# Cartes non-orientables, polynômes de Jack, et $b$-positivité

Guillaume Chapuy
CNRS – IRIF – Université de Paris – ERC CombiTop

joint work with

Maciej Dołęga
Polish Academy of Sciences, Kraków

arXiv:submit/3132718 [math.CO] 15 Apr 2020

NON-ORIENTABLE BRANCHED COVERINGS, $b$-HURWITZ NUMBERS, AND POSITIVITY FOR MULTIPARAMETRIC JACK EXPANSIONS

GUILLAUME CHAPUY AND MACIEJ DOŁĘGA

Abstract. We introduce a one-parameter deformation of the 2-Toda tau-function of (weighted) Hurwitz numbers, obtained by deforming Schur functions into Jack symmetric functions. We show that its coefficients are polynomials in the deformation parameter $b$ with nonnegative integer coefficients. These coefficients count generalized branched coverings of the sphere by an arbitrary surface, orientable or not, with an appropriate $b$-weighting that "measures" in some sense their non-orientability.

Notable special cases include non-orientable *dessins d'enfants* for which we prove the most general result so far towards the Matching-Jack conjecture and the "$b$-conjecture" of Goulden and Jackson from 1996, expansions of the $\beta$-ensemble matrix model, deformations of the HCIZ integral, and $b$-Hurwitz numbers that we introduce here and that are $b$-deformations of classical (single or double) Hurwitz numbers obtained for $b = 0$.

A key role in our proof is played by a combinatorial model of non-orientable constellations equipped with a suitable $b$-weighting, whose partition function satisfies an infinite set of PDEs. These PDEs have two definitions, one given by Lax equations, the other one following an explicit combinatorial decomposition.

1. Introduction

**Hurwitz numbers and tau-functions.** Hurwitz numbers, in their most general sense, count the number of combinatorially inequivalent branched coverings of the sphere by an orientable surface with a given number of branchpoints and given ramification profiles. Hurwitz numbers and their variants (dessins d'enfants, weighted, monotone, orbifold Hurwitz numbers) have numerous connections to mathematical physics, combinatorics, and the moduli spaces of curves [Kon92, GJ97, ELSV01, GV03, GJV05, OP06, Mir07, GPH17].

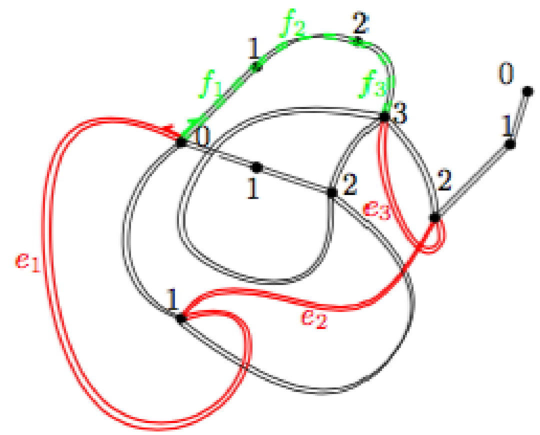
Hurwitz himself [Hur91] showed that Hurwitz numbers can be expressed in terms of characters of the symmetric group. Equivalently, generating functions of Hurwitz numbers can be expressed explicitly in terms of Schur functions, which gives them a rich structure. A fundamental fact in the field, going back to Pandariphande [Pan00] and Okounkov [Oko00] and now understood in a wide generality (see e.g. [GJ08, GPH17]) is that Hurwitz numbers can be used to define a formal power series which is a tau-function of the KP, or more generally 2-Toda hierarchy [MJD00]. Explicitly, in the case of $k+2$ branchpoints, this tau-function has the form

$$(1)\quad \tau^{(k)}(t;\mathbf{p},\mathbf{q},u_1,\dots,u_k) := \sum_{n\geq 0} t^n \sum_{\lambda\vdash n} \left(\frac{f_\lambda}{n!}\right)^2 \tilde{s}_\lambda(\mathbf{p})\tilde{s}_\lambda(\mathbf{q})s_\lambda(\underline{u_1})\tilde{s}_\lambda(\underline{u_2})\dots\tilde{s}_\lambda(\underline{u_k}),$$

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. ERC-2016-STG 716083 "CombiTop"). MD is supported from *Narodowe Centrum Nauki*, grant UMO-2017/26/D/ST1/00186. Emails: guillaume.chapuy@irif.fr, mdolega@impan.pl.

1

→ https://arxiv.org/abs/2004.07824




Séminaire CEA du LaBRI, avril 2020

# Cartes non-orientables, polynômes de Jack, et $b$-positivité

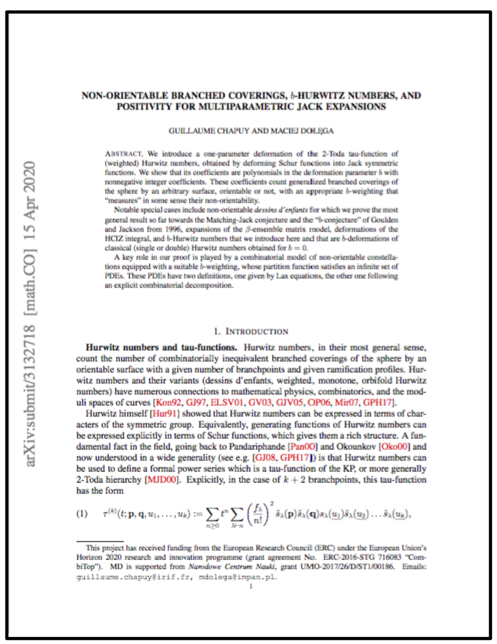

Guillaume Chapuy

CNRS – IRIF – Université de Paris – ERC CombiTop

joint work with

Maciej Dołęga

Polish Academy of Sciences, Kraków

arXiv:submit/3132718 [math.CO] 15 Apr 2020

**NON-ORIENTABLE BRANCHED COVERINGS, $b$-HURWITZ NUMBERS, AND POSITIVITY FOR MULTIPARAMETRIC JACK EXPANSIONS**

GUILLAUME CHAPUY AND MACIEJ DOŁĘGA


ABSTRACT. We introduce a one-parameter deformation of the 2-Toda tau-function of (weighted) Hurwitz numbers, obtained by deforming Schur functions into Jack symmetric functions. We show that its coefficients are polynomials in the deformation parameter $b$ with nonnegative integer coefficients. These coefficients count generalized branched coverings of the sphere by an arbitrary surface, orientable or not, with an appropriate $b$-weighting that "measures" in some sense their non-orientability.

Notable special cases include non-orientable *dessins d'enfants* for which we prove the most general result so far towards the Matching-Jack conjecture and the "$b$-conjecture" of Goulden and Jackson from 1996, expansions of the $\beta$-ensemble matrix model, deformations of the HCIZ integral, and $b$-Hurwitz numbers that we introduce here and that are $b$-deformations of classical (single or double) Hurwitz numbers obtained for $b = 0$.

A key role in our proof is played by a combinatorial model of non-orientable constellations equipped with a suitable $b$-weighting, whose partition function satisfies an infinite set of PDEs. These PDEs have two definitions, one given by Lax equations, the other one following an explicit combinatorial decomposition.


1. INTRODUCTION

**Hurwitz numbers and tau-functions.** Hurwitz numbers, in their most general sense, count the number of combinatorially inequivalent branched coverings of the sphere by an orientable surface with a given number of branchpoints and given ramification profiles. Hurwitz numbers and their variants (dessins d'enfants, weighted, monotone, orbifold Hurwitz numbers) have numerous connections to mathematical physics, combinatorics, and the moduli spaces of curves [Kon92, GJ97, ELSV01, GV03, GJV05, OP06, Mir07, GPH17].

Hurwitz himself [Hur91] showed that Hurwitz numbers can be expressed in terms of characters of the symmetric group. Equivalently, generating functions of Hurwitz numbers can be expressed explicitly in terms of Schur functions, which gives them a rich structure. A fundamental fact in the field, going back to Pandariphande [Pan00] and Okounkov [Oko00] and now understood in a wide generality (see e.g. [GJ08, GPH17]) is that Hurwitz numbers can be used to define a formal power series which is a tau-function of the KP, or more generally 2-Toda hierarchy [MJD00]. Explicitly, in the case of $k+2$ branchpoints, this tau-function has the form

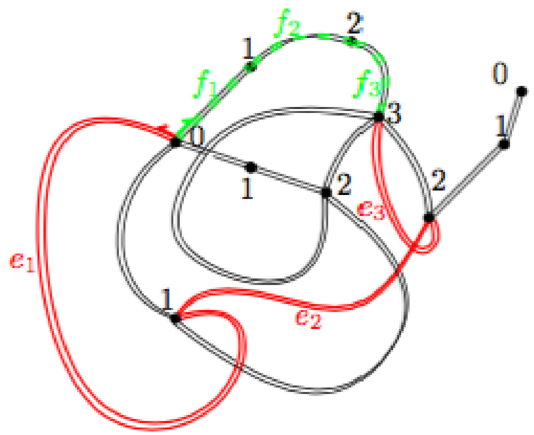
$$\tau^{(k)}(t; \mathbf{p}, \mathbf{q}, u_1, \dots, u_k) := \sum_{n \geq 0} t^n \sum_{\lambda \vdash n} \left(\frac{f_\lambda}{n!}\right)^2 \tilde{s}_\lambda(\mathbf{p}) \tilde{s}_\lambda(\mathbf{q}) s_\lambda(\underline{u_1}) \tilde{s}_\lambda(\underline{u_2}) \dots \tilde{s}_\lambda(\underline{u_k}), \quad (1)$$


This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. ERC-2016-STG 716083 "CombiTop"). MD is supported from *Narodowe Centrum Nauki*, grant UMO-2017/26/D/ST1/00186. Emails: guillaume.chapuy@irif.fr, mdolega@impan.pl.


1

Mes excuses pour les transparents en anglais

→ https://arxiv.org/abs/2004.07824




Séminaire CEA du LaBRI, avril 2020

# Symmetric functions

# Symmetric functions and their bases.

- Let $\Lambda_n$ be the vector space of formal power series in $x_1, x_2, \ldots$ which are symmetric, and homogeneous of degree $n$.

  Examples: $1 \in \Lambda_0$, $\sum_i x_i \in \Lambda_1$, $\sum_{i,j} x_i x_j \in \Lambda_2$, $\sum_i x_i^2 - 2\sum_{i,j} x_i x_j \in \Lambda_2$

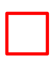
# Symmetric functions and their bases.

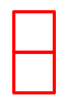
- Let $\Lambda_n$ be the vector space of formal power series in $x_1, x_2, \ldots$ which are symmetric, and homogeneous of degree $n$.

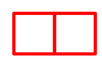
  Examples: $1 \in \Lambda_0$, $\sum_i x_i \in \Lambda_1$, $\sum_{i,j} x_i x_j \in \Lambda_2$, $\sum_i x_i^2 - 2\sum_{i,j} x_i x_j \in \Lambda_2$

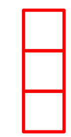
- $\Lambda_n$ has a basis which is naturally indexed by partitions of $n$

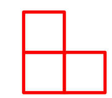
$m_{\varnothing} = 1$

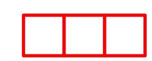

$m_{[1]} = \sum_i x_i$


$m_{[1,1]} = \sum_{i<j} x_i x_j$


$m_{[2]} = \sum_i x_i^2$


$m_{[1,1,1]} = \sum_{i<j<k} x_i x_j x_k$


$m_{[2,1]} = \sum_{i,j} x_i^2 x_j$


$m_{[3]} = \sum_i x_i^3$

($m_\lambda$: monomial symmetric functions)
(group together all monomials "of re-ordered exponents $\lambda$")

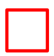
# Symmetric functions and their bases.

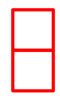
- Let $\Lambda_n$ be the vector space of formal power series in $x_1, x_2, \ldots$ which are symmetric, and homogeneous of degree $n$.

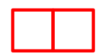
  Examples: $1 \in \Lambda_0$, $\sum_i x_i \in \Lambda_1$, $\sum_{i,j} x_i x_j \in \Lambda_2$, $\sum_i x_i^2 - 2\sum_{i,j} x_i x_j \in \Lambda_2$

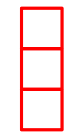
- $\Lambda_n$ has a basis which is naturally indexed by partitions of $n$

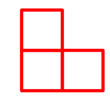
$m_{\varnothing} = 1$

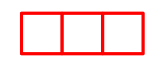
 $m_{[1]} = \sum_i x_i$

 $m_{[1,1]} = \sum_{i<j} x_i x_j$

 $m_{[2]} = \sum_i x_i^2$

 $m_{[1,1,1]} = \sum_{i<j<k} x_i x_j x_k$

 $m_{[2,1]} = \sum_{i,j} x_i^2 x_j$

 $m_{[3]} = \sum_i x_i^3$

($m_\lambda$: monomial symmetric functions)
(group together all monomials "of re-ordered exponents $\lambda$")

- $\Lambda_n$ has many other nice bases, all indexed by partitions.

# Symmetric functions and their bases.

- Let $\Lambda_n$ be the vector space of formal power series in $x_1, x_2, \ldots$ which are symmetric, and homogeneous of degree $n$.

