Holonomy and combinatorics

From automatic identities to systematic enumeration

Marni Mishna

LaBRI, Université Bordeaux I

http://www.labri.fr/~mishna
A grab bag of problems

Prove Dixon's identity: \[ \sum_{k} (-1)^k \frac{(n+k)(n+b)(a+b)}{(b+k)(a+k)} = \frac{(n+a+b)!}{n!a!b!} \]

Calculate the dimension of one representation of the symmetric group in another

Find a formula for standard Young tableaux of size \( n \)

Find a recursive formula for the number of 4-regular graphs of size \( n \)

Enumerate plane partitions of shape \( 2 \times n \)

\( \zeta(2, 2) = 1/2 \left( \zeta(2)^2 - \zeta(4) \right) \)

Prove \( q \)-Hypergeometric identities

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Basic Idea

Set of easy identities
Basic Idea

Set of easy identities → Elimination in operator algebra
Basic Idea

Set of easy identities $\rightarrow$ Elimination in operator algebra $\rightarrow$ Profound identity

- A grab bag of problems
- Summary of how it works

Creative telescoping

A combinatorial framework

The scalar product

Conclusion
Basic Idea

Set of easy identities

Elimination in operator algebra

Profound identity

A set of recurrences for $p(n, k)$

A recurrence for $\phi(p(n, k))$
Basic Idea

Set of easy identities \( p(n, k) \) \( \rightarrow \) Elimination in operator algebra \( \rightarrow \) Profound identity

A set of recurrences for \( p(n, k) \)

Systems of differential equations satisfied by \( f \) and \( g \)

A recurrence for \( \phi(p(n, k)) \)

A differential equation for \( \psi(f, g) \)

The recurrences and DEs are useful! They give sequences, can be solved, allow asymptotic analysis, ...
<table>
<thead>
<tr>
<th>Dixon’s Identity</th>
<th>4-Regular Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ \sum_k (-1)^k \frac{(n+a)(n+b)(a+b)}{(n+a+b)!} ]</td>
<td></td>
</tr>
</tbody>
</table>

### Summary of how it works

<table>
<thead>
<tr>
<th>Operator Algebra</th>
<th>Shift</th>
<th>Differential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closure Operation</td>
<td>+, *</td>
<td>⟨, ⟩</td>
</tr>
</tbody>
</table>

- **“Easy” Identity**: recurrence for \( n! \)
- **Elimination**: Answer is in a left ideal. Find a Gröbner basis for this ideal
- **Output**: Recurrences for hypergeometric sums
- **Output**: Diff eq. satisfied by generating function
Creative telescoping

- P-Recursive Functions
- Express recurrences as operators
- Elimination
- Zeilberger's Creative Telescoping
- How does this work in general?

A combinatorial framework

The scalar product

Conclusion
P-Recursive Functions

Def: A function $f : \mathbb{N} \rightarrow R$ is P-recursive if it satisfies a finite linear recurrence with polynomial coefficients. There exists polynomials $\gamma_i$ such that:

$$\gamma_d(n)f(n+d) + \ldots + \gamma_1(n)f(n+1) + \gamma_0(n)f(n) = 0$$

Ex: $f(n) = n!$ is P-recursive: $1(n+1)! - (n+1)n! = 0$
Ex: Rational: $p(x)/q(x) = \sum_n f(n)x^n \implies \gamma_i = \text{const.}$
Express recurrences as operators

Shift operator:

\[ S_n P(n, k) = P(n + 1, k), \quad S_k P(n, k) = P(n, k + 1) \]

Define a (non-commutative) algebra:

\[ B_{n,k} := \mathbb{C}\langle n, k, S_n, S_k \rangle \]

\[ S_n n = (n + 1) S_n, \quad S_k k = (k + 1) S_k, \quad \text{all other variables commute} \]
Express recurrences as operators

Shift operator:

\[ S_n P(n, k) = P(n + 1, k), \quad S_k P(n, k) = P(n, k + 1) \]

Define a (non-commutative) algebra:

\[ B_{n,k} := \mathbb{C}\langle n, k, S_n, S_k \rangle \]

\[ S_n n = (n + 1)S_n, \quad S_k k = (k + 1)S_k, \quad \text{all other variables commute} \]

\[ 0 = P(n, k) + P(n, k + 1) + (3n^2 + k)P(n + 2, k + 1) \]

\[ = P(n, k) + S_k P(n, k) + (3n^2 + k)S_n^2 S_k P(n, k) \]

\[ = (1 + S_k + (3n^2 + k)S_n^2 S_k) \cdot P(n, k) \]

\[ = \psi(n, k, S_n, S_k) \cdot P \]
Express recurrences as operators

Shift operator:

\[ S_n P(n, k) = P(n + 1, k), \quad S_k P(n, k) = P(n, k + 1) \]

Define a (non-commutative) algebra:

\[ B_{n,k} := \mathbb{C}\langle n, k, S_n, S_k \rangle \]
\[ S_n n = (n + 1)S_n, \quad S_k k = (k + 1)S_k, \text{ all other variables commute} \]

\[ 0 = P(n, k) + P(n, k + 1) + (3n^2 + k)P(n + 2, k + 1) \]
\[ = P(n, k) + S_k P(n, k) + (3n^2 + k)S_n^2 S_k P(n, k) \]
\[ = (1 + S_k + (3n^2 + k)S_n^2 S_k) \cdot P(n, k) \]
\[ = \psi(n, k, S_n, S_k) \cdot P \]

- If \( \psi, \xi \in I_P \), then \( A(n, k, S_n, S_k)\psi + B(n, k, S_n, S_k)\xi \in I_P \).
- \( \psi \) is in the annihilating (left) ideal \( I_P \) of \( P \).
Making new identities using elimination

Recall from your youth:

\[ n \] equations, \( m \) unknowns variables + linear algebra \( \implies \)

eliminate some unknown variables.

