# Power series of non linear operators, effective actions and some combinatorial illustrations

Krajewski Thomas

Centre de Physique Théorique, Marseille
krajew@cpt.univ-mrs.fr

*Conférence Algèbre combinatoire et Arbres*
*Lyon, May 2008*

# Power series of non linear operators, effective actions and some combinatorial illustrations

Krajewski Thomas

Centre de Physique Théorique, Marseille
krajew@cpt.univ-mrs.fr

*Conférence Algèbre combinatoire et Arbres*
*Lyon, May 2008*

## Rooted trees and power series of non linear operators

# Power series of non linear operators, effective actions and some combinatorial illustrations

Krajewski Thomas

Centre de Physique Théorique, Marseille
krajew@cpt.univ-mrs.fr

*Conférence Algèbre combinatoire et Arbres*
*Lyon, May 2008*

**Rooted trees and power series of non linear operators**

- Geometric and binomial series

# Power series of non linear operators, effective actions and some combinatorial illustrations

Krajewski Thomas

Centre de Physique Théorique, Marseille
krajew@cpt.univ-mrs.fr

*Conférence Algèbre combinatoire et Arbres*
*Lyon, May 2008*

**Rooted trees and power series of non linear operators**

- Geometric and binomial series
- Postnikov's hook length formula

# Power series of non linear operators, effective actions and some combinatorial illustrations

Krajewski Thomas

Centre de Physique Théorique, Marseille
krajew@cpt.univ-mrs.fr

*Conférence Algèbre combinatoire et Arbres*
*Lyon, May 2008*

**Rooted trees and power series of non linear operators**

- Geometric and binomial series
- Postnikov's hook length formula

**Feynman diagrams and iterations of effective actions**

# Power series of non linear operators, effective actions and some combinatorial illustrations

Krajewski Thomas

Centre de Physique Théorique, Marseille
krajew@cpt.univ-mrs.fr

*Conférence Algèbre combinatoire et Arbres*
*Lyon, May 2008*

**Rooted trees and power series of non linear operators**

- Geometric and binomial series
- Postnikov's hook length formula

**Feynman diagrams and iterations of effective actions**

- Some properties of the Tutte polynomial

# Power series of non linear operators, effective actions and some combinatorial illustrations

Krajewski Thomas

Centre de Physique Théorique, Marseille
krajew@cpt.univ-mrs.fr

*Conférence Algèbre combinatoire et Arbres*
*Lyon, May 2008*

**Rooted trees and power series of non linear operators**

- Geometric and binomial series
- Postnikov's hook length formula

**Feynman diagrams and iterations of effective actions**

- Some properties of the Tutte polynomial
- Loop decomposition of the Symanzik polynomial

# Perturbative solution of a differential equation

# Perturbative solution of a differential equation

Expand the solution of **differential equation**

$$\frac{dx}{ds} = X(x), \qquad x(s_0) = x_0,$$

in powers of $(s - s_0)$

# Perturbative solution of a differential equation

Expand the solution of **differential equation**

$$\frac{dx}{ds} = X(x), \qquad x(s_0) = x_0,$$

in powers of $(s - s_0)$ with

$$\begin{aligned}
\frac{dx^i}{ds} &= X^i \\
\frac{d^2x^i}{ds^2} &= \sum_j \frac{\partial X^i}{\partial x^j} X^j \\
\frac{d^3x^i}{ds^3} &= \sum_{j,k} \frac{\partial X^i}{\partial x^j} \frac{\partial X^j}{\partial x^k} X^k + \frac{\partial^2 X^i}{\partial x^j \partial x^k} X^j X^k \\
\frac{d^4x^i}{ds^4} &= \sum_{j,k,l} \frac{\partial X^i}{\partial x^j} \frac{\partial X^j}{\partial x^k} \frac{\partial X^k}{\partial x^l} X^l + 3 \frac{\partial^2 X^i}{\partial x^j \partial x^k} \frac{\partial X^k}{\partial x^l} X^j X^l \\
&\quad + \frac{\partial^3 X^i}{\partial x^j \partial x^k \partial x^l} X^j X^k X^l + \frac{\partial X^i}{\partial x^j} \frac{\partial^2 X^j}{\partial x^k \partial x^l} X^k X^l
\end{aligned}$$

# Perturbative solution of a differential equation

Expand the solution of **differential equation**

$$\frac{dx}{ds} = X(x), \qquad x(s_0) = x_0,$$

in powers of $(s - s_0)$ with

$$\begin{aligned}
\frac{dx^i}{ds} &= X^i \\
\frac{d^2x^i}{ds^2} &= \sum_j \frac{\partial X^i}{\partial x^j} X^j \\
\frac{d^3x^i}{ds^3} &= \sum_{j,k} \frac{\partial X^i}{\partial x^j} \frac{\partial X^j}{\partial x^k} X^k + \frac{\partial^2 X^i}{\partial x^j \partial x^k} X^j X^k \\
\frac{d^4x^i}{ds^4} &= \sum_{j,k,l} \frac{\partial X^i}{\partial x^j} \frac{\partial X^j}{\partial x^k} \frac{\partial X^k}{\partial x^l} X^l + 3 \frac{\partial^2 X^i}{\partial x^j \partial x^k} \frac{\partial X^k}{\partial x^l} X^j X^l \\
&\quad + \frac{\partial^3 X^i}{\partial x^j \partial x^k \partial x^l} X^j X^k X^l + \frac{\partial X^i}{\partial x^j} \frac{\partial^2 X^j}{\partial x^k \partial x^l} X^k X^l
\end{aligned}$$

Each of these terms correspond to **rooted trees** with various weights.

# Runge-Kutta methods

# Runge-Kutta methods

Numerical algorithm based on the **Runge-Kutta method**, given by a square matrix $(a_{ij})_{1\leq i,j\leq n}$ and a vector $(b_i)_{1\leq i\leq n}$ of real numbers,

$$x(s_1) = x(s_0) + h\sum_{i=1}^{n} b_i X(y_i),$$

where $y_i$ is determined by $y_i = x_0 + h\sum_{j=1}^{n} a_{ij}X(y_j)$ and $h = s_1 - s_0$.

# Runge-Kutta methods

Numerical algorithm based on the **Runge-Kutta method**, given by a square matrix $(a_{ij})_{1\leq i,j\leq n}$ and a vector $(b_i)_{1\leq i\leq n}$ of real numbers,

$$x(s_1) = x(s_0) + h\sum_{i=1}^{n} b_i X(y_i),$$

where $y_i$ is determined by $y_i = x_0 + h\sum_{j=1}^{n} a_{ij}X(y_j)$ and $h = s_1 - s_0$.

The method is of order $n$ if it agrees with the exact expansion up to terms of order $h^n$.

# Runge-Kutta methods

Numerical algorithm based on the **Runge-Kutta method**, given by a square matrix $(a_{ij})_{1\leq i,j\leq n}$ and a vector $(b_i)_{1\leq i\leq n}$ of real numbers,

$$x(s_1) = x(s_0) + h\sum_{i=1}^{n} b_i X(y_i),$$

where $y_i$ is determined by $y_i = x_0 + h\sum_{j=1}^{n} a_{ij}X(y_j)$ and $h = s_1 - s_0$.

The method is of order $n$ if it agrees with the exact expansion up to terms of order $h^n$.

Runge-Kutta methods can be **composed** (Butcher's B-series):

# Runge-Kutta methods

Numerical algorithm based on the **Runge-Kutta method**, given by a square matrix $(a_{ij})_{1\leq i,j\leq n}$ and a vector $(b_i)_{1\leq i\leq n}$ of real numbers,

$$x(s_1) = x(s_0) + h\sum_{i=1}^{n} b_i X(y_i),$$

where $y_i$ is determined by $y_i = x_0 + h\sum_{j=1}^{n} a_{ij} X(y_j)$ and $h = s_1 - s_0$.

The method is of order $n$ if it agrees with the exact expansion up to terms of order $h^n$.

Runge-Kutta methods can be **composed** (Butcher's B-series):

- compute $x(s_1)$ starting at $x(s_0)$ by a first method

# Runge-Kutta methods

Numerical algorithm based on the **Runge-Kutta method**, given by a square matrix $(a_{ij})_{1\leq i,j\leq n}$ and a vector $(b_i)_{1\leq i\leq n}$ of real numbers,

$$x(s_1) = x(s_0) + h\sum_{i=1}^{n} b_i X(y_i),$$

where $y_i$ is determined by $y_i = x_0 + h\sum_{j=1}^{n} a_{ij} X(y_j)$ and $h = s_1 - s_0$.

The method is of order $n$ if it agrees with the exact expansion up to terms of order $h^n$.

Runge-Kutta methods can be **composed** (Butcher's B-series):

- compute $x(s_1)$ starting at $x(s_0)$ by a first method
- compute $x(s_s)$ starting at $x_(s_1)$ by a second method

# Runge-Kutta methods

Numerical algorithm based on the **Runge-Kutta method**, given by a square matrix $(a_{ij})_{1\leq i,j\leq n}$ and a vector $(b_i)_{1\leq i\leq n}$ of real numbers,

$$x(s_1) = x(s_0) + h\sum_{i=1}^{n} b_i X(y_i),$$

where $y_i$ is determined by $y_i = x_0 + h\sum_{j=1}^{n} a_{ij}X(y_j)$ and $h = s_1 - s_0$.

The method is of order $n$ if it agrees with the exact expansion up to terms of order $h^n$.

Runge-Kutta methods can be **composed** (Butcher's B-series):

- compute $x(s_1)$ starting at $x(s_0)$ by a first method
- compute $x(s_s)$ starting at $x_(s_1)$ by a second method

amounts to compute $x(s_2)$ starting at $x(s_0)$ by a third method.

# Runge-Kutta methods

Numerical algorithm based on the **Runge-Kutta method**, given by a square matrix $(a_{ij})_{1\leq i,j\leq n}$ and a vector $(b_i)_{1\leq i\leq n}$ of real numbers,

$$x(s_1) = x(s_0) + h\sum_{i=1}^{n} b_i X(y_i),$$

where $y_i$ is determined by $y_i = x_0 + h\sum_{j=1}^{n} a_{ij}X(y_j)$ and $h = s_1 - s_0$.

The method is of order $n$ if it agrees with the exact expansion up to terms of order $h^n$.

Runge-Kutta methods can be **composed** (Butcher's B-series):

- compute $x(s_1)$ starting at $x(s_0)$ by a first method
- compute $x(s_s)$ starting at $x_(s_1)$ by a second method

amounts to compute $x(s_2)$ starting at $x(s_0)$ by a third method.

This can be formalized using the **Hopf algebra** of rooted trees.

# Hopf algebra of rooted trees

# Hopf algebra of rooted trees

# Hopf algebra of rooted trees

$\mathcal{H}_T$ = commutative algebra generated by all (isomorphism classes) of **rooted trees**.

