Symbolic calculations with FORM
– A practitioner's point of view –

Sven-Olaf Moch

DESY Zeuthen

1. **What is FORM ?**
2. **What is FORM doing ?**
3. **What is FORM missing ?**
4. **What is the future of FORM ?**

What is FORM ?

– The Free On-line Dictionary of Computing
  http ://computing-dictionary.thefreedictionary.com/FORM

```
form :

A system written by Jos Vermaseren in 1989 for fast handling of very
large-scale symbolic mathematics problems. FORM is a descendant of
Schoonschip and is available for many personal computers and
workstations.
```

Literature on Form

- Standard reference for scientific community
  - *New features of* Form J.A.M. Vermaseren `math-ph/0010025`
- Manual & Tutorial available from www.nikhef.nl/~form
  - Form *manual* J.A.M. Vermaseren

    Form *for Pedestrians* A. Heck
- Special features
  - Parallel Form

    D. Fliegner, A. Retey, J.A.M. Vermaseren `hep-ph/0007221`
  - Table bases

    J.A.M. Vermaseren `hep-ph/0211297`

What is FORM doing ?

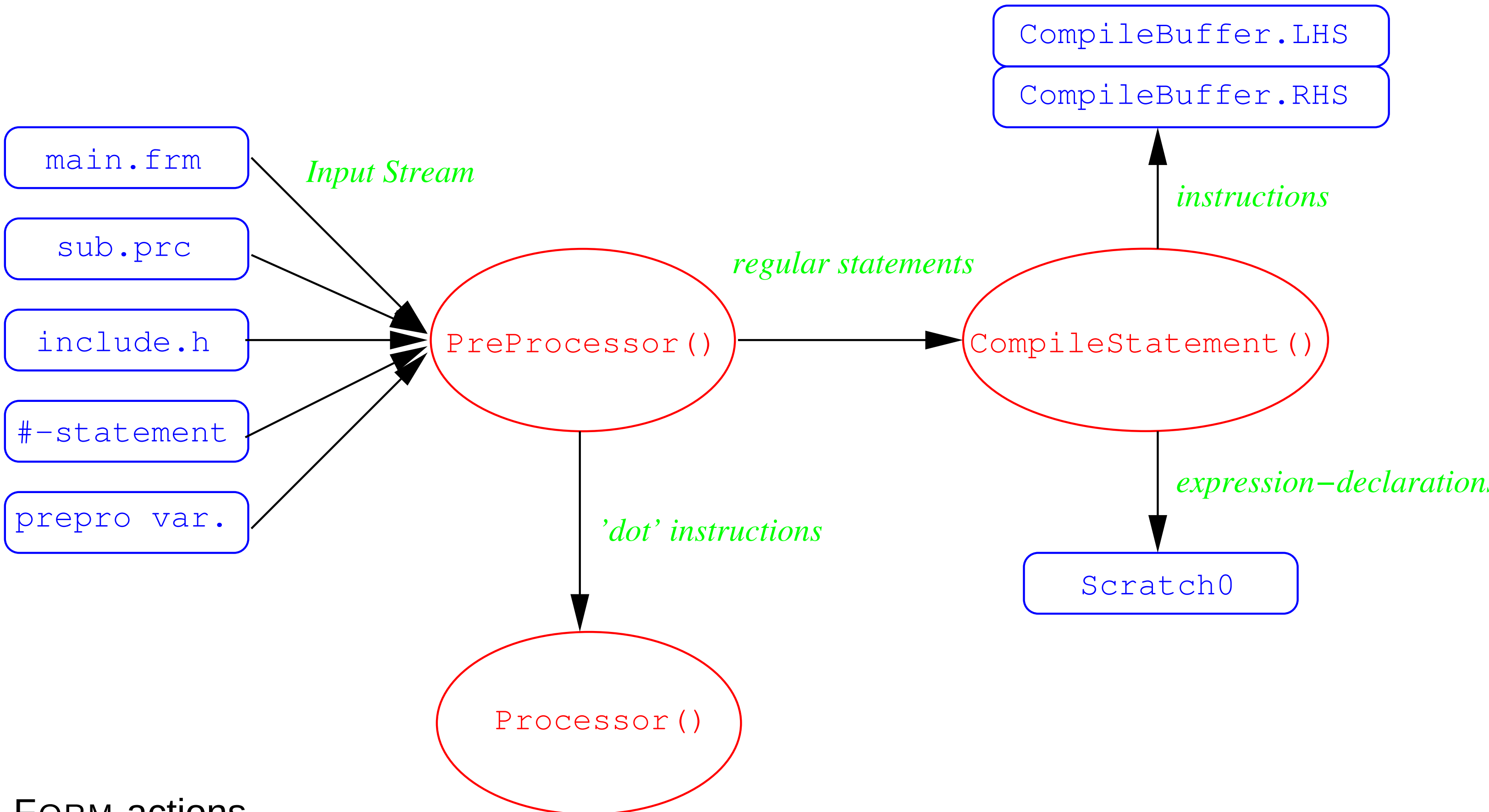
The structure of FORM

- FORM is **type oriented**
  ⟶ declarations required (symbols, functions, . . .)
- FORM is **term oriented**
  ⟶ local operations on expressions
- FORM works on partions
  ⟶ **modules** terminated with `.sort` and `.end`

```
local expression =      a*x + x^2;
id x = a + b;
                        a^2 + a*b + a^2 + 2*a*b + b^2
.sort                   2*a^2 + 3*a*b + b^2

if(count(b,1)==1);
multiply 4*a/b;
endif;
                        2*a^2 + 12*a^2 + b^2

print;
.end                    14*a^2 + b^2
```



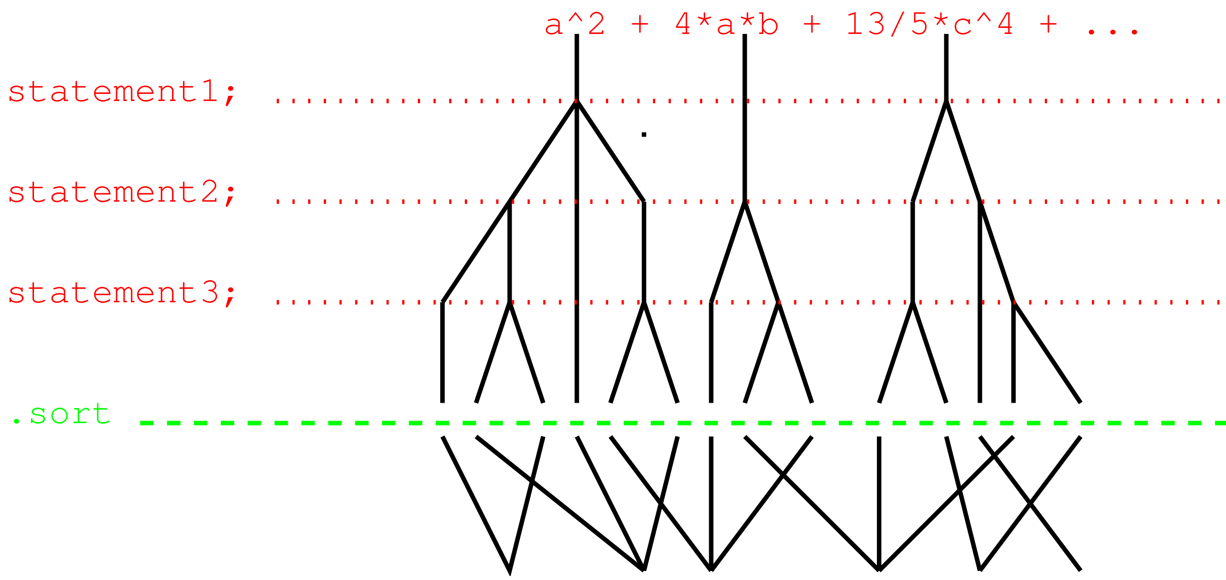
– FORM actions

1. preprocessor ⟶ preparation and filter of input
2. compiler ⟶ expressions, set of instructions in internal representation (flat data structure)
3. processor ⟶ generation of terms, execution of instructions, sorting