\[
\begin{align*}
3x + 4y + z &= 2 \quad (1) \\
5x - 2y + 2z &= 3 \quad (2)
\end{align*}
\]

\[
5 \times (1) - 3 \times (2) \implies \\
26y - z &= 1
\]

We can do this in the setting of non-commutative algebras. (Ore algebra)

- Left ideals will have Gröbner basis (Galligo)
- Modified traditional tools work (Buchbergers + Euclidean algorithm)
Zeilberger’s Creative Telescoping

- Easy identities:
  \[
  \frac{F(n+1,k)}{F(n,k)} = \frac{n+1}{n-k+1}, \quad \frac{F(n,k+1)}{F(n,k)} = \frac{n-k}{k+1}
  \]
Zeilberger’s Creative Telescoping

The Set Up

Creative telescoping
- P-Recursive Functions
- Express recurrences as operators
- Elimination
- Zeilberger’s Creative Telescoping
- How does this work in general?

A combinatorial framework

The scalar product

Conclusion

\[ F(n, k) = \binom{n}{k} \]

- Easy identities:
  \[
  \frac{F(n+1, k)}{F(n, k)} = \frac{n+1}{n-k+1} \quad \frac{F(n, k+1)}{F(n, k)} = \frac{n-k}{k+1}
  \]

- Cross Multiply:
  \[
  ((n - k + 1)S_n - (n + 1)) F = 0 \quad \heartsuit
  
  ((k + 1)S_k - (n - k)) F = 0 \quad \clubsuit
  \]
Zeilberger’s Creative Telescoping

- **Easy identities:**
  \[
  \frac{F(n+1,k)}{F(n,k)} = \frac{n+1}{n-k+1} \\
  \frac{F(n,k+1)}{F(n,k)} = \frac{n-k}{k+1}
  \]

- **Cross Multiply:**
  \[
  \begin{align*}
  ((n - k + 1)S_n - (n + 1)) F &= 0 \heartsuit \\
  ((k + 1)S_k - (n - k)) F &= 0 \diamondsuit
  \end{align*}
  \]

- **A linear combination:**
  \[
  (S_k + 1) \heartsuit + S_n \diamondsuit = 0 \\
  (n + 1) (S_n S_k - S_k - 1) F = 0
  \]
Zeilberger’s Creative Telescoping

\[ F(n, k) = \binom{n}{k} \]

- **Easy identities:**
  \[ \frac{F(n+1,k)}{F(n,k)} = \frac{n+1}{n-k+1} \]
  \[ \frac{F(n,k+1)}{F(n,k)} = \frac{n-k}{k+1} \]

- **Cross Multiply:**
  \[ ((n - k + 1)S_n - (n + 1))F = 0 \]
  \[ (((k + 1)S_k - (n - k))F = 0 \]

- **A linear combination:**
  \[ (S_k + 1)\heartsuit + S_n\clubsuit = 0 \]
  \[ (n + 1)(S_nS_k - S_k - 1)F = 0 \]

- **Reorganize this to have only** \( S_n \) **and** \( n \) **on the left, and a factor of** \( (S_k - 1) \) **on the right:**
  \[ (n + 1)(S_n - 2)F = (S_k - 1)G(n, k), \text{ for some } G(n, k) \]
Zeilberger’s Creative Telescoping

The Set Up
Creative telescoping
- P-Recursive Functions
- Express recurrences as operators
- Elimination
- Zeilberger’s Creative Telescoping
- How does this work in general?

A combinatorial framework
The scalar product
Conclusion

\[ F(n, k) = \binom{n}{k} \]

\[(n + 1)(S_n - 2)F = (S_k - 1)G(n, k) \quad (\ast)\]

Take the sum:

\[ a(n) = \sum_{k \geq 0} F(n, k) \]

\[ a(n + 1) - 2a(n) = 0 \]

\[ a(n) = 2n \]
Zeilberger’s Creative Telescoping

\[ F(n, k) = \binom{n}{k} \]

\[(n + 1)(S_n - 2)F = (S_k - 1)G(n, k) \quad (\star)\]

- Apply the linear combination to this sum:

\[(n + 1)(S_n - 2)a(n) = \sum_{k \geq 0} (n + 1)(S_n - 2)F(n, k)\]
Zeilberger’s Creative Telescoping

\[ F(n, k) = \binom{n}{k} \]

\[ (n + 1)(S_n - 2)F = (S_k - 1)G(n, k) \quad (\ast) \]

Apply (\ast):

\[ (n + 1)(S_n - 2)a(n) = \sum_{k \geq 0} (n + 1)(S_n - 2)F(n, k) \]

\[ = \sum_{k \geq 0} (S_k - 1)(G(n, k)) \]
Zeilberger’s Creative Telescoping

\[ F(n, k) = \binom{n}{k} \]

\[(n + 1)(S_n - 2)F = (S_k - 1)G(n, k) \quad (\ast)\]

**Expand:**

\[
(n + 1)(S_n - 2)a(n) = \sum_{k \geq 0} (n + 1)(S_n - 2)F(n, k)
\]

\[
= \sum_{k \geq 0} (S_k - 1)(G(n, k))
\]

\[
= \sum_{k \geq 0} (G(n, k + 1) - G(n, k)) = 0
\]
Zeilberger’s Creative Telescoping

\[ F(n, k) = \binom{n}{k} \]

\[(n + 1)(S_n - 2)F = (S_k - 1)G(n, k) \quad (\ast)\]

**Expand:**

\[
(n + 1)(S_n - 2)a(n) = \sum_{k \geq 0} (n + 1)(S_n - 2)F(n, k)
\]

\[
= \sum_{k \geq 0} (S_k - 1)(G(n, k))
\]

\[
= \sum_{k \geq 0} (G(n, k + 1) - G(n, k)) = 0
\]

**Gives result about the sum:**

\[ a(n + 1) - 2a(n) = 0 \implies a(n) = 2^n \implies \sum_k \binom{n}{k} = 2^n \]
How does this work in general?