$\mathcal{H}_T$ = commutative algebra generated by all (isomorphism classes) of **rooted trees**. Hopf algebra structure:

# Hopf algebra of rooted trees

$\mathcal{H}_T$ = commutative algebra generated by all (isomorphism classes) of **rooted trees**. Hopf algebra structure:

- coproduct

$$\Delta(t) = t \otimes 1 + 1 \otimes t + \sum_{c \text{ admissible cut}} P_c(t) \otimes R_c(t) \qquad (1)$$

admissible cut : any path from any leaf to the root is cut at most once
$R_c(t)$ = connected component of the root
$P_c(t)$ = product of remaining trees.

# Hopf algebra of rooted trees

$\mathcal{H}_T$ = commutative algebra generated by all (isomorphism classes) of **rooted trees**. Hopf algebra structure:

- coproduct

$$\Delta(t) = t \otimes 1 + 1 \otimes t + \sum_{c \text{ admissible cut}} P_c(t) \otimes R_c(t) \qquad (1)$$

  admissible cut : any path from any leaf to the root is cut at most once
  $R_c(t)$ = connected component of the root
  $P_c(t)$ = product of remaining trees.

- counit: $\epsilon(t) = 0$ unless $t = 1$

# Hopf algebra of rooted trees

$\mathcal{H}_T$ = commutative algebra generated by all (isomorphism classes) of **rooted trees**. Hopf algebra structure:

- coproduct

$$\Delta(t) = t \otimes 1 + 1 \otimes t + \sum_{c \text{ admissible cut}} P_c(t) \otimes R_c(t) \qquad (1)$$

  admissible cut : any path from any leaf to the root is cut at most once
  $R_c(t)$ = connected component of the root
  $P_c(t)$ = product of remaining trees.

- counit: $\epsilon(t) = 0$ unless $t = 1$
- antipode:

$$S(t) = -\sum_{c \text{ cut}} (-1)^{n_c(t)} \Pi_c(t), \qquad (2)$$

  $n_c(t)$ = number of cuts.

# Hopf algebra of rooted trees

$\mathcal{H}_T$ = commutative algebra generated by all (isomorphism classes) of **rooted trees**. Hopf algebra structure:

- coproduct

$$\Delta(t) = t \otimes 1 + 1 \otimes t + \sum_{c \text{ admissible cut}} P_c(t) \otimes R_c(t) \qquad (1)$$

admissible cut : any path from any leaf to the root is cut at most once
$R_c(t)$ = connected component of the root
$P_c(t)$ = product of remaining trees.

- counit: $\epsilon(t) = 0$ unless $t = 1$
- antipode:

$$S(t) = -\sum_{c \text{ cut}} (-1)^{n_c(t)} \Pi_c(t), \qquad (2)$$

$n_c(t)$ = number of cuts.

$\mathcal{H}_T$ is graded by the number of vertices $|t|$.

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

**Characters** of $\mathcal{H}_T$ form a group $G_T$ for the convolution product

$$\alpha * \beta = (\alpha \otimes \beta) \circ \Delta \tag{3}$$

with unit $\epsilon$ and inverse $\alpha^{-1} = \alpha \circ S$.

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

**Characters** of $\mathcal{H}_T$ form a group $G_T$ for the convolution product

$$\alpha * \beta = (\alpha \otimes \beta) \circ \Delta \tag{3}$$

with unit $\epsilon$ and inverse $\alpha^{-1} = \alpha \circ S$.

**Infinitesimal characters** (derivations with values in the trivial bimodule provided by $\epsilon$) define its Lie algebra $\mathcal{G}_T$.

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

**Characters** of $\mathcal{H}_T$ form a group $G_T$ for the convolution product

$$\alpha * \beta = (\alpha \otimes \beta) \circ \Delta \qquad (3)$$

with unit $\epsilon$ and inverse $\alpha^{-1} = \alpha \circ S$.

**Infinitesimal characters** (derivations with values in the trivial bimodule provided by $\epsilon$) define its Lie algebra $\mathcal{G}_T$.

- Lie bracket only involves single cuts ($n_c$=1)

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

**Characters** of $\mathcal{H}_T$ form a group $G_T$ for the convolution product

$$\alpha * \beta = (\alpha \otimes \beta) \circ \Delta \tag{3}$$

with unit $\epsilon$ and inverse $\alpha^{-1} = \alpha \circ S$.

**Infinitesimal characters** (derivations with values in the trivial bimodule provided by $\epsilon$) define its Lie algebra $\mathcal{G}_T$.

- Lie bracket only involves single cuts ($n_c{=}1$)
- $\exp_*(\delta) = \sum_n \frac{\delta^{*n}}{n!}$ is a bijection between $\mathcal{G}_T$ and $G_T$

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

**Characters** of $\mathcal{H}_T$ form a group $G_T$ for the convolution product

$$\alpha * \beta = (\alpha \otimes \beta) \circ \Delta \tag{3}$$

with unit $\epsilon$ and inverse $\alpha^{-1} = \alpha \circ S$.

**Infinitesimal characters** (derivations with values in the trivial bimodule provided by $\epsilon$) define its Lie algebra $\mathcal{G}_T$.

- Lie bracket only involves single cuts ($n_c$=1)
- $\exp_*(\delta) = \sum_n \frac{\delta^{*n}}{n!}$ is a bijection between $\mathcal{G}_T$ and $G_T$

Powerful technique for proving general results.

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

**Characters** of $\mathcal{H}_T$ form a group $G_T$ for the convolution product

$$\alpha * \beta = (\alpha \otimes \beta) \circ \Delta \tag{3}$$

with unit $\epsilon$ and inverse $\alpha^{-1} = \alpha \circ S$.

**Infinitesimal characters** (derivations with values in the trivial bimodule provided by $\epsilon$) define its Lie algebra $\mathcal{G}_T$.

- Lie bracket only involves single cuts ($n_c$=1)
- $\exp_*(\delta) = \sum_n \frac{\delta^{*n}}{n!}$ is a bijection between $\mathcal{G}_T$ and $G_T$

Powerful technique for proving general results.

$G_T$ is a projective limit of finite dimensional nilpotent Lie group $G_T^n$ with Lie algebra $\mathcal{G}_T^n$ both obtained by truncation at order $n$.

# Lie groups and Lie algebras associated to $\mathcal{H}_T$

**Characters** of $\mathcal{H}_T$ form a group $G_T$ for the convolution product

$$\alpha * \beta = (\alpha \otimes \beta) \circ \Delta \tag{3}$$

with unit $\epsilon$ and inverse $\alpha^{-1} = \alpha \circ S$.

**Infinitesimal characters** (derivations with values in the trivial bimodule provided by $\epsilon$) define its Lie algebra $\mathcal{G}_T$.

- Lie bracket only involves single cuts ($n_c$=1)
- $\exp_*(\delta) = \sum_n \frac{\delta^{*n}}{n!}$ is a bijection between $\mathcal{G}_T$ and $G_T$

Powerful technique for proving general results.

$G_T$ is a projective limit of finite dimensional nilpotent Lie group $G_T^n$ with Lie algebra $\mathcal{G}_T^n$ both obtained by truncation at order $n$.

Results also valid for the more general graded and commutative Hopf algebras based on **Feynman diagrams**.

# Power series of non-linear operators

# Power series of non-linear operators

Smooth map $X$ raised to the power of the tree $t$:

$$X^t = \overrightarrow{\prod_{v \in t}} X^{(n_v)}$$

$X^{(n)}$: $n^{\text{th}}$ order differential of $X$ with $n_v$ the number of edges leaving $v$

# Power series of non-linear operators

Smooth map $X$ raised to the power of the tree $t$:

$$X^t = \overrightarrow{\prod_{v \in t}} X^{(n_v)}$$

$X^{(n)}$: $n^{\text{th}}$ order differential of $X$ with $n_v$ the number of edges leaving $v$

Examples: $X$ $= X'\big[X'[X]\big]$, $X$ $= X''\big[X, X\big]$.

# Power series of non-linear operators

Smooth map $X$ raised to the power of the tree $t$:

$$X^t = \overrightarrow{\prod_{v \in t}} X^{(n_v)}$$

$X^{(n)}$: $n^{\text{th}}$ order differential of $X$ with $n_v$ the number of edges leaving $v$

Examples: $X$ $= X'\big[X'[X]\big]$, $X$ $= X''\big[X, X\big]$.

Formal power series of non-linear operators:

$$\Psi_\alpha(X) = \sum_t \alpha(t)\,\frac{X^t}{\mathrm{S}_t}, \quad \alpha \in G_T.$$

with $S_t$ the symmetry factor of $t$ (cardinal of the automorphism group).

# Power series of non-linear operators

Smooth map $X$ raised to the power of the tree $t$:

$$X^t = \overrightarrow{\prod_{v \in t}} X^{(n_v)}$$

$X^{(n)}$: $n^{\text{th}}$ order differential of $X$ with $n_v$ the number of edges leaving $v$

Examples: $X$ $= X'\big[X'[X]\big]$, $X$ $= X''\big[X, X\big]$.

Formal power series of non-linear operators:

$$\Psi_\alpha(X) = \sum_t \alpha(t) \, \frac{X^t}{\mathrm{S}_t}, \quad \alpha \in G_T.$$

with $S_t$ the symmetry factor of $t$ (cardinal of the automorphism group).

Composition law (B-series):

$$\Psi_\alpha(X) \circ \Psi_\beta(X) = \Psi_{\beta * \alpha}(X), \quad \alpha, \beta \in G_T.$$

# Geometric series

# Geometric series

If $\alpha$ is the character that takes the value $-1$ on the tree with one vertex and 0 otherwise, $\alpha^{-1} = \alpha \circ S$ takes the value 1 on all trees and defines the **geometric series**

$$(\mathrm{id} - X)^{-1} = \sum_t \frac{X^t}{S_t}$$

# Geometric series

If $\alpha$ is the character that takes the value $-1$ on the tree with one vertex and 0 otherwise, $\alpha^{-1} = \alpha \circ S$ takes the value 1 on all trees and defines the **geometric series**

$$(\mathrm{id} - X)^{-1} = \sum_t \frac{X^t}{S_t}$$

Perturbative solution of the **fixed point equation**

$$x = x_0 + X(x) \quad \rightarrow \quad x = (\mathrm{id} - X)^{-1}(x_0)$$

# Geometric series

If $\alpha$ is the character that takes the value $-1$ on the tree with one vertex and 0 otherwise, $\alpha^{-1} = \alpha \circ S$ takes the value 1 on all trees and defines the **geometric series**

$$(\mathrm{id} - X)^{-1} = \sum_t \frac{X^t}{S_t}$$

Perturbative solution of the **fixed point equation**

$$x = x_0 + X(x) \quad \to \quad x = (\mathrm{id} - X)^{-1}(x_0)$$

Applications:

# Geometric series

If $\alpha$ is the character that takes the value $-1$ on the tree with one vertex and 0 otherwise, $\alpha^{-1} = \alpha \circ S$ takes the value 1 on all trees and defines the **geometric series**

$$(\mathrm{id} - X)^{-1} = \sum_t \frac{X^t}{S_t}$$

Perturbative solution of the **fixed point equation**

$$x = x_0 + X(x) \quad \rightarrow \quad x = (\mathrm{id} - X)^{-1}(x_0)$$

Applications:

