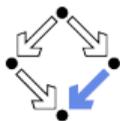


# FORMAL MODELING

## Algebra and Logic in Mathematical Modeling



Carsten Schneider    Wolfgang Schreiner    Wolfgang Windsteiger  
Research Institute for Symbolic Computation (RISC)  
Johannes Kepler University, Linz, Austria  
*FirstName.LastName@risc.jku.at*



## 1. Introduction

## 2. Symbolic Summation and the Modeling of Sequences

## 3. Logical Models of Problems and Computations

## 4. Modeling Problems in Geometry and Discrete Mathematics

- Problem 1: How to get from  $A$  to  $B$ ?
- Problem 2: How to efficiently use resources?

## 5. Organization

# Introduction

What is this course about?

- Application of techniques from symbolic computation.
  - Rooted in computer algebra, algebraic geometry, computational logic.
  - Focus is on correct formalization, precise analysis, exact solving (rather than on fast but numerically approximated computations).
- Modeling and analysis of problems in various application domains.
  - Symbolic summation and sequences, geometry and discrete mathematics, programs and computational systems, ...
- Theoretical frameworks and practical tools.
  - Computer algebra and automated reasoning software.

Prerequisites for the algorithmization and automation of mathematics.

# Contents

What are you going to see?

- Symbolic Summation and the Modeling of Sequences.
  - Carsten Schneider.
- Logic Models of Problems and Computations.
  - Wolfgang Schreiner.
- Modeling Problems in Geometry and Discrete Mathematics.
  - Wolfgang Windsteiger.

A (non-exhaustive) selection of topics pursued at the RISC institute.

## 1. Introduction

## 2. Symbolic Summation and the Modeling of Sequences

## 3. Logical Models of Problems and Computations

## 4. Modeling Problems in Geometry and Discrete Mathematics

- Problem 1: How to get from  $A$  to  $B$ ?
- Problem 2: How to efficiently use resources?

## 5. Organization

## Example: a challenging email

From: Doron Zeilberger  
To: Robin Pemantle, Herbert Wilf  
CC:Carsten Schneider

Robin and Herb,

I am willing to bet that Carsten Schneider's SIGMA package for handling sums with harmonic numbers (among others) can do it in a jiffy. I am Cc-ing this to Carsten.

Carsten: please do it, and Cc- the answer to me.

-Doron

# The problem

From: Robin Pemantle [University of Pennsylvania]

To: herb wilf; doron zeilberger

Herb, Doron,

I have a sum that, when I evaluate numerically, looks suspiciously like it comes out to exactly 1.

Is there a way I can automatically decide this?

The sum may be written in many ways, but one is:

$$\sum_{k=1}^{\infty} \frac{H_{k+1} - 1}{k(k+1)} \boxed{\sum_{j=1}^{\infty} \frac{H_j}{j(j+k)}}$$

with

$$H_j := \sum_{i=1}^j \frac{1}{i} (= H_j)$$

## The summation paradigms

$$\mathbf{A}(k) = \sum_{j=1}^{\infty} \frac{H_j}{j(j+k)}$$

## The summation paradigms

$$k^2 \mathbf{A}(k) - (k+1)(2k+1) \mathbf{A}(k+1) + (k+1)(k+2) \mathbf{A}(k+2) = \frac{1}{k+1}$$

Recurrence finder

$$\mathbf{A}(k) = \sum_{j=1}^{\infty} \frac{H_j}{j(j+k)}$$

## The summation paradigms

$$k^2 \mathbf{A}(k) - (k+1)(2k+1)\mathbf{A}(k+1) + (k+1)(k+2)\mathbf{A}(k+2) = \frac{1}{k+1}$$

Recurrence solver

$$\mathbf{A}(k) = \sum_{j=1}^{\infty} \frac{H_j}{j(j+k)} \in \left\{ c_1 \frac{H_k}{k} + c_2 \frac{1}{k} + \frac{kH_k^2 - 2H_k + kH_k^{(2)}}{2k^2} \mid c_1, c_2 \in \mathbb{R} \right\}$$

## The summation paradigms

$$k^2 \mathbf{A}(k) - (k+1)(2k+1)\mathbf{A}(k+1) + (k+1)(k+2)\mathbf{A}(k+2) = \frac{1}{k+1}$$

Recurrence solver

$$\mathbf{A}(k) = \sum_{j=1}^{\infty} \frac{H_j}{j(j+k)} = 0 \frac{H_k}{k} + \zeta(2) \frac{1}{k} + \frac{kH_k^2 - 2H_k + kH_k^{(2)}}{2k^2}$$

where

$$\zeta(z) = \sum_{i=1}^{\infty} \frac{1}{i^z} \quad H_k^{(2)} = \sum_{i=1}^k \frac{1}{i^2}$$

In[1]:= << **Sigma.m**

Sigma - A summation package by Carsten Schneider © RISC-Linz

In[2]:= **mySum** =  $\sum_{k=1}^a \frac{H_k}{k(k+n)}$

In[1]:= << Sigma.m

Sigma - A summation package by Carsten Schneider © RISC-Linz

$$\text{In[2]:= mySum} = \sum_{k=1}^a \frac{H_k}{k(k+n)}$$

In[3]:= rec = GenerateRecurrence[mySum, n][[1]]

$$\text{Out[3]= } n^2 \text{SUM}[n] - (n+1)(2n+1) \text{SUM}[n+1] + (n+1)(n+2) \text{SUM}[n+2] == \\ \frac{(-a-1)H_a}{(a+n+1)(a+n+2)} + \frac{a}{(n+1)(a+n+1)}$$

In[1]:= << Sigma.m

Sigma - A summation package by Carsten Schneider © RISC-Linz

$$\text{In[2]:= mySum} = \sum_{k=1}^a \frac{H_k}{k(k+n)}$$

In[3]:= rec = GenerateRecurrence[mySum, n][[1]]

$$\begin{aligned} \text{Out[3]= } n^2 \text{SUM}[n] - (n+1)(2n+1) \text{SUM}[n+1] + (n+1)(n+2) \text{SUM}[n+2] == \\ \frac{(-a-1)H_a}{(a+n+1)(a+n+2)} + \frac{a}{(n+1)(a+n+1)} \end{aligned}$$

In[4]:= rec = LimitRec[rec, SUM[n], {n}, a]

$$\text{Out[4]= } n^2 \text{SUM}[n] - (n+1)(2n+1) \text{SUM}[n+1] + (n+1)(n+2) \text{SUM}[n+2] = \frac{1}{n+1}$$

In[1]:= << Sigma.m

Sigma - A summation package by Carsten Schneider © RISC-Linz

$$\text{In[2]:= mySum} = \sum_{k=1}^a \frac{H_k}{k(k+n)}$$

In[3]:= rec = GenerateRecurrence[mySum, n][[1]]

$$\text{Out[3]= } n^2 \text{SUM}[n] - (n+1)(2n+1)\text{SUM}[n+1] + (n+1)(n+2)\text{SUM}[n+2] == \\ \frac{(-a-1)H_a}{(a+n+1)(a+n+2)} + \frac{a}{(n+1)(a+n+1)}$$

In[4]:= rec = LimitRec[rec, SUM[n], {n}, a]

$$\text{Out[4]= } n^2 \text{SUM}[n] - (n+1)(2n+1)\text{SUM}[n+1] + (n+1)(n+2)\text{SUM}[n+2] = \frac{1}{n+1}$$

In[5]:= recSol = SolveRecurrence[rec, SUM[n]]

$$\text{Out[5]= } \left\{ \left\{ 0, \frac{1}{n} \right\}, \left\{ 0, \frac{\sum_{i=1}^n \frac{1}{i}}{n} - \frac{1}{n^2} \right\}, \left\{ 1, \frac{\left(\sum_{i=1}^n \frac{1}{i}\right)^2}{2n} - \frac{\sum_{i=1}^n \frac{1}{i}}{n^2} + \frac{\sum_{i=1}^n \frac{1}{i^2}}{2n} \right\} \right\}$$

In[1]:= << Sigma.m

Sigma - A summation package by Carsten Schneider © RISC-Linz

$$\text{In[2]:= mySum} = \sum_{k=1}^a \frac{H_k}{k(k+n)}$$

In[3]:= rec = GenerateRecurrence[mySum, n][[1]]

$$\text{Out[3]= } n^2 \text{SUM}[n] - (n+1)(2n+1)\text{SUM}[n+1] + (n+1)(n+2)\text{SUM}[n+2] == \\ \frac{(-a-1)H_a}{(a+n+1)(a+n+2)} + \frac{a}{(n+1)(a+n+1)}$$

In[4]:= rec = LimitRec[rec, SUM[n], {n}, a]

$$\text{Out[4]= } n^2 \text{SUM}[n] - (n+1)(2n+1)\text{SUM}[n+1] + (n+1)(n+2)\text{SUM}[n+2] = \frac{1}{n+1}$$

In[5]:= recSol = SolveRecurrence[rec, SUM[n]]

$$\text{Out[5]= } \left\{ \left\{ 0, \frac{1}{n} \right\}, \left\{ 0, \frac{\sum_{i=1}^n \frac{1}{i}}{n} - \frac{1}{n^2} \right\}, \left\{ 1, \frac{\left(\sum_{i=1}^n \frac{1}{i}\right)^2}{2n} - \frac{\sum_{i=1}^n \frac{1}{i}}{n^2} + \frac{\sum_{i=1}^n \frac{1}{i^2}}{2n} \right\} \right\}$$