- Begin with a set of “independent” relations
  
  \[ P(n, k, S_n, S_k) F(n, k) = 0 \]
  
  \[ Q(n, k, S_n, S_k) F(n, k) = 0 \]

- Find the good linear combination which eliminates \( k \) and has a factor of \( S_k - 1 \)

  \[
  R(n, k, S_n, S_k) = A(n, k, S_n, S_k) P + B(n, k, S_n, S_k) Q
  \]
  
  \[
  R(n, k, S_n, S_k) = S(S_n, n) + (S_k - 1) R'(n, S_n, S_k)
  \]

- Take the infinite sum over \( k \). We have a telescoping sum:

  \[
  0 = \sum_k R(n, k, S_n, S_k) F(n, k)
  \]
  
  \[
  = \sum_k S(S_n, n) F + \sum_k (S_k - 1) R'(N, K, n) F
  \]

  \[
  = \underbrace{G(n, k)}_{\text{...And we get a recurrence for the sum}}
  \]
A combinatorial framework
Combinatorial generating functions

Counting information about combinatorial objects can be encoded in *formal* power series:

Ex. exponential generating function (egf):

\[ C(z) = \sum_{n} \text{(number of objects in } C \text{ of size } n) \frac{z^n}{n!} \]

Ex. \( C = \) \[ \text{(...)} \]

\[ \implies C(z) = z + \frac{z^2}{2} + 2\frac{z^3}{3!} + 3\frac{x^4}{4!} + \ldots \]
Combinatorial generating functions

Counting information about combinatorial objects can be encoded in formal power series:

**Ex. exponential generating function (egf):**

\[
C(z) = \sum_n \left( \text{number of objects in } C \text{ of size } n \right) \frac{z^n}{n!}
\]

**Ex.** 

\[C = \ldots\]  
\[\implies C(z) = z + \frac{z^2}{2} + \frac{2z^3}{3!} + \frac{3z^4}{4!} + \ldots\]

**Nice:** Algebraic operations correspond with combinatorial operations.

- \[C = A \cup B \implies C(z) = A(z) + B(z)\]
- \[C = \{(a, b), a \in A, b \in B\} \implies C(z) = A(z)B(z)\]
D-finite functions

Def. A function $f(x_1, x_2, \ldots, x_n)$ is \textbf{Differentially finite} (D-finite) with respect to $X = x_1, \ldots, x_n$ if

- For $1 \leq j \leq n$, $f$ satisfies $n$ linear differential equations with polynomial coefficients:

$$
\phi_0(X)f(X) + \phi_1(X) \frac{\partial f(X)}{\partial x_j} + \ldots + \phi_k(X) \frac{\partial^k f(X)}{\partial x_j^k} = 0
$$
D-finite functions

Def. A function \( f(x_1, x_2, \ldots, x_n) \) is **Differentially finite** (D-finite) with respect to \( X = x_1, \ldots, x_n \) if

- For \( 1 \leq j \leq n \), \( f \) satisfies \( n \) linear differential equations with polynomial coefficients:

\[
\phi_0(X)f(X) + \phi_1(X)\frac{\partial f(X)}{\partial x_j} + \ldots + \phi_k(X)\frac{\partial^k f(X)}{\partial x_j^k} = 0
\]

Examples

- \( f = \sin(2x^3 + 3yz) \) is D-finite with respect to \( \{x, y, z\} \).
D-finite functions

Def. A function $f(x_1, x_2, \ldots, x_n)$ is **Differentially finite** (D-finite) with respect to $X = x_1, \ldots, x_n$ if

- For $1 \leq j \leq n$, $f$ satisfies $n$ linear differential equations with polynomial coefficients:

$$\phi_0(X)f(X) + \phi_1(X)\frac{\partial f(X)}{\partial x_j} + \ldots + \phi_k(X)\frac{\partial^k f(X)}{\partial x_j^k} = 0$$

**Examples**

- $f = \sin(2x^3 + 3yz)$ is D-finite with respect to $\{x, y, z\}$.
  
  $xf_{xx} - 2f_x + 9x^5f = 0$, 

The scalar product

Conclusion
D-finite functions

Def. A function $f(x_1, x_2, \ldots, x_n)$ is Differentiably finite (D-finite) with respect to $X = x_1, \ldots, x_n$ if

- For $1 \leq j \leq n$, $f$ satisfies $n$ linear differential equations with polynomial coefficients:

$$\phi_0(X)f(X) + \phi_1(X)\frac{\partial f(X)}{\partial x_j} + \ldots + \phi_k(X)\frac{\partial^k f(X)}{\partial x_j^k} = 0$$

Examples

- $f = \sin(2x^3 + 3yz)$ is D-finite with respect to $\{x, y, z\}$.
  $$xf_{xx} - 2f_x + 9x^5f = 0, \quad f_{yy} - 9z^2f = 0,$$
D-finite functions

Def. A function $f(x_1, x_2, \ldots, x_n)$ is Differentially finite (D-finite) with respect to $X = x_1, \ldots, x_n$ if

- For $1 \leq j \leq n$, $f$ satisfies $n$ linear differential equations with polynomial coefficients:

$$\phi_0(X)f(X) + \phi_1(X)\frac{\partial f(X)}{\partial x_j} + \ldots + \phi_k(X)\frac{\partial^k f(X)}{\partial x_j^k} = 0$$

Examples

- $f = \sin(2x^3 + 3yz)$ is D-finite with respect to $\{x, y, z\}$.
  \[xf_{xx} - 2f_x + 9x^5f = 0, \quad ff_{yy} - 9z^2f = 0, \quad f_{zz} - 9y^2f = 0\]
D-finite functions