- solution of **differential equations** written in integral form,

# Geometric series

If $\alpha$ is the character that takes the value $-1$ on the tree with one vertex and 0 otherwise, $\alpha^{-1} = \alpha \circ S$ takes the value 1 on all trees and defines the **geometric series**

$$(\mathrm{id} - X)^{-1} = \sum_t \frac{X^t}{S_t}$$

Perturbative solution of the **fixed point equation**

$$x = x_0 + X(x) \quad \rightarrow \quad x = (\mathrm{id} - X)^{-1}(x_0)$$

Applications:

- solution of **differential equations** written in integral form,
- **counting trees** ($X(x) = x^2$ and $x_0 = 1 \rightarrow$ Catalan numbers),

# Geometric series

If $\alpha$ is the character that takes the value $-1$ on the tree with one vertex and 0 otherwise, $\alpha^{-1} = \alpha \circ S$ takes the value 1 on all trees and defines the **geometric series**

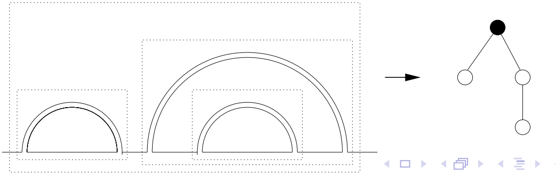
$$(\mathrm{id} - X)^{-1} = \sum_t \frac{X^t}{S_t}$$

Perturbative solution of the **fixed point equation**

$$x = x_0 + X(x) \quad \rightarrow \quad x = (\mathrm{id} - X)^{-1}(x_0)$$

Applications:

- solution of **differential equations** written in integral form,
- **counting trees** ($X(x) = x^2$ and $x_0 = 1 \rightarrow$ Catalan numbers),
- **resummation** of tree-like structures (example: planar diagrams)



# Tree ordered products

# Tree ordered products

Pertubative solution of a non-linear time dependent **differential equation**

$$\frac{dx}{ds} = X_s(x), \quad x(s_0) = x_0,$$

# Tree ordered products

Pertubative solution of a non-linear time dependent **differential equation**

$$\frac{dx}{ds} = X_s(x), \quad x(s_0) = x_0,$$

as $R_{s,s_0}(x_0)$ with

$$R_{s,s_0} = \sum_t \frac{1}{s_t} \int_{I^t_{s,s_0}} d^{|t|}s \overrightarrow{\prod_{v \in t}} X^{(n_v)}_{s^v}.$$

$I^t_{s,s_0} \subset \mathbb{R}^{|t|}$ is a **treeplex** (generalization of a simplex) obtained by assigning real numbers $s^v$ to the vertices in decreasing order from the root to the leaves, with $s^{\text{root}} \leq s$ and $s^{\text{leaf}} \geq s_0$.

# Tree ordered products

Pertubative solution of a non-linear time dependent **differential equation**

$$\frac{dx}{ds} = X_s(x), \quad x(s_0) = x_0,$$

as $R_{s,s_0}(x_0)$ with

$$R_{s,s_0} = \sum_t \frac{1}{s_t} \int_{I^t_{s,s_0}} d^{|t|}s \overrightarrow{\prod_{v \in t}} X^{(n_v)}_{s^v}.$$

$I^t_{s,s_0} \subset \mathbb{R}^{|t|}$ is a **treeplex** (generalization of a simplex) obtained by assigning real numbers $s^v$ to the vertices in decreasing order from the root to the leaves, with $s^{\text{root}} \leq s$ and $s^{\text{leaf}} \geq s_0$.

For a time independent equation we the **tree factorial** $t!$

$$\int_{I^t_{s,s_0}} d^{|t|}s = \frac{1}{t!}.$$

# Geometric interpretation of the coproduct

Comparing both sides of $R_{s_2,s_1} \circ R_{s_1,s_0} = R_{s_2,s_0}$ yields a disjoint union

$$I^{t}_{s_2,s_0} = \bigcup_{c \text{ admissible cut}} \mathfrak{S}_c \left( I^{t'_1}_{s_1,s_0} \times \cdots \times I^{t_n}_{s_1,s_0} \times I^{t''}_{s_2,s_1} \right) \quad (4)$$

with $P_c(t) = t'_1 \dots t'_n$ and $R_c(t) = t''$ and $\mathfrak{S}_c$ a permutation of the labels preserving the ordering of the tree.

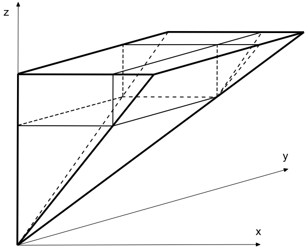
# Geometric interpretation of the coproduct

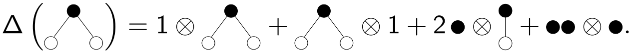
Comparing both sides of $R_{s_2,s_1} \circ R_{s_1,s_0} = R_{s_2,s_0}$ yields a disjoint union

$$I^t_{s_2,s_0} = \bigcup_{c \text{ admissible cut}} \mathfrak{S}_c \left( I^{t'_1}_{s_1,s_0} \times \cdots \times I^{t_n}_{s_1,s_0} \times I^{t''}_{s_2,s_1} \right) \quad (4)$$

with $P_c(t) = t'_1 \ldots t'_n$ and $R_c(t) = t''$ and $\mathfrak{S}_c$ a permutation of the labels preserving the ordering of the tree.

Example:



# The binomial series

# The binomial series

Expanding $(\alpha)^a = (\epsilon + \alpha - \epsilon)^a$ for $\alpha$ the character that takes the value 1 on the tree with one vertex and vanishes otherwise, we obtain the **binomial series** for a non linear operator

$$(\mathrm{id} + X)^a = \sum_t \left( \sum_{n=d_t}^{|t|} N(n,t) \frac{a(a-1)\cdots(a-n+1)}{n!} \right) \frac{X^t}{\mathrm{S}_t}$$

where $N(n,t)$ is the number of surjective maps from the vertices of $t$ to $\{1, \cdots, n\}$, strictly increasing from the root to the leaves (heaps) and $d_t$ is the height of the tree, i.e. the length of the longest path from the root to the leaves.

# The binomial series

Expanding $(\alpha)^a = (\epsilon + \alpha - \epsilon)^a$ for $\alpha$ the character that takes the value 1 on the tree with one vertex and vanishes otherwise, we obtain the **binomial series** for a non linear operator

$$(\mathrm{id} + X)^a = \sum_t \left( \sum_{n=d_t}^{|t|} N(n,t) \frac{a(a-1)\cdots(a-n+1)}{n!} \right) \frac{X^t}{\mathrm{S}_t}$$

where $N(n,t)$ is the number of surjective maps from the vertices of $t$ to $\{1, \cdots, n\}$, strictly increasing from the root to the leaves (heaps) and $d_t$ is the height of the tree, i.e. the length of the longest path from the root to the leaves.

**Square root** of a diffeomorphism close to the identity

$$\sqrt{\mathrm{id} + X} = \mathrm{id}\tfrac{1}{2}X - \tfrac{1}{8}X'[X] + \tfrac{1}{16}X'[X'[X]] - \tfrac{5}{128}X'[X'[X'[X]]] \\ + \tfrac{1}{128}X''[X, X'[X]] - \tfrac{1}{2\cdot 64}X'[X''[X,X]] + \tfrac{1}{6\cdot 64}X'''[X,X,X] + \cdots$$

which fulfills $\sqrt{\mathrm{id} + X} \circ \sqrt{\mathrm{id} + X} = \mathrm{id} + X$ up to terms of fifth order.

# Combinatorial applications of the binomial series

# Combinatorial applications of the binomial series

If $\beta$ takes the value 1 on all trees, expanding $(\beta)^{-a} = (\epsilon - \alpha)^a$ yields

$$(\mathrm{id} - X)^{-a} = \sum_t \left( \sum_{n=d_t}^{|t|} \widetilde{N}(n,t) \frac{a(a-1)\cdots(a-n+1)}{n!} \right) \frac{X^t}{\mathrm{S}_t}$$

with $\widetilde{N}(n,t)$ the number of surjective maps from the vertices of $t$ to $\{1, \cdots, n\}$, non decreasing from the root to the leaves.

# Combinatorial applications of the binomial series

If $\beta$ takes the value 1 on all trees, expanding $(\beta)^{-a} = (\epsilon - \alpha)^a$ yields

$$(\mathrm{id} - X)^{-a} = \sum_t \left( \sum_{n=d_t}^{|t|} \widetilde{N}(n,t) \frac{a(a-1)\cdots(a-n+1)}{n!} \right) \frac{X^t}{\mathrm{S}_t}$$

with $\widetilde{N}(n,t)$ the number of surjective maps from the vertices of $t$ to $\{1, \cdots, n\}$, non decreasing from the root to the leaves.

Equating $(\mathrm{id} - X)^a$ with a binomial series we get identities between $N(n,t)$ and $\widetilde{N}(n,t)$. For instance, for $a = 1, 2$

$$\begin{array}{rcl} 1 & = & \sum_{n=d_t}^{|t|} (-1)^{n+|t|} N(n,t) \\ \widetilde{N}(2,t) & = & \sum_{n=d_t}^{|t|} (-1)^{n+|t|} (n-1) N(t,n) \end{array}$$

# Combinatorial applications of the binomial series

If $\beta$ takes the value 1 on all trees, expanding $(\beta)^{-a} = (\epsilon - \alpha)^a$ yields

$$(\mathrm{id} - X)^{-a} = \sum_t \left( \sum_{n=d_t}^{|t|} \widetilde{N}(n,t) \frac{a(a-1)\cdots(a-n+1)}{n!} \right) \frac{X^t}{\mathrm{S}_t}$$

with $\widetilde{N}(n,t)$ the number of surjective maps from the vertices of $t$ to $\{1, \cdots, n\}$, non decreasing from the root to the leaves.

Equating $(\mathrm{id} - X)^a$ with a binomial series we get identities between $N(n,t)$ and $\widetilde{N}(n,t)$. For instance, for $a = 1, 2$

$$\begin{aligned} 1 &= \textstyle\sum_{n=d_t}^{|t|} (-1)^{n+|t|} N(n,t) \\ \widetilde{N}(2,t) &= \textstyle\sum_{n=d_t}^{|t|} (-1)^{n+|t|} (n-1) N(t,n) \end{aligned}$$

Computing the exponential as $\mathrm{e}^X = \lim_{n\to\infty} \left(\mathrm{id} + \frac{X}{n}\right)^n$ we obtain a combinatorial formula for the **tree factorial**

$$\frac{|t|!}{t!} = N(|t|, t).$$

# Postnikov's hook length formula

# Postnikov's hook length formula

In his study of the permutohedron, Postnikov's introduced the following formula

$$\sum_{\substack{\text{plane binary trees}\\ \text{of order } n}} \prod_v \left(1 + \frac{1}{h_v}\right) = (n+1)^{n-1}\frac{2^n}{n!}$$

where the **hook length** $h_v$ is the number of vertices below $v$ ($v$ included).