In[6]:= FindLinearCombination[recSol, {1, {\zeta\_2, 1/2 + \zeta\_2/2}}, n, 2]

$$\text{Out[6]= } -\frac{\sum_{i=1}^n \frac{1}{i}}{n^2} + \frac{\left(\sum_{i=1}^n \frac{1}{i}\right)^2}{2n} + \frac{\sum_{i=1}^n \frac{1}{i^2}}{2n} + \frac{\zeta_2}{n}$$

## Example

$$S = \sum_{k=1}^{\infty} \frac{H_{k+1} - 1}{k(k+1)} \underbrace{\left[ \sum_{j=1}^{\infty} \frac{H_j}{j(j+k)} \right]}_{= \frac{\zeta(2)}{k} + \frac{kH_k^2 - 2H_k + kH_k^{(2)}}{2k^2}}$$

## Example

$$\begin{aligned} S &= \sum_{k=1}^{\infty} \frac{H_{k+1} - 1}{k(k+1)} \underbrace{\left[ \sum_{j=1}^{\infty} \frac{H_j}{j(j+k)} \right]} \\ &= \frac{\zeta(2)}{k} + \frac{kH_k^2 - 2H_k + kH_k^{(2)}}{2k^2} \\ &= -4\zeta(2) + (\zeta(2) - 1) \sum_{i=1}^{\infty} \frac{H_i}{i^2} - \sum_{i=1}^{\infty} \frac{H_i^2}{i^3} + \frac{1}{2} \sum_{i=1}^{\infty} \frac{H_i^3}{i^2} + \frac{1}{2} \sum_{i=1}^{\infty} \frac{H_i H_i^{(2)}}{i^2} \end{aligned}$$

## Example

$$\begin{aligned} S &= \sum_{k=1}^{\infty} \frac{H_{k+1} - 1}{k(k+1)} \underbrace{\left[ \sum_{j=1}^{\infty} \frac{H_j}{j(j+k)} \right]} \\ &= \frac{\zeta(2)}{k} + \frac{kH_k^2 - 2H_k + kH_k^{(2)}}{2k^2} \\ &= -4\zeta(2) + (\zeta(2) - 1) \sum_{i=1}^{\infty} \frac{H_i}{i^2} - \sum_{i=1}^{\infty} \frac{H_i^2}{i^3} + \frac{1}{2} \sum_{i=1}^{\infty} \frac{H_i^3}{i^2} + \frac{1}{2} \sum_{i=1}^{\infty} \frac{H_i H_i^{(2)}}{i^2} \\ &= -4\zeta(2) - 2\zeta(3) + 4\zeta(2)\zeta(3) + 2\zeta(5) = 0.999222\dots \end{aligned}$$

## The basic idea (special case telescoping)

FIND  $g(k)$ :

$$H_k = g(k+1) - g(k)$$

## The basic idea (special case telescoping)

FIND  $g(k)$ :

$$H_k = g(k+1) - g(k)$$

A difference ring for the **summand**:

Construct a formal ring

$$\mathbb{A} := \underbrace{\mathbb{Q}(x)}_{\text{rat. fu. field}} [s]$$

polynomial ring

## The basic idea (special case telescoping)

FIND  $g(k)$ :

$$H_k = g(k+1) - g(k)$$

A difference ring for the **summand**:

Construct a formal ring

$$\mathbb{A} := \mathbb{Q}(x)[s]$$

and a ring automorphism  $\sigma : \mathbb{A} \rightarrow \mathbb{A}$  defined by

$$\sigma(c) = c \quad \forall c \in \mathbb{Q},$$

$$\sigma(x) = x + 1,$$

$$\sigma(s) = s + \frac{1}{x+1},$$

$$S k = k + 1,$$

$$S H_k = H_k + \frac{1}{k+1}.$$

## The basic idea (special case telescoping)

FIND  $g(k)$ :

$$H_k = g(k+1) - g(k)$$

A difference ring for the **summand**:

Construct a formal ring

$(\mathbb{A}, \sigma)$  *RΠΣ*-ring

$$\mathbb{A} := \mathbb{Q}(x)[s]$$

Karr 1981, Schneider 2001–

and a ring automorphism  $\sigma : \mathbb{A} \rightarrow \mathbb{A}$  defined by

$$\sigma(c) = c \quad \forall c \in \mathbb{Q},$$

$$\sigma(x) = x + 1,$$

$$\sigma(s) = s + \frac{1}{x+1},$$

$$S k = k + 1,$$

$$S H_k = H_k + \frac{1}{k+1}.$$

with

$$\text{const}_\sigma(\mathbb{A}) = \{c \in \mathbb{A} \mid \sigma(c) = c\} = \mathbb{Q}$$

## The basic idea (special case telescoping)

GIVEN FIND  $g(k)$ :

$$H_k = g(k+1) - g(k)$$

FIND  $g \in \mathbb{A}$ :

$$s = \sigma(g) - g$$

## The basic idea (special case telescoping)

GIVEN FIND  $g(k)$ :

$$H_k = g(k+1) - g(k)$$

FIND  $g \in \mathbb{A}$ :

$$s = \sigma(g) - g$$

recursive ansatz

$$g = x s - x$$

## The basic idea (special case telescoping)

GIVEN FIND  $g(k)$ :

$$H_k = g(k+1) - g(k)$$

FIND  $g \in \mathbb{A}$ :

$$x \equiv k$$

$$s \equiv H_k$$

$$s = \sigma(g) - g$$

recursive ansatz

$$g = x s - x$$

## The basic idea (special case telescoping)

GIVEN FIND  $g(k)$ :

$$H_k = g(k + 1) - g(k)$$

Summation of the telescoping equation over  $k$  from 1 to  $n$  yields

$$\sum_{k=1}^n H_k = g(n + 1) - g(1)$$

## The basic idea (special case telescoping)

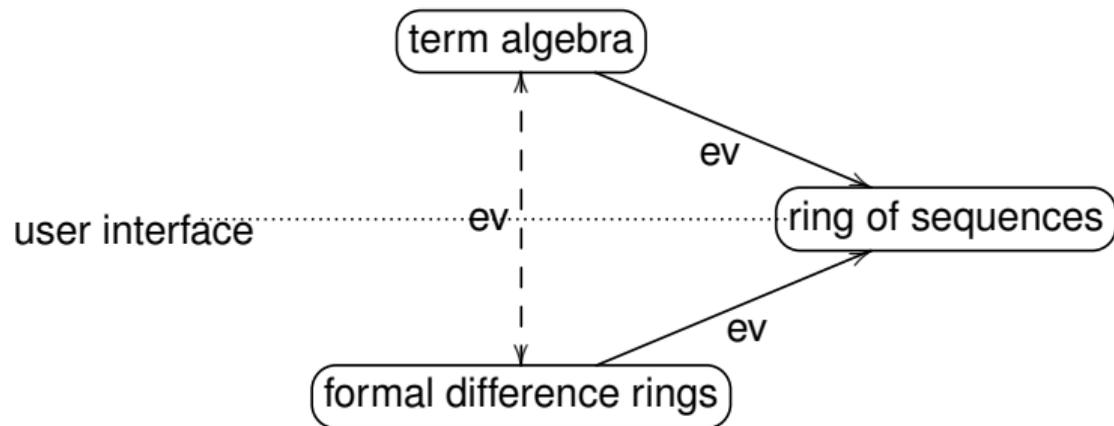
GIVEN FIND  $g(k)$ :

$$H_k = g(k+1) - g(k)$$

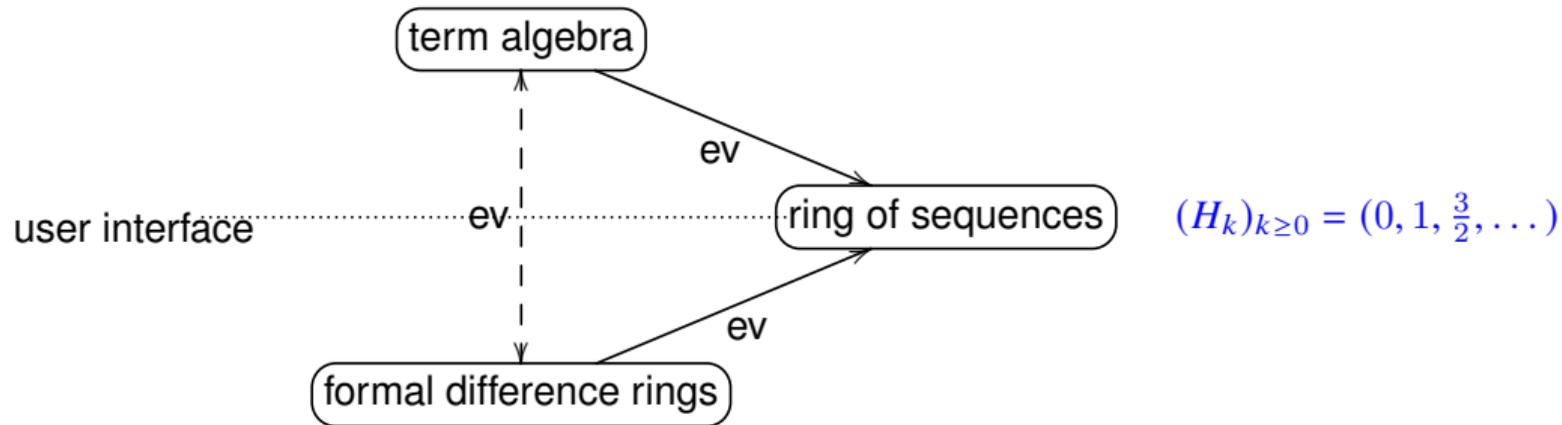
Summation of the telescoping equation over  $k$  from 1 to  $n$  yields