Def. A function \( f(x_1, x_2, \ldots, x_n) \) is **Differentially finite** (D-finite) with respect to \( X = x_1, \ldots, x_n \) if

- For \( 1 \leq j \leq n \), \( f \) satisfies \( n \) linear differential equations with polynomial coefficients:

\[
\phi_0(X)f(X) + \phi_1(X)\frac{\partial f(X)}{\partial x_j} + \ldots + \phi_k(X)\frac{\partial^k f(X)}{\partial x_j^k} = 0
\]

**Examples**

- \( f = \sin(2x^3 + 3yz) \) is D-finite with respect to \( \{x, y, z\} \).

\[
x f_{xx} - 2f_x + 9x^5 f = 0, \quad f_{yy} - 9z^2 f = 0, \quad f_{zz} - 9y^2 f = 0
\]

- \( \exp(\text{polynomial}) \) is D-finite
**D-finite functions**

Def. A function $f(x_1, x_2, \ldots, x_n)$ is **Differentiably finite** (D-finite) with respect to $X = x_1, \ldots, x_n$ if

- For $1 \leq j \leq n$, $f$ satisfies $n$ linear differential equations with polynomial coefficients:

$$
\phi_0(X)f(X) + \phi_1(X)\frac{\partial f(X)}{\partial x_j} + \ldots + \phi_k(X)\frac{\partial^k f(X)}{\partial x_j^k} = 0
$$

**Examples**

- $f = \sin(2x^3 + 3yz)$ is D-finite with respect to $\{x, y, z\}$.
  $$xf_{xx} - 2f_x + 9x^5f = 0, \quad f_{yy} - 9z^2f = 0, \quad f_{zz} - 9y^2f = 0$$
- $\exp(\text{polynomial})$ is D-finite
- $\sum f(n)z^n$ D-finite $\iff f(n)$ P-recursive
### Hierarchy of Power Series

#### The Set Up
- Creative telescoping
- A combinatorial framework
- Generating functions
- D-finite functions
- D-finiteness

#### Creative Telescoping

<table>
<thead>
<tr>
<th>D-finite</th>
<th>$k$-regular graphs</th>
<th>partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algebraic</td>
<td>$k$-regular graphs</td>
<td>partitions</td>
</tr>
<tr>
<td>Rational</td>
<td>$k$-regular graphs</td>
<td>partitions</td>
</tr>
<tr>
<td>Restricted walks</td>
<td>$k$-regular graphs</td>
<td>partitions</td>
</tr>
<tr>
<td>in the plane</td>
<td>$k$-regular graphs</td>
<td>partitions</td>
</tr>
<tr>
<td>Regular expressions</td>
<td>$k$-regular graphs</td>
<td>partitions</td>
</tr>
<tr>
<td>Motzkin Paths</td>
<td>$k$-regular graphs</td>
<td>partitions</td>
</tr>
</tbody>
</table>

#### Conclusion

- Marni Mishna, Freaky Friday the 13th, 2004 Carleton University

---

**Differentiably Algebraic**

1. **Algebraic**
   - **D-finite**: Generating functions, D-finite functions, D-finiteness
   - **Algebraic**
   - **Rational**: Restricted walks in the plane
   - **Dyck Paths**: $\frac{1 - \sqrt{1 - 4x}}{2x}$
   - **Motzkin Paths**
   - $k \times n$-latin squares
   - Grid walks in the first quadrant

2. **Partitions**
   - $e^x$
   - $e^e - 1$
Hierarchy of Power Series

Differentiably Algebraic

D-finite

Algebraic

Rational

unrestricted walks in the plane

Dyck Paths

\[ \frac{1 - \sqrt{1 - 4x}}{2x} \]

\[ e^x \]

grid walks in the first quadrant

\[ e^{e^x - 1} \]

k-regular graphs

partitions

Algebraic closure

Functional Inverse

Integrals

Hadamard product

Diagonals

Multiplicative inverse

Algebraic substitution

Hadamard product

Marni Mishna, Freaky Friday the 13th, 2004 Carleton University
Hierarchy of Power Series

Differentiably Algebraic

- D-finite
  - Algebraic
    - Rational
      - unrestricted walks
        - in the plane
      - Regular expressions
      - Motzkin Paths
    - $k \times n$-latin squares
  - Dyck Paths
    - $1 - \sqrt{1 - 4x} / 2x$
  - Grid walks in the first quadrant
    - $e^x$
    - $e^{e^x - 1}$

- $k$-regular graphs

- Algebraic closure
- Functional Inverse
- Integrals
- Hadamard product
- Diagonals
- Multiplicative inverse
- Algebraic substitution
- Hadamard product
The scalar product
Symmetric Series

- **power** $p_k$
  
  $p_3 = x_1^3 + x_2^3 + x_3^3 + \ldots$

- **homogeneous** $h_k$
  
  $h_3 = x_1^3 + x_2^3 + \ldots + x_1^2 x_2 + \ldots + x_1 x_2 x_3 + \ldots$

- **elementary** $e_n$
  
  $e_3 = x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_3 x_4 + \ldots$

- **monomial** $m_{\lambda}$
  
  $m_{(3,2,1)} = x_1^3 x_2^2 x_3 + x_1^3 x_2^2 x_3 + x_3^3 x_2^2 x_6 + \ldots$
Symmetric Series

- power $p_k$
  $p_3 = x_1^3 + x_2^3 + x_3^3 + \ldots$

- homogeneous $h_k$
  $h_3 = x_1^3 + x_2^3 + \ldots + x_1^2x_2 + \ldots + x_1x_2x_3 + \ldots$

- elementary $e_n$
  $e_3 = x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + \ldots$

- monomial $m_\lambda$
  $m_{(3,2,1)} = x_1^3x_2^2x_3 + x_2^3x_1^2x_3 + x_3^3x_2^2x_6 + \ldots$