Generating terms

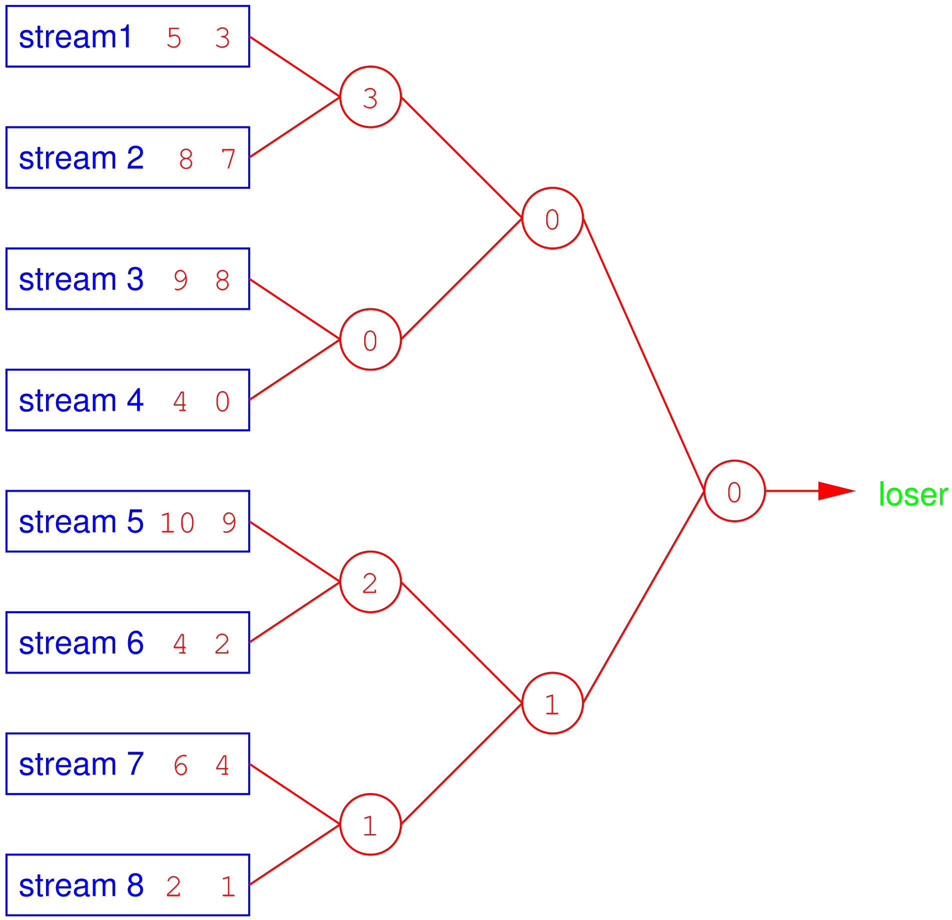
– Local recursive operation on individual term

1. check term for subterms to be inserted
2. put term in normal form by ordering subterms
3. apply set of instructions for given recursion depth
4. put term in normal form by expanding subterms



– FORM process of generation is term oriented (MAPLE or Mathematica work line-oriented)