$$\sum_{k=1}^n H_k = g(n+1) - g(1)$$
$$= (n+1)H_{n+1} - (n+1).$$

## Modeling of sequences (built by nested sums/products)

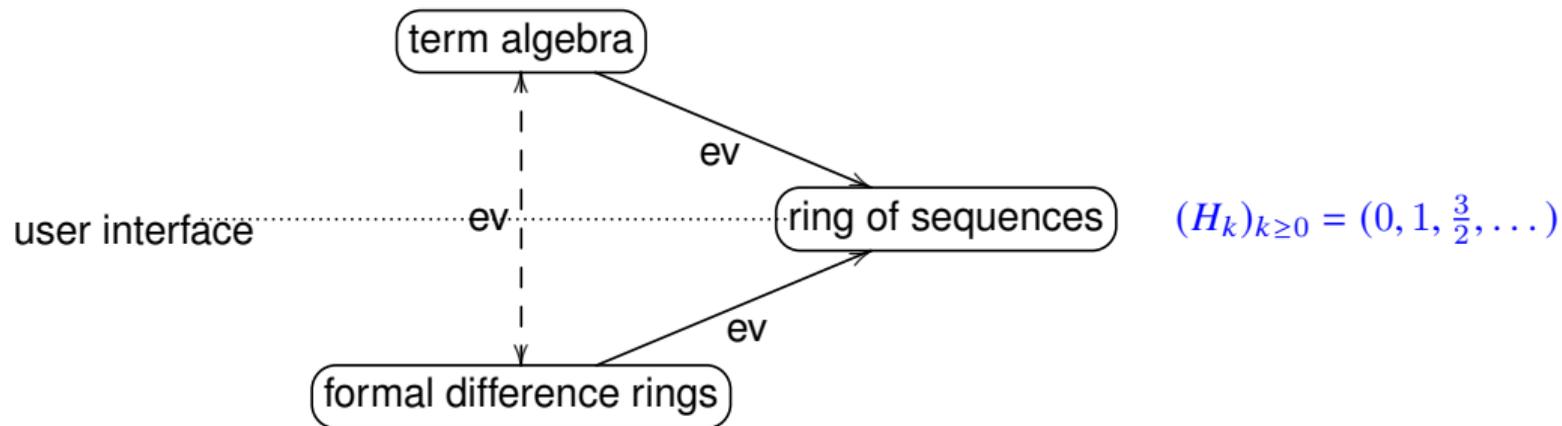


# Modeling of sequences (built by nested sums/products)



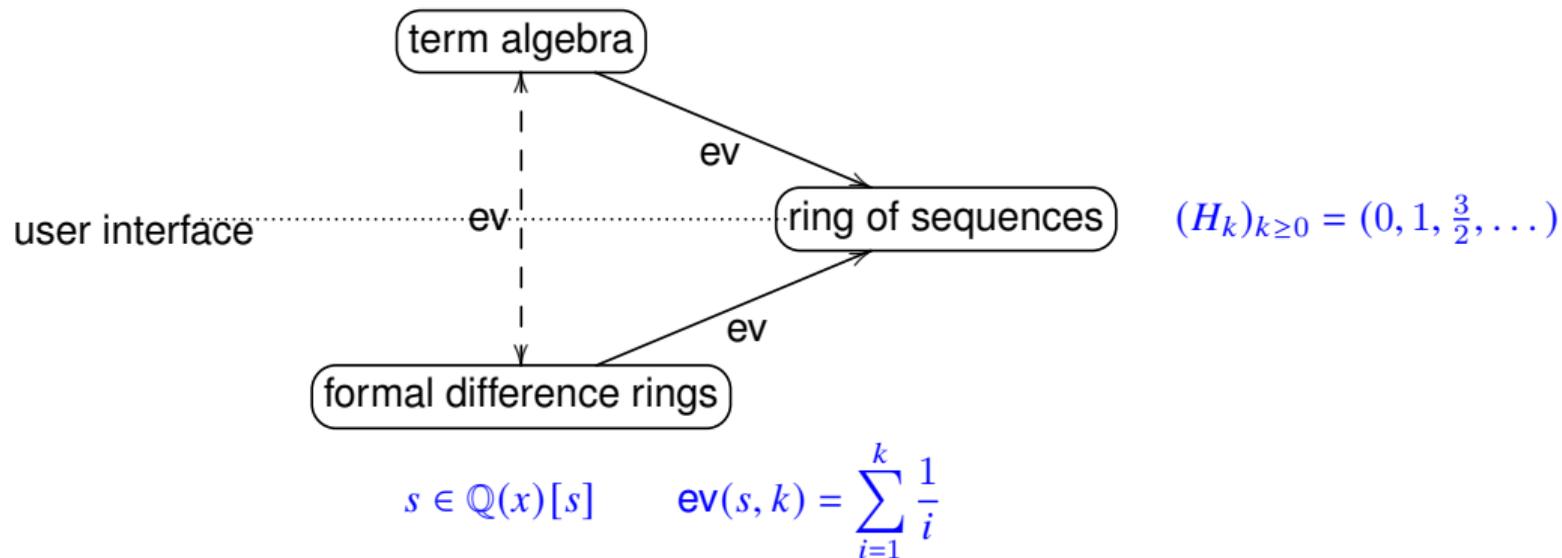
# Modeling of sequences (built by nested sums/products)

$$H = \text{Sum}(1, \frac{1}{x}) \quad \text{ev}(H, k) = \sum_{i=1}^k \frac{1}{i}$$



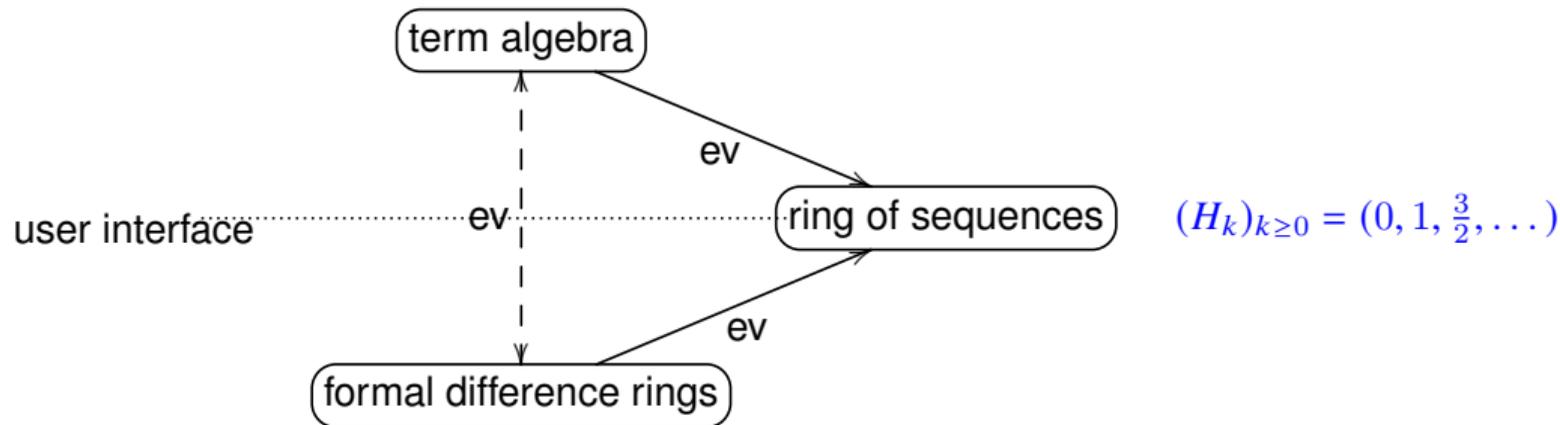
# Modeling of sequences (built by nested sums/products)

$$H = \text{Sum}(1, \frac{1}{x}) \quad \text{ev}(H, k) = \sum_{i=1}^k \frac{1}{i}$$



# Modeling of sequences (built by nested sums/products)

$$H = \text{Sum}(1, \frac{1}{x}) \quad \text{ev}(H, k) = \sum_{i=1}^k \frac{1}{i}$$



$$s \in \mathbb{Q}(x)[s] \quad \text{ev}(s, k) = \sum_{i=1}^k \frac{1}{i}$$

computer algebra algorithms  
(for unique representations,  
recurrence finding and solving)

## Possible proseminar topics

1. Carry out a concrete (non-trivial example) and present details on the different modeling layers
2. Elaborate on canonical simplifiers (unique representation) and the simplification of summation objects in the difference ring setting
3. The interaction of term algebras and computer algebra
4. Elaborate on the modeling of sequences with the holonomic approach (representation of sequences by recurrences and initial values)