- $p_\lambda = p_{\lambda_1}p_{\lambda_2} \cdots p_{\lambda_k}$ (e.g. $p_{(3,2,2,1)} = p_3p_2^2p_1$)
Symmetric Series

- **power** $p_k$
  
  $p_3 = x_1^3 + x_2^3 + x_3^3 + \ldots$

- **homogeneous** $h_k$
  
  $h_3 = x_1^3 + x_2^3 + \ldots + x_1^2 x_2 + \ldots + x_1 x_2 x_3 + \ldots$

- **elementary** $e_n$
  
  $e_3 = x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_3 x_4 + \ldots$

- **monomial** $m_\lambda$
  
  $m_{(3,2,1)} = x_1^3 x_2^2 x_3 + x_2^3 x_1^2 x_3 + x_3^3 x_2^2 x_6 + \ldots$

- $p_\lambda = p_{\lambda_1} p_{\lambda_2} \cdots p_{\lambda_k}$ (e.g. $p_{(3,2,2,1)} = p_3 p_2^2 p_1$)

- **Our algebra of interest**: $K[[p_1, p_2, \ldots]]$
Symmetric Series

- **power** $p_k$
  \[ p_3 = x_1^3 + x_2^3 + x_3^3 + \ldots \]

- **homogeneous** $h_k$
  \[ h_3 = x_1^3 + x_2^3 + \ldots + x_1^2x_2 + \ldots + x_1x_2x_3 + \ldots \]

- **elementary** $e_n$
  \[ e_3 = x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + \ldots \]

- **monomial** $m_\lambda$
  \[ m_{(3,2,1)} = x_1^3x_2^2x_3 + x_2^3x_1^2x_3 + x_3^3x_2^2x_6 + \ldots \]

- $p_\lambda = p_{\lambda_1}p_{\lambda_2} \cdots p_{\lambda_k}$ (e.g. $p_{(3,2,2,1)} = p_3p_2^2p_1$)

- **Our algebra of interest**: $K[[p_1, p_2, \ldots]]$

- **Ex**:  
  \[
  H = \sum_n h_n = \exp\left(\sum_n \frac{p_n}{n}\right) \\
  E = \sum_n e_n = \exp\left(\sum_n (-1)^{n+1} \frac{p_n}{n}\right)
  \]
Symmetric Series

- **power** \( p_k \)
  \[ p_3 = x_1^3 + x_2^3 + x_3^3 + \ldots \]

- **homogeneous** \( h_k \)
  \[ h_3 = x_1^3 + x_2^3 + \ldots + x_1^2x_2 + \ldots + x_1x_2x_3 + \ldots \]

- **elementary** \( e_n \)
  \[ e_3 = x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + \ldots \]

- **monomial** \( m_\lambda \)
  \[ m_{(3,2,1)} = x_1^3x_2^2x_3 + x_2^3x_1^2x_3 + x_3^3x_2^2x_6 + \ldots \]

- \( p_\lambda = p_{\lambda_1}p_{\lambda_2} \cdots p_{\lambda_k} \) (e.g. \( p_{(3,2,2,1)} = p_3p_2^2p_1 \))

- Our algebra of interest: \( K[[p_1, p_2, \ldots ]] \)

- Ex:
  \[
  H = \sum_n h_n = \exp(\sum_n \frac{p_n}{n}) \\
  E = \sum_n e_n = \exp(\sum_n (-1)^{n+1} \frac{p_n}{n})
  \]

- Scalar product: \( \langle p_\lambda, p_\mu \rangle = \delta_{\lambda \mu} z_\lambda \)
  \[
  \lambda = (1^{m_1}2^{m_2} \cdots k^{m_k}) \implies z_\lambda = 1^{m_1}m_1!2^{m_2}m_2! \cdots k^{m_k}m_k!
  \]
D-finite symmetric series

Def. A symmetric series $F \in K[[p_1, p_2, \ldots ; t]]$ is D-finite if for any $n$,

- $F(p_1, ..., p_n, 0, 0, \ldots ; t)$ is D-finite wrt $p_1, \ldots, p_n, t$. 
D-finite symmetric series

Def. A symmetric series $F \in K[[p_1, p_2, \ldots; t]]$ is D-finite if for any $n$,
- $F(p_1, \ldots, p_n, 0, 0, \ldots; t)$ is D-finite wrt $p_1, \ldots, p_n, t$.

Examples:

- $H = \sum h_n t^n = \exp(\sum_n p_n t^n / n)$
- $E = \sum e_n t^n = \exp(\sum_n (-1)^{n+1} p_n t^n / n)$
- $S = \sum \lambda s_{\lambda} t^{\lambda} = \exp(\sum_n \frac{p_n^2 t^{2n}}{2n} + \frac{p_{2n-1} t^{2n-1}}{2n-1})$

- $\sum \lambda$ all parts odd $s_{\lambda} t^{\lambda} = SE^{-1}$
The scalar product, (symmetric, bilinear form) is defined by

\[ \langle p_\lambda, p_\mu \rangle = \delta_{\lambda\mu} z_\lambda. \]

\[ \lambda = (1^{m_1} 2^{m_2} \cdots k^{m_k}) \implies z_\lambda = 1^{m_1} m_1!2^{m_2} m_2! \cdots k^{m_k} m_k! \]
The scalar product, (symmetric, bilinear form) is defined by

\[ \langle p_\lambda, p_\mu \rangle = \delta_{\lambda \mu} z_\lambda. \]

\[ \lambda = (1^m 2^m \cdots k^m) \implies z_\lambda = 1^m m_1! 2^m m_2! \cdots k^m m_k! \]

- Equivalent to coefficient extraction:

\[ [x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}] F = \langle F, h_{k_1} h_{k_2} \cdots h_{k_n} \rangle \]
The Scalar Product