  Examples: $1 \in \Lambda_0$, $\sum_i x_i \in \Lambda_1$, $\sum_{i,j} x_i x_j \in \Lambda_2$, $\sum_i x_i^2 - 2\sum_{i,j} x_i x_j \in \Lambda_2$

- $\Lambda_n$ has a basis which is naturally indexed by partitions of $n$

$m_{\varnothing} = 1$

$m_{[1]} = \sum_i x_i$

$m_{[1,1]} = \sum_{i<j} x_i x_j$

$m_{[2]} = \sum_i x_i^2$

$m_{[1,1,1]} = \sum_{i<j<k} x_i x_j x_k$

$m_{[2,1]} = \sum_{i,j} x_i^2 x_j$

$m_{[3]} = \sum_i x_i^3$

($m_\lambda$: monomial symmetric functions)
(group together all monomials "of re-ordered exponents $\lambda$")

- $\Lambda_n$ has many other nice bases, all indexed by partitions.

$$p_k(\mathbf{x}) = \sum_i x_i^k \quad \text{and} \quad p_\lambda = p_{\lambda_1} p_{\lambda_2} \ldots p_{\lambda_{\ell(\lambda)}}$$

($p_\lambda$: powersums)

Example: $p_{[2,2,1]} = p_2^2 p_1 = \sum_{i,j,k} x_i^2 x_j^2 x_k$
$= \ldots$

# Schur functions (I)

- If $\lambda$ is a partition, a semistandard Young tableau of shape $\lambda$ (SSYT) is a filling of $\lambda$ which is $\leq$ on rows and $\vee$ on columns.

$$\lambda = (4,4,2,1) \qquad T = \begin{array}{|c|c|c|c|} \hline 7 \\ \cline{1-1} 5 & 12 \\ \cline{1-2} 3 & 9 & 9 & 9 \\ \hline 2 & 2 & 3 & 7 \\ \hline \end{array}$$

$$x^T = x_2^2 x_3^2 x_5 x_7^2 x_9^3 x_{12}$$

$$s_{[2]} = \sum_{i \leq j} x_i x_j \qquad \begin{array}{|c|c|} \hline i & j \\ \hline \end{array}$$

$$s_{[1,1]} = \sum_{i < j} x_i x_j \qquad \begin{array}{|c|} \hline j \\ \hline i \\ \hline \end{array}$$

$$s_{[3,1]} = \sum_{\substack{i \leq j \leq k \\ m > i}} x_i x_j x_k x_m \qquad \begin{array}{|c|c|c|} \hline m \\ \cline{1-1} i & j & k \\ \hline \end{array}$$

The Schur function $s_\lambda$ is the generating function of SSYT's of shape $\lambda$.

$$s_\lambda(x) = \sum_{T:SSYT(\lambda)} x^T$$

**Thm:** The $s_\lambda$ for $\lambda \vdash n$ are a basis of $\Lambda_n$.

(yes, in particular they are symmetric functions)

# Schur functions (bis)

- Cool fact: Viewed as polynomials in the powersums $\mathbf{p} = (p_k)_{k\geq 1}$, Schur functions generate characters of the symmetric group

$$\boxed{s_\lambda =: s_\lambda(\mathbf{p}) = \frac{1}{n!}\sum_{\mu\vdash n} |C_\mu|\chi^\lambda(\mu)p_\mu}$$

(here $n = |\lambda|$)

$\chi^\lambda(\mu)$: trace of a permutation of type $\mu$ acting on the representation $V^\lambda$ of $\mathfrak{S}_n$

- We use powersums to equip $\Lambda_n$ with the Hall scalar product:

$$\boxed{\langle p_\lambda, p_\mu\rangle = z_\lambda \delta_{\lambda,\mu} \quad z_\lambda = \frac{n!}{|\mathcal{C}_\lambda|}}$$

$$p_k = \sum_i x_i^k, \qquad p_\lambda = p_{\lambda_1}p_{\lambda_2}\dots\lambda_{\lambda_{\ell(\lambda)}}$$

# Schur functions (bis)

- Cool fact: Viewed as polynomials in the powersums $\mathbf{p} = (p_k)_{k\geq 1}$, Schur functions generate characters of the symmetric group

$$\boxed{s_\lambda =: s_\lambda(\mathbf{p}) = \frac{1}{n!}\sum_{\mu\vdash n} |C_\mu|\chi^\lambda(\mu)p_\mu}$$

(here $n = |\lambda|$)

$\chi^\lambda(\mu)$: trace of a permutation of type $\mu$ acting on the representation $V^\lambda$ of $\mathfrak{S}_n$

- We use powersums to equip $\Lambda_n$ with the Hall scalar product:

$$\boxed{\langle p_\lambda, p_\mu\rangle = z_\lambda \delta_{\lambda,\mu} \qquad z_\lambda = \frac{n!}{|\mathcal{C}_\lambda|}}$$

$$p_k = \sum_i x_i^k, \qquad p_\lambda = p_{\lambda_1}p_{\lambda_2}\dots\lambda_{\lambda_{\ell(\lambda)}}$$

- A characterisation of Schur functions:

$$\boxed{\begin{cases} \text{orthonormal for Hall: } \langle s_\lambda, s_\mu\rangle = \delta_{\lambda,\mu} \\ \text{triangular w.r.t. to monomials: } s_\lambda = m_\lambda + \sum_{\mu<\lambda} a_{\lambda,\mu}m_\mu \end{cases}}$$

[recall dominance order: $\mu \leq \lambda : \mu_1 + \dots + \mu_i \leq \lambda_1 + \dots + \lambda_i \quad \forall i$]

# Jack polynomials

- Piotr Śniady: "*Jack polynomials are Schur functions under steroids.*"
- More precisely: we deform the Hall scalar product and keep triangularity

$$\langle p_\lambda, p_\mu \rangle_\alpha = z_\lambda \alpha^{\ell(\lambda)} \delta_{\lambda,\mu}$$

- Jack polynomials:

$$\boxed{\begin{cases} \text{orthogonal for Hall}_\alpha\text{:} \quad \langle J_\lambda^{(\alpha)}, J_\mu^{(\alpha)} \rangle_\alpha = j_\lambda^{(\alpha)} \delta_{\lambda,\mu} \\ \text{triangular w.r.t. to monomials:} \quad J_\lambda^{(\alpha)} = g_\lambda^{(\alpha)} m_\lambda + \sum_{\mu<\lambda} a'_{\lambda,\mu} m_\mu \end{cases}}$$

(for us: normalization coefficients $j_\lambda^{(\alpha)}$ and $g_\lambda^{(\alpha)}$ chosen s.t. $[p_1^n] J_\lambda^{(\alpha)} = 1$.)

# Jack polynomials

- Piotr Śniady: *"Jack polynomials are Schur functions under steroids."*
- More precisely: we deform the Hall scalar product and keep triangularity

$$\langle p_\lambda, p_\mu \rangle_\alpha = z_\lambda \alpha^{\ell(\lambda)} \delta_{\lambda,\mu}$$

- Jack polynomials:

$$\boxed{\begin{cases} \text{orthogonal for Hall}_\alpha\text{:} & \langle J_\lambda^{(\alpha)}, J_\mu^{(\alpha)} \rangle_\alpha = j_\lambda^{(\alpha)} \delta_{\lambda,\mu} \\ \text{triangular w.r.t. to monomials:} & J_\lambda^{(\alpha)} = g_\lambda^{(\alpha)} m_\lambda + \sum_{\mu<\lambda} a'_{\lambda,\mu} m_\mu \end{cases}}$$

(for us: normalization coefficients $j_\lambda^{(\alpha)}$ and $g_\lambda^{(\alpha)}$ chosen s.t. $[p_1^n] J_\lambda^{(\alpha)} = 1$.)

- We choose to view Jacks as polynomials in the powersums

for example $J_{3,1}^{(\alpha)}(\mathbf{p}) = {p_1}^4 + (3\alpha - 1) p_2 {p_1}^2 + (2\alpha^2 - 2\alpha) p_3 p_1 - 2\alpha^2 p_4 - \alpha {p_2}^2$

# Jack polynomials

- Piotr Śniady: *"Jack polynomials are Schur functions under steroids."*
- More precisely: we deform the Hall scalar product and keep triangularity

$$\langle p_\lambda, p_\mu \rangle_\alpha = z_\lambda \alpha^{\ell(\lambda)} \delta_{\lambda,\mu}$$

- Jack polynomials:

$$\boxed{\begin{cases} \text{orthogonal for Hall}_\alpha: & \langle J_\lambda^{(\alpha)}, J_\mu^{(\alpha)} \rangle_\alpha = j_\lambda^{(\alpha)} \delta_{\lambda,\mu} \\ \text{triangular w.r.t. to monomials:} & J_\lambda^{(\alpha)} = g_\lambda^{(\alpha)} m_\lambda + \sum_{\mu<\lambda} a'_{\lambda,\mu} m_\mu \end{cases}}$$

(for us: normalization coefficients $j_\lambda^{(\alpha)}$ and $g_\lambda^{(\alpha)}$ chosen s.t. $[p_1^n] J_\lambda^{(\alpha)} = 1$.)

- We choose to view Jacks as polynomials in the powersums

for example $\quad J_{3,1}^{(\alpha)}(\mathbf{p}) = {p_1}^4 + (3\alpha - 1) p_2 {p_1}^2 + (2\alpha^2 - 2\alpha) p_3 p_1 - 2\alpha^2 p_4 - \alpha {p_2}^2$

- For $\alpha = 1$, Jacks are (normalized) Schur: $J_\lambda^{(1)} = H_\lambda s_\lambda =: \tilde{s}_\lambda$

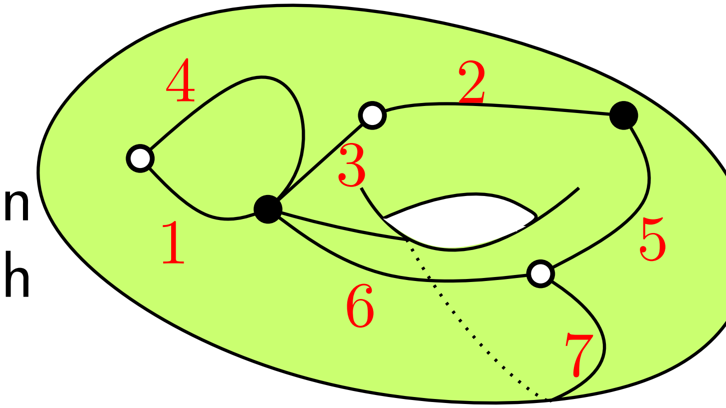
hook product

For $\alpha = 2$, Jacks are zonal polynomials: $J_\lambda^{(2)}$ is related to representation theory of the Gelfand pair $(\mathfrak{S}_{2n}, B_n)$ "in the same way as" $\tilde{s}_\lambda$ is to representation theory of $\mathfrak{S}_n$.

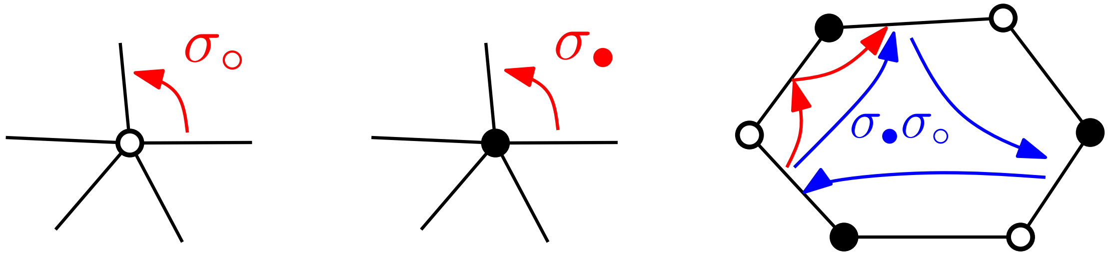
(a bit more later)

# Maps and factorizations

# Maps on orientable surfaces




- Bipartite map: bipartite ($\circ/\bullet$) graph embedded on an oriented surface with edges labelled $\{1, 2, \ldots, n\}$, with simply connected faces, considered up to homeomorphism.