⟶ **tree structure for generation of terms**



$\longrightarrow$ **tree structure for elimination of terms**

Sorting terms

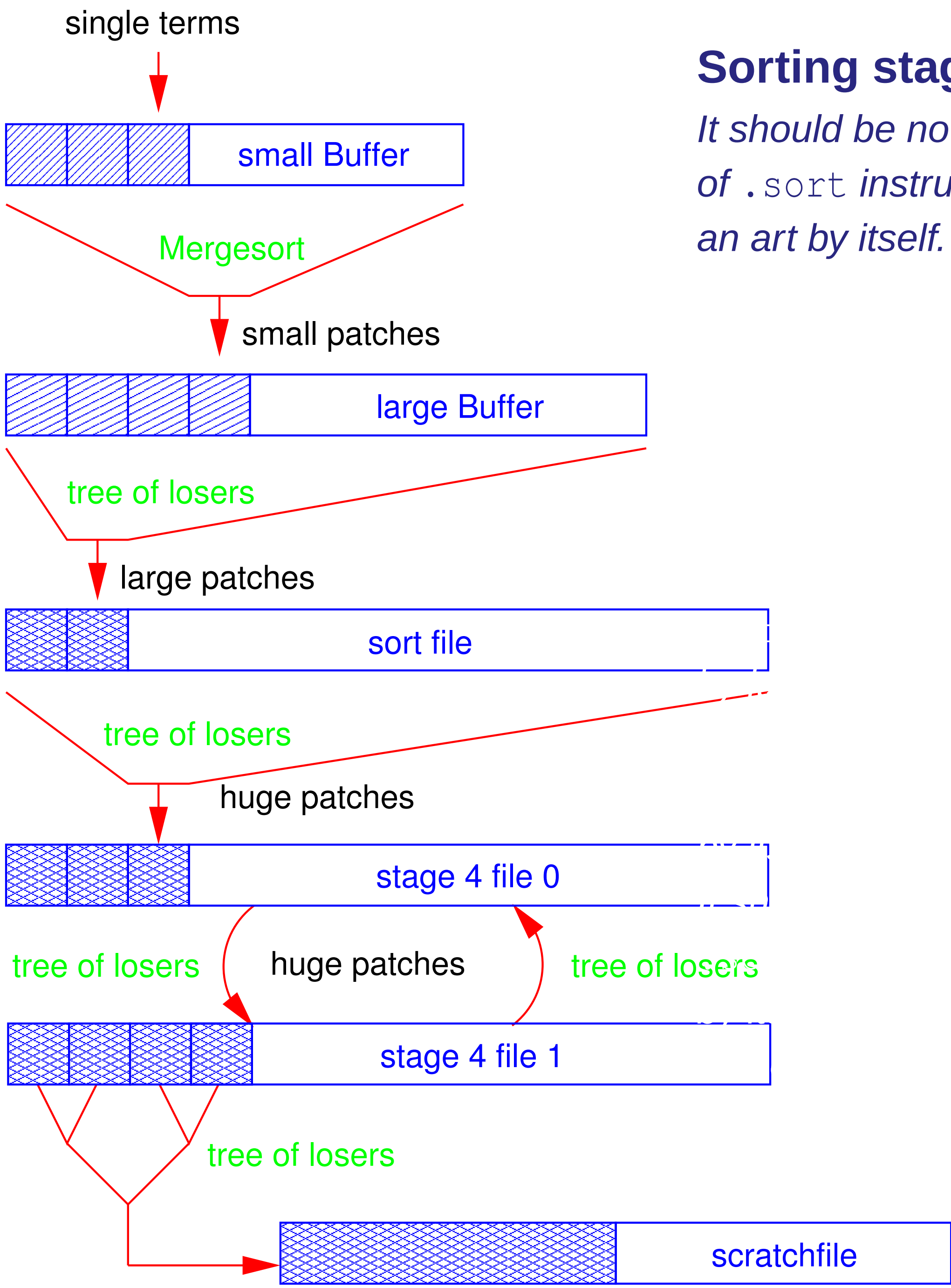
– Output stream from generation in unsorted and redundant form

– Result in standard form (summation of equivalent terms)

– `Mergesort` on patches in small buffer
$\longrightarrow$ presorted patches
$\longrightarrow$ computational cost $n \log n$ for n terms

– File-to-file sort
$\longrightarrow$ tree of losers

– Example :
merging of presorted integer stream with smallest element surviving at root



Sorting stages

It should be noted that the proper placement of `.sort` *instructions in a* FORM *program is an art by itself.*

J.A.M. Vermaseren

Internal data structure

– Representation of subterm $-\frac{2}{5}a^5$ as series of short words (integer numbers)

– 8 $\longrightarrow$ total length
– 1 $\longrightarrow$ type symbol
– 4 $\longrightarrow$ total length of symbol
– 3 $\longrightarrow$ # of a in table of symbols
– 5 $\longrightarrow$ exponent of a
– 2 $\longrightarrow$ numerator of coeff.
– 5 $\longrightarrow$ denominator of coeff.
– -3 $\longrightarrow$ total length of coeff.
(with negative sign)

expression

prototype	term	term	term	NULL

term

length	subterm	subterm	$\cdots$	coeff.