- The scalar product, (symmetric, bilinear form) is defined by

\[ \langle p_{\lambda}, p_{\mu} \rangle = \delta_{\lambda \mu} z_{\lambda}. \]

\[ \lambda = (1^{m_1} 2^{m_2} \cdots k^{m_k}) \implies z_{\lambda} = 1^{m_1} m_1! 2^{m_2} m_2! \cdots k^{m_k} m_k! \]

- Equivalent to coefficient extraction:

\[ \left[ x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n} \right] F = \langle F, h_{k_1} h_{k_2} \cdots h_{k_n} \rangle \]

- Adjunction:

\[ \langle p_n F, G \rangle = \langle F, n \frac{\partial}{\partial p_n} G \rangle. \]
The Scalar Product

Theorem (Gessel, 90): If $F \in K[[p_1, p_2, \ldots, p_n; t]]$ and $G \in K[[p_1, p_2, \ldots; t]]$ are both D-finite symmetric series then the scalar product

$$\langle F, G \rangle = \phi(t)$$

is a D-finite function with respect to $t$. 

Termination guaranteed by D-finiteness.

Ex:

```plaintext
> scalar_de(exp(p1+p1^2/2+p2+p2^2), exp((p2^2+p2/2)*t), [t], f);
```

```plaintext
(-126t + 13 - 16t^2)f(t) + (-1 + 32t - 256t^2)d/dt f(t)
```
The Scalar Product

**Theorem (Gessel, 90):** If \( F \in K[[p_1, p_2, \ldots, p_n; t]] \) and \( G \in K[[p_1, p_2, \ldots; t]] \) are both D-finite symmetric series then the scalar product

\[
\langle F, G \rangle = \phi(t)
\]

is a D-finite function with respect to \( t \).

That is, \( \phi(t) \) satisfies a linear DE with polynomial coefficients. Infact, this DE can be calculated (Chyzak, M., Salvy 02)

Termination guaranteed by D-finiteness.

Ex:

\[
\begin{align*}
> \text{scalar\_de}(& \exp(p1+p1^2/2+p2+p2^2), \exp((p2^2+p2/2)*t), [t], f) ; \\
& \left[ (-126t + 13 - 16t^2) f(t) + (-1 + 32t - 256t^2) \frac{d}{dt} f(t) \right]
\end{align*}
\]
## Encoding combinatorial objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Encoding $\pi$</th>
<th>Sum over objects in the class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3</td>
<td>$x_1^3 x_2^4 x_3^2 x_4^5 x_5^5 x_6^2 x_7^2$</td>
<td>$G = \sum g$ a graph $\pi(g)$</td>
</tr>
<tr>
<td>4 5 6 7</td>
<td></td>
<td>$= \exp(\sum_n \frac{p_n^2}{2n} - \frac{p_{2n}}{2n})$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set($C$)</th>
<th>$x_1^2 x_2^2 x_3^2 x_4 x_5$</th>
<th>$H[Z_C], E[Z_C]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$ a labelled, finite comb. class</td>
<td>$x_i^k \Rightarrow k$ occurrences of label $i$</td>
<td>$H = \sum h_n, E = \sum e_n, Z = \text{Polya cycle index}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$[] = \text{plethystic substitution}$</td>
</tr>
</tbody>
</table>
## Encoding combinatorial objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Encoding $\pi$</th>
<th>Generating function of regular subclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4</td>
<td>$x_1^2x_2^2x_3^2x_4^2$</td>
<td>$G_k(t) = \langle G, \sum h_n^2t^n \rangle$</td>
</tr>
</tbody>
</table>

- **Idea:** Encode all objects, use scalar product to extract the number with a certain regularity.
- **Scalar product with a series gives the egf of a “regular” subclass.**
Weyl Algebras

\[ A_p := K \langle p_1, \ldots, p_n, \partial_1, \ldots, \partial_n \rangle \]

\[ \partial_i p_i = p_i \partial_i + 1, \text{ all other variables commute} \]
Weyl Algebras

\[ A_p := K \langle p_1, \ldots, p_n, \partial_1, \ldots, \partial_n \rangle \]

\[ \partial_i p_i = p_i \partial_i + 1, \text{ all other variables commute} \]

\[ f(p_1, \ldots, p_n) \]

Left annihilating ideal: \( I_f \)
\[ = \{ \phi \in A_p | \phi \cdot f = 0 \} \]

Left module: \( A_p f \)
\[ = \{ \psi \cdot f | \psi \in A_p \} \cong A_p / I_f \]
Weyl Algebras

\[ A_p := K\langle p_1, \ldots, p_n, \partial_1, \ldots, \partial_n \rangle \]

\[ \partial_i p_i = p_i \partial_i + 1, \text{ all other variables commute} \]

\[ f(p_1, \ldots, p_n) \quad \text{and} \quad f = \sin(p_1 + p_2/2) \]

Left annihilating ideal: \( I_f \)

\[ = \{ \phi \in A_p | \phi \cdot f = 0 \} \quad \text{and} \quad I_f = A_p\langle \partial_1^2 + 1, 4\partial_2^2 + 1, 2\partial_1 - \partial_2 \rangle \]

Left module: \( A_p f \)

\[ = \{ \psi \cdot f | \psi \in A_p \} \simeq A_p/I_f \quad \text{and} \quad A_p f = K[p_1, p_2] \sin(p_1 + p_2/2) \oplus K[p_1, p_2] \cos(p_1 + p_2/2) \]
How does it work?

Like with creative telescoping, we take linear combinations of equations to eliminate variables and generate new equations.

\[ 0 - 0 = 0 \]

\[ \langle 0, G \rangle - \langle F, 0 \rangle = 0 \]
How does it work?