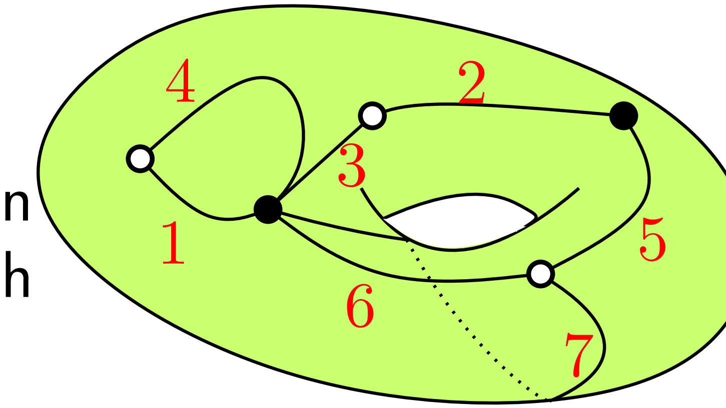

cycles of $\sigma_\circ$ cycles of $\sigma_\bullet$ cycles of $\sigma_\bullet\sigma_\circ$

$$\sigma_\circ = (1,4)(2,3)(5,6,7)$$
$$\sigma_\bullet = (1,6,7,3,4)(2,5)$$
$$\sigma_\diamond^{-1} = \sigma_\bullet\sigma_\circ = (1)(2,4,6,3,5,7)$$

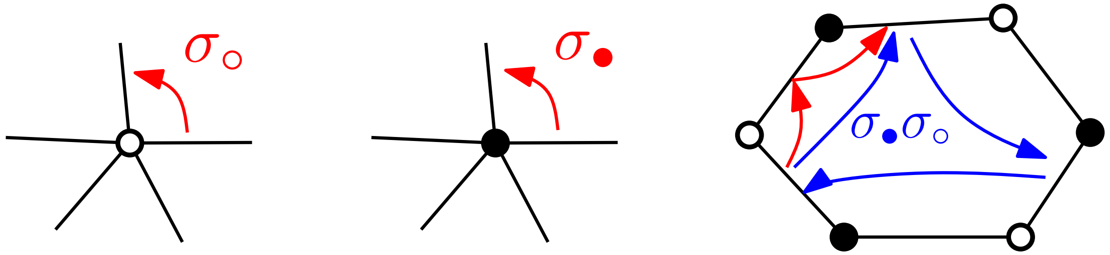
$\rightarrow$ the same as a triple of permutations $(\sigma_\circ, \sigma_\bullet, \sigma_\diamond)$ such that $\sigma_\circ\sigma_\bullet\sigma_\diamond = id$.

cf. [Cori-Machí 80s]

# Maps on orientable surfaces




- Bipartite map: bipartite ($\circ/\bullet$) graph embedded on an oriented surface with edges labelled $\{1, 2, \ldots, n\}$, with simply connected faces, considered up to homeomorphism.




cycles of $\sigma_\circ$ cycles of $\sigma_\bullet$ cycles of $\sigma_\bullet \sigma_\circ$

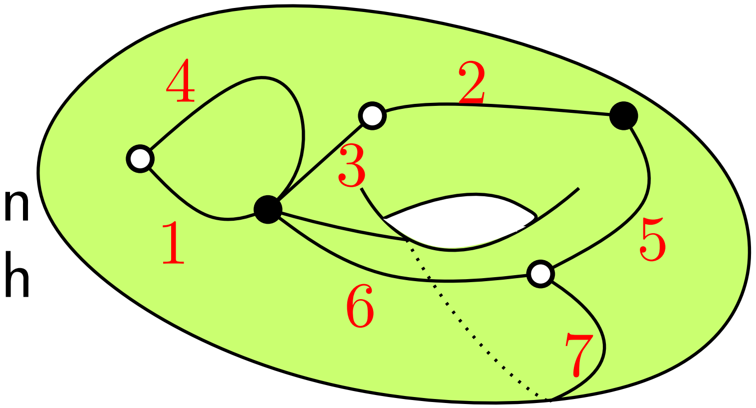
$\sigma_\circ = (1,4)(2,3)(5,6,7)$
$\sigma_\bullet = (1,6,7,3,4)(2,5)$
$\sigma_\diamond^{-1} = \sigma_\bullet \sigma_\circ = (1)(2,4,6,3,5,7)$

$\rightarrow$ the same as a triple of permutations $(\sigma_\circ, \sigma_\bullet, \sigma_\diamond)$ such that $\sigma_\circ \sigma_\bullet \sigma_\diamond = id$.

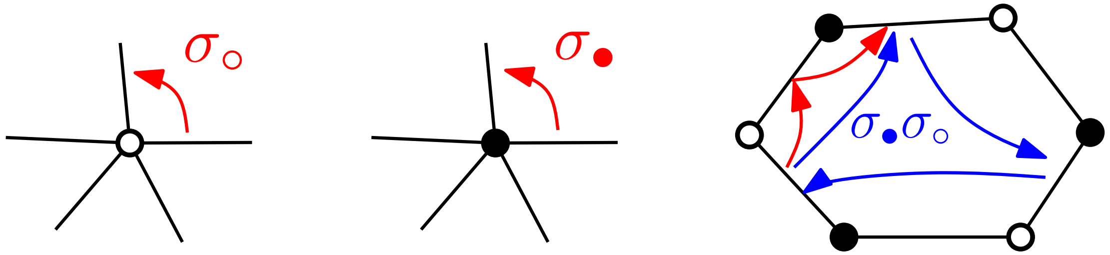
cf. [Cori-Machí 80s]

- Cool fact (Frobenius). The number of factorizations of the identity in a finite group into factors of given conjugacy classes, can be expressed in terms of irreducible characters of the group. As a consequence we have:

# Maps on orientable surfaces




- Bipartite map: bipartite ($\circ/\bullet$) graph embedded on an oriented surface with edges labelled $\{1, 2, \ldots, n\}$, with simply connected faces, considered up to homeomorphism.




cycles of $\sigma_\circ$ cycles of $\sigma_\bullet$ cycles of $\sigma_\bullet\sigma_\circ$

$\sigma_\circ = (1,4)(2,3)(5,6,7)$

$\sigma_\bullet = (1,6,7,3,4)(2,5)$

$\sigma_\diamond^{-1} = \sigma_\bullet\sigma_\circ = (1)(2,4,6,3,5,7)$

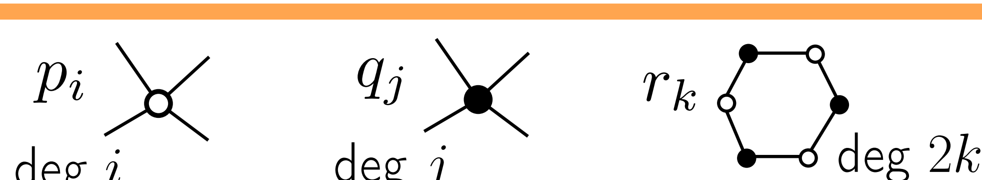
$\rightarrow$ the same as a triple of permutations $(\sigma_\circ, \sigma_\bullet, \sigma_\diamond)$ such that $\sigma_\circ\sigma_\bullet\sigma_\diamond = id$.

cf. [Cori-Machí 80s]

- Cool fact (Frobenius). The number of factorizations of the identity in a finite group into factors of given conjugacy classes, can be expressed in terms of irreducible characters of the group. As a consequence we have:

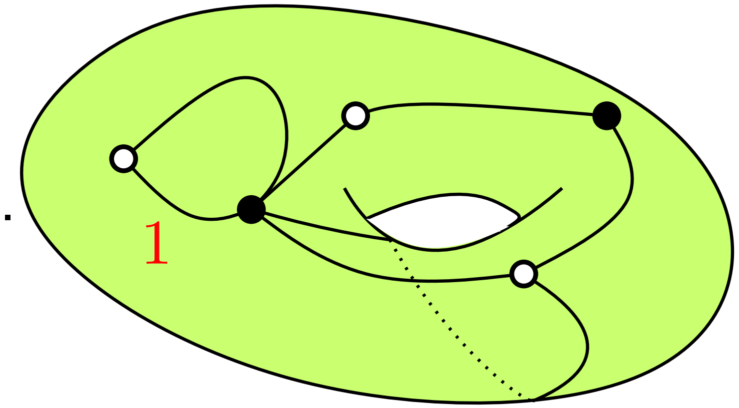
**"Character formula" for map generating function**

$$\sum_{\substack{\mathbf{m}:\\ bip.map}} \frac{t^n}{n!} p_{\lambda^\circ(\mathbf{m})} q_{\lambda^\bullet(\mathbf{m})} r_{\lambda^\diamond(\mathbf{m})} = \sum_{\lambda \in \mathcal{P}} t^{|\lambda|} H_\lambda s_\lambda(\mathbf{p}) s_\lambda(\mathbf{q}) s_\lambda(\mathbf{r})$$




[proof: put the two cool facts together]

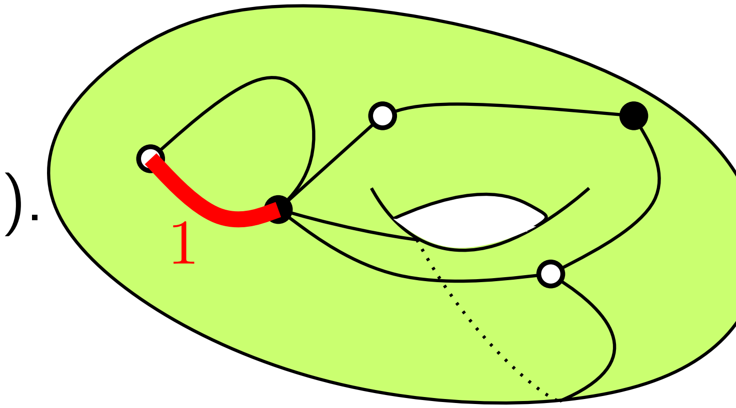
## Variant: rooted maps




- We only remember the position of the label $1$ ("root edge").

We ask the surface to be connected.

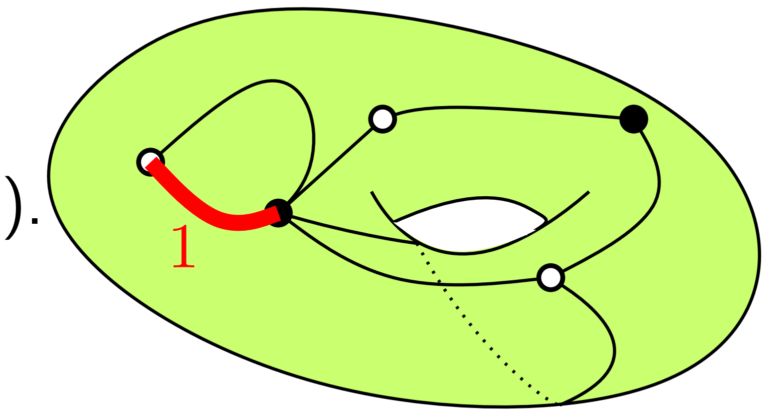
# Variant: rooted maps




- We only remember the position of the label $1$ ("root edge").

We ask the surface to be connected.

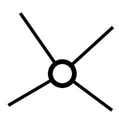
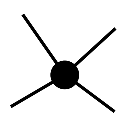
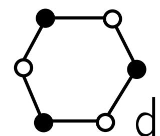
# Variant: rooted maps




- We only remember the position of the label $1$ ("root edge").
We ask the surface to be connected.

**"Character formula" for rooted maps**

$$\sum_{\substack{\mathbf{m}:\\ \textit{rooted bip. map}}} t^n p_{\lambda^\circ(\mathbf{m})} q_{\lambda^\bullet(\mathbf{m})} r_{\lambda^\diamond(\mathbf{m})} = t\frac{\partial}{\partial t} \log \sum_{\lambda \in \mathcal{P}} t^{|\lambda|} H_\lambda s_\lambda(\mathbf{p}) s_\lambda(\mathbf{q}) s_\lambda(\mathbf{r})$$

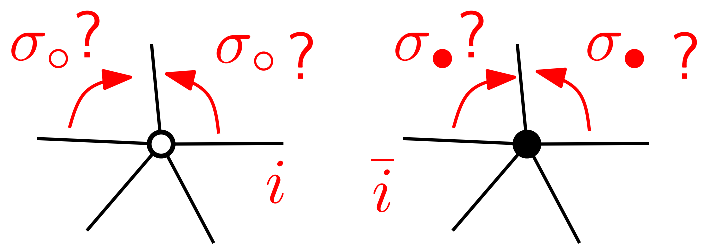
$p_i$  $q_j$  $r_k$ 


Note: we now have coefficients in $\mathbb{N}$ (no more labels, usual g.f. instead of exponential g.f.)

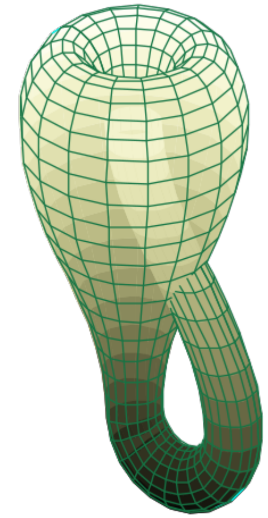
(this is not obvious from the RHS!)

# Maps on non (necessarily) orientable surfaces

- We now want to look at bipartite maps on non-necessarily orientable surfaces.



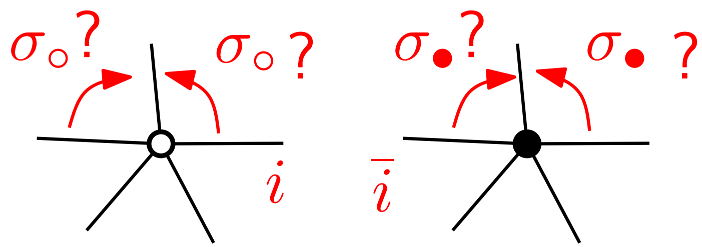

An encoding by permutations still works but everything is defined up to change of local orientation around each vertex.... so it's more complicated



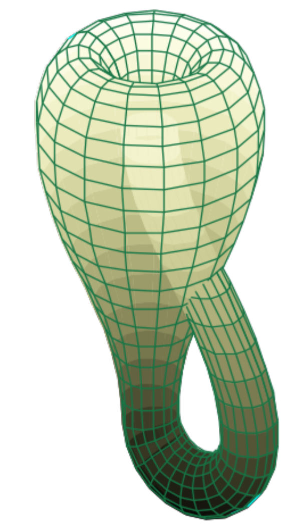
$\rightarrow$ still works but now bipartite maps have to do with factorisations in the double coset algebra $B_n \setminus \mathfrak{S}_{2n} / B_n$.