# Postnikov's hook length formula

In his study of the permutohedron, Postnikov's introduced the following formula

$$\sum_{\substack{\text{plane binary trees} \\ \text{of order } n}} \prod_{v} \left(1 + \frac{1}{h_v}\right) = (n+1)^{n-1} \frac{2^n}{n!}$$

where the **hook length** $h_v$ is the number of vertices below $v$ ($v$ included).

Since $\prod_v \frac{1}{h_v} = \frac{1}{t!}$, this correspond to a **geometric series** evaluated at 1 for the functional operator

$$X[f](s) = sf^2(s) + \int_0^s ds' f^2(s').$$

The associated **differential equation** is solved by $f(s)$ satisfying $f(s) = \mathrm{e}^{2sf(s)}$ whose power expansion is computed using the Lagrange inversion formula.

# Postnikov's hook length formula

In his study of the permutohedron, Postnikov's introduced the following formula

$$\sum_{\substack{\text{plane binary trees} \\ \text{of order } n}} \prod_v \left(1 + \frac{1}{h_v}\right) = (n+1)^{n-1} \frac{2^n}{n!}$$

where the **hook length** $h_v$ is the number of vertices below $v$ ($v$ included).

Since $\prod_v \frac{1}{h_v} = \frac{1}{t!}$, this correspond to a **geometric series** evaluated at 1 for the functional operator

$$X[f](s) = sf^2(s) + \int_0^s ds' f^2(s').$$

The associated **differential equation** is solved by $f(s)$ satisfying $f(s) = \mathrm{e}^{2sf(s)}$ whose power expansion is computed using the Lagrange inversion formula.

Combinatorial intepretation:

$$\frac{1}{|t|!} \prod_v \left(1 + \frac{1}{h_v}\right) = N(t, |t|) \prod_v (1 + h_v)$$
$$= \#\{\text{tree ordered lists of paths from vertices to leaves}\}$$

# Wilsonian effective action

# Wilsonian effective action

Path integral in quantum field theory:

$$\mathcal{Z} = \int [D\phi]\, \mathrm{e}^{-\frac{1}{2}\chi \cdot A_{\Lambda_0}^{-1} \cdot \chi + S_0[\phi]}$$

$A_{\Lambda_0}^{-1}$ implements the integration over field whose Fourier transform vanish for momenta above the cut-off $\Lambda_0$ and $S_0$ is the bare action.

# Wilsonian effective action

Path integral in quantum field theory:

$$\mathcal{Z} = \int [D\phi]\, \mathrm{e}^{-\frac{1}{2}\chi\cdot A_{\Lambda_0}^{-1}\cdot\chi + S_0[\phi]}$$

$A_{\Lambda_0}^{-1}$ implements the integration over field whose Fourier transform vanish for momenta above the cut-off $\Lambda_0$ and $S_0$ is the bare action.

Wilsonian renormalization:

Introduce a lower cut-off $\Lambda$ and first integrate over fields with momenta between $\Lambda$ and $\Lambda_0$ to obtain an **effective action** $S_\Lambda$

$$\mathcal{Z} = \int [D\phi]\, \mathrm{e}^{-\frac{1}{2}\phi\cdot A_{\Lambda}^{-1}\cdot\phi + S_\Lambda[\phi]} \quad \text{with} \quad S_\Lambda[\phi] = \log \int [D\chi]\, \mathrm{e}^{-\frac{1}{2}\chi\cdot A_{\Lambda,\Lambda_0}^{-1}\cdot\chi + S_\Lambda[\phi+\chi]}$$

# Wilsonian effective action

Path integral in quantum field theory:

$$\mathcal{Z} = \int [D\phi]\, \mathrm{e}^{-\frac{1}{2}\chi \cdot A_{\Lambda_0}^{-1} \cdot \chi + S_0[\phi]}$$

$A_{\Lambda_0}^{-1}$ implements the integration over field whose Fourier transform vanish for momenta above the cut-off $\Lambda_0$ and $S_0$ is the bare action.

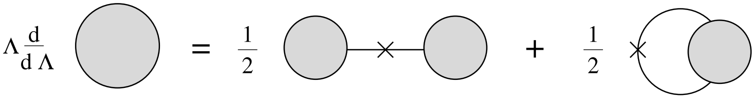
Wilsonian renormalization:

Introduce a lower cut-off $\Lambda$ and first integrate over fields with momenta between $\Lambda$ and $\Lambda_0$ to obtain an **effective action** $S_\Lambda$

$$\mathcal{Z} = \int [D\phi]\, \mathrm{e}^{-\frac{1}{2}\phi \cdot A_{\Lambda}^{-1} \cdot \phi + S_\Lambda[\phi]} \quad \text{with} \quad S_\Lambda[\phi] = \log \int [D\chi]\, \mathrm{e}^{-\frac{1}{2}\chi \cdot A_{\Lambda,\Lambda_0}^{-1} \cdot \chi + S_\Lambda[\phi+\chi]}$$

Polchinski's equation:

**Differential equation** for $S_\Lambda$ that generate **Feynman diagrams**



# Feynman diagram expansion

# Feynman diagram expansion

**Feynman rules** for the perturbed Gaußian integral

$$S'[\phi] = \log\left\{ \int [D\chi]\, \mathrm{e}^{-\frac{1}{2}\chi\cdot A^{-1}\cdot\chi}\ \mathrm{e}^{S[\phi+\chi]} \right\} = \sum_{\gamma \text{ connected diagram}} \frac{A^{\gamma}(S)}{\mathrm{S}_{\gamma}}[\phi]$$

# Feynman diagram expansion

**Feynman rules** for the perturbed Gaußian integral

$$S'[\phi] = \log\left\{ \int [D\chi]\, \mathrm{e}^{-\frac{1}{2}\chi\cdot A^{-1}\cdot\chi}\ \mathrm{e}^{S[\phi+\chi]} \right\} = \sum_{\gamma \text{ connected diagram}} \frac{A^{\gamma}(S)}{\mathrm{S}_{\gamma}}[\phi]$$

- associate a matrix element $A_{ij}$ to **edges**

# Feynman diagram expansion

**Feynman rules** for the perturbed Gaußian integral

$$S'[\phi] = \log\left\{ \int [D\chi]\, \mathrm{e}^{-\frac{1}{2}\chi\cdot A^{-1}\cdot\chi}\ \mathrm{e}^{S[\phi+\chi]} \right\} = \sum_{\gamma \text{ connected diagram}} \frac{A^{\gamma}(S)}{\mathrm{S}_{\gamma}}[\phi]$$

- associate a matrix element $A_{ij}$ to **edges**
- associate $\frac{\partial^n S}{\partial\phi_{j_1}\cdots\partial\phi_{j_n}}(\phi)$ to **vertices** of valence $n$

# Feynman diagram expansion

**Feynman rules** for the perturbed Gaußian integral

$$S'[\phi] = \log\left\{ \int [D\chi]\, \mathrm{e}^{-\frac{1}{2}\chi\cdot A^{-1}\cdot\chi}\ \mathrm{e}^{S[\phi+\chi]} \right\} = \sum_{\gamma \text{ connected diagram}} \frac{A^{\gamma}(S)}{\mathrm{S}_{\gamma}}[\phi]$$

- associate a matrix element $A_{ij}$ to **edges**
- associate $\frac{\partial^n S}{\partial\phi_{j_1}\cdots\partial\phi_{j_n}}(\phi)$ to **vertices** of valence $n$
- sum over all free indices

# Feynman diagram expansion

**Feynman rules** for the perturbed Gaußian integral

$$S'[\phi] = \log \left\{ \int [D\chi] \, \mathrm{e}^{-\frac{1}{2}\chi \cdot A^{-1} \cdot \chi} \; \mathrm{e}^{S[\phi+\chi]} \right\} = \sum_{\gamma \text{ connected diagram}} \frac{A^{\gamma}(S)}{\mathrm{S}_{\gamma}}[\phi]$$

- associate a matrix element $A_{ij}$ to **edges**
- associate $\frac{\partial^n S}{\partial \phi_{j_1} \cdots \partial \phi_{j_n}}(\phi)$ to **vertices** of valence $n$
- sum over all free indices
- divide by the symmetry factor $\mathrm{S}_{\gamma}$

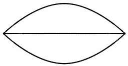
# Feynman diagram expansion

**Feynman rules** for the perturbed Gaußian integral

$$S'[\phi] = \log \left\{ \int [D\chi] \, \mathrm{e}^{-\frac{1}{2}\chi \cdot A^{-1} \cdot \chi} \; \mathrm{e}^{S[\phi+\chi]} \right\} = \sum_{\gamma \text{ connected diagram}} \frac{A^{\gamma}(S)}{\mathrm{S}_{\gamma}}[\phi]$$

- associate a matrix element $A_{ij}$ to **edges**
- associate $\frac{\partial^n S}{\partial \phi_{j_1} \cdots \partial \phi_{j_n}}(\phi)$ to **vertices** of valence $n$
- sum over all free indices
- divide by the symmetry factor $\mathrm{S}_{\gamma}$

A simple example:



$$\frac{1}{12} \sum_{\substack{i_1, i_2, i_3 \\ j_1, j_2, j_3}} \frac{\partial^3 S}{\partial \phi_{i_1} \partial \phi_{i_2} \partial \phi_{i_3}}[\phi] \, A_{i_1, j_1} A_{i_2, j_2} A_{i_3, j_3} \, \frac{\partial^3 S}{\partial \phi_{j_1} \partial \phi_{j_2} \partial \phi_{j_3}}[\phi]$$

# Hopf algebra of connected diagrams

# Hopf algebra of connected diagrams

$\mathcal{H}_F$ free commutative algebra generated by all **connected Feynman diagrams** with vertices of arbirary valence and coproduct

$$\Delta(\Gamma) = \Gamma \otimes 1 + 1 \otimes \Gamma + \sum_{\gamma_i \cap \gamma_j = \emptyset} \gamma_1 \cdots \gamma_n \otimes \Gamma/(\gamma_1 \cdots \gamma_n),$$

# Hopf algebra of connected diagrams

$\mathcal{H}_F$ free commutative algebra generated by all **connected Feynman diagrams** with vertices of arbirary valence and coproduct

$$\Delta(\Gamma) = \Gamma \otimes 1 + 1 \otimes \Gamma + \sum_{\gamma_i \cap \gamma_j = \emptyset} \gamma_1 \cdots \gamma_n \otimes \Gamma/(\gamma_1 \cdots \gamma_n),$$

- sum over all possible mutually **disjoint connected subdiagrams**,

# Hopf algebra of connected diagrams

$\mathcal{H}_F$ free commutative algebra generated by all **connected Feynman diagrams** with vertices of arbirary valence and coproduct

$$\Delta(\Gamma) = \Gamma \otimes 1 + 1 \otimes \Gamma + \sum_{\gamma_i \cap \gamma_j = \emptyset} \gamma_1 \cdots \gamma_n \otimes \Gamma/(\gamma_1 \cdots \gamma_n),$$

- sum over all possible mutually **disjoint connected subdiagrams**,
- **subdiagram** = subset lines and vertices related to them.