Like with creative telescoping, we take linear combinations of equations to eliminate variables and generate new equations.

\[ 0 - 0 = 0 \]
\[ \langle 0, G \rangle - \langle F, 0 \rangle = 0 \]
\[ \langle \phi \cdot F, G \rangle - \langle F, \psi \cdot G \rangle = 0 \]  \[\phi \in I_{F}, \psi \in I_{G}\]
How does it work?

Like with creative telescoping, we take linear combinations of equations to eliminate variables and generate new equations.

\[0 - 0 = 0\]
\[\langle 0, G \rangle - \langle F, 0 \rangle = 0\]
\[\langle \phi \cdot F, G \rangle - \langle F, \psi \cdot G \rangle = 0\]
\[\langle F, \phi^\perp \cdot G \rangle - \langle F, \psi \cdot G \rangle = 0\]

\[\phi \in I_F, \psi \in I_G\]
\[p_n^\perp = n\partial_n, \partial_n^\perp = p_n / n\]
How does it work?

Like with creative telescoping, we take linear combinations of equations to eliminate variables and generate new equations.

\[ 0 - 0 = 0 \]
\[ \langle 0, G \rangle - \langle F, 0 \rangle = 0 \]
\[ \langle \phi \cdot F, G \rangle - \langle F, \psi \cdot G \rangle = 0 \]
\[ \langle F, \phi^\perp \cdot G \rangle - \langle F, \psi \cdot G \rangle = 0 \]
\[ \langle F, (\phi^\perp - \psi) \cdot G \rangle = 0 \]

\[ \phi \in I_F, \psi \in I_G \]
\[ p_n^\perp = n \partial_n, \partial_n^\perp = p_n/n \]

by linearity
How does it work?

Like with creative telescoping, we take linear combinations of equations to eliminate variables and generate new equations.

\[ 0 - 0 = 0 \]
\[ \langle 0, G \rangle - \langle F, 0 \rangle = 0 \]
\[ \langle \phi \cdot F, G \rangle - \langle F, \psi \cdot G \rangle = 0 \]
\[ \phi \in I_F, \psi \in I_G \]
\[ p_n^\perp = n \partial_n, \partial_n^\perp = p_n/n \]
\[ \langle F, (\phi^\perp - \psi) \cdot G \rangle = 0 \]

If, \((\phi^\perp - \psi) = \gamma(t, \partial_t) \in C[t, \partial_t], \) then...

\[ \langle F, \gamma(t, \partial_t) \cdot G \rangle = 0 \]
How does it work?

Like with creative telescoping, we take linear combinations of equations to eliminate variables and generate new equations.

\[ 0 - 0 = 0 \]
\[ \langle 0, G \rangle - \langle F, 0 \rangle = 0 \]
\[ \langle \phi \cdot F, G \rangle - \langle F, \psi \cdot G \rangle = 0 \]
\[ \phi \in I_F, \psi \in I_G \]
\[ p_n = n \partial_n, \partial_n^\perp = p_n/n \]

by linearity

\[ \langle F, \phi^\perp - \psi \rangle \cdot G \rangle = 0 \]
\[ \langle F, (\phi^\perp - \psi) \cdot G \rangle = 0 \]

If, \( (\phi^\perp - \psi) = \gamma(t, \partial_t) \in C[t, \partial_t] \), then...

\[ \langle F, \gamma(t, \partial_t) \cdot G \rangle = 0 \implies \gamma(t, \partial_t) \cdot \langle F, G \rangle = 0 \]

This gives a differential equation satisfied by \( \langle F, G \rangle \).
How does it work?

Like with creative telescoping, we take linear combinations of equations to eliminate variables and generate new equations.

\[ 0 - 0 = 0 \]
\[ \langle 0, G \rangle - \langle F, 0 \rangle = 0 \]
\[ \langle \phi \cdot F, G \rangle - \langle F, \psi \cdot G \rangle = 0 \]
\[ \phi \in I_F, \psi \in I_G \]
\[ p_n^\perp = n\partial_n, \partial_n^\perp = p_n/n \]
\[ p_n = n\partial_n, \partial_n^\perp = p_n/n \]
\[ \text{by linearity} \]
\[ \langle F, \gamma(t, \partial_t) \cdot G \rangle = 0 \]

\[ \text{If, } (\phi^\perp - \psi) = \gamma(t, \partial_t) \in C[t, \partial_t], \text{ then...} \]

\[ \langle F, \gamma(t, \partial_t) \cdot G \rangle = 0 \implies \gamma(t, \partial_t) \cdot \langle F, G \rangle = 0 \]

This gives a differential equation satisfied by \( \langle F, G \rangle \).

Any non-trivial element \( \gamma \) of \( (I_F^\perp + I_g) \cap C[t, \partial_t] \) provides a differential equation satisfied by \( \langle f, g \rangle \).
Holonomic Modules

\[ A_x := K \langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \rangle \]
\[ \partial_i x_i = x_i \partial_i + 1, \text{ all other variables commute} \]

Filtration: \( A_x^{(d)} = \{ \psi \in A_x \mid \deg \psi \leq d \} \)

Def. A module \( A_x f \) is **holonomic** if

\[ \dim A_x^{(d)} \cdot f = O(d^n). \]
Holonomic Modules

$\mathcal{A}_x := K\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \rangle$

$\partial_i x_i = x_i \partial_i + 1$, all other variables commute

Filtration: $A_x^{(d)} = \{ \psi \in A_x | \deg \psi \leq d \}$

Def. A module $A_x f$ is holonomic if

$\dim A_x^{(d)} \cdot f = O(d^n)$.