# Maps on non (necessarily) orientable surfaces

- We now want to look at bipartite maps on non-necessarily orientable surfaces.



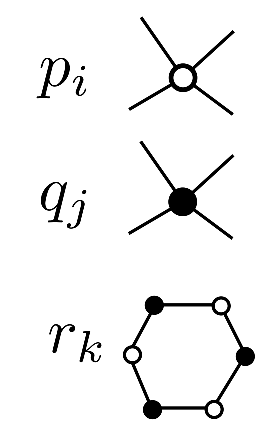

An encoding by permutations still works but everything is defined up to change of local orientation around each vertex.... so it's more complicated



$\rightarrow$ still works but now bipartite maps have to do with factorisations in the double coset algebra $B_n \setminus \mathfrak{S}_{2n}/B_n$.

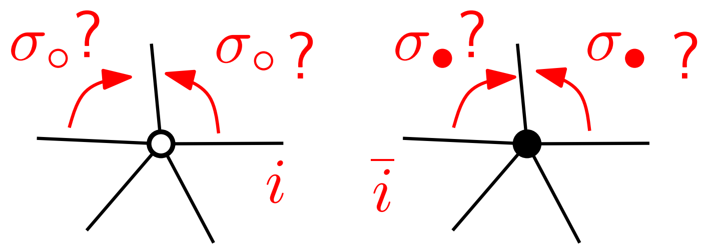
**"Non-orientable character formula"**[Hanlon, Stanley, Stembridge '92, Goulden, Jackson '96]

$$\sum_{\substack{\mathbf{m}:\\ \textit{rooted bip. map}\\ \textit{orientable or not}}} t^n p_{\lambda^\circ(\mathbf{m})} q_{\lambda^\bullet(\mathbf{m})} r_{\lambda^\diamond(\mathbf{m})} = 2t\frac{\partial}{\partial t} \log \sum_{\lambda \in \mathcal{P}} t^{|\lambda|} \frac{J_\lambda^{(2)}(\mathbf{p}) J_\lambda^{(2)}(\mathbf{q}) J_\lambda^{(2)}(\mathbf{r})}{j_\lambda^{(2)}}$$

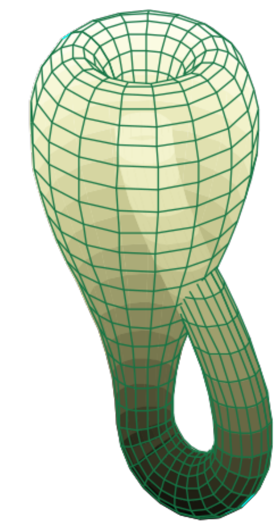


# Maps on non (necessarily) orientable surfaces

- We now want to look at bipartite maps on non-necessarily orientable surfaces.




An encoding by permutations still works but everything is defined up to change of local orientation around each vertex.... so it's more complicated



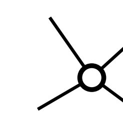
$\rightarrow$ still works but now bipartite maps have to do with factorisations in the double coset algebra $B_n \setminus \mathfrak{S}_{2n} / B_n$.

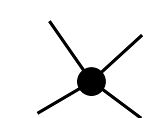
**"Non-orientable character formula"** [Hanlon, Stanley, Stembridge '92, Goulden, Jackson '96]

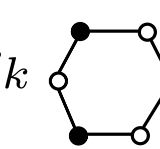
$$\sum_{\substack{\mathbf{m}: \\ \text{rooted bip. map} \\ \text{orientable or not}}} t^n p_{\lambda^\circ(\mathbf{m})} q_{\lambda^\bullet(\mathbf{m})} r_{\lambda^\diamond(\mathbf{m})} = 2t \frac{\partial}{\partial t} \log \sum_{\lambda \in \mathcal{P}} t^{|\lambda|} \frac{J_\lambda^{(2)}(\mathbf{p}) J_\lambda^{(2)}(\mathbf{q}) J_\lambda^{(2)}(\mathbf{r})}{j_\lambda^{(2)}}$$

**"Orientable character formula"** (rewritten in Jack notation)

$$\sum_{\substack{\mathbf{m}: \\ \text{rooted bip. map} \\ \text{orientable}}} t^n p_{\lambda^\circ(\mathbf{m})} q_{\lambda^\bullet(\mathbf{m})} r_{\lambda^\diamond(\mathbf{m})} = t \frac{\partial}{\partial t} \log \sum_{\lambda \in \mathcal{P}} t^{|\lambda|} \frac{J_\lambda^{(1)}(\mathbf{p}) J_\lambda^{(1)}(\mathbf{q}) J_\lambda^{(1)}(\mathbf{r})}{j_\lambda^{(1)}}$$

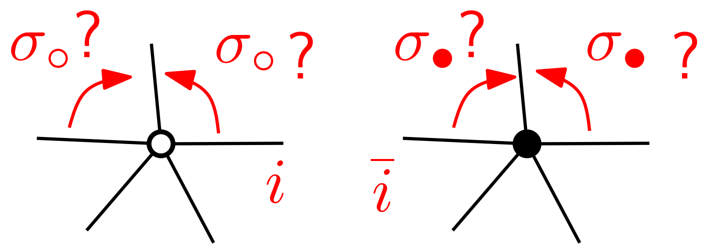
$p_i$ 

$q_j$ 

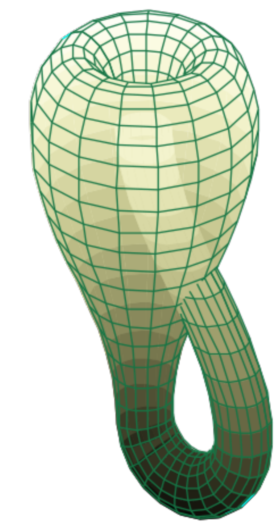
$r_k$ 

# Maps on non (necessarily) orientable surfaces

- We now want to look at bipartite maps on non-necessarily orientable surfaces.




An encoding by permutations still works but everything is defined up to change of local orientation around each vertex.... so it's more complicated



$\rightarrow$ still works but now bipartite maps have to do with factorisations in the double coset algebra $B_n \setminus \mathfrak{S}_{2n}/B_n$.

Zonal

$\alpha = 2$

**"Non-orientable character formula"**[Hanlon, Stanley, Stembridge '92, Goulden, Jackson '96]

$$\sum_{\substack{\mathbf{m}:\\ \textit{rooted bip. map}\\ \textit{orientable or not}}} t^n p_{\lambda^\circ(\mathbf{m})} q_{\lambda^\bullet(\mathbf{m})} r_{\lambda^\diamond(\mathbf{m})} = 2t\frac{\partial}{\partial t} \log \sum_{\lambda \in \mathcal{P}} t^{|\lambda|} \frac{J_\lambda^{(2)}(\mathbf{p}) J_\lambda^{(2)}(\mathbf{q}) J_\lambda^{(2)}(\mathbf{r})}{j_\lambda^{(2)}}$$

??

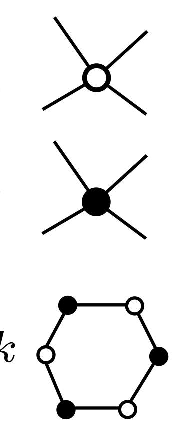
$\alpha = 1$

Schur

**"Orientable character formula"** (rewritten in Jack notation)

$$\sum_{\substack{\mathbf{m}:\\ \textit{rooted bip. map}\\ \textit{orientable}}} t^n p_{\lambda^\circ(\mathbf{m})} q_{\lambda^\bullet(\mathbf{m})} r_{\lambda^\diamond(\mathbf{m})} = t\frac{\partial}{\partial t} \log \sum_{\lambda \in \mathcal{P}} t^{|\lambda|} \frac{J_\lambda^{(1)}(\mathbf{p}) J_\lambda^{(1)}(\mathbf{q}) J_\lambda^{(1)}(\mathbf{r})}{j_\lambda^{(1)}}$$

$p_i$ $q_j$ $r_k$



# $b$-positivity

# The Goulden-Jackson $b$-conjecture (1996)

- Conjecture: The generating function

$$(1+b)t\frac{\partial}{\partial t}\log\sum_{\lambda\in\mathcal{P}} t^{|\lambda|}\frac{J_\lambda^{(1+b)}(\mathbf{p})J_\lambda^{(1+b)}(\mathbf{q})J_\lambda^{(1+b)}(\mathbf{r})}{j_\lambda^{(1+b)}}$$

is $b$-positive, with integer coefficients!! It counts bipartite maps!!!

## The Goulden-Jackson $b$-conjecture (1996)

- Conjecture: We have

$$(1+b)t\frac{\partial}{\partial t}\log\sum_{\lambda\in\mathcal{P}} t^{|\lambda|}\frac{J_\lambda^{(1+b)}(\mathbf{p})J_\lambda^{(1+b)}(\mathbf{q})J_\lambda^{(1+b)}(\mathbf{r})}{j_\lambda^{(1+b)}} = \sum_{\substack{\mathbf{m}:\\ \textit{bip. rooted map}\\ \textit{orientable or not}}} t^{|\mathbf{m}|}p_{\lambda^\circ(\mathbf{m})}q_{\lambda^\bullet(\mathbf{m})}r_{\lambda^\diamond(\mathbf{m})}b^{\nu(\mathbf{m})}$$

The coefficients count non-orientable bipartite maps with a weight $b^{\nu(\mathbf{m})}$ where $\nu(\mathbf{m})=0$ iff $\mathbf{m}$ is orientable,

# The Goulden-Jackson $b$-conjecture (1996)

- Conjecture: We have

$$(1+b)t\frac{\partial}{\partial t}\log\sum_{\lambda\in\mathcal{P}} t^{|\lambda|}\frac{J_\lambda^{(1+b)}(\mathbf{p})J_\lambda^{(1+b)}(\mathbf{q})J_\lambda^{(1+b)}(\mathbf{r})}{j_\lambda^{(1+b)}} = \sum_{\substack{\mathbf{m}:\\ \textit{bip. rooted map}\\ \textit{orientable or not}}} t^{|\mathbf{m}|}p_{\lambda^\circ(\mathbf{m})}q_{\lambda^\bullet(\mathbf{m})}r_{\lambda^\diamond(\mathbf{m})}b^{\nu(\mathbf{m})}$$

The coefficients count non-orientable bipartite maps with a weight $b^{\nu(\mathbf{m})}$ where $\nu(\mathbf{m})=0$ iff $\mathbf{m}$ is orientable,

$\rightarrow$ $b=0$: (Schur) character formula for orientable maps

$\rightarrow$ $b=1$: (Zonal) character formula for non-(necessarily)-orientable maps

# The Goulden-Jackson $b$-conjecture (1996)

- Conjecture: We have

$$(1+b)t\frac{\partial}{\partial t}\log\sum_{\lambda\in\mathcal{P}} t^{|\lambda|}\frac{J_\lambda^{(1+b)}(\mathbf{p})J_\lambda^{(1+b)}(\mathbf{q})J_\lambda^{(1+b)}(\mathbf{r})}{j_\lambda^{(1+b)}} = \sum_{\substack{\mathbf{m}:\\ \text{bip. rooted map} \\ \text{orientable or not}}} t^{|\mathbf{m}|} p_{\lambda^\circ(\mathbf{m})} q_{\lambda^\bullet(\mathbf{m})} r_{\lambda^\diamond(\mathbf{m})} b^{\nu(\mathbf{m})}$$

The coefficients count non-orientable bipartite maps with a weight $b^{\nu(\mathbf{m})}$ where $\nu(\mathbf{m})=0$ iff $\mathbf{m}$ is orientable,

$\rightarrow$ $b=0$: (Schur) character formula for orientable maps

$\rightarrow$ $b=1$: (Zonal) character formula for non-(necessarily)-orientable maps

- Why is it so interesting? Because we have (had?) NO tools to attack it!

for $b\notin\{0,1\}$ there is no "character" technology. Progress is rare.

[Lacroix '10] $\rightarrow$ OK if we keep ONE full set of variables ("times"):

$$\mathbf{p}=(p_i)_{i\geq 1},\ \mathbf{q}=(\delta_{i,2})_{i\geq 1},\ \mathbf{r}=\underline{u}=(u,u,\dots)$$

Uses [Okounkov'97] about (linear) expectations of Jacks under $\beta$-ensembles.