subterm

type	length	#var.	exponent

coefficient

numerat.	denom.	length

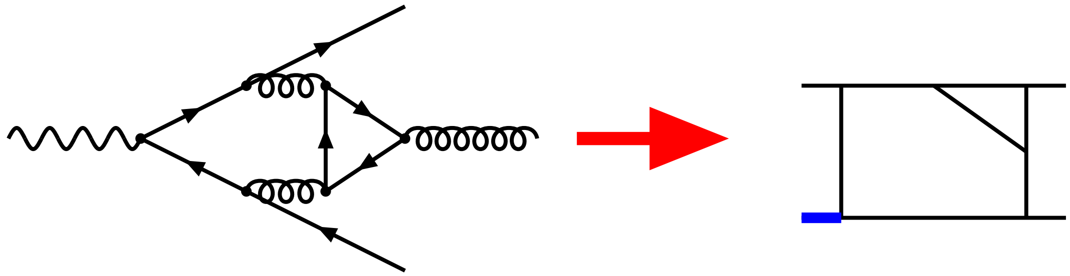
`-2/5*a^5` = 8 1 4 3 5 2 5 –3

– Maximum size of terms limited by maximum number of words (maximum size for integers) on given architecture, e.g 32568 on 32 bit system

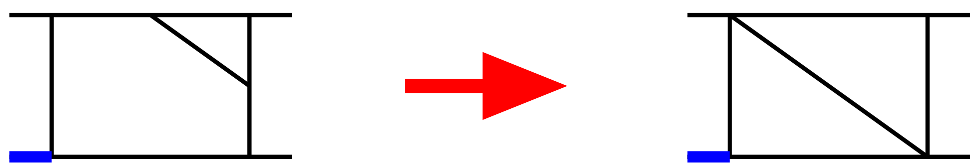
$\longrightarrow$ **big gain from 64 bit architecture**

An application (calculation of radiative corrections) S.M., P. Uwer, S. Weinzierl '02

– Two-loop corrections to amplitude for $e^+e^- \to 3$ jets (n_f-terms)
  S.M., P. Uwer, S. Weinzierl '02



– Identification of scalar integrals : penta-box $\longrightarrow$ C-topology



– Result for two-loop C-topology as two-fold nested sum S.M., Uwer, Weinzierl '01
  – in $D = 2m - 2\varepsilon$ dimensions and with arbitrary powers of the propagators
  – with scales $x_1 = (-s_{12})/(-s_{123})$ and $x_2 = (-s_{23})/(-s_{123})$