# Hopf algebra of connected diagrams

$\mathcal{H}_F$ free commutative algebra generated by all **connected Feynman diagrams** with vertices of arbirary valence and coproduct

$$\Delta(\Gamma) = \Gamma \otimes 1 + 1 \otimes \Gamma + \sum_{\gamma_i \cap \gamma_j = \emptyset} \gamma_1 \cdots \gamma_n \otimes \Gamma/(\gamma_1 \cdots \gamma_n),$$

- sum over all possible mutually **disjoint connected subdiagrams**,
- **subdiagram** = subset lines and vertices related to them.
- **reduced diagram** $\Gamma/(\gamma_1 \cdots \gamma_n)$ obtained by shrinking all $\gamma_i$ to single vertices in $\Gamma$.

# Hopf algebra of connected diagrams

$\mathcal{H}_F$ free commutative algebra generated by all **connected Feynman diagrams** with vertices of arbirary valence and coproduct

$$\Delta(\Gamma) = \Gamma \otimes 1 + 1 \otimes \Gamma + \sum_{\gamma_i \cap \gamma_j = \emptyset} \gamma_1 \cdots \gamma_n \otimes \Gamma/(\gamma_1 \cdots \gamma_n),$$

- sum over all possible mutually **disjoint connected subdiagrams**,
- **subdiagram** = subset lines and vertices related to them.
- **reduced diagram** $\Gamma/(\gamma_1 \cdots \gamma_n)$ obtained by shrinking all $\gamma_i$ to single vertices in $\Gamma$.

$\mathcal{H}_T$ is bigraded by the number of internal lines (edges) and loops (cycles).

# Hopf algebra of connected diagrams

$\mathcal{H}_F$ free commutative algebra generated by all **connected Feynman diagrams** with vertices of arbirary valence and coproduct

$$\Delta(\Gamma) = \Gamma \otimes 1 + 1 \otimes \Gamma + \sum_{\gamma_i \cap \gamma_j = \emptyset} \gamma_1 \cdots \gamma_n \otimes \Gamma/(\gamma_1 \cdots \gamma_n),$$

- sum over all possible mutually **disjoint connected subdiagrams**,
- **subdiagram** = subset lines and vertices related to them.
- **reduced diagram** $\Gamma/(\gamma_1 \cdots \gamma_n)$ obtained by shrinking all $\gamma_i$ to single vertices in $\Gamma$.

$\mathcal{H}_T$ is bigraded by the number of internal lines (edges) and loops (cycles).

Any $\alpha \in G_T$ define the weights of the diagram in the expansion

$$\mathcal{S}' = \Psi_\alpha(\mathcal{S}) = \sum_{\gamma} \alpha(\gamma) \frac{A^\gamma(\mathcal{S})}{\mathsf{S}_\gamma}$$

# Hopf algebra of connected diagrams

$\mathcal{H}_F$ free commutative algebra generated by all **connected Feynman diagrams** with vertices of arbirary valence and coproduct

$$\Delta(\Gamma) = \Gamma \otimes 1 + 1 \otimes \Gamma + \sum_{\gamma_i \cap \gamma_j = \emptyset} \gamma_1 \cdots \gamma_n \otimes \Gamma/(\gamma_1 \cdots \gamma_n),$$

- sum over all possible mutually **disjoint connected subdiagrams**,
- **subdiagram** = subset lines and vertices related to them.
- **reduced diagram** $\Gamma/(\gamma_1 \cdots \gamma_n)$ obtained by shrinking all $\gamma_i$ to single vertices in $\Gamma$.

$\mathcal{H}_T$ is bigraded by the number of internal lines (edges) and loops (cycles).

Any $\alpha \in G_T$ define the weights of the diagram in the expansion

$$\mathcal{S}' = \Psi_\alpha(\mathcal{S}) = \sum_\gamma \alpha(\gamma) \frac{A^\gamma(\mathcal{S})}{\mathsf{S}_\gamma}$$

Composition law analogous to B-series:

$$\underbrace{\Psi_\alpha \circ \Psi_\beta}_{\text{composition}} = \underbrace{\Psi_{\beta * \alpha}}_{\text{convolution}} \qquad \alpha, \beta \in G_F.$$

# Ordered diagrams

# Ordered diagrams

Ordered diagram= diagram with a hierarchy of boxes drawn on the subdiagrams s.t.

# Ordered diagrams

Ordered diagram= diagram with a hierarchy of boxes drawn on the subdiagrams s.t.

- boxes are either nested or disjoint,

# Ordered diagrams

Ordered diagram= diagram with a hierarchy of boxes drawn on the subdiagrams s.t.

- boxes are either nested or disjoint,
- boxes differ from their followers by one internal line.

# Ordered diagrams

Ordered diagram= diagram with a hierarchy of boxes drawn on the subdiagrams s.t.

- boxes are either nested or disjoint,
- boxes differ from their followers by one internal line.

Ordered diagram generate a commutative Hopf algebra $\mathcal{H}_F^{\text{ord}}$ with coproduct similar to $\mathcal{H}_F$ except that it has to preserve the box structure.

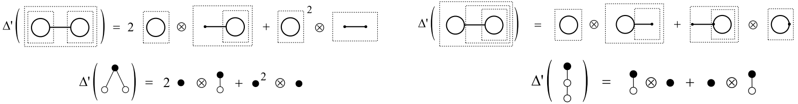
# Ordered diagrams

Ordered diagram= diagram with a hierarchy of boxes drawn on the subdiagrams s.t.

- boxes are either nested or disjoint,
- boxes differ from their followers by one internal line.

Ordered diagram generate a commutative Hopf algebra $\mathcal{H}_F^{\text{ord}}$ with coproduct similar to $\mathcal{H}_F$ except that it has to preserve the box structure.

Examples (with $\Delta'$ the non trivial part of $\Delta$):



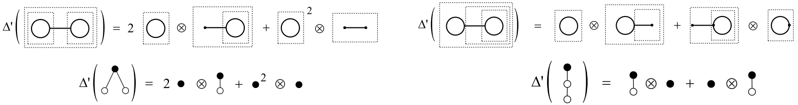
# Ordered diagrams

Ordered diagram= diagram with a hierarchy of boxes drawn on the subdiagrams s.t.

- boxes are either nested or disjoint,
- boxes differ from their followers by one internal line.

Ordered diagram generate a commutative Hopf algebra $\mathcal{H}_F^{\text{ord}}$ with coproduct similar to $\mathcal{H}_F$ except that it has to preserve the box structure.

Examples (with $\Delta'$ the non trivial part of $\Delta$):



Hopf algebra morphisms:

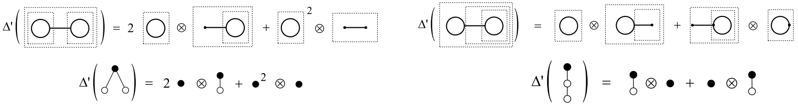
# Ordered diagrams

Ordered diagram= diagram with a hierarchy of boxes drawn on the subdiagrams s.t.

- boxes are either nested or disjoint,
- boxes differ from their followers by one internal line.

Ordered diagram generate a commutative Hopf algebra $\mathcal{H}_F^{\text{ord}}$ with coproduct similar to $\mathcal{H}_F$ except that it has to preserve the box structure.

Examples (with $\Delta'$ the non trivial part of $\Delta$):



Hopf algebra morphisms:

- ordered diagrams $\longrightarrow$ trees: $\gamma^{\text{ord}} \longrightarrow t$,

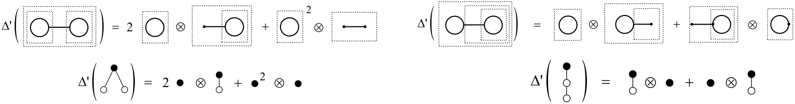
# Ordered diagrams

Ordered diagram= diagram with a hierarchy of boxes drawn on the subdiagrams s.t.

- boxes are either nested or disjoint,
- boxes differ from their followers by one internal line.

Ordered diagram generate a commutative Hopf algebra $\mathcal{H}_F^{\text{ord}}$ with coproduct similar to $\mathcal{H}_F$ except that it has to preserve the box structure.

Examples (with $\Delta'$ the non trivial part of $\Delta$):



Hopf algebra morphisms:

- ordered diagrams $\longrightarrow$ trees: $\gamma^{\text{ord}} \longrightarrow t$,
- diagrams $\rightarrow$ ordered diagrams: $\frac{\gamma}{S_\gamma} \longrightarrow \sum_{\text{orders on } \gamma} \frac{\gamma^{\text{ord}}}{S_\gamma^{\text{ord}}}$.

# Deletion/contraction of edges

# Deletion/contraction of edges

In the Lie algebra of $\mathcal{G}_T$ define the infinitesimal characters such that $\delta_{\text{tree}}($$) = 1$ and $\delta_{\text{loop}}($$) = 1$ and vanish otherwise.

# Deletion/contraction of edges

In the Lie algebra of $\mathcal{G}_T$ define the infinitesimal characters such that $\delta_{\mathrm{tree}}($$) = 1$ and $\delta_{\mathrm{loop}}($$) = 1$ and vanish otherwise.

$\alpha = \exp_* \{sa\,\delta_{\mathrm{tree}} + sb\,\delta_{\mathrm{loop}}\}(\gamma) = s^{I_\gamma}\, a^{I_\gamma - L_\gamma}\, b^{L_\gamma}$ obeys the differential equation

$$\frac{d\alpha}{ds} = (a\delta_{\mathrm{tree}} * \alpha + b\delta_{\mathrm{loop}}) * \alpha$$

# Deletion/contraction of edges

In the Lie algebra of $\mathcal{G}_T$ define the infinitesimal characters such that $\delta_{\mathrm{tree}}(\bullet\!\!-\!\!\bullet) = 1$ and $\delta_{\mathrm{loop}}(\bullet\!\!\bigcirc) = 1$ and vanish otherwise.

$\alpha = \exp_* \{sa\, \delta_{\mathrm{tree}} + sb\, \delta_{\mathrm{loop}}\} (\gamma) = s^{I_\gamma}\, a^{I_\gamma - L_\gamma}\, b^{L_\gamma}$ obeys the differential equation

$$\frac{d\alpha}{ds} = (a\delta_{\mathrm{tree}} * \alpha + b\delta_{\mathrm{loop}}) * \alpha$$

Any $\delta \in \mathcal{G}_T$ defines two derivations $f \lhd \delta = (\delta \otimes \mathrm{Id}) \circ \Delta(f)$ and $\delta \rhd f = (\mathrm{Id} \otimes \delta) \circ \Delta(f)$.

# Deletion/contraction of edges

In the Lie algebra of $\mathcal{G}_T$ define the infinitesimal characters such that $\delta_{\mathrm{tree}}(\bullet\!\!-\!\!\bullet) = 1$ and $\delta_{\mathrm{loop}}(\bullet\!\!\!\bigcirc) = 1$ and vanish otherwise.

