,5,3,3,3,3}, H_{7,3,3,3,3,3} \\
&\quad H_{6,4,5,5,1,1}, H_{6,6,3,5,1,1}, H_{6,6,5,3,1,1}, H_{8,2,3,7,1,1}, \\
P_{23} &= H_{23}, H_{11,7,5}, H_{11,9,3}, H_{13,3,7}, H_{13,5,5}, H_{13,7,3}, H_{15,3,5}, H_{15,5,3}, \\
&\quad H_{17,3,3}, H_{5,5,5,5,3}, H_{7,3,7,3,3}, H_{7,3,5,5,3}, H_{7,5,3,5,3}, H_{7,5,5,3,3}, \\
&\quad H_{7,7,3,3,3}, H_{9,3,3,3,5}, H_{9,3,3,5,3}, H_{9,3,5,3,3}, H_{9,5,3,3,3}, H_{11,3,3,3,3}, \\
&\quad H_{8,6,7,1,1}, H_{8,8,5,1,1}, H_{10,2,9,1,1}, H_{10,4,7,1,1}, \\
&\quad H_{5,3,3,3,3,3,3} H_{6,2,3,5,5,1,1}, H_{6,2,5,3,5,1,1}, H_{6,4,3,3,5,1,1},
\end{aligned}
$$

$$
\begin{aligned}
P_{24} = \; & H_{17,7}, H_{19,5}, H_{21,3}, H_{7,7,7,3}, H_{9,7,3,5}, H_{9,7,5,3}, H_{9,9,3,3}, \\
& H_{11,3,3,7}, H_{11,3,5,5}, H_{11,3,7,3}, H_{11,5,3,5}, H_{11,5,5,3}, H_{11,7,3,3}, H_{13,3,3,5}, \\
& H_{13,3,5,3}, H_{13,5,3,3}, H_{15,3,3,3}, H_{12,10,1,1}, H_{14,8,1,1}, H_{5,5,3,3,5,3}, \\
& H_{5,5,3,5,3,3}, H_{5,5,5,3,3,3}, H_{7,3,3,3,5,3}, H_{7,3,3,5,3,3}, H_{7,3,5,3,3,3}, H_{7,5,3,3,3,3}, \\
& H_{9,3,3,3,3,3}, H_{6,6,5,5,1,1}, H_{8,2,5,7,1,1}, H_{8,2,7,5,1,1}, H_{8,4,3,7,1,1}, H_{8,4,5,5,1,1}, \\
& H_{8,4,7,3,1,1}, H_{6,2,3,3,3,5,1,1} \\
P_{25} = \; & H_{25}, H_{11,11,3}, H_{13,5,7}, H_{13,7,5}, H_{13,9,3}, H_{15,3,7}, H_{15,5,5}, H_{15,7,3}, \\
& H_{17,3,5}, H_{17,5,3}, H_{19,3,3}, H_{7,3,7,3,5}, H_{7,5,3,7,3}, H_{7,5,7,3,3}, \\
& H_{9,3,3,3,7}, H_{9,3,3,5,5}, H_{9,3,3,7,3}, H_{9,3,5,3,5}, H_{9,3,5,5,3}, H_{9,3,7,3,3}, \\
& H_{9,5,3,3,5}, H_{9,5,3,5,3}, H_{9,5,5,3,3}, H_{9,7,3,3,3}, H_{11,3,3,3,5}, H_{11,3,3,5,3}, \\
& H_{11,3,5,3,3}, H_{11,5,3,3,3}, H_{13,3,3,3,3}, H_{8,8,7,1,1}, H_{10,4,9,1,1}, \\
& H_{10,6,7,1,1}, H_{10,8,5,1,1}, H_{12,2,9,1,1}, H_{5,3,3,5,3,3,3}, H_{5,3,5,3,3,3,3}, \\
& H_{5,5,3,3,3,3,3}, H_{7,3,3,3,3,3,3}, H_{6,2,5,5,5,1,1}, H_{6,4,3,5,5,1,1}, H_{6,4,5,3,5,1,1}, \\
& H_{6,4,5,5,3,1,1}, H_{6,6,3,3,5,1,1}, H_{6,6,3,5,3,1,1}, H_{6,6,5,3,3,1,1}
\end{aligned}
$$

Currently we don't know how to work the prescription backward. Starting from the complete set of Lyndon words we don't have a way of telling which elements should be selected to remain as they are and which ones should be extended, and by how much.

This means that we cannot predict the complete basis and we need the computer runs.

Yet it looks like progress.

It seems important to have more insight in the embedding of the MZVs in the Euler sums. The role that the doubling relations and the GDR's play here is crucial and should be understood.

Conclusions

We have now complete and partial tables for the MZV's and Euler sums that will cover many more values of the weight and depth than were previously available.

These results will be publicly accessible soon under the name "MZV Datamine". It should be linked in the FORM pages and the home pages of the authors of the paper (http://www.nikhef.nl/∼form).

FORM programs to allow one to manipulate this data are available in the Datamine as well. So is FORM. One is advised to use computers with a 64-bits architecture for this.

We have conjectured a new type of basis which seems to be connected to an embedding of the MZVs in the Euler sums.

The holy grail in the field of MZV's is an algorithm to express each MZV into an unique basis in a constructive way. That way we would have a (hopefully) small procedure rather than giant tables. Thus far this has not been found.