Other cases proved for some particular coefficients [Kanunnikov,Promyslov,Vassilieva '18] [Dołęga '17]
That coefficients are in $\mathbb{Q}[b]$ (not $\mathbb{Q}(b)$) is proved in [Dołęga-Féray '17]

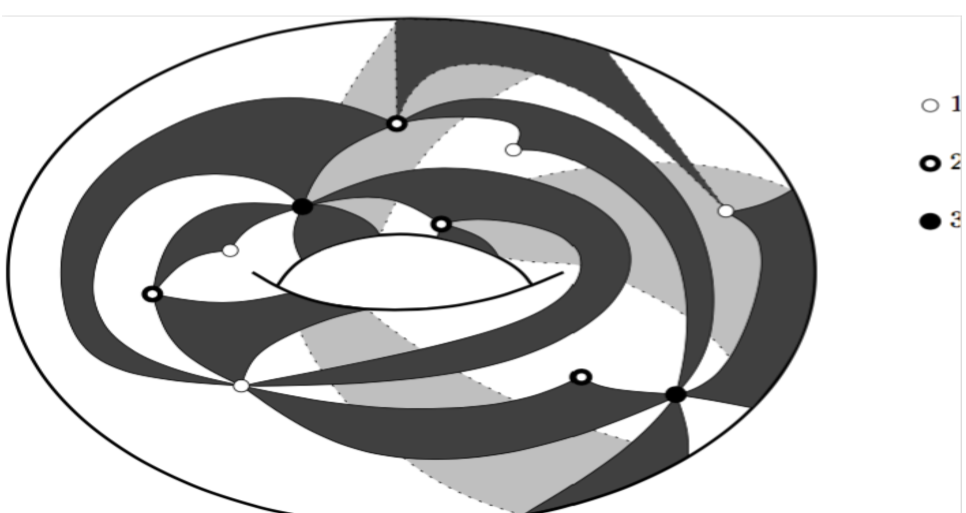
- One of our results: $\rightarrow$ OK if we keep TWO sets of times.

$$\mathbf{p}=(p_i)_{i\geq 1},\ \mathbf{q}=(q_i)_{i\geq 1},\ \mathbf{r}=\underline{u}=(u,u,\dots)$$

# Even better: constellations and the tau-function

- Factorisations of the form $\sigma_\circ\sigma_\bullet\sigma_1\sigma_2\dots\sigma_k = id$ in $\mathfrak{S}_n$ are in bijection with generalizations of bipartite maps called $k$-constellations.

The character approach is still valid to count them.



# Even better: constellations and the tau-function

- Factorisations of the form $\sigma_\circ \sigma_\bullet \sigma_1 \sigma_2 \dots \sigma_k = id$ in $\mathfrak{S}_n$ are in bijection with generalizations of bipartite maps called $k$-constellations.

  The character approach is still valid to count them.

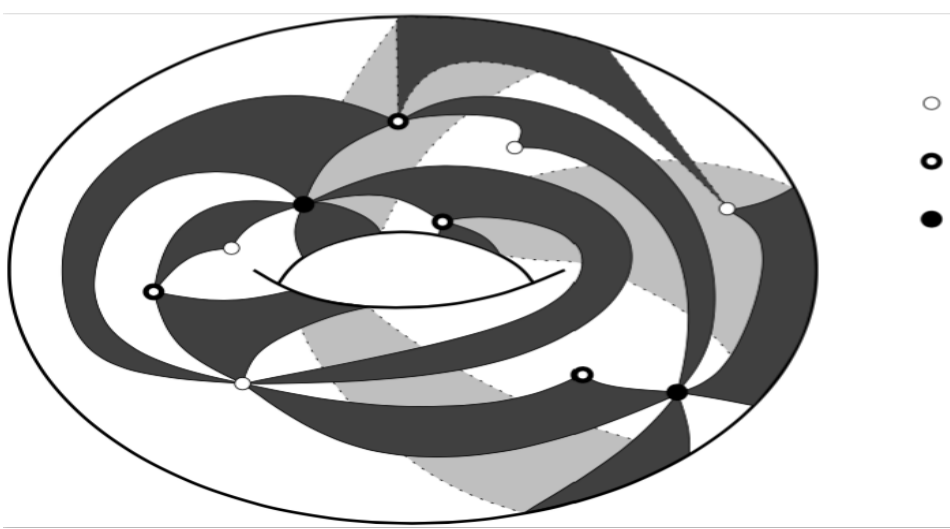
- Fact: The coolest object in the orientable $(b = 0)$ literature is

$$\tau^{(k)}(t; \mathbf{p}, \mathbf{q}, u_1, \dots, u_k) = \sum_\lambda t^{|\lambda|} \frac{\tilde{s}_\lambda(\mathbf{p}) \tilde{s}_\lambda(\mathbf{q}) \tilde{s}_\lambda(\underline{u_1}) \tilde{s}_\lambda(\underline{u_2}) \dots \tilde{s}_\lambda(\underline{u_k})}{j_\lambda^{(1)}}$$

$$= \sum_{\mathfrak{m}} \frac{t^n}{n!} p_{\sigma_\circ} q_{\sigma_\bullet} u_1^{\ell(\sigma_1)} \dots u_k^{\ell(\sigma_k)}$$

This is a tau-function of the $2$-Toda (and KP) hierarchy.

[Goulden-Jackson'09,Okounkov'00]




This is a central object in enumerative geometry (it counts branched coverings of the sphere).

# Our main result

- Theorem[Chapuy-Dołęga'20] Consider the $b$-deformed tau-function

$$\tau_b^{(k)}(t;\mathbf{p},\mathbf{q},u_1,\dots,u_k) = \sum_{\lambda\in\mathcal{P}} t^{|\lambda|} \frac{J_\lambda^{(1+b)}(\mathbf{p}) J_\lambda^{(1+b)}(\mathbf{q}) J_\lambda^{(1+b)}(\underline{u_1}) \dots J_\lambda^{(1+b)}(\underline{u_k})}{j_\lambda^{(1+b)}}$$

  Then $\quad (1+b)t\dfrac{\partial}{\partial t}\log\tau_b^{(k)} \quad$ is $b$-positive.

  Its coefficients count (properly defined) $k$-constellations on non-orientable surfaces with a weight $b^{\nu(\mathbf{m})}$ where $\nu(\mathbf{m}) = 0$ iff $\mathbf{m}$ is orientable.

- our result has three sets of parameters $\mathbf{p} = (p_i)_{i\geq 1}$, $\mathbf{q} = (q_i)_{i\geq 1}$, $\mathbf{u} = (u_i)_{i\leq k}$
- the case $k = 1$ or our result is the case $\mathbf{r} = \underline{u_1}$ of the $b$-conjecture.

# Our main result

- Theorem[Chapuy-Dołęga'20] Consider the $b$-deformed tau-function

$$\tau_b^{(k)}(t;\mathbf{p},\mathbf{q},u_1,\dots,u_k)=\sum_{\lambda\in\mathcal{P}} t^{|\lambda|}\frac{J_\lambda^{(1+b)}(\mathbf{p})J_\lambda^{(1+b)}(\mathbf{q})J_\lambda^{(1+b)}(\underline{u_1})\dots J_\lambda^{(1+b)}(\underline{u_k})}{j_\lambda^{(1+b)}}$$

  Then $(1+b)t\dfrac{\partial}{\partial t}\log\tau_b^{(k)}$ is $b$-positive.

  Its coefficients count (properly defined) $k$-constellations on non-orientable surfaces with a weight $b^{\nu(\mathbf{m})}$ where $\nu(\mathbf{m})=0$ iff $\mathbf{m}$ is orientable.

- our result has three sets of parameters $\mathbf{p}=(p_i)_{i\geq 1}$, $\mathbf{q}=(q_i)_{i\geq 1}$, $\mathbf{u}=(u_i)_{i\leq k}$
- the case $k=1$ or our result is the case $\mathbf{r}=\underline{u_1}$ of the $b$-conjecture.
- The case $b=0$ is the classical tau function.

# Our main result

- Theorem[Chapuy-Dołęga'20] Consider the $b$-deformed tau-function

$$\tau_b^{(k)}(t;\mathbf{p},\mathbf{q},u_1,\ldots,u_k)=\sum_{\lambda\in\mathcal{P}} t^{|\lambda|}\frac{J_\lambda^{(1+b)}(\mathbf{p})J_\lambda^{(1+b)}(\mathbf{q})J_\lambda^{(1+b)}(\underline{u_1})\ldots J_\lambda^{(1+b)}(\underline{u_k})}{j_\lambda^{(1+b)}}$$

Then $\quad (1+b)t\dfrac{\partial}{\partial t}\log\tau_b^{(k)}\quad$ is $b$-positive.

Its coefficients count (properly defined) $k$-constellations on non-orientable surfaces with a weight $b^{\nu(\mathbf{m})}$ where $\nu(\mathbf{m})=0$ iff $\mathbf{m}$ is orientable.

- our result has three sets of parameters $\mathbf{p}=(p_i)_{i\geq 1}$, $\mathbf{q}=(q_i)_{i\geq 1}$, $\mathbf{u}=(u_i)_{i\leq k}$
- the case $k=1$ or our result is the case $\mathbf{r}=\underline{u_1}$ of the $b$-conjecture.
- The case $b=0$ is the classical tau function.
- By letting $k\to\infty$ we can do $b$-analogues of Hurwitz numbers (factorisations in transpositions) and in fact, general weighted Hurwitz numbers ($k=\infty$).

## Elements of proof (?)

# Proof structure

- Our proof has three halves:

$1/2$ – If you are a map expert, you can, in principle, write some sort of linear PDE for the g.f. of constellations that reflects a "root-edge" decomposition. You can hope to do it by controlling the variables $\mathbf{p}$ and $\mathbf{q}$ and $(u_i)$.

– do it. And do it also for the non-orientable case.

– there is (seems to be) a natural way to put the $b$-parameter in these PDEs.

# Proof structure

- Our proof has three halves:

$1/2$ – If you are a map expert, you can, in principle, write some sort of linear PDE for the g.f. of constellations that reflects a "root-edge" decomposition. You can hope to do it by controlling the variables $\mathbf{p}$ and $\mathbf{q}$ and $(u_i)$.

  - do it. And do it also for the non-orientable case.
  - there is (seems to be) a natural way to put the $b$-parameter in these PDEs.

$2/2$ – If you are a Jack polynomial expert, you know that there are nice rules, such as the Pieri rule, or the $\alpha$-hook-content formula in [Stanley'89]. Also the Laplace Beltrami operator acts nicely on Jack polynomials.
Use your creativity + commutator magic (applying these rules in all sort of orders) to construct by induction a set of PDEs that cancel the function $\tau_b^{(k)}$.

# Proof structure

- Our proof has three halves:

$1/2$ – If you are a map expert, you can, in principle, write some sort of linear PDE for the g.f. of constellations that reflects a "root-edge" decomposition. You can hope to do it by controlling the variables $\mathbf{p}$ and $\mathbf{q}$ and $(u_i)$.

– do it. And do it also for the non-orientable case.

– there is (seems to be) a natural way to put the $b$-parameter in these PDEs.

$2/2$ – If you are a Jack polynomial expert, you know that there are nice rules, such as the Pieri rule, or the $\alpha$-hook-content formula in [Stanley'89]. Also the Laplace Beltrami operator acts nicely on Jack polynomials.
Use your creativity + commutator magic (applying these rules in all sort of orders) to construct by induction a set of PDEs that cancel the function $\tau_b^{(k)}$.

$3/2$ – Suffer proving that the combinatorial (explicit) PDEs and the recursively defined (Lax type) PDEs are in fact the same.

This part of the proof is long *and* difficult, at least in the way to do it. For $b \in \{0, 1\}$ we have a combinatorial proof. We develop some sort of operator calculus that "lifts the combinatorial proof" to the world of differential operators, and the lifted proof works for general $b$.

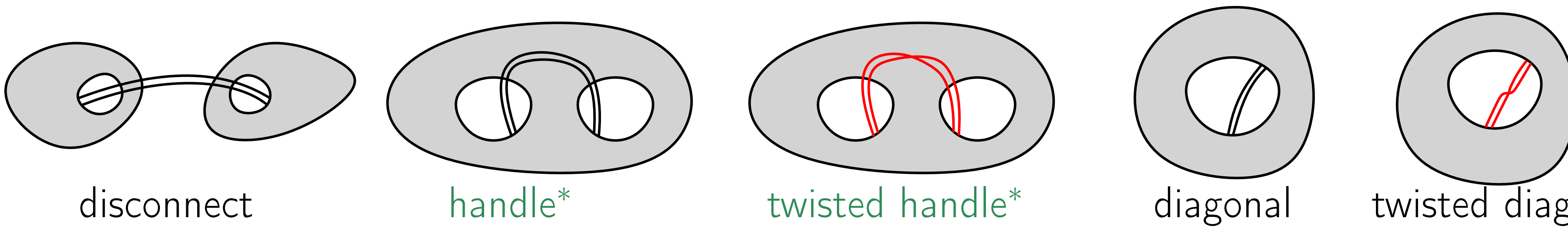
# The combinatorial equations for $k = 1$ (bipartite maps) -I

# The combinatorial equations for $k = 1$ (bipartite maps) -I

- Given a rooted bipartite (say) map, perform a root-edge decomposition:
  - remove the root
  - iterate on each remaining connected component

# The combinatorial equations for $k=1$ (bipartite maps) -I

- Given a rooted bipartite (say) map, perform a root-edge decomposition:
  - remove the root
  - iterate on each remaining connected component
- When you remove an edge there are several topological situations:

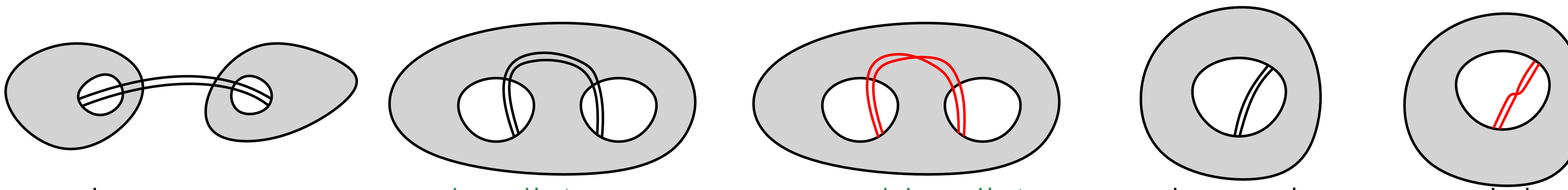



weights: $1$ $1$ $b$ $1$ $b$

these two cases happen at least once in the decomposition iff the starting map is non-orientable

# The combinatorial equations for $k=1$ (bipartite maps) -I

- Given a rooted bipartite (say) map, perform a root-edge decomposition:
  - remove the root
  - iterate on each remaining connected component
- When you remove an edge there are several topological situations:



disconnect — handle* — twisted handle* — diagonal — twisted diagonal

weights: $1$ $1$ $b$ $1$ $b$

these two cases happen at least once in the decomposition iff the starting map is non-orientable

- Claim: the following operator takes all these cases into account:

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j \frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i \frac{i\partial}{\partial y_i},$$

($y_i$: root-face of degree $i$; $p_i$: non-root-face of degree $i$; also extra weight $1/(1+b)$ per cc.)