$\alpha = \exp_* \left\{ sa\, \delta_{\mathrm{tree}} + sb\, \delta_{\mathrm{loop}} \right\}(\gamma) = s^{I_\gamma}\, a^{I_\gamma - L_\gamma}\, b^{L_\gamma}$ obeys the differential equation

$$\frac{d\alpha}{ds} = (a\delta_{\mathrm{tree}} * \alpha + b\delta_{\mathrm{loop}}) * \alpha$$

Any $\delta \in \mathcal{G}_T$ defines two derivations $f \triangleleft \delta = (\delta \otimes \mathrm{Id}) \circ \Delta(f)$ and $\delta \triangleright f = (\mathrm{Id} \otimes \delta) \circ \Delta(f)$.

Deletion/contraction interpretation:

# Deletion/contraction of edges

In the Lie algebra of $\mathcal{G}_T$ define the infinitesimal characters such that $\delta_{\text{tree}}(\bullet\!\!-\!\!\bullet) = 1$ and $\delta_{\text{loop}}(\bullet\!\!\bigcirc) = 1$ and vanish otherwise.

$\alpha = \exp_* \{ sa\, \delta_{\text{tree}} + sb\, \delta_{\text{loop}} \} (\gamma) = s^{I_\gamma} a^{I_\gamma - L_\gamma} b^{L_\gamma}$ obeys the differential equation

$$\frac{d\alpha}{ds} = (a\delta_{\text{tree}} * \alpha + b\delta_{\text{loop}}) * \alpha$$

Any $\delta \in \mathcal{G}_T$ defines two derivations $f \lhd \delta = (\delta \otimes \mathrm{Id}) \circ \Delta(f)$ and $\delta \rhd f = (\mathrm{Id} \otimes \delta) \circ \Delta(f)$.

Deletion/contraction interpretation:

- $\delta_{\text{tree}} \rhd \gamma$ is a sum over all the diagrams obtained from $\gamma$ by cutting a bridge.

# Deletion/contraction of edges

In the Lie algebra of $\mathcal{G}_T$ define the infinitesimal characters such that $\delta_{\mathrm{tree}}($$) = 1$ and $\delta_{\mathrm{loop}}($$) = 1$ and vanish otherwise.

$\alpha = \exp_* \{sa\, \delta_{\mathrm{tree}} + sb\, \delta_{\mathrm{loop}}\}(\gamma) = s^{I_\gamma} a^{I_\gamma - L_\gamma} b^{L_\gamma}$ obeys the differential equation

$$\frac{d\alpha}{ds} = (a\delta_{\mathrm{tree}} * \alpha + b\delta_{\mathrm{loop}}) * \alpha$$

Any $\delta \in \mathcal{G}_T$ defines two derivations $f \triangleleft \delta = (\delta \otimes \mathrm{Id}) \circ \Delta(f)$ and $\delta \triangleright f = (\mathrm{Id} \otimes \delta) \circ \Delta(f)$.

Deletion/contraction interpretation:

- $\delta_{\mathrm{tree}} \triangleright \gamma$ is a sum over all the diagrams obtained from $\gamma$ by cutting a bridge.
- $\delta_{\mathrm{loop}} \triangleright \gamma$ is a sum over all the diagrams obtained from $\gamma$ by cutting a line which is not a bridge.

# Deletion/contraction of edges

In the Lie algebra of $\mathcal{G}_T$ define the infinitesimal characters such that $\delta_{\text{tree}}(\bullet\!\!-\!\!\bullet) = 1$ and $\delta_{\text{loop}}(\bullet\!\!\!\bigcirc) = 1$ and vanish otherwise.

$\alpha = \exp_* \{sa\,\delta_{\text{tree}} + sb\,\delta_{\text{loop}}\}(\gamma) = s^{I_\gamma}\, a^{I_\gamma - L_\gamma}\, b^{L_\gamma}$ obeys the differential equation

$$\frac{d\alpha}{ds} = (a\delta_{\text{tree}} * \alpha + b\delta_{\text{loop}}) * \alpha$$

Any $\delta \in \mathcal{G}_T$ defines two derivations $f \lhd \delta = (\delta \otimes \mathrm{Id}) \circ \Delta(f)$ and $\delta \rhd f = (\mathrm{Id} \otimes \delta) \circ \Delta(f)$.

Deletion/contraction interpretation:

- $\delta_{\text{tree}} \rhd \gamma$ is a sum over all the diagrams obtained from $\gamma$ by cutting a bridge.
- $\delta_{\text{loop}} \rhd \gamma$ is a sum over all the diagrams obtained from $\gamma$ by cutting a line which is not a bridge.
- $\gamma \lhd \delta_{\text{loop}}$ is a sum over all the diagrams obtained from $\gamma$ by contracting a self-loop with one edge.

# Deletion/contraction of edges

In the Lie algebra of $\mathcal{G}_T$ define the infinitesimal characters such that $\delta_{\rm tree}($$) = 1$ and $\delta_{\rm loop}($$) = 1$ and vanish otherwise.

$\alpha = \exp_* \{sa\, \delta_{\rm tree} + sb\, \delta_{\rm loop}\}(\gamma) = s^{I_\gamma} a^{I_\gamma - L_\gamma} b^{L_\gamma}$ obeys the differential equation

$$\frac{d\alpha}{ds} = (a\delta_{\rm tree} * \alpha + b\delta_{\rm loop}) * \alpha$$

Any $\delta \in \mathcal{G}_T$ defines two derivations $f \lhd \delta = (\delta \otimes \mathrm{Id}) \circ \Delta(f)$ and $\delta \rhd f = (\mathrm{Id} \otimes \delta) \circ \Delta(f)$.

Deletion/contraction interpretation:

- $\delta_{\rm tree} \rhd \gamma$ is a sum over all the diagrams obtained from $\gamma$ by cutting a bridge.
- $\delta_{\rm loop} \rhd \gamma$ is a sum over all the diagrams obtained from $\gamma$ by cutting a line which is not a bridge.
- $\gamma \lhd \delta_{\rm loop}$ is a sum over all the diagrams obtained from $\gamma$ by contracting a self-loop with one edge.
- $\gamma \lhd \delta_{\rm tree}$ is a sum over all the diagrams obtained from $\gamma$ by contracting a line which is not a self-loop.

# The Tutte polynomial

# The Tutte polynomial

The **Tutte polynomial** is a two variable polynomial attached to graphs

$$P_\gamma(x, y) = \sum_{A \subset E} (y-1)^{n(A)} (x-1)^{r(E)-r(A)},$$

where the sum runs over all subsets of the set of edges $E$ of $\gamma$. In the QFT language, the nullity and the rank of a connected diagram can be expressed in terms of the number of internal lines and loops $n(\gamma) = I_\gamma - L_\gamma$ and $r(\gamma) = L_\gamma$.

# The Tutte polynomial

The **Tutte polynomial** is a two variable polynomial attached to graphs

$$P_\gamma(x,y) = \sum_{A\subset E} (y-1)^{n(A)}(x-1)^{r(E)-r(A)},$$

where the sum runs over all subsets of the set of edges $E$ of $\gamma$. In the QFT language, the nullity and the rank of a connected diagram can be expressed in terms of the number of internal lines and loops $n(\gamma) = I_\gamma - L_\gamma$ and $r(\gamma) = L_\gamma$.

Therefore, it can be expressed as the evaluation at $s = 1$ of the character

$$\alpha = \exp_* s\left\{\delta_{\rm tree} + (y-1)\,\delta_{\rm loop}\right\} * \exp_* s\left\{(x-1)\,\delta_{\rm tree} + \delta_{\rm loop}\right\},$$

solution of the **differential equation** with boundary condition $\alpha(0) = \epsilon$,

$$\frac{d\alpha}{ds} = x\,\alpha * \delta_{\rm tree} + y\,\delta_{\rm loop} * \alpha + [\delta_{\rm tree}, \alpha]_* - [\delta_{\rm loop}, \alpha]_*\,.$$

# Universality of the Tutte polynomial

# Universality of the Tutte polynomial

Consider a four variable graph polynomial $Q_\gamma(x, y, a, b)$ that is multiplicative on disjoint unions and one vertex unions and obeys **the following contraction/deletion** rules

# Universality of the Tutte polynomial

Consider a four variable graph polynomial $Q_\gamma(x, y, a, b)$ that is multiplicative on disjoint unions and one vertex unions and obeys **the following contraction/deletion** rules

- $Q_\gamma(x, y, a, b) = x\, Q_{\gamma-e}(x, y, a, b)$ if $e$ is a bridge

# Universality of the Tutte polynomial

Consider a four variable graph polynomial $Q_\gamma(x, y, a, b)$ that is multiplicative on disjoint unions and one vertex unions and obeys **the following contraction/deletion** rules

- $Q_\gamma(x, y, a, b) = x\, Q_{\gamma - e}(x, y, a, b)$ if $e$ is a bridge
- $Q_\gamma(x, y, a, b) = y\, Q_{\gamma / e}(x, y, a, b)$ if $e$ is a self-loop

# Universality of the Tutte polynomial

Consider a four variable graph polynomial $Q_\gamma(x, y, a, b)$ that is multiplicative on disjoint unions and one vertex unions and obeys **the following contraction/deletion** rules

- $Q_\gamma(x, y, a, b) = x\, Q_{\gamma - e}(x, y, a, b)$ if $e$ is a bridge
- $Q_\gamma(x, y, a, b) = y\, Q_{\gamma / e}(x, y, a, b)$ if $e$ is a self-loop
- $Q_\gamma(x, y, a, b) = a\, Q_{\gamma - e}(x, y, a, b, ) + b\, Q_{\gamma / e}(x, y, a, b)$
  if $e$ is neither a bridge nor a self-loop

# Universality of the Tutte polynomial

Consider a four variable graph polynomial $Q_\gamma(x, y, a, b)$ that is multiplicative on disjoint unions and one vertex unions and obeys **the following contraction/deletion** rules

- $Q_\gamma(x, y, a, b) = x\, Q_{\gamma-e}(x, y, a, b)$ if $e$ is a bridge
- $Q_\gamma(x, y, a, b) = y\, Q_{\gamma/e}(x, y, a, b)$ if $e$ is a self-loop
- $Q_\gamma(x, y, a, b) = a\, Q_{\gamma-e}(x, y, a, b, ) + b\, Q_{\gamma/e}(x, y, a, b)$ if $e$ is neither a bridge nor a self-loop

The character $\beta(\gamma) = s^{l_\gamma}\, Q_\gamma(x, y, a, b)$ obeys the **differential equation**

$$\frac{d\beta}{ds} = x\, \beta * \delta_{\text{tree}} + y\, \delta_{\text{loop}} * \beta + a\, [\delta_{\text{tree}}, \beta]_* - b\, [\delta_{\text{loop}}, \beta]_* \, .$$

# Universality of the Tutte polynomial

Consider a four variable graph polynomial $Q_\gamma(x, y, a, b)$ that is multiplicative on disjoint unions and one vertex unions and obeys **the following contraction/deletion** rules

- $Q_\gamma(x, y, a, b) = x\, Q_{\gamma-e}(x, y, a, b)$ if $e$ is a bridge
- $Q_\gamma(x, y, a, b) = y\, Q_{\gamma/e}(x, y, a, b)$ if $e$ is a self-loop
- $Q_\gamma(x, y, a, b) = a\, Q_{\gamma-e}(x, y, a, b, ) + b\, Q_{\gamma/e}(x, y, a, b)$ if $e$ is neither a bridge nor a self-loop

The character $\beta(\gamma) = s^{I_\gamma}\, Q_\gamma(x, y, a, b)$ obeys the **differential equation**

$$\frac{d\beta}{ds} = x\, \beta * \delta_{\rm tree} + y\, \delta_{\rm loop} * \beta + a\, [\delta_{\rm tree}, \beta]_* - b\, [\delta_{\rm loop}, \beta]_* \,.$$

Acting with the automorphism $\varphi_{a^{-1}, b^{-1}}(\gamma) = a^{-(I_\gamma - L_\gamma)}\, b^{-L_\gamma}\, \gamma$, we obtain the **Tutte polynomial differential equation** with modified parameters $\frac{x}{a}$ and $\frac{y}{b}$, so that

$$Q_\gamma(x, y, a, b) = a^{I_\gamma - L_\gamma}\, b^{L_\gamma}\, P_\gamma\left(\frac{x}{a}, \frac{y}{b}\right)$$

# Composition of effective actions and the Tutte polynomial

# Composition of effective actions and the Tutte polynomial

The Tutte polynomial is obtained by a composition of **effective action computations** starting with a universal action $S[\phi] = e^{\phi}$ that generate all diagrams.