– Analytical expression as two-fold nested sum

$$
\begin{aligned}
\text{Ctopo} = {} & \frac{\Gamma(2m-2\varepsilon-\nu_{1235})\Gamma(1+\nu_{1235}-2m+2\varepsilon)\Gamma(2m-2\varepsilon-\nu_{2345})\Gamma(1+\nu_{2345}-2m+2\varepsilon)}{\Gamma(\nu_1)\Gamma(\nu_2)\Gamma(\nu_3)\Gamma(\nu_4)\Gamma(\nu_5)\Gamma(3m-3\varepsilon-\nu_{12345})} \\
& \cdot \frac{\Gamma(m-\varepsilon-\nu_5)\Gamma(m-\varepsilon-\nu_{23})}{\Gamma(2m-2\varepsilon-\nu_{235})}(-s_{123})^{2m-2\varepsilon-\nu_{12345}} \sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty}\frac{x_1^{i_1}}{i_1!}\frac{x_2^{i_2}}{i_2!} \\
& \cdot \Bigg[\frac{\Gamma(i_1+\nu_3)\Gamma(i_2+\nu_2)\Gamma(i_1+i_2-2m+2\varepsilon+\nu_{12345})\Gamma(i_1+i_2-m+\varepsilon+\nu_{235})}{\Gamma(i_1+1-2m+2\varepsilon+\nu_{1235})\Gamma(i_2+1-2m+2\varepsilon+\nu_{2345})\Gamma(i_1+i_2+\nu_{23})} \\
& -x_1^{2m-2\varepsilon-\nu_{1235}}\frac{\Gamma(i_1+2m-2\varepsilon-\nu_{125})\Gamma(i_2+\nu_2)\Gamma(i_1+i_2+\nu_4)\Gamma(i_1+i_2+m-\varepsilon-\nu_1)}{\Gamma(i_1+1+2m-2\varepsilon-\nu_{1235})\Gamma(i_2+1-2m+2\varepsilon+\nu_{2345})\Gamma(i_1+i_2+2m-2\varepsilon-\nu_{15})} \\
& -x_2^{2m-2\varepsilon-\nu_{2345}}\frac{\Gamma(i_1+\nu_3)\Gamma(i_2+2m-2\varepsilon-\nu_{345})\Gamma(i_1+i_2+\nu_1)\Gamma(i_1+i_2+m-\varepsilon-\nu_4)}{\Gamma(i_1+1-2m+2\varepsilon+\nu_{1235})\Gamma(i_2+1+2m-2\varepsilon-\nu_{2345})\Gamma(i_1+i_2+2m-2\varepsilon-\nu_{45})} \\
& +x_1^{2m-2\varepsilon-\nu_{1235}}x_2^{2m-2\varepsilon-\nu_{2345}}\frac{\Gamma(i_1+2m-2\varepsilon-\nu_{125})\Gamma(i_2+2m-2\varepsilon-\nu_{345})}{\Gamma(i_1+1+2m-2\varepsilon-\nu_{1235})\Gamma(i_2+1+2m-2\varepsilon-\nu_{2345})} \\
& \cdot\frac{\Gamma(i_1+i_2+2m-2\varepsilon-\nu_{235})\Gamma(i_1+i_2+3m-3\varepsilon-\nu_{12345})}{\Gamma(i_1+i_2+4m-4\varepsilon-\nu_{12345}-\nu_5)}\Bigg]
\end{aligned}
$$

– systematic expansion of sum in ε possible

How to do the nested sums ?

– Hard task in general

$\longrightarrow$ for single-parameter nested sums closed analytical solutions possible

– Example
$$\sum_{j=1}^{\infty}\frac{x^j}{j!}\Gamma(j-\varepsilon) = \sum_{j=1}^{\infty}\frac{x^j}{j} - \varepsilon\sum_{j=1}^{\infty}\frac{x^j}{j}S_1(j-1) + \varepsilon^2 \ldots$$
$$= -\ln(1-x) - \varepsilon\frac{1}{2}\ln(1-x)^2 + \varepsilon^2 \ldots$$

– Expansion of Γ-functions in powers of ε

$\longrightarrow$ harmonic sums Gonzalez-Arroyo, Lopez, Ynduráin '79 ; Vermaseren '98 ; Blümlein, Kurth '98

$$S_{\pm m_1,m_2,\ldots,m_k}(M) = \sum_{i=1}^{M}\frac{(\pm 1)^i}{i^{m_1}}S_{m_2,\ldots,m_k}(i) \ .$$

Multi-scale nested sums

– Definition of nested S-sums S.M., Uwer, Weinzierl '01

$$S(n;m_1,\ldots,m_k;x_1,\ldots,x_k) = \sum_{i=1}^{n}\frac{x_1^i}{i^{m_1}}S(i;m_2,\ldots,m_k;x_2,\ldots,x_k)$$

– multiple scales $x_1,\ldots,x_k$

– depth k, weight $w = m_1+\ldots+m_k$

Algorithms

$\longrightarrow$ all algorithms implemented in FORM S.M., P. Uwer to be published

– Multiplication

$$S(n;m_1,\ldots;x_1,\ldots)\cdot S(n;m'_1,\ldots;x'_1,\ldots)$$

– Sums involving i and $n-i$

$$\sum_{i=1}^{n-1}\frac{x_1^i}{i^{m_1}}S(i;m_2\ldots;x_2,\ldots)\frac{{x'_1}^{n-i}}{(n-i)^{m'_1}}S(n-i;m'_2,\ldots;x'_2,\ldots)$$

– Conjugations

$$-\sum_{i=1}^{n}\left(\begin{array}{c} n \\ i \end{array}\right)(-1)^i\frac{x_0^i}{i^{m_0}}S(i;m_1,\ldots,m_k;x_1,\ldots,x_k)$$