# The combinatorial equations for $k=1$ (bipartite maps) - II

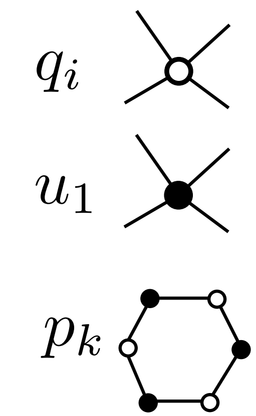
$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i \frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i \frac{\partial}{\partial y_i},$$

- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

$$m\frac{\partial}{q_m}F = t^m \cdot \Theta_Y\Big(Y_+\prod_{i=1}^{k}(\Lambda_Y + u_i)\Big)^m \frac{y_0}{1+b}F.$$

mark a vertex of degree $m$



# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i\frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i\frac{\partial}{\partial y_i},$$

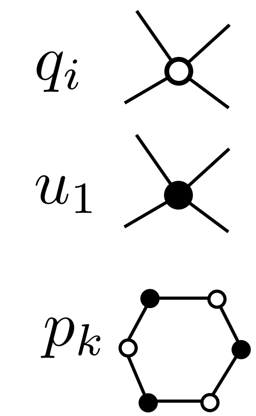
- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

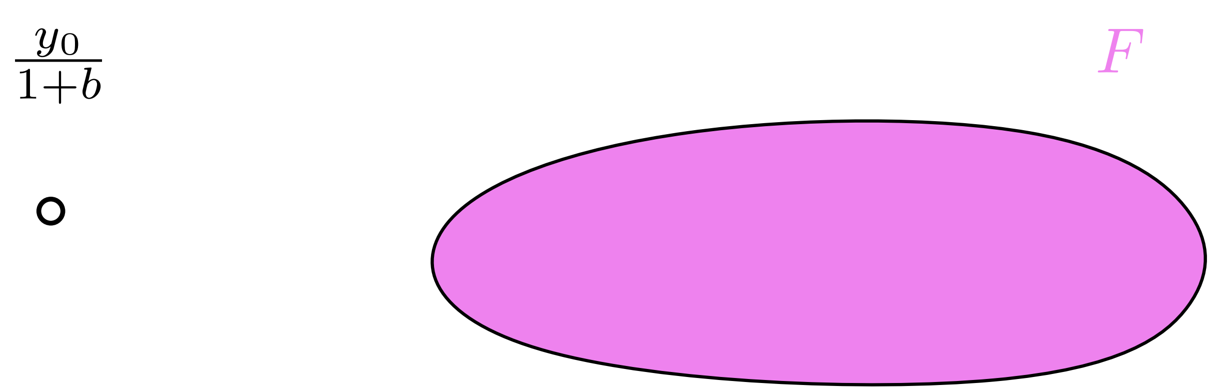
$$m\frac{\partial}{q_m}F = t^m\cdot\Theta_Y\Big(Y_+\prod_{i=1}^{k}(\Lambda_Y+u_i)\Big)^m\frac{y_0}{1+b}F.$$

take a smaller map

build a "fake" root face of degree 0 (track root-face degree with variables $y_i$)

mark a vertex of degree $m$



# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot \sum_{i\geq 0} y_i \frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i \frac{\partial}{\partial y_i},$$

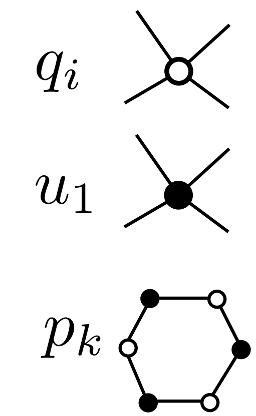
- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

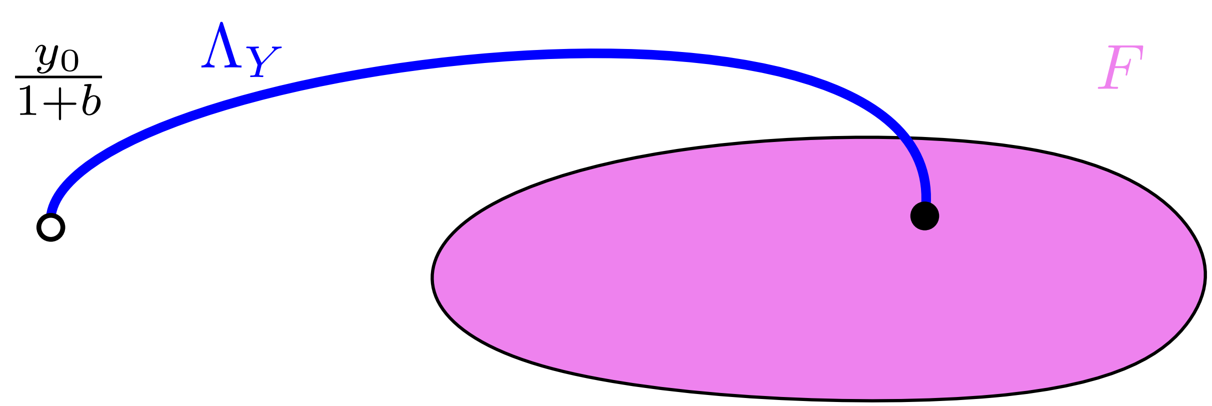
$$m\frac{\partial}{q_m}F = t^m\cdot\Theta_Y\Big(Y_+\underbrace{\prod_{i=1}^{k}(\Lambda_Y+u_i)}_{\text{build a } k\text{-hyperedge inside the root face}}\Big)^m\frac{y_0}{1+b}F.$$

mark a vertex of degree $m$

build a "fake" root face of degree 0 (track root-face degree with variables $y_i$)

take a smaller map



# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i\frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i\frac{\partial}{\partial y_i},$$

- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

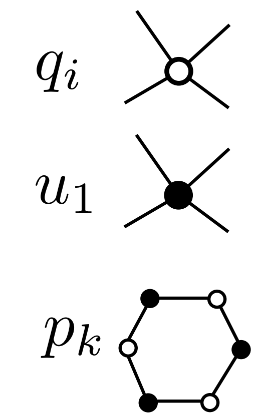
$$m\frac{\partial}{q_m}F = t^m \cdot \Theta_Y\Big(\underbrace{Y_+\prod_{i=1}^{k}(\Lambda_Y+u_i)}_{\text{build a }k\text{-hyperedge inside the root face}}\Big)^m \frac{y_0}{1+b}F.$$

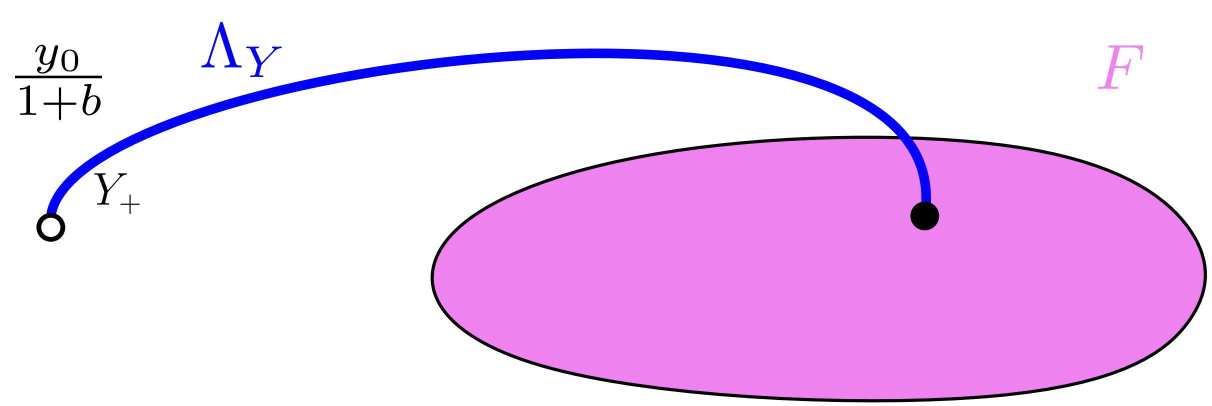
mark a vertex of degree $m$

build a $k$-hyperedge inside the root face

build a "fake" root face of degree 0 (track root-face degree with variables $y_i$)

take a smaller map



# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i\frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i\frac{\partial}{\partial y_i},$$

- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

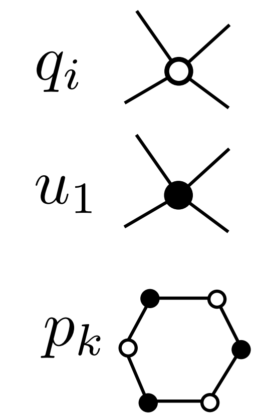
$$m\frac{\partial}{q_m}F = t^m\cdot\Theta_Y\Big(Y_+\prod_{i=1}^{k}(\Lambda_Y+u_i)\Big)^m\frac{y_0}{1+b}F.$$

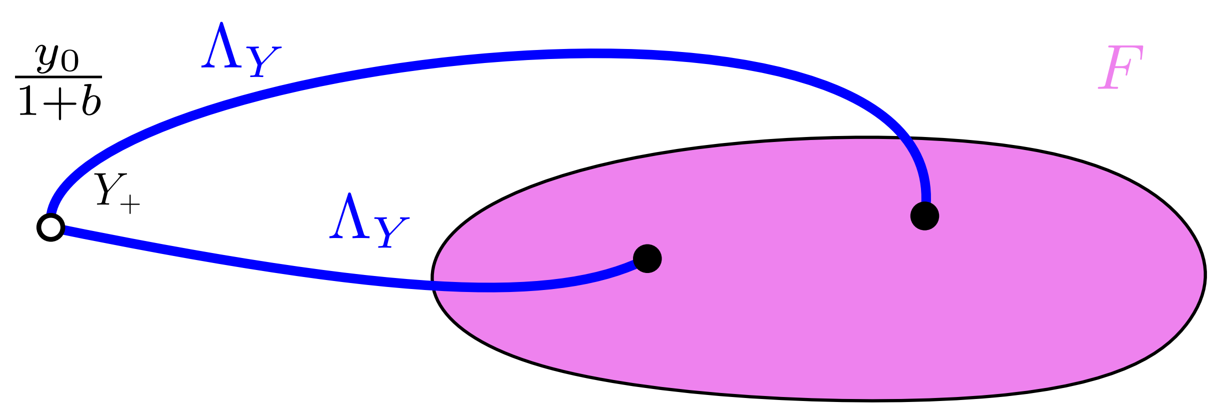
mark a vertex of degree $m$

build a $k$-hyperedge inside the root face

build a "fake" root face of degree 0 (track root-face degree with variables $y_i$)

take a smaller map



# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i\frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i\frac{\partial}{\partial y_i},$$

- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

$$m\frac{\partial}{q_m}F = t^m\cdot\Theta_Y\Big(Y_+\underbrace{\prod_{i=1}^{k}(\Lambda_Y+u_i)}_{\text{build a } k\text{-hyperedge inside the root face}}\Big)^m\frac{y_0}{1+b}F.$$

mark a vertex of degree $m$

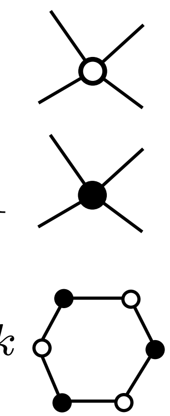
take a smaller map

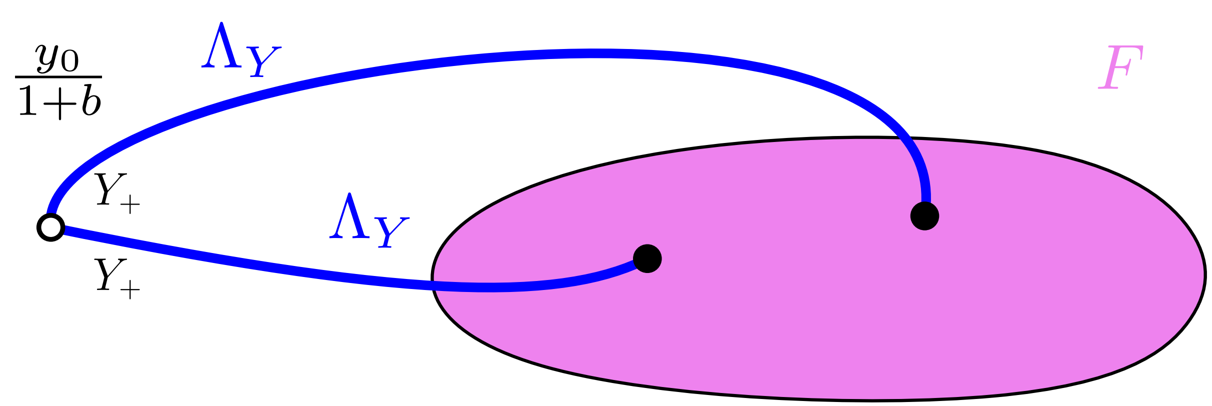
build a "fake" root face of degree $0$ (track root-face degree with variables $y_i$)

$q_i$

$u_1$

$p_k$





# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i\frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i\frac{\partial}{\partial y_i},$$

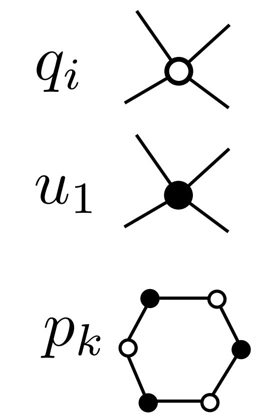
- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

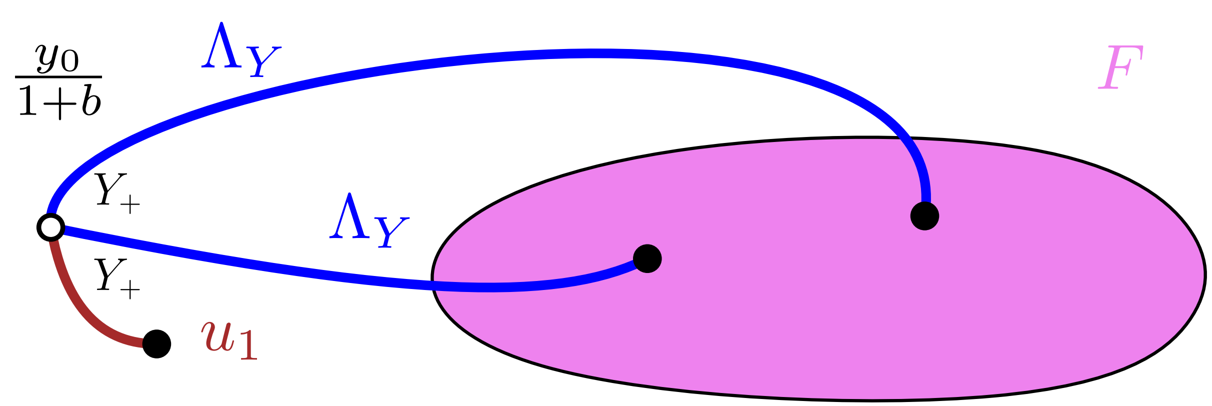
$$m\frac{\partial}{q_m}F = t^m\cdot\Theta_Y\Big(Y_+\underbrace{\prod_{i=1}^{k}(\Lambda_Y+u_i)}_{\text{build a } k\text{-hyperedge inside the root face}}\Big)^m\frac{y_0}{1+b}F.$$

mark a vertex of degree $m$

take a smaller map

build a "fake" root face of degree $0$ (track root-face degree with variables $y_i$)



# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i\frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i\frac{\partial}{\partial y_i},$$

- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

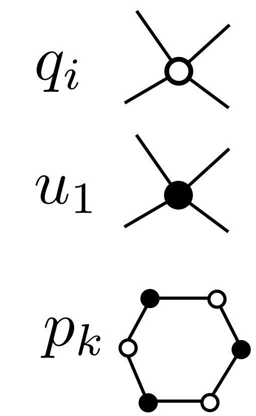
$$\underbrace{m\frac{\partial}{q_m}F}_{\text{mark a vertex of degree } m} = t^m\cdot\Theta_Y\Big(Y_+\underbrace{\prod_{i=1}^{k}(\Lambda_Y+u_i)}_{\text{build a } k\text{-hyperedge inside the root face}}\Big)^m\frac{y_0}{1+b}F.$$

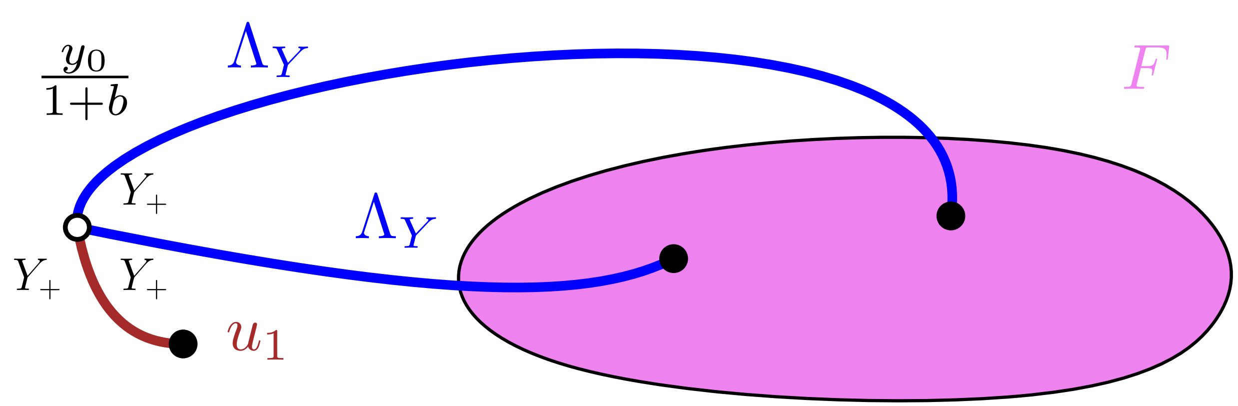
mark a vertex of degree $m$

build a $k$-hyperedge inside the root face

build a "fake" root face of degree 0 (track root-face degree with variables $y_i$)

take a smaller map



# The combinatorial equations for $k = 1$ (bipartite maps) - II

$$\Lambda_Y := (1+b) \sum_{i,j\geq 1} y_{i+j-1} \frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1} p_j \frac{\partial}{\partial y_{i+j-1}} + b \cdot \sum_{i \geq 0} y_i \frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1} \frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i \geq 0} p_i \frac{\partial}{\partial y_i},$$

- Let $F(\mathbf{p}, \mathbf{q}, u_1)$ be the g.f. of labelled bipartite maps. Then:

$$m \frac{\partial}{q_m} F = t^m \cdot \Theta_Y \Big(Y_+ \prod_{i=1}^{k} (\Lambda_Y + u_i)\Big)^m \frac{y_0}{1+b} F.$$

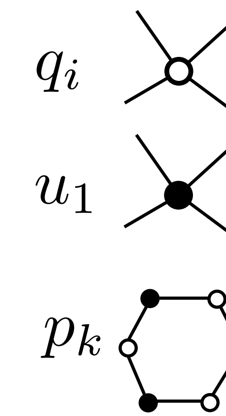
take a smaller map

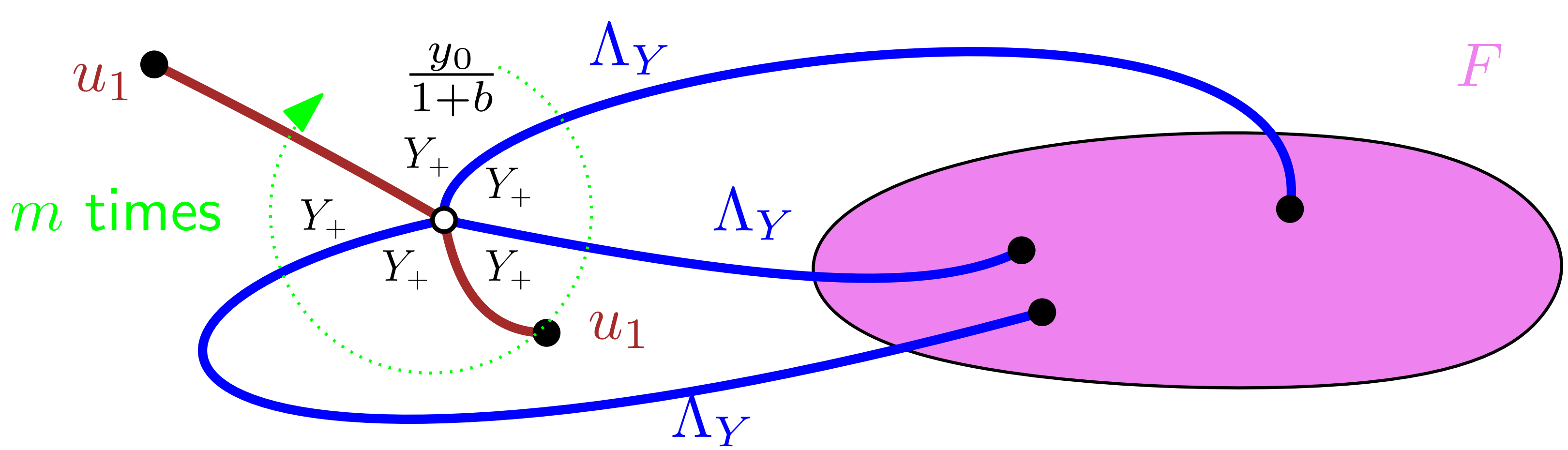
mark a vertex of degree $m$

build a $k$-hyperedge inside the root face

do it $m$ times to create "root" vertex of degree $m$

build a "fake" root face of degree 0 (track root-face degree with variables $y_i$)



# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i\frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \qquad \Theta_Y := \sum_{i\geq 0} p_i\frac{\partial}{\partial y_i},$$

- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

$$m\frac{\partial}{q_m}F = t^m\cdot\Theta_Y\Big(Y_+\prod_{i=1}^{k}(\Lambda_Y+u_i)\Big)^m\frac{y_0}{1+b}F.$$

mark a vertex of degree $m$

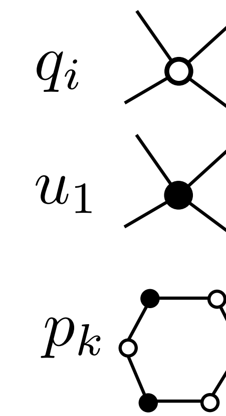
anonymise root face

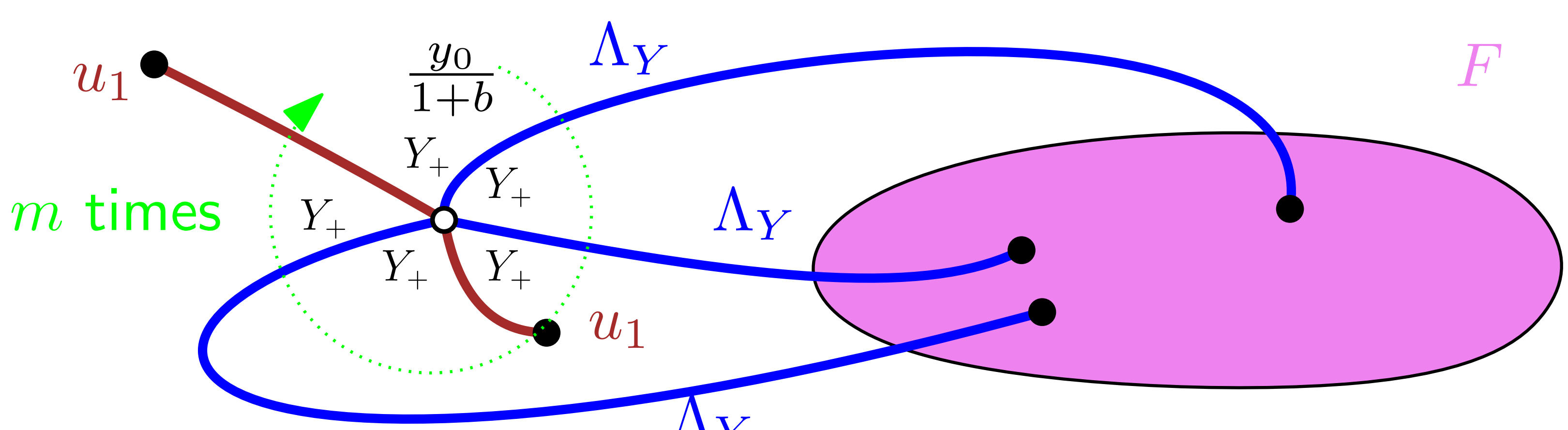
build a $k$-hyperedge inside the root face

do it $m$ times to create "root" vertex of degree $m$

build a "fake" root face of degree 0 (track root-face degree with variables $y_i$)

take a smaller map



# The combinatorial equations for $k=1$ (bipartite maps) - II

$$\Lambda_Y := (1+b)\sum_{i,j\geq 1} y_{i+j-1}\frac{i\partial^2}{\partial p_i \partial y_{j-1}} + \sum_{i,j\geq 1} y_{i-1}p_j\frac{\partial}{\partial y_{i+j-1}} + b\cdot\sum_{i\geq 0} y_i\frac{i\partial}{\partial y_i},$$

$$Y_+ := \sum_{i\geq 0} y_{i+1}\frac{\partial}{\partial y_i}, \quad \Theta_Y := \sum_{i\geq 0} p_i \frac{\partial}{\partial y_i},$$