# Composition of effective actions and the Tutte polynomial

The Tutte polynomial is obtained by a composition of **effective action computations** starting with a universal action $S[\phi] = \mathrm{e}^{\phi}$ that generate all diagrams.

- To generate $v^{L_\gamma}$, we weight loops by $h = v$ (with $v = y - 1$),

$$S'[\phi] = v \log \left\{ \int [D\chi] \, \mathrm{e}^{-\frac{1}{2v}\chi^2} \; \mathrm{e}^{\frac{S[\phi+\chi]}{v}} \right\}$$

# Composition of effective actions and the Tutte polynomial

The Tutte polynomial is obtained by a composition of **effective action computations** starting with a universal action $S[\phi] = \mathrm{e}^{\phi}$ that generate all diagrams.

- To generate $v^{L_\gamma}$, we weight loops by $h = v$ (with $v = y - 1$),

$$S'[\phi] = v \log \left\{ \int [D\chi] \, \mathrm{e}^{-\frac{1}{2v}\chi^2} \, \mathrm{e}^{\frac{S[\phi+\chi]}{v}} \right\}$$

- To generate $u^{I_\gamma - L_\gamma}$, we weight internal lines by $u$ (with $u = x - 1$) and loops by $\frac{1}{u}$, so that

$$S''[\psi] = \frac{1}{u} \log \left\{ \int [D\phi] \, \mathrm{e}^{-\frac{1}{2}\phi^2} \, \mathrm{e}^{uS[\psi+\phi]} \right\}.$$

# Composition of effective actions and the Tutte polynomial

The Tutte polynomial is obtained by a composition of **effective action computations** starting with a universal action $S[\phi] = \mathrm{e}^{\phi}$ that generate all diagrams.

- To generate $v^{L_\gamma}$, we weight loops by $h = v$ (with $v = y - 1$),

$$S'[\phi] = v \log \left\{ \int [D\chi] \, \mathrm{e}^{-\frac{1}{2v}\chi^2} \; \mathrm{e}^{\frac{S[\phi+\chi]}{v}} \right\}$$

- To generate $u^{I_\gamma - L_\gamma}$, we weight internal lines by $u$ (with $u = x - 1$) and loops by $\frac{1}{u}$, so that

$$S''[\psi] = \frac{1}{u} \log \left\{ \int [D\phi] \, \mathrm{e}^{-\frac{1}{2}\phi^2} \; \mathrm{e}^{uS[\psi+\phi]} \right\}.$$

- Then, we substitute $S'$ into the expression of $S''$ and evaluate at $\psi = 0$,

$$G_{\text{Tutte}}(u, v) = \frac{1}{u} \log I = \frac{1}{u} \log \left\{ \int [D\phi] \, \mathrm{e}^{-\frac{1}{2}\phi^2} \left( \int [D\chi] \, \mathrm{e}^{-\frac{1}{2v}\chi^2} \; \mathrm{e}^{\frac{S[\phi+\chi]}{v}} \right)^q \right\}$$

with $q = uv$.

# Generating function for the Tutte polynomial

# Generating function for the Tutte polynomial

When $q$ is an integer, we introduce $q$ independent fields $\chi_i$

$$I = \int [D\phi] \int \prod_{1 \leq i \leq q} [D\chi_i] \, \mathrm{e}^{-\frac{1}{2}\phi^2} \mathrm{e}^{-\frac{1}{2v}\sum_i (\chi_i)^2} \mathrm{e}^{\frac{1}{v}\sum_i S[\chi_i + \phi]}$$

# Generating function for the Tutte polynomial

When $q$ is an integer, we introduce $q$ independent fields $\chi_i$

$$I = \int [D\phi] \int \prod_{1 \leq i \leq q} [D\chi_i] \, \mathrm{e}^{-\frac{1}{2}\phi^2} \mathrm{e}^{-\frac{1}{2v}\sum_i (\chi_i)^2} \mathrm{e}^{\frac{1}{v}\sum_i S[\chi_i + \phi]}$$

It is convenient to trade $\chi_i$ for $\xi_i = \chi_i + \phi$ so that the integral over $\phi$ is Gaußian and can be performed

$$I = \int \prod_{1 \leq i \leq q} [D\xi_i] \, \mathrm{e}^{-\frac{1}{2v}\left\{\left(\sum_i (\xi_i)^2 - \frac{1}{v(1+u)}\left(\sum_i \xi_i\right)^2\right)\right\}} \times \mathrm{e}^{\frac{1}{v}\sum_i S[\xi_i]}.$$

# Generating function for the Tutte polynomial

When $q$ is an integer, we introduce $q$ independent fields $\chi_i$

$$I = \int [D\phi] \int \prod_{1 \leq i \leq q} [D\chi_i] \, \mathrm{e}^{-\frac{1}{2}\phi^2} \mathrm{e}^{-\frac{1}{2v}\sum_i (\chi_i)^2} \mathrm{e}^{\frac{1}{v}\sum_i S[\chi_i + \phi]}$$

It is convenient to trade $\chi_i$ for $\xi_i = \chi_i + \phi$ so that the integral over $\phi$ is Gaußian and can be performed

$$I = \int \prod_{1 \leq i \leq q} [D\xi_i] \, \mathrm{e}^{-\frac{1}{2v}\left\{\left(\sum_i (\xi_i)^2 - \frac{1}{v(1+u)}\left(\sum_i \xi_i\right)^2\right)\right\}} \times \mathrm{e}^{\frac{1}{v}\sum_i S[\xi_i]}.$$

By expanding the integral over a multiplet of fields $\xi = (\xi_i)$ using Feynman diagrams, we generate the Tutte polynomials,

$$G_{\text{Tutte}}(u, v) = \frac{1}{u} \log\left\{\int [D\xi] \, \mathrm{e}^{-\frac{1}{2}\xi \cdot A^{-1}\xi} \, \mathrm{e}^{V(\xi)}\right\}$$

with a $q \times q$ propagator $A = v + M$, where $M$ is the $q \times q$ matrix whose entries are all equal to 1, and an interaction $V(\xi) = \frac{1}{v}\sum_i S(\xi_i)$.

# Relation to the Potts model partition function

# Relation to the Potts model partition function

The **Feynman rules** for this expansion are as follows:

# Relation to the Potts model partition function

The **Feynman rules** for this expansion are as follows:

- each **vertex** contributes to a factor $\frac{1}{v}$ and is equipped with an index $i$ taking $q$ values, because of the $q$ independent fields in the interaction;

# Relation to the Potts model partition function

The **Feynman rules** for this expansion are as follows:

- each **vertex** contributes to a factor $\frac{1}{v}$ and is equipped with an index $i$ taking $q$ values, because of the $q$ independent fields in the interaction;
- each **line** contributes a factor of $1 + v$ if it connects vertices with the same index and a factor of 1 otherwise,

# Relation to the Potts model partition function

The **Feynman rules** for this expansion are as follows:

- each **vertex** contributes to a factor $\frac{1}{v}$ and is equipped with an index $i$ taking $q$ values, because of the $q$ independent fields in the interaction;
- each **line** contributes a factor of $1 + v$ if it connects vertices with the same index and a factor of 1 otherwise,
- sum over all the indices.

# Relation to the Potts model partition function

The **Feynman rules** for this expansion are as follows:

- each **vertex** contributes to a factor $\frac{1}{v}$ and is equipped with an index $i$ taking $q$ values, because of the $q$ independent fields in the interaction;
- each **line** contributes a factor of $1+v$ if it connects vertices with the same index and a factor of 1 otherwise,
- sum over all the indices.

The evaluation of a graph $\gamma$ is proportional to the $q$-state **Potts model** partition function on $\gamma$,

$$Z(\beta, J, \gamma) = \sum_{\sigma} \mathrm{e}^{-\beta H(\sigma)}$$

where the sum runs over all states and $\beta$ is such that $v = \mathrm{e}^{-\beta J} - 1$. A state $\sigma$ is a assigment of spin in $q$ element set to each vertex of the graph and the Hamiltonian is

$$H(\sigma) = -\#\left\{\text{edges joining identical spins}\right\}$$

# Tree level Feynman diagrams and Postnikov's formula

# Tree level Feynman diagrams and Postnikov's formula

By Cayley's formula, the number of **labelled non rooted trees** $\tau$ with $|\tau| = n$ edges is $(n+1)^{n-1}$. Accordingly,

$$\sum_{|\tau|=n} \frac{(2s)^n}{\mathrm{S}_\tau} = \frac{(n+1)^{n-1}(2s)^n}{(n+1)!}$$

# Tree level Feynman diagrams and Postnikov's formula

By Cayley's formula, the number of **labelled non rooted trees** $\tau$ with $|\tau| = n$ edges is $(n+1)^{n-1}$. Accordingly,

$$\sum_{|\tau|=n} \frac{(2s)^n}{\mathrm{S}_\tau} = \frac{(n+1)^{n-1}(2s)^n}{(n+1)!}$$

This sum is generated by the **tree level** part of an equation of the **Polchinski type** (note the absence of $\frac{1}{2}$ prefactor)

$$\frac{\partial S}{\partial s} = \left(\frac{\partial S}{\partial \phi}\right)^2 \qquad S_0[\phi] = \mathrm{e}^{\phi}$$

# Tree level Feynman diagrams and Postnikov's formula

By Cayley's formula, the number of **labelled non rooted trees** $\tau$ with $|\tau| = n$ edges is $(n+1)^{n-1}$. Accordingly,

$$\sum_{|\tau|=n} \frac{(2s)^n}{\mathrm{S}_\tau} = \frac{(n+1)^{n-1}(2s)^n}{(n+1)!}$$