In[1]:= << **Sigma.m**

Sigma - A summation package by Carsten Schneider © RISC-Linz

In[2]:= << **HarmonicSums.m**

HarmonicSums by Jakob Ablinger © RISC-Linz

In[3]:= << **EvaluateMultiSums.m**

EvaluateMultiSums by Carsten Schneider © RISC-Linz

In[4]:= **EvaluateMultiSum** $\left[ \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \frac{H_k (H_{n+1} - 1)}{kn(n+1)(k+n)} \right]$

In[1]:= << **Sigma.m**

Sigma - A summation package by Carsten Schneider © RISC-Linz

In[2]:= << **HarmonicSums.m**

HarmonicSums by Jakob Ablinger © RISC-Linz

In[3]:= << **EvaluateMultiSums.m**

EvaluateMultiSums by Carsten Schneider © RISC-Linz

In[4]:= **EvaluateMultiSum** $\left[ \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \frac{H_k (H_{n+1} - 1)}{kn(n+1)(k+n)} \right]$

Out[4]=  $-4\zeta_2 - 2\zeta_3 + 4\zeta_2\zeta_3 + 2\zeta_5$

In[1]:= << **Sigma.m**

Sigma - A summation package by Carsten Schneider © RISC-Linz

In[2]:= << **HarmonicSums.m**

HarmonicSums by Jakob Ablinger © RISC-Linz

In[3]:= << **EvaluateMultiSums.m**

EvaluateMultiSums by Carsten Schneider © RISC-Linz

In[4]:= **EvaluateMultiSum** $\left[ \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \frac{H_k (H_{n+1} - 1)}{kn(n+1)(k+n)} \right]$

Out[4]=  $-4\zeta_2 - 2\zeta_3 + 4\zeta_2\zeta_3 + 2\zeta_5$

In[5]:= **EvaluateMultiSum** $\left[ \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \frac{H_k^2 (H_{n+1} - 1)^2}{k(k+n)n} \right]$

Out[5]=  $-10\zeta_3 + \zeta_2^2 \left( \frac{58\zeta_3}{5} - \frac{29}{5} \right) - 10\zeta_5 + \zeta_2(-\zeta_3 + 13\zeta_5 - 4) + \frac{457\zeta_7}{8}$

In[1]:= << **Sigma.m**

Sigma - A summation package by Carsten Schneider © RISC-Linz

In[2]:= << **HarmonicSums.m**

HarmonicSums by Jakob Ablinger © RISC-Linz

In[3]:= << **EvaluateMultiSums.m**

EvaluateMultiSums by Carsten Schneider © RISC-Linz

In[4]:= **EvaluateMultiSum** $\left[ \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \frac{H_k (H_{n+1} - 1)}{kn(n+1)(k+n)} \right]$

Out[4]=  $-4\zeta_2 - 2\zeta_3 + 4\zeta_2\zeta_3 + 2\zeta_5$

In[5]:= **EvaluateMultiSum** $\left[ \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \frac{H_k (H_{n+1} - 1)}{k(k+n)^2 n^2} \right]$

Out[5]=  $2\zeta_3 + \zeta_2^2 \left( \frac{17\zeta_3}{10} + \frac{17}{10} \right) + \zeta_2(2\zeta_3 - 3\zeta_5 - 4) - \frac{9\zeta_5}{2} + \frac{3\zeta_7}{16}$

In[1]:= << **Sigma.m**

Sigma - A summation package by Carsten Schneider © RISC-Linz

In[2]:= << **HarmonicSums.m**

HarmonicSums by Jakob Ablinger © RISC-Linz

In[3]:= << **EvaluateMultiSums.m**

EvaluateMultiSums by Carsten Schneider © RISC-Linz

In[4]:= **EvaluateMultiSum** $\left[ \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \frac{H_k (H_{n+1} - 1)}{kn(n+1)(k+n)} \right]$

Out[4]=  $-4\zeta_2 - 2\zeta_3 + 4\zeta_2\zeta_3 + 2\zeta_5$

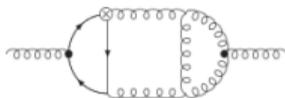
In[5]:= **EvaluateMultiSum** $\left[ \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \sum_{l=1}^{\infty} \frac{H_k H_n H_{n+l+k}}{k(k+n)(k+n+l+1)^2} \right]$

Out[5]=  $3\zeta_3^2 - \frac{15\zeta_5}{2} + \zeta_2(9\zeta_5 - 6\zeta_3) + \frac{149\zeta_7}{16} + \frac{114}{35}\zeta_2^3$

# Applications:

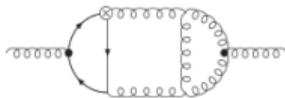
- analysis of algorithms (QuickSort, AVL trees, ...)
- combinatorial problems
- number theory
- numerics
- statistics
- special functions
- complex analysis
- ⋮
- **particle physics**

# Application: Evaluation of Feynman integrals



Behavior of particles

# Application: Evaluation of Feynman integrals



Behavior of particles



$$\int \Phi(N, \epsilon, x) dx$$

Feynman integrals

# Feynman integrals

$$\int_0^1 x^N dx$$

# Feynman integrals

$$\int_0^1 x^N (1+x)^N dx$$

# Feynman integrals

$$\int_0^1 \frac{x^N (1+x)^N}{(1-x)^{1+\varepsilon}} dx$$

# Feynman integrals

$$\int_0^1 \int_0^1 \frac{x_1^N (1+x_1)^N}{(1-x_1)^{1+\varepsilon}} \dots dx_1 dx_2$$

# Feynman integrals

$$\int_0^1 \int_0^1 \int_0^1 \frac{x_1^N (1+x_1)^N}{(1-x_1)^{1+\varepsilon}} \dots dx_1 dx_2 dx_3$$

# Feynman integrals

$$\int_0^1 \int_0^1 \int_0^1 \int_0^1 \frac{x_1^N (1+x_1)^N}{(1-x_1)^{1+\varepsilon}} \dots dx_1 dx_2 dx_3 dx_4$$

# Feynman integrals

$$\int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \frac{x_1^N (1+x_1)^N}{(1-x_1)^{1+\varepsilon}} \dots dx_1 dx_2 dx_3 dx_4 dx_5$$

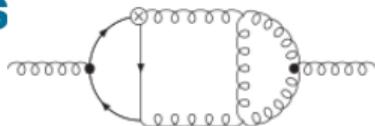
# Feynman integrals

$$\int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \frac{x_1^N (1+x_1)^N}{(1-x_1)^{1+\varepsilon}} \dots dx_1 dx_2 dx_3 dx_4 dx_5 dx_6$$

# Feynman integrals

$$\sum_{j=0}^{N-3} \sum_{k=0}^j \binom{N-1}{j+2} \binom{j+1}{k+1} \\ \times \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \frac{x_1^N (1+x_1)^{N-j+k}}{(1-x_1)^{1+\varepsilon}} \dots dx_1 dx_2 dx_3 dx_4 dx_5 dx_6$$

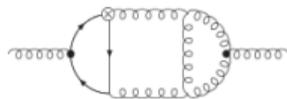
# Feynman integrals



a 3-loop massive ladder diagram  
[arXiv:1509.08324]

$$\begin{aligned}
 & \sum_{j=0}^{N-3} \sum_{k=0}^j \binom{N-1}{j+2} \binom{j+1}{k+1} \quad || \\
 & \times \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \theta(1-x_5-x_6)(1-x_2)(1-x_4)x_2^{-\varepsilon} \\
 & (1-x_2)^{-\varepsilon} x_4^{\varepsilon/2-1} (1-x_4)^{\varepsilon/2-1} x_5^{\varepsilon-1} x_6^{-\varepsilon p/2} \\
 & \left[ \begin{aligned} & [-x_3(1-x_4) - x_4(1-x_5-x_6+x_5x_1+x_6x_3)]^k \\ & + [x_3(1-x_4) - (1-x_4)(1-x_5-x_6+x_5x_1+x_6x_3)]^k \end{aligned} \right] \\
 & \times (1-x_5-x_6+x_5x_1+x_6x_3)^{j-k} (1-x_2)^{N-3-j}
 \end{aligned}$$