$K[x_1, \ldots, x_n] = A_x \cdot 1$: $\dim A_x^{(d)} \cdot 1 = O(d^n)$
Holonomic Modules

\[ A_x := K\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \rangle \]
\[ \partial_i x_i = x_i \partial_i + 1, \text{ all other variables commute} \]

Filtration: \( A_x^{(d)} = \{ \psi \in A_x | \deg \psi \leq d \} \)

Def. A module \( A_x f \) is holonomic if

\[ \dim A_x^{(d)} \cdot f = O(d^n). \]

✓ \( K[x_1, \ldots, x_n] = A_x \cdot 1: \dim A_x^{(d)} \cdot 1 = O(d^n) \)

✗ \( A_x: \dim A_x^{(d)} = O(d^{2n}) \)
Holonomic Modules

\[ A_x := K\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \rangle \]
\[ \partial_i x_i = x_i \partial_i + 1, \text{ all other variables commute} \]

Filtration: \( A_x^{(d)} = \{ \psi \in A_x \mid \deg \psi \leq d \} \)

Def. A module \( A_x f \) is holonomic if

\[ \dim A_x^{(d)} \cdot f = O(d^n). \]

✔ \( K[x_1, \ldots, x_n] = A_x \cdot 1: \dim A_x^{(d)} \cdot 1 = O(d^n) \)

✗ \( A_x: \dim A_x^{(d)} = O(d^{2n}) \)

✔ \( A_x \cdot \sin(x) \simeq K[x] \sin(x) \oplus K[x] \cos(x) \)
\[ \dim A_x^{(d)} \cdot \sin(x) = 2 \dim K[x]^{(d)} = O(d^1) \]
Holonomic Modules

\[ A_x := K\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n \rangle \]
\[ \partial_i x_i = x_i \partial_i + 1, \text{ all other variables commute} \]

Filtration: \[ A_x^{(d)} = \{ \psi \in A_x \mid \deg \psi \leq d \} \]

Def. A module \( A_x f \) is holonomic if

\[ \dim A_x^{(d)} \cdot f = O(d^n). \]

- \( K[x_1, \ldots, x_n] = A_x \cdot 1: \dim A_x^{(d)} \cdot 1 = O(d^n) \)
- \( A_x: \dim A_x^{(d)} = O(d^{2n}) \)
- \( A_x \cdot \sin(x) \simeq K[x] \sin(x) \oplus K[x] \cos(x) \)
  \[ \dim A_x^{(d)} \cdot \sin(x) = 2 \dim K[x]^{(d)} = O(d^1) \]

**Theorem: (Kashiwara)** \( f(x) \) D-finite \( \iff \) \( A_x f \) holonomic
\[(I_f^\perp + I_g) \cap A_t \text{ is not empty!}\]

Idea: model adjunction \( p_n^\perp = n\partial p_n \) by a tensor product

- Define surjection:

\[
A_{p,t} f \otimes_{C[t]} A_{p,t} g \rightarrow A_{p,t} \langle f, g \rangle
\]
\[
a \otimes_{C[t]} b \mapsto \langle a, b \rangle
\]

Prop: \( I_f^\perp + I_g \) contains \( \text{Ann}_{A_t}(f \otimes_{C[t]} g) \)
\[(I_f^\perp + I_g) \cap A_t \text{ is not empty!}\]

**Idea:** model adjunction \( p_n^\perp = n\partial p_n \) by a tensor product

- **Define surjection:**
  \[ A_{p,t} f \otimes_{\mathbb{C}[t]}^\perp A_{p,t} g \rightarrow A_{p,t} \langle f, g \rangle \]
  \[ a \otimes_{\mathbb{C}[t]}^\perp b \mapsto \langle a, b \rangle \]

**Prop:** \( I_f^\perp + I_g \) contains \( \text{Ann}_{A_t} (f \otimes_{\mathbb{C}[t]}^\perp g) \)

- **Thm:** \( A_{p,t} f \otimes_{\mathbb{C}[t]}^\perp A_{p,t} g \) is a holonomic \( A_{p,t} \)-module

**key fact:** \( \perp \) is degree preserving
\((I_f^\perp + I_g) \cap A_t\) is not empty!

Idea: model adjunction \(p_n^\perp = n\partial_{p_n}\) by a tensor product

- Define surjection:
  \[ A_{p,t}f \otimes_{C[t]}^\perp A_{p,t}g \rightarrow A_{p,t} \langle f, g \rangle \]
  \[ a \otimes_{C[t]}^\perp b \mapsto \langle a, b \rangle \]

Prop: \(I_f^\perp + I_g\) contains \(\text{Ann}_{A_t}(f \otimes_{C[t]}^\perp g)\)

- Thm: \(A_{p,t}f \otimes_{C[t]}^\perp A_{p,t}g\) is a holonomic \(A_{p,t}\)-module

- \( \left( A_{p,t}f \otimes_{C[t]}^\perp A_{p,t}g \right) \cap "A_t" \) is a holonomic \(A_t\)-module isomorphic to \(A_t / \text{Ann}_{A_t}(f \otimes_{C[t]}^\perp g)\) as \(A_t\)-modules

Cor: \(\text{Ann}_{A_t}(f \otimes_{C[t]}^\perp g)\) is non-trivial.
Algorithm computes scalar products whose adjoint operation is a degree preserving operation:

➢ **Related to Macdonald Polynomials**

\[
\langle p_\lambda, p_\mu \rangle_{q,t} = z_\lambda \delta_{\mu\lambda} \prod_{\lambda_i \in \lambda} \frac{1 - q^{\lambda_i}}{1 - t^{\lambda_i}}
\]

➢ **MacMahon symmetric functions**

\[
\langle P(a,b)^2(c,d), P(a,b)^2(c,d) \rangle = 2 \left( \frac{a!b!}{(a + b - 1)!} \right)^2 \left( \frac{c!d!}{(c + d - 1)!} \right)
\]
Lots to do...

- Other D-finite closure properties?
- Effective closure properties??
- Classifying families of combinatorial objects e.g. walks in the plane, families of generalized partitions
- Improving the Gröbner basis calculations