This sum is generated by the **tree level** part of an equation of the **Polchinski type** (note the absence of $\frac{1}{2}$ prefactor)

$$\frac{\partial S}{\partial s} = \left(\frac{\partial S}{\partial \phi}\right)^2 \qquad S_0[\phi] = \mathrm{e}^{\phi}$$

This non linear equation can be solved using sum over **plane binary trees**:

# Tree level Feynman diagrams and Postnikov's formula

By Cayley's formula, the number of **labelled non rooted trees** $\tau$ with $|\tau| = n$ edges is $(n+1)^{n-1}$. Accordingly,

$$\sum_{|\tau|=n} \frac{(2s)^n}{\mathrm{S}_\tau} = \frac{(n+1)^{n-1}(2s)^n}{(n+1)!}$$

This sum is generated by the **tree level** part of an equation of the **Polchinski type** (note the absence of $\frac{1}{2}$ prefactor)

$$\frac{\partial S}{\partial s} = \left(\frac{\partial S}{\partial \phi}\right)^2 \qquad S_0[\phi] = \mathrm{e}^{\phi}$$

This non linear equation can be solved using sum over **plane binary trees**:

- $\frac{1}{t!} = \prod_v \frac{1}{h_v}$ is generated by the iterated integrals

# Tree level Feynman diagrams and Postnikov's formula

By Cayley's formula, the number of **labelled non rooted trees** $\tau$ with $|\tau| = n$ edges is $(n+1)^{n-1}$. Accordingly,

$$\sum_{|\tau|=n} \frac{(2s)^n}{\mathrm{S}_\tau} = \frac{(n+1)^{n-1}(2s)^n}{(n+1)!}$$

This sum is generated by the **tree level** part of an equation of the **Polchinski type** (note the absence of $\frac{1}{2}$ prefactor)

$$\frac{\partial S}{\partial s} = \left(\frac{\partial S}{\partial \phi}\right)^2 \qquad S_0[\phi] = \mathrm{e}^{\phi}$$

This non linear equation can be solved using sum over **plane binary trees**:

- $\frac{1}{t!} = \prod_v \frac{1}{h_v}$ is generated by the iterated integrals
- $\prod_{v \neq \mathrm{root}}(1 + h_v)$ is generated by iterations of $\frac{\partial}{\partial \phi}$

# Tree level Feynman diagrams and Postnikov's formula

By Cayley's formula, the number of **labelled non rooted trees** $\tau$ with $|\tau| = n$ edges is $(n+1)^{n-1}$. Accordingly,

$$\sum_{|\tau|=n} \frac{(2s)^n}{\mathrm{S}_\tau} = \frac{(n+1)^{n-1}(2s)^n}{(n+1)!}$$

This sum is generated by the **tree level** part of an equation of the **Polchinski type** (note the absence of $\frac{1}{2}$ prefactor)

$$\frac{\partial S}{\partial s} = \left(\frac{\partial S}{\partial \phi}\right)^2 \qquad S_0[\phi] = \mathrm{e}^{\phi}$$

This non linear equation can be solved using sum over **plane binary trees**:

- $\frac{1}{t!} = \prod_v \frac{1}{h_v}$ is generated by the iterated integrals
- $\prod_{v \neq \mathrm{root}}(1 + h_v)$ is generated by iterations of $\frac{\partial}{\partial \phi}$

We recover Postnikov's formula by evaluating $\frac{\partial S}{\partial \phi}$ at $s = 1$ and $\phi = 0$.

# Feynman graphs and their Symanzik polynomials

In **quantum field theory**, a Feynman diagram $\gamma$ with $n$ edges can be evaluated, in dimension $D$ as

$$\int \frac{d^n\alpha}{(U_\gamma(\alpha))^{\frac{D}{2}}}\, \mathrm{e}^{-\frac{V_\gamma(\alpha,p)}{U_\gamma(\alpha)}}$$

where $U_\gamma(\alpha)$ is the (first) **Symanzik polynomial**

$$U_\gamma(\alpha) = \sum_{\substack{t \\ \text{spanning trees}}} \prod_{i \notin t} \alpha_i$$

and $\alpha_i$ are variables associated to the edges.

# Feynman graphs and their Symanzik polynomials

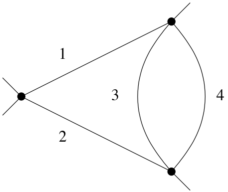
In **quantum field theory**, a Feynman diagram $\gamma$ with $n$ edges can be evaluated, in dimension $D$ as

$$\int \frac{d^n\alpha}{(U_\gamma(\alpha))^{\frac{D}{2}}}\, \mathrm{e}^{-\frac{V_\gamma(\alpha,p)}{U_\gamma(\alpha)}}$$

where $U_\gamma(\alpha)$ is the (first) **Symanzik polynomial**

$$U_\gamma(\alpha) = \sum_{\substack{t \\ \text{spanning trees}}} \prod_{i \notin t} \alpha_i$$

and $\alpha_i$ are variables associated to the edges. For example,




yields $U_\gamma(\alpha) = \alpha_1\alpha_3 + \alpha_1\alpha_4 + \alpha_2\alpha_3 + \alpha_2\alpha_4 + \alpha_3\alpha_4$.

# Loop decomposition of the Symanzik polynomial

# Loop decomposition of the Symanzik polynomial

In the Hopf algebra of Feynman diagrams with **labelled edges** consider the infinitesimal characters

$\delta_{\mathrm{tree}}($  $) = 1 \qquad \text{and} \qquad \delta_{\mathrm{loop}}($  $) = \alpha_i$

and vanish otherwise.

# Loop decomposition of the Symanzik polynomial

In the Hopf algebra of Feynman diagrams with **labelled edges** consider the infinitesimal characters

$$\delta_{\text{tree}}(\bullet\!\!-\!\!\bullet) = 1 \qquad \text{and} \qquad \delta_{\text{loop}}(\bullet\!\!\!\bigcirc) = \alpha_i$$

and vanish otherwise.

$$\begin{aligned} U_\gamma(\alpha) &= \mathrm{e}^{\delta_{\text{tree}}} * \mathrm{e}^{\delta_{\text{loop}}} \\ &= \mathrm{e}^{\delta_{\text{tree}}} * \mathrm{e}^{\delta_{\text{loop}}} * \mathrm{e}^{-\delta_{\text{tree}}} * \mathrm{e}^{\delta_{\text{tree}}} \\ &= \mathrm{e}^{\sum_n \delta_{n\,\text{loop}}} * \mathrm{e}^{\delta_{\text{tree}}} \end{aligned}$$

since, in the Lie algebra of Feynman diagrams,

$$\frac{1}{n!} \underbrace{\Big[\delta_{\text{tree}}, \big[\cdots[\delta_{\text{tree}}, \delta_{\text{loop}}]\cdots\big]\Big]}_{n\,\text{iterations}} = \delta_{n\,\text{loop}}$$

with $\delta_{n\,\text{loop}}$ taking the value $\sum_i \alpha_i$ on the one loop diagram with $n$ edges and vanishes otherwise. Thus, $U_\gamma(\alpha)$ can be evaluated by summing over all **contraction schemes of the loops.**

# Loop decomposition of the Symanzik polynomial

In the Hopf algebra of Feynman diagrams with **labelled edges** consider the infinitesimal characters

$\delta_{\text{tree}}($  $) = 1$ and $\delta_{\text{loop}}($  $) = \alpha_i$

and vanish otherwise.

$$\begin{aligned} U_\gamma(\alpha) &= \mathrm{e}^{\delta_{\text{tree}}} * \mathrm{e}^{\delta_{\text{loop}}} \\ &= \mathrm{e}^{\delta_{\text{tree}}} * \mathrm{e}^{\delta_{\text{loop}}} * \mathrm{e}^{-\delta_{\text{tree}}} * \mathrm{e}^{\delta_{\text{tree}}} \\ &= \mathrm{e}^{\sum_n \delta_{n\,\text{loop}}} * \mathrm{e}^{\delta_{\text{tree}}} \end{aligned}$$

since, in the Lie algebra of Feynman diagrams,

$$\frac{1}{n!} \underbrace{\Big[\delta_{\text{tree}}, \big[\cdots[\delta_{\text{tree}}, \delta_{\text{loop}}]\cdots\big]\Big]}_{n\,\text{iterations}} = \delta_{n\,\text{loop}}$$

with $\delta_{n\,\text{loop}}$ taking the value $\sum_i \alpha_i$ on the one loop diagram with $n$ edges and vanishes otherwise. Thus, $U_\gamma(\alpha)$ can be evaluated by summing over all **contraction schemes of the loops.** For example,

$U$  $(\alpha) = \frac{1}{2}\big\{(\alpha_1+\alpha_2+\alpha_3)\alpha_4 + (\alpha_1+\alpha_2+\alpha_4)\alpha_3 + (\alpha_1+\alpha_2)(\alpha_3+\alpha_4)\big\}$

# Conclusion

# Conclusion

Common framework for perturbative resolution of **non linear equations** and **effective actions** based on Hopf algebras of **rooted trees** and **Feynman diagrams**.

# Conclusion

Common framework for perturbative resolution of **non linear equations** and **effective actions** based on Hopf algebras of **rooted trees** and **Feynman diagrams**.

| rooted trees | Feynman diagrams |
|---|---|
| non linear analysis | perturbative path integrals |
| fixed point equation | renormalization group equation |
| Powers of non linear operators $X^t(x)$ | background field technique $A^\gamma(S)$ |
| $x' = (\mathrm{id} - X)(x)$ <br> $= \sum_t \frac{X^t}{\mathrm{S}_t}$ | $S'[\phi] = \log \int [D\chi] \mathrm{e}^{-\frac{1}{2}\chi\cdot A\cdot\chi + S[\phi+\chi]}$ <br> $= \sum_\gamma \frac{A^\gamma(S)}{\mathrm{S}_\gamma}[\phi]$ |
| composition | successive integrations |

# Conclusion

Common framework for perturbative resolution of **non linear equations** and **effective actions** based on Hopf algebras of **rooted trees** and **Feynman diagrams**.

| rooted trees | Feynman diagrams |
|---|---|
| non linear analysis | perturbative path integrals |
| fixed point equation | renormalization group equation |
| Powers of non linear operators $X^t(x)$ | background field technique $A^\gamma(S)$ |
| $x' = (\mathrm{id} - X)(x)$ <br> $= \sum_t \frac{X^t}{\mathrm{S}_t}$ | $S'[\phi] = \log \int [D\chi] \mathrm{e}^{-\frac{1}{2}\chi \cdot A \cdot \chi + S[\phi+\chi]}$ <br> $= \sum_\gamma \frac{A^\gamma(S)}{\mathrm{S}_\gamma}[\phi]$ |
| composition | successive integrations |

Derivation of **combinatorial identities** (hook length formula, properties of the tutte polynomial, ...) inspired by **effective action computations**.