# Application: Evaluation of Feynman integrals



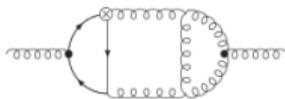
Behavior of particles



$$\int \Phi(N, \epsilon, x) dx$$

Feynman integrals

# Application: Evaluation of Feynman integrals



Behavior of particles



$$\int \Phi(N, \epsilon, x) dx$$

Feynman integrals

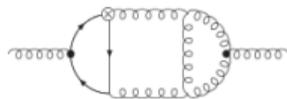


**DESY**

$$\sum f(N, \epsilon, k)$$

complicated  
multi-sums

# Application: Evaluation of Feynman integrals



Behavior of particles



$$\int \Phi(N, \epsilon, x) dx$$

Feynman integrals

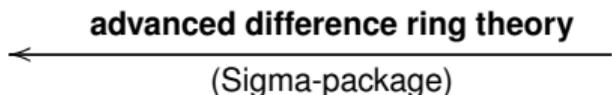


**DESY**

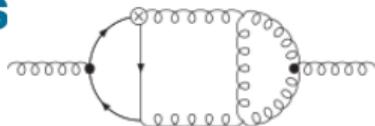
$$\sum f(N, \epsilon, k)$$

complicated  
multi-sums

expression in  
special functions

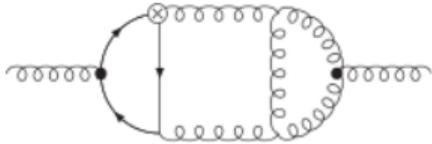


# Feynman integrals

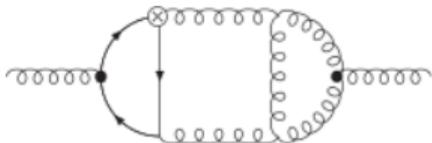


a 3-loop massive ladder diagram  
[arXiv:1509.08324]

$$\begin{aligned}
 & \sum_{j=0}^{N-3} \sum_{k=0}^j \binom{N-1}{j+2} \binom{j+1}{k+1} \quad || \\
 & \times \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \theta(1-x_5-x_6)(1-x_2)(1-x_4)x_2^{-\varepsilon} \\
 & (1-x_2)^{-\varepsilon} x_4^{\varepsilon/2-1} (1-x_4)^{\varepsilon/2-1} x_5^{\varepsilon-1} x_6^{-\varepsilon p/2} \\
 & \left[ \begin{aligned} & [-x_3(1-x_4) - x_4(1-x_5-x_6+x_5x_1+x_6x_3)]^k \\ & + [x_3(1-x_4) - (1-x_4)(1-x_5-x_6+x_5x_1+x_6x_3)]^k \end{aligned} \right] \\
 & \times (1-x_5-x_6+x_5x_1+x_6x_3)^{j-k} (1-x_2)^{N-3-j}
 \end{aligned}$$



$$= F_{-3}(N)\varepsilon^{-3} + F_{-2}(N)\varepsilon^{-2} + F_{-1}(N)\varepsilon^{-1} + \boxed{F_0(N)}$$



$$= F_{-3}(N)\varepsilon^{-3} + F_{-2}(N)\varepsilon^{-2} + F_{-1}(N)\varepsilon^{-1} + \boxed{F_0(N)}$$

Simplify

||

$$\sum_{j=0}^{N-3} \sum_{k=0}^j \sum_{l=0}^k \sum_{q=0}^{-j+N-3} \sum_{s=1}^{-l+N-q-3} \sum_{r=0}^{-l+N-q-s-3} (-1)^{-j+k-l+N-q-3} \times$$

$$\times \frac{\binom{j+1}{k+1} \binom{k}{l} \binom{N-1}{j+2} \binom{-j+N-3}{q} \binom{-l+N-q-3}{s} \binom{-l+N-q-s-3}{r} r! (-l+N-q-r-s-3)! (s-1)!}{(-l+N-q-2)! (-j+N-1) (N-q-r-s-2) (q+s+1)}$$

$$\left[ 4H_{-j+N-1} - 4H_{-j+N-2} - 2H_k - (H_{-l+N-q-2} + H_{-l+N-q-r-s-3} - 2H_{r+s}) \right.$$

$$\left. + 2H_{s-1} - 2H_{r+s} \right] + \mathbf{3 \text{ further 6-fold sums}}$$

$$\boxed{F_0(N)} =$$

$$\begin{aligned}
& \frac{7}{12}S_1(N)^4 + \frac{(17N+5)S_1(N)^3}{3N(N+1)} + \left( \frac{35N^2-2N-5}{2N^2(N+1)^2} + \frac{13S_2(N)}{2} + \frac{5(-1)^N}{2N^2} \right) S_1(N)^2 \\
& + \left( -\frac{4(13N+5)}{N^2(N+1)^2} + \left( \frac{4(-1)^N(2N+1)}{N(N+1)} - \frac{13}{N} \right) S_2(N) + \left( \frac{29}{3} - (-1)^N \right) S_3(N) \right. \\
& + (2+2(-1)^N)S_{2,1}(N) - 28S_{-2,1}(N) + \frac{20(-1)^N}{N^2(N+1)} S_1(N) + \left( \frac{3}{4} + (-1)^N \right) S_2(N)^2 \\
& - 2(-1)^N S_{-2}(N)^2 + S_{-3}(N) \left( \frac{2(3N-5)}{N(N+1)} + (26+4(-1)^N)S_1(N) + \frac{4(-1)^N}{N+1} \right) \\
& + \left( \frac{(-1)^N(5-3N)}{2N^2(N+1)} - \frac{5}{2N^2} \right) S_2(N) + S_{-2}(N)(10S_1(N)^2 + \left( \frac{8(-1)^N(2N+1)}{N(N+1)} \right. \\
& + \frac{4(3N-1)}{N(N+1)} S_1(N) + \frac{8(-1)^N(3N+1)}{N(N+1)^2} + (-22+6(-1)^N)S_2(N) - \frac{16}{N(N+1)} \left. \right) \\
& + \left( \frac{(-1)^N(9N+5)}{N(N+1)} - \frac{29}{3N} \right) S_3(N) + \left( \frac{19}{2} - 2(-1)^N \right) S_4(N) + (-6+5(-1)^N)S_{-4}(N) \\
& + \left( -\frac{2(-1)^N(9N+5)}{N(N+1)} - \frac{2}{N} \right) S_{2,1}(N) + (20+2(-1)^N)S_{2,-2}(N) + (-17+13(-1)^N)S_{3,1}(N) \\
& - \frac{8(-1)^N(2N+1)+4(9N+1)}{N(N+1)} S_{-2,1}(N) - (24+4(-1)^N)S_{-3,1}(N) + (3-5(-1)^N)S_{2,1,1}(N) \\
& + 32S_{-2,1,1}(N) + \left( \frac{3}{2}S_1(N)^2 - \frac{3S_1(N)}{N} + \frac{3}{2}(-1)^N S_{-2}(N) \right) \zeta(2)
\end{aligned}$$

$$\boxed{F_0(N)} =$$

$$\begin{aligned}
& \frac{7}{12}S_1(N)^4 + \frac{(17N+5)S_1(N)^3}{N^2(N+1)} \left( \frac{35N^2-2N-5}{2N^2(N+1)^2} + \frac{13S_2(N)}{2} + \frac{5(-1)^N}{2N^2} \right) S_1(N)^2 \\
& + \left( -\frac{4(13N+1)}{N^2(N+1)} + 1 - \frac{13}{N} \right) S_2(N) + \left( \frac{29}{3} - (-1)^N \right) S_3(N) \\
& + (2 + 2(-1)^N)S_{2,1}(N) - 28S_{-2,1}(N) + \frac{20(-1)^N}{N^2(N+1)} S_1(N) + \left( \frac{3}{4} + (-1)^N \right) S_2(N)^2 \\
& - 2(-1)^N S_{-2}(N)^2 + S_{-3}(N) \left( \frac{2(3N-5)}{N(N+1)} + (26 + 4(-1)^N)S_1(N) + \frac{4(-1)^N}{N+1} \right) \\
& + \left( \frac{(-1)^N(5-3N)}{2N^2(N+1)} - \frac{5}{2N^2} \right) S_2(N) + S_{-2}(N) (10S_1(N)^2 + \frac{8(-1)^N(2N+1)}{N(N+1)}) \\
& + \frac{4(3N-1)}{N(N+1)} S_1(N) + \frac{8(-1)^N(3N+1)}{N(N+1)^2} + \left( -22 + 6(-1)^N \right) S_2(N) - \frac{16}{N(N+1)} \\
& + \left( \frac{(-1)^N(9N+5)}{N(N+1)} - \frac{29}{3N} \right) S_3(N) + \left( \frac{19}{2} - 2(-1)^N \right) S_4(N) + \left( -6 + 5(-1)^N \right) S_{-4}(N) \\
& + \left( -\frac{2(-1)^N(9N+5)}{N(N+1)} - \frac{2}{N} \right) S_{2,1}(N) + (20 + 2(-1)^N)S_{2,-2}(N) + \left( -17 + 13(-1)^N \right) S_{3,1}(N) \\
& - \frac{8(-1)^N(2N+1) + 4(9N+1)}{N(N+1)} S_{-2,1}(N) - (24 + 4(-1)^N)S_{-3,1}(N) + (3 - 5(-1)^N)S_{2,1,1}(N) \\
& + 32S_{-2,1,1}(N) + \left( \frac{3}{2}S_1(N)^2 - \frac{3S_1(N)}{N} + \frac{3}{2}(-1)^N S_{-2}(N) \right) \zeta(2)
\end{aligned}$$