– Sums involving binomials, i and $n-i$

$$-\sum_{i=1}^{n-1}\left(\begin{array}{c} n \\ i \end{array}\right)(-1)^i\frac{x_1^i}{i^{m_1}}S(i;m_2\ldots;x_2,\ldots)\frac{{x'_1}^{n-i}}{(n-i)^{m'_1}}S(n-i;m'_2,\ldots;x'_2,\ldots)$$

Higher transcendental functions

- Expansion of higher transcendental functions
  - expansion parameter ε occurs in the Pochhammer symbols $(a)_n = \Gamma(a+n)/\Gamma(a)$

- Hypergeometric function

$$ {}_2F_1(a,b;c,x_0) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{x_0^n}{n!} $$

- First Appell function

$$ F_1(a,b_1,b_2;c;x_1,x_2) = \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \frac{(a)_{m_1+m_2} (b_1)_{m_1} (b_2)_{m_2}}{(c)_{m_1+m_2}} \frac{x_1^{m_1}}{m_1!} \frac{x_2^{m_2}}{m_2!} $$

- Second Appell function

$$ F_2(a,b_1,b_2;c_1,c_2;x_1,x_2) = \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \frac{(a)_{m_1+m_2} (b_1)_{m_1} (b_2)_{m_2}}{(c_1)_{m_1} (c_2)_{m_2}} \frac{x_1^{m_1}}{m_1!} \frac{x_2^{m_2}}{m_2!} $$

$\longrightarrow$ **task is beyond capabilities of MAPLE or Mathematica**

Packages for Form

All packages are available from www.nikhef.nl/~form.

- color.h

  library for evaluating group invariants

  T. van Ritbergen, A.N. Schellekens, J.A.M. Vermaseren `hep-ph/9802376`
- harmpol.h

  library for the manipulation of harmonic polylogarithms

  E. Remiddi, J.A.M. Vermaseren `hep-ph/9905237`
- mincer.h

  library for the calculation of three loop massless propagator graphs

  S.A. Larin, F.V. Tkachov and J.A.M. Vermaseren preprint NIKHEF-H/91-18
- summer6.h

  library for the manipulation of harmonic sums

  J.A.M. Vermaseren `hep-ph/9806280`

What is FORM missing ?

Factorization

– Good factorization offers an enourmous potential for improving algorithms
– However, it is a non-local operation

Gröbner bases

– Solving systems of nonlinear and/or differential equations
– Constructive solution of system of equations

  – example for nonlinear equations

$$
\begin{aligned}
0 &= x^2 + xy + y^2 - 4 \\
0 &= 5x^2 + 2xy + y^2 - 14
\end{aligned}
$$

  – corresponding Gröbner basis

$$
\begin{aligned}
0 &= 4x^2 + xy - 10 \\
0 &= 13x^4 - 74x^2 + 100
\end{aligned}
$$

Improved parallelization

– Better parallelization for most popular architectures
  $\longrightarrow$ computers with two processors
  $\longrightarrow$ computers with > 100 processors
– Major achievements for multi-processor SMP architectures
  TTP, Karlsruhe University

Miscellaneous

– Local variables
  $\longrightarrow$ precompiled subroutines or namespace protection
– System call
  $\longrightarrow$ interface e.g. with MAPLE or Mathematica
– Better documentation
– Useful standard libraries

What is the future of FORM ?

Eventually I will retire and not do any research any longer.

J.A.M. Vermaseren

OpenFORM

- Make FORM an open source program
- Good programmers can make improvements

Prerequistites

- Several improvements needed for OpenFORM
  - Improved internal documentation
  - Internal coherency
  - Update to current programming practises
  - Definition of set of rules for additions and extensions

Strategy

- Make FORM useful for a large enough community in science
- Rewrite FORM in C++
  $\longrightarrow$ layout of objects and structure like C++ from outside
  $\longrightarrow$ keep flat (and fast) data structure internally
- Implement wish-list of additions and new features

Epilogue

FORM *is very versatile and powerful and you have to make everything yourself.*