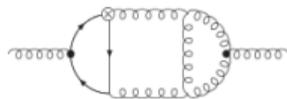
$$F_0(N) =$$

$$\begin{aligned}
& \frac{7}{12} S_1(N)^4 + \frac{(17N+5)S_1(N)^3}{2N^2(N+1)^2} + \left( \frac{35N^2 - 2N - 5}{2} + \frac{13S_2(N)}{2} + \frac{5(-1)^N}{2N^2} \right) S_1(N)^2 \\
& + \left( -\frac{4(13N+1)}{N^2(N+1)} S_1(N) = \sum_{i=1}^N \frac{1}{i} + 1 - \frac{13}{N} \right) S_2(N) + \left( \frac{29}{3} - (-1)^N \right) S_3(N) \\
& + (2 + 2(-1)^N) S_{2,1}(N) - 28S_{-2,1}(N) + \frac{20(-1)^N}{N^2(N+1)} S_1(N) + S_2(N) = \sum_{i=1}^N \frac{1}{i^2} \\
& - 2(-1)^N S_{-2}(N)^2 + S_{-3}(N) \left( \frac{2(3N-5)}{N(N+1)} + (26 + 4(-1)^N) S_1(N) \right) \\
& + \left( \frac{(-1)^N(5-3N)}{2N^2(N+1)} - \frac{5}{2N^2} \right) S_2(N) + S_{-2}(N) (10S_1(N)^2 + \frac{8(-1)^N(2N+1)}{N(N+1)}) \\
& + \frac{4(3N-1)}{N(N+1)} S_1(N) + \frac{8(-1)^N(3N+1)}{N(N+1)^2} + \left( -22 + 6(-1)^N \right) S_2(N) - \frac{16}{N(N+1)} \\
& + \left( \frac{(-1)^N(9N+5)}{N(N+1)} - \frac{29}{3N} \right) S_3(N) + \left( \frac{19}{2} - 2(-1)^N \right) S_4(N) + \left( -6 + 5(-1)^N \right) S_{-4}(N) \\
& + \left( -\frac{2(-1)^N(9N+5)}{N(N+1)} - \frac{2}{N} \right) S_{2,1}(N) + (20 + 2(-1)^N) S_{2,-2}(N) + (-17 + 13(-1)^N) S_{3,1}(N) \\
& - \frac{8(-1)^N(2N+1) + 4(9N+1)}{N(N+1)} S_{-2,1}(N) - (24 + 4(-1)^N) S_{-3,1}(N) + (3 - 5(-1)^N) S_{2,1,1}(N) \\
& + 32S_{-2,1,1}(N) + \left( \frac{3}{2} S_1(N)^2 - \frac{3S_1(N)}{N} + \frac{3}{2} (-1)^N S_{-2}(N) \right) \zeta(2)
\end{aligned}$$

$$F_0(N) =$$

$$\begin{aligned}
 & \frac{7}{12} S_1(N)^4 + \frac{(17N+5)S_1(N)^3}{2N^2(N+1)^2} + \frac{(35N^2 - 2N - 5)}{2} + \frac{13S_2(N)}{2} + \frac{5(-1)^N}{2N^2} S_1(N)^2 \\
 & + \left( -\frac{4(13N+1)}{N^2(N+1)} S_1(N) = \sum_{i=1}^N \frac{1}{i} + \frac{13}{N} \right) S_2(N) + \left( \frac{29}{3} - (-1)^N \right) S_3(N) \\
 & + (2 + 2(-1)^N) S_{2,1}(N) - 28S_{-2,1}(N) + \frac{20(-1)^N}{N^2(N+1)} S_1(N) + S_2(N) = \sum_{i=1}^N \frac{1}{i^2} \\
 & - 2(-1)^N S_{-2}(N)^2 + S_{-3}(N) \left( \frac{2(3N-5)}{N(N+1)} + (26 + 4(-1)^N) S_1(N) \right) \\
 & + \left( \frac{(-1)^N(5-3N)}{2N^2(N+1)} - \frac{5}{2N^2} \right) S_2(N) \\
 & + \frac{4(3N-1)}{N(N+1)} S_1(N) + \frac{8(-1)^N(3N-1)}{N(N+1)} \\
 & + \left( \frac{(-1)^N(9N+5)}{N(N+1)} - \frac{29}{3N} \right) S_3(N) + S_{-2,1,1}(N) = \sum_{i=1}^N \frac{(-1)^i \sum_{j=1}^i \frac{1}{k}}{i^2} \\
 & + \left( -\frac{2(-1)^N(9N+5)}{N(N+1)} - \frac{2}{N} \right) S_{2,1}(N) - (-1)^N S_{3,1}(N) \\
 & - \frac{8(-1)^N(2N+1) + 4(9N+1)}{N(N+1)} S_{-2,1}(N) - (24 + 4(-1)^N) S_{-3,1}(N) + (3 - 5(-1)^N) S_{2,1,1}(N) \\
 & + 32S_{-2,1,1}(N) + \left( \frac{3}{2} S_1(N)^2 - \frac{3S_1(N)}{N} + \frac{3}{2} (-1)^N S_{-2}(N) \right) \zeta(2)
 \end{aligned}$$

# Application: Evaluation of Feynman integrals



Behavior of particles



$$\int \Phi(N, \epsilon, x) dx$$

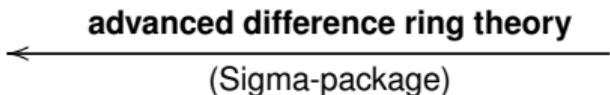
Feynman integrals



**DESY**

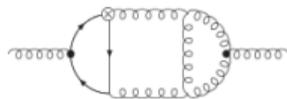
$$\sum f(N, \epsilon, k)$$

complicated  
multi-sums



expression in  
special functions

# Application: Evaluation of Feynman integrals



Behavior of particles



$$\int \Phi(N, \epsilon, x) dx$$

Feynman integrals



LHC at CERN



**DESY**

$$\sum f(N, \epsilon, k)$$

complicated  
multi-sums

applicable

expression in  
special functions

advanced difference ring theory

(Sigma-package)

## 1. Introduction

## 2. Symbolic Summation and the Modeling of Sequences

## 3. Logical Models of Problems and Computations

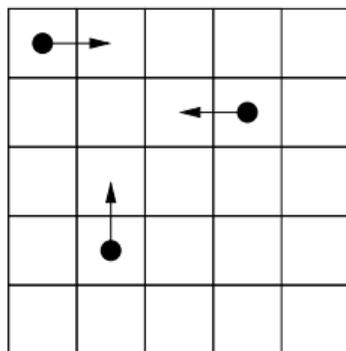
## 4. Modeling Problems in Geometry and Discrete Mathematics

- Problem 1: How to get from  $A$  to  $B$ ?
- Problem 2: How to efficiently use resources?

## 5. Organization

## Example: A Simple Robotic System

A grid in which multiple robots move around.



- **Control software:** move some robot one cell in some direction.
- **Safety property:** the robots shall not collide with the walls or with each other.

Our task is to model an adequate policy for the control software: given the current situation of the system, select a robot and a safe direction for its movement.

# A Model of the System

A hybrid logical/operational system description.

```
shared system Robots
{
  var x: Positions; var y: Positions;
  invariant noCollision(x, y);

  init(x0:Positions, y0:Positions) with initialState(x0, y0);
  {
    x := x0; y := y0;
  }

  action move(r:Robot, d:Direction) with nextDir(x, y, r, d);
  {
    x := moveX(x, r, d); y := moveY(y, r, d);
  }
}
```

The policy of the control software is defined by predicate `nextDir()`.

# Auxiliary Definitions

```
val R: N; // number of robots
val P: N; // number of positions
axiom notzero  $\Leftrightarrow R \geq 1 \wedge P \geq 1$ ;

type Robot = N[R-1];
type Position = N[P-1];
type Positions = Array[R,Position];
type Direction = N[4];
val Stop = 0; val Left = 1; val Right = 2; val Up = 3; val Down = 4;

fun moveX(x:Positions, r:Robot, d:Direction): Positions =
  if d = Left then x with [r] = x[r]-1 else
  if d = Right then x with [r] = x[r]+1 else x;
fun moveY(y:Positions, r:Robot, d:Direction): Positions =
  if d = Up then y with [r] = y[r]-1 else
  if d = Down then y with [r] = y[r]+1 else y;

pred noCollision(x:Positions, y:Positions)  $\Leftrightarrow$ 
   $\forall r1:Robot, r2:Robot \text{ with } r1 < r2. x[r1] \neq x[r2] \vee y[r1] \neq y[r2];$ 
```

# The Control Policy

```
// an initial state of the system
pred initialState(x: Positions, y: Positions)  $\Leftrightarrow$  noCollision(x, y);

// the relation between the current system state and the new direction d of robot r
pred nextDir(x: Positions, y: Positions, r: Robot, d: Direction)  $\Leftrightarrow$ 
  (d = Left  $\Rightarrow$  x[r] > 0  $\wedge$   $\neg$ anyOtherAt(x, y, r, x[r]-1, y[r]))
 $\wedge$  (d = Right  $\Rightarrow$  x[r] < P-1  $\wedge$   $\neg$ anyOtherAt(x, y, r, x[r]+1, y[r]))
 $\wedge$  (d = Up  $\Rightarrow$  y[r] > 0  $\wedge$   $\neg$ anyOtherAt(x, y, r, x[r], y[r]-1))
 $\wedge$  (d = Down  $\Rightarrow$  y[r] < P-1  $\wedge$   $\neg$ anyOtherAt(x, y, r, x[r], y[r]+1));

// some robot different from r is at position xr, yr
pred anyOtherAt(x: Positions, y: Positions, r: Robot, xr: Position, yr: Position)  $\Leftrightarrow$ 
   $\exists$ r0: Robot with r0  $\neq$  r. xr = x[r0]  $\wedge$  yr = y[r0];
```

The robot may move within the grid to any unoccupied position.

# Verifying the Safety of the System

Using  $R=3$ .

Using  $P=5$ .

Executing system Robots.

13800 system states found with search depth 13062.

Execution completed (576 ms).

Checking the safety of all reachable states of the systems.

# Safety Conditions

Rather than exploring the state space, we may verify some safety conditions.

```
theorem _Robots_6_initPre_cverify_0(x:Positions, y:Positions)
  ⇔ ∀x0:Map[ℤ[0,2],ℤ[0,4]], y0:Map[ℤ[0,2],ℤ[0,4]].
    (initialState(x0,y0) ⇒ (let x = x0 in (let y = y0 in noCollision(x,y))));
```

```
theorem _Robots_6_actionPre_0_cverify_0(x:Positions, y:Positions)
  requires noCollision(x, y);
  ⇔ ∀r:ℤ[0,2], d:ℤ[0,4]. (nextDir(x,y,r,d) ⇒
    (let x = moveX(x,r,d) in (let y = moveY(y,r,d) in noCollision(x,y))));
```

A generalization of mathematical induction: if every initial state of a system is safe and safety is preserved by every system transition, then every reachable state of the system is safe, i.e., safety is a *system invariant*.

# Verifying the Safety Conditions

- Check validity for say  $N = 5$  and  $P = 3$ :

The SMT solver Yices started execution.

Theorem `_Robots_6_initPre_cverify_0` is valid.

Theorem `_Robots_6_actionPre_0_cverify_0` is valid.

Total time: 15 ms, translation: 3 ms, decision: 10 ms.

- Prove validity for arbitrary  $N \in \mathbb{N}$  and  $P \in \mathbb{N}$ :

Proving theorem `_Robots_6_initPre_cverify_0...`

SUCCESS: theorem was proved (22 ms, see 'Print Prover Output').

Proving theorem `_Robots_6_actionPre_0_cverify_0...`

SUCCESS: theorem was proved (26 ms, see 'Print Prover Output').

Via safety conditions, also *infinite* state systems can be verified.

# A Purely Logical Model of the System

```
shared system RobotsLogical
{
  var x: Positions; var y: Positions;
  invariant noCollision(x, y);
  init()  $\Leftrightarrow$  initialState(x0, y0);
  action move()  $\Leftrightarrow$  nextState(x, y, x0, y0);
}

// the system's transition relation
pred nextState(x:Positions, y:Positions, x0:Positions, y0:Positions)  $\Leftrightarrow$ 
   $\exists r$ :Robot, d:Direction with nextDir(x, y, r, d).
  x0 = moveX(x, r, d)  $\wedge$  y0 = moveY(y, r, d);
```

Actually, a system can be defined in a *purely logical* way by its initial state condition and transition relation.

# Course Contents

Formulating logical formulas that characterize computational problems/systems.

- Logical specification of computational problems.

- Pre- and post-conditions.
- Validation of specifications according to various criteria.
- Computation of results by evaluation of logical formulas.

- Logical modeling of computational systems.

- Initial state conditions, transition relations.
- Modeling safety properties respectively goal states.
- Computation of results by state space exploration.

Software: the “mathematical model checker” RISCAL.

# The RISCAL Software

The screenshot displays the RISCAL software interface. The main window is titled "RISC Algorithm Language (RISCAL)". It features a menu bar with "File", "Edit", "SMT", and "Help". The file path is "/usr2/schreine/papers/RISCALBook/src/./models/gcd.bt".

The code editor shows three versions of Euclid's algorithm:

```
91 // Euclid's algorithm as a recursive function
92 fun gcdf(m:Num,n:Num): Num
93   requires hasgcd(m,n); // m ≠ 0 ∨ n ≠ 0
94   ensures isgcd(result,m,n);
95   decreases m+n;
96 = if m = 0 then n
97   else if n = 0 then m
98   else if n ≤ m then gcdf(m-n,n)
99   else gcdf(m,m-n);
100
101 // simplified version
102 fun gcdf0(m:Num,n:Num): Num
103   requires hasgcd(m,n); // m ≠ 0 ∨ n ≠ 0
104   ensures isgcd(result,m,n);
105   decreases m+n,n;
106 = if n = 0 then m else gcdf0(m,n-m);
107
108 // Euclid's algorithm as an iterative procedure
109 proc gcdp(m:Num,n:Num): Num
110   requires hasgcd(m,n); // m ≠ 0 ∨ n ≠ 0
111   ensures isgcd(result,m,n);
112 {
113   var a:Num = m;
114   var b:Num = n;
115   while a ≠ 0 ∧ b ≠ 0 do
116     invariant hasgcd(a,b); // a ≠ 0 ∨ b ≠ 0;
117     invariant vr:Num. isgcd(r,a,b) = isgcd(r,m,n);
118     decreases a+b;
119   {
120     if b ≤ a then
121       a = a-b;
122     else
123       b = b-a;
124   }
125   return if a = 0 then b else a;
126 }
127
128 // simplified version
129 proc gcdp0(m:Num,n:Num): Num
```

The "Analysis" panel on the right contains the following settings:

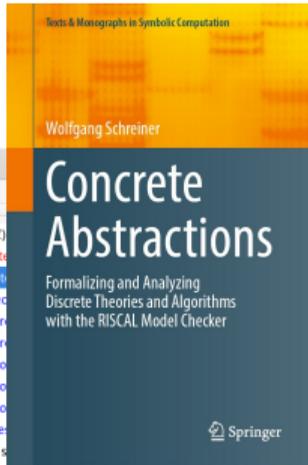
- Translation:  Nondeterminism Default Value: 0 Other Values: [...]
- Execution:  Silent Inputs: Per Mille: Branches: Depth:
- Visualization:  Trace  Tree Width: 400 Height: 400
- Parallelism:  Multi-Threaded Threads: 4  Distributed Servers: [...]
- Operation: gcdp(Z,Z)

The bottom panel shows the execution results for "gcdp(Z,Z)":

```
RISC ALGORITHM LANGUAGE 3.1 APRIL 20, 2021
http://www.risc.jku.at/research/formal/software/RISCAL
(C) 2016-, Research Institute for Symbolic Computation (RISC)
This is free software distributed under the terms of the GNU GPL.
Execute "RISCAL -h" to see the available command line options.
-----
Reading file /usr2/schreine/papers/RISCALBook/src/./models/gcd.txt
Using N=10.
Computing the truth value of notZero...
Computing the value of zero...
Computing the truth value of twoNums...
Computing the truth value of zeroMin...
Type checking and translation completed.
Executing gcdp 27 Spec(Z,Z) with all 121 inputs.
Execution completed for ALL inputs (148 ms, 120 checked, 1 inadmissible).
Executing gcdp 27 PreSat().
Execution completed (1 ms).
Executing gcdp 27 PreNotTrivial().
Execution completed (0 ms).
Executing gcdp 27 PostSat(Z,Z) with all 121 inputs.
Execution completed for ALL inputs (112 ms, 120 checked, 1 inadmissible).
Executing gcdp 27 PostMotTrivialAll(Z,Z) with all 121 inputs.
Execution completed for ALL inputs (68 ms, 120 checked, 1 inadmissible).
Executing gcdp 27 PostMotTrivialSome().
Execution completed (0 ms).
Executing gcdp 27 PostUnique(Z,Z) with all 121 inputs.
Execution completed for ALL inputs (355 ms, 120 checked, 1 inadmissible).
```

The "Tasks" panel on the right shows a list of tasks for "gcdp(Z,Z)":

- Verify correctness of result
  - Is result correct?
- Verify iteration and recursion
  - Does loop invariant initially hold?
  - Does loop invariant initially hold?
  - Is loop measure non-negative?
  - Is loop invariant preserved?
  - Is loop measure decreased?
  - Is loop measure decreased?
- Verify implementation preconditions
  - Does operation precondition hold?
  - Does operation precondition hold?



## 1. Introduction

## 2. Symbolic Summation and the Modeling of Sequences

## 3. Logical Models of Problems and Computations

## 4. Modeling Problems in Geometry and Discrete Mathematics

- Problem 1: How to get from  $A$  to  $B$ ?
- Problem 2: How to efficiently use resources?

## 5. Organization

# MODELING PROBLEMS IN GEOMETRY AND DISCRETE MATHEMATICS



**PROBLEM 1: HOW TO GET FROM  $A$  TO  $B$ ?**

## Navigation Systems (trivial approach)

What is the mathematics behind navigation systems in modern cars?

## Navigation Systems (trivial approach)

What is the mathematics behind navigation systems in modern cars?

**Given:** start address  $A$ , destination address  $B$ .

**Find:** “shortest route” from  $A$  to  $B$ .

## Navigation Systems (trivial approach)

What is the mathematics behind navigation systems in modern cars?

**Given:** start address  $A$ , destination address  $B$ .

**Find:** “shortest route” from  $A$  to  $B$ .

Real world:  $A$  and  $B$  are given by geographical coordinates (2D or even 3D).

## Navigation Systems (trivial approach)

What is the mathematics behind navigation systems in modern cars?

**Given:** start address  $A$ , destination address  $B$ .

**Find:** “shortest route” from  $A$  to  $B$ .

Real world:  $A$  and  $B$  are given by geographical coordinates (2D or even 3D).

Solution: if no further restrictions are given, the solution is trivial: the shortest connection from  $A$  to  $B$  is the straight line from  $A$  to  $B$ , the “shortest route” is given by  $(A, B)$  with length

$$d_{\min} = \|B - A\|$$

with some appropriate norm  $\|\cdot\|$ .

# Navigation Systems (realistic approach)

**Given:** start address  $A$ , destination address  $B$ , “network” of streets  $S$ .

**Find:** “shortest route” from  $A$  to  $B$  on  $S$ .



## Navigation Systems (realistic approach)

**Given:** start address  $A$ , destination address  $B$ , “network” of streets  $S$ .

**Find:** “shortest route” from  $A$  to  $B$  on  $S$ .

Mathematical model: Given  $n$  Streets  $S_i$  with  $i = 1, \dots, n$ . Streets  $S_i$  and  $S_j$  intersect at crossing  $C_{ij}$ . Two crossings  $c$  and  $d$  are adjacent iff  $c \neq d$  and there is no crossing between them on the same street. Two adjacent crossings are connected by a street segment.

Network of streets is characterized by

- crossings  $V = \{C_{ij} \mid i, j = 1, \dots, n\}$ ,
- adjacency relation between crossings  $E = \{\{v_1, v_2\} \mid v_1 \text{ and } v_2 \text{ are adjacent}\}$ ,
- length of street segments  $w: E \rightarrow \mathbb{R}^+$ .

# Undirected Weighted Graphs

The triple  $G = (V, E, w)$  is called an undirected weighted graph iff

- $V$  is some non-empty finite set,
- $E \subset P(V)$  with  $|e| = 2$  for all  $e \in E$ , and
- $w: E \rightarrow \mathbb{R}$ .

# Undirected Weighted Graphs

The triple  $G = (V, E, w)$  is called an undirected weighted graph iff

- $V$  is some non-empty finite set,
- $E \subset P(V)$  with  $|e| = 2$  for all  $e \in E$ , and
- $w: E \rightarrow \mathbb{R}$ .

Our problem now becomes

**Given:** an undirected weighted graph  $G = (V, E, w)$ ,  $A, B \in V$ .

**Find:** a sequence  $P$  of some length  $n$  in  $V$  such that

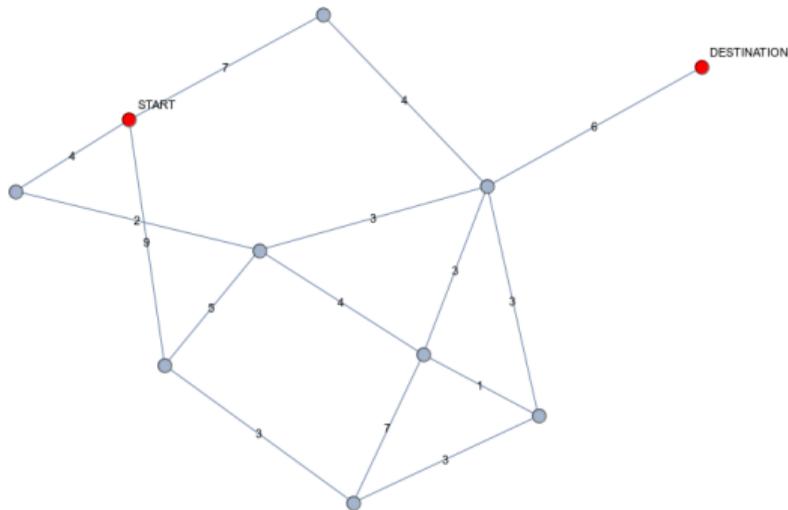
$$P_1 = A, P_n = B$$

$$\forall 1 \leq i \leq n - 1 : \{P_i, P_{i+1}\} \in E$$

$$\sum_{i=1}^n w(\{P_i, P_{i+1}\}) = \min\{w(Q) \mid Q \text{ is a path from } A \text{ to } B \text{ in } G\}$$

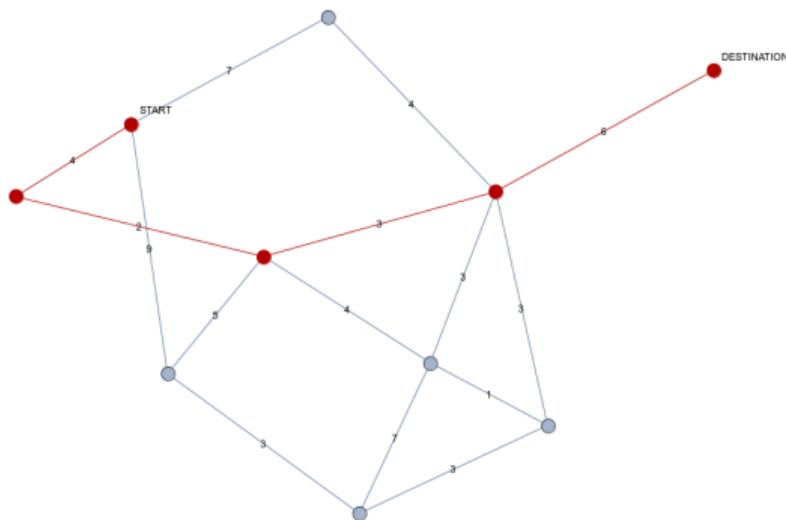
# Solution

The above problem is a well-known and well-studied problem in graph theory called the Shortest Path Problem.



# Solution

The above problem is a well-known and well-studied problem in graph theory called the Shortest Path Problem.



There are several algorithms to solve the Shortest Path Problem, e.g. Dijkstra's Algorithm or the Bellman-Ford-Algorithm.

# MODELING PROBLEMS IN GEOMETRY AND DISCRETE MATHEMATICS



**PROBLEM 2: HOW TO EFFICIENTLY USE RESOURCES?**

# A Planning Problem

## Real Life Situation

A factory has 10 production stations with equal capabilities. Each machine can be operated for at most 9 hours per day, production may start at 8:30. Every station needs two workers for operation, if a station stays closed the two employees can be used for other useful tasks. There are 160 orders with different production duration that have to be processed on a certain day. Each order can be processed on any of the stations. The delivery of the final products is scheduled on the night train leaving the factory no earlier than 18:00. Time for packing the products on the train is less than half an hour.

Design a “good” production schedule for that day.

## Problem Analysis

- Every station can be utilized for the whole 9 hours from 8:30–17:30.

## Problem Analysis

- Every station can be utilized for the whole 9 hours from 8:30–17:30.
- Production order does not play a role.

## Problem Analysis

- Every station can be utilized for the whole 9 hours from 8:30–17:30.
- Production order does not play a role.
- Every order has to be processed.

## Problem Analysis

- Every station can be utilized for the whole 9 hours from 8:30–17:30.
- Production order does not play a role.
- Every order has to be processed.
- There is no need to finish production as early as possible, finishing by 17:30 is all that is required so that the train is readily packed by 18:00.

## Problem Analysis

- Every station can be utilized for the whole 9 hours from 8:30–17:30.
- Production order does not play a role.
- Every order has to be processed.
- There is no need to finish production as early as possible, finishing by 17:30 is all that is required so that the train is readily packed by 18:00.
- Fast production is not the criterion for a “good” production schedule, but the number of open production stations.

## Mathematical Model

**Given:** orders  $O = \{1, \dots, n\}$ , duration  $d: O \rightarrow \mathbb{R}$ , stations  $S = \{1, \dots, m\}$ , maximal operation time on stations  $D: S \rightarrow \mathbb{R}$ .

**Find:** number of open stations  $k$  and assignment of orders to stations  $s: O \rightarrow \{1, \dots, k\}$  such that

$$k \leq m, \quad (1)$$

$$\forall j \in S: \sum_{\substack{i \in O \\ s(i)=j}} d(i) \leq D(j), \quad (2)$$

$$\forall l < k \nexists t: O \rightarrow \{1, \dots, l\} \forall j \in S: \sum_{\substack{i \in O \\ t(i)=j}} d(i) \leq D(j) \quad (3)$$

(2) means assignment obeys limit on every station.

(3) means that no assignment with less stations is possible.

# Solution

The above problem is a well-known and well-studied problem in combinatorial optimization called the Bin Packing Problem.

## Solution

The above problem is a well-known and well-studied problem in combinatorial optimization called the Bin Packing Problem.

There are several algorithms to solve the Bin Packing Problem, e.g. Branch-and-Bound or various Heuristic Approximation Methods, because finding the minimal  $k$  can be very time consuming.

## 1. Introduction

## 2. Symbolic Summation and the Modeling of Sequences

## 3. Logical Models of Problems and Computations

## 4. Modeling Problems in Geometry and Discrete Mathematics

- Problem 1: How to get from  $A$  to  $B$ ?
- Problem 2: How to efficiently use resources?

## 5. Organization

# Organization

- This course (VO)
  - Grading based on three home assignments ( $3 \times 100$  grade points).
  - Each assignment deals with the elaboration of a small model.
  - Minimum requirement to pass the course:  $3 \times 50$  grade points.
  - Extra assignments: if the minimum requirements are not met.
- Accompanying proseminar (PS)
  - Deals with the kind of models treated in this course.
  - Additionally discusses the basics of “mathematical practice”.
  - Each participant selects an individual problem to be modeled/analyzed.
  - Requirement is to write a small paper and prepare/give a small presentation.
  - Some topics are also suitable for a **bachelor thesis**.

This course and the proseminar are not formally linked: they can be independently pursued and are independently graded.

# Moodle Course

Central point of electronic interaction.

- Forum “Discussions”: your questions and answers.
  - Anyone can post a question or an answer.
- Forum “Announcements”: our messages.
  - Only we (the lecturers) can post here.
- Various “Assignments”: your submissions.
  - Email submissions are not accepted.
- Personal messages/emails: only for confidential matters.
  - Everything else all lecturers and students should see.

See the link in the KUSSS page of this course.