- Let $F(\mathbf{p},\mathbf{q},u_1)$ be the g.f. of labelled bipartite maps. Then:

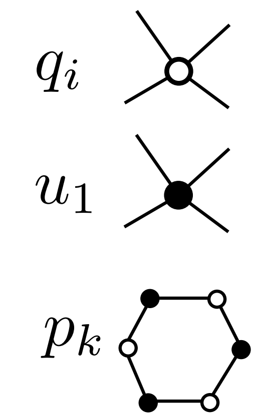
$$m\frac{\partial}{q_m}F = t^m \cdot \Theta_Y \Big(\underbrace{\underbrace{Y_+ \prod_{i=1}^{k}(\Lambda_Y + u_i)}_{\text{build a } k\text{-hyperedge inside the root face}}\Big)^m}_{\text{do it } m \text{ times to create ``root'' vertex of degree } m}\frac{y_0}{1+b}F.$$

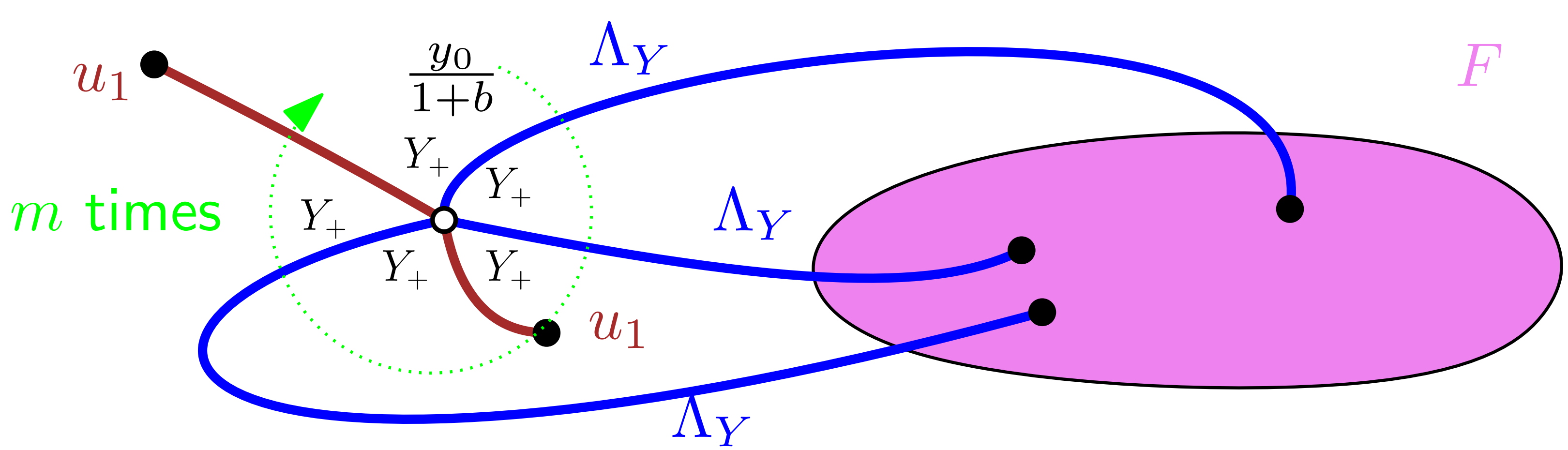
mark a vertex of degree $m$

anonymise root face

build a "fake" root face of degree $0$ (track root-face degree with variables $y_i$)

take a smaller map




Note: this corresponds to a randomized decomposition and proves only rational weights. One can be more precise and do a deterministic rooted decomposition instead.

# How do the “Lax pair” equations look like?

- Define the Laplace-Beltrami operator:

$$D_\alpha = \frac{1}{2}\left( (1+b)\sum_{i,j\geq 1} p_{i+j}\frac{ij\partial^2}{\partial p_i \partial p_j} + \sum_{i,j\geq 1} p_i p_j \frac{(i+j)\partial}{\partial p_{i+j}} + b\cdot \sum_{i\geq 1} p_i \frac{i(i-1)\partial}{\partial p_i}\right).$$

Fact: it acts diagonally on Jacks (multiplies by the sum of $\alpha$-contents).

# How do the "Lax pair" equations look like?

- Define the Laplace-Beltrami operator:

$$D_\alpha = \frac{1}{2}\left( (1+b) \sum_{i,j\geq 1} p_{i+j} \frac{ij\partial^2}{\partial p_i \partial p_j} + \sum_{i,j\geq 1} p_i p_j \frac{(i+j)\partial}{\partial p_{i+j}} + b \cdot \sum_{i\geq 1} p_i \frac{i(i-1)\partial}{\partial p_i} \right).$$

Fact: it acts diagonally on Jacks (multiplies by the sum of $\alpha$-contents).

- Define the operators:

$$A_1 = p_1/(1+b) \quad , \quad A_{j+1} = [D_\alpha, A_j], \text{ , for } j \geq 1$$

$$\Omega_Y^{(k)} = \textstyle\sum_{1\leq i \leq k+1} A_{i+1} e_{k+1-i}(u_1, \ldots, u_k)$$

$$B_1^{(k)} = \sum_{1\leq i\leq k+1} A_i e_{k+1-i}(u_1, \ldots, u_k) \quad , \quad B_{m+1}^{(k)} = [\Omega_Y^{(k)}, B_m^{(k)}] \text{ , for } m \geq 1.$$

- Then one has the equation, for $m \geq 1$:

$$m! \frac{\partial}{\partial q_m} \tau_b^{(k)} = t^m B_m^{(k)} \tau_b^{(k)}$$

(proof uses $\alpha$-Pieri rule + $\alpha$-hook content rule+ Laplace-Beltrami rule + commutator magic)

# How do the "Lax pair" equations look like?

- Define the Laplace-Beltrami operator:

$$D_\alpha = \frac{1}{2}\left( (1+b) \sum_{i,j\geq 1} p_{i+j} \frac{ij\partial^2}{\partial p_i \partial p_j} + \sum_{i,j\geq 1} p_i p_j \frac{(i+j)\partial}{\partial p_{i+j}} + b \cdot \sum_{i\geq 1} p_i \frac{i(i-1)\partial}{\partial p_i} \right).$$

Fact: it acts diagonally on Jacks (multiplies by the sum of $\alpha$-contents).

- Define the operators:

$$A_1 = p_1/(1+b) \quad , \boxed{A_{j+1} = [D_\alpha, A_j], \text{ , for } j \geq 1}$$

$$\Omega_Y^{(k)} = \sum_{1\leq i\leq k+1} A_{i+1} e_{k+1-i}(u_1, \ldots, u_k)$$

$$B_1^{(k)} = \sum_{1\leq i\leq k+1} A_i e_{k+1-i}(u_1, \ldots, u_k) \quad , \boxed{B_{m+1}^{(k)} = [\Omega_Y^{(k)}, B_m^{(k)}] \text{ , for } m \geq 1.}$$

- Then one has the equation, for $m \geq 1$:

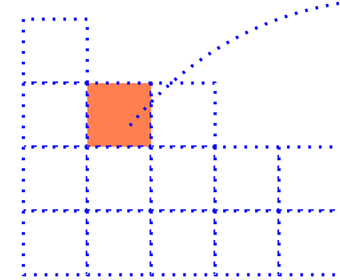
$$\boxed{m! \frac{\partial}{\partial q_m} \tau_b^{(k)} = t^m B_m^{(k)} \tau_b^{(k)}}$$

(proof uses $\alpha$-Pieri rule + $\alpha$-hook content rule+ Laplace-Beltrami rule + commutator magic)

# How to construct PDE's for $\tau_b^{(k)}$? ($m = 1, k = 1$, Schur case).

- Hook content formula:

$$\tilde{s}_\lambda(\underline{u}) = \prod_{\square \in \lambda} (u + c(\square))$$

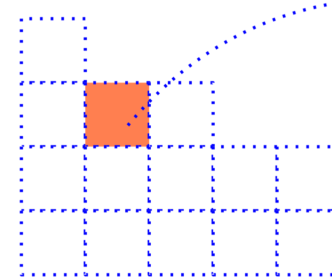


$c(\square) := x(\square) - y(\square) = -1$

$$\longrightarrow B(\mathbf{p}, \mathbf{q}, u) = \sum_{\lambda} \Big( \prod_{\square \in \lambda} (u + c(\square)) \Big) s_\lambda(\mathbf{p}) s_\lambda(\mathbf{q})$$

# How to construct PDE's for $\tau_b^{(k)}$? ($m = 1, k = 1$, Schur case).

- Hook content formula:



$c(\square) := x(\square) - y(\square) = -1$

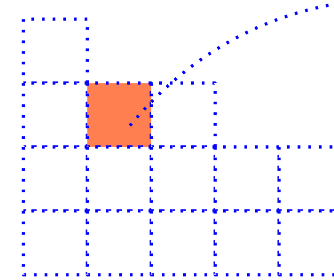
$$\tilde{s}_\lambda(\underline{u}) = \prod_{\square \in \lambda} (u + c(\square))$$

$$\longrightarrow B(\mathbf{p}, \mathbf{q}, u) = \sum_\lambda \Big( \prod_{\square \in \lambda} (u + c(\square)) \Big) s_\lambda(\mathbf{p}) s_\lambda(\mathbf{q})$$

- Pieri rule: multiply by $p_1$ adds a box to a Schur function: $p_1 s_\lambda(\mathbf{p}) = \sum_{\square} s_{\lambda \uplus \square}(\mathbf{p})$.
  $\partial/\partial p_1$ removes a box.

# How to construct PDE's for $\tau_b^{(k)}$? ($m=1, k=1$, Schur case).

- Hook content formula:



$c(\square) := x(\square) - y(\square) = -1$

$$\tilde{s}_\lambda(\underline{u}) = \prod_{\square\in\lambda} (u + c(\square))$$

$$\longrightarrow B(\mathbf{p}, \mathbf{q}, u) = \sum_\lambda \Big( \prod_{\square\in\lambda} (u + c(\square)) \Big) s_\lambda(\mathbf{p}) s_\lambda(\mathbf{q})$$

- Pieri rule: multiply by $p_1$ adds a box to a Schur function: $p_1 s_\lambda(\mathbf{p}) = \sum_\square s_{\lambda\uplus\square}(\mathbf{p})$.

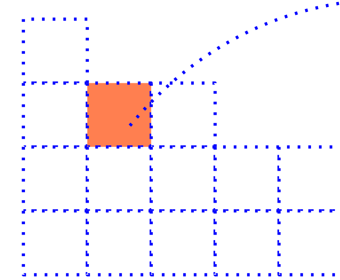
  $\partial/\partial p_1$ removes a box.

- Another rule $D_\alpha s_\lambda = \Big( \sum_{\square\in\lambda} c(\square) \Big) s_\lambda$

$$\longrightarrow [D_\alpha, p_1] s_\lambda = \sum_\square c(\square) s_{\lambda\uplus\square}(\mathbf{p})$$

## How to construct PDE's for $\tau_b^{(k)}$? ($m=1, k=1$, Schur case).

- Hook content formula:



$$c(\square) := x(\square) - y(\square) = -1$$

$$\tilde{s}_\lambda(\underline{u}) = \prod_{\square\in\lambda}(u + c(\square))$$

$$\longrightarrow B(\mathbf{p},\mathbf{q},u) = \sum_\lambda \Big(\prod_{\square\in\lambda}(u+c(\square))\Big) s_\lambda(\mathbf{p}) s_\lambda(\mathbf{q})$$

- Pieri rule: multiply by $p_1$ adds a box to a Schur function: $p_1 s_\lambda(\mathbf{p}) = \sum_{\square} s_{\lambda\uplus\square}(\mathbf{p})$.
  $\partial/\partial p_1$ removes a box.

- Another rule $D_\alpha s_\lambda = \Big(\sum_{\square\in\lambda} c(\square)\Big) s_\lambda$

$$\longrightarrow [D_\alpha, p_1] s_\lambda = \sum_{\square} c(\square) s_{\lambda\uplus\square}(\mathbf{p})$$

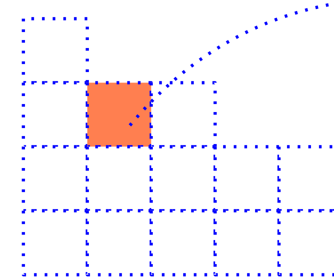
- From this we obtain a differential equation for our function!!!

$$\frac{\partial}{\partial q_1} B(\mathbf{p},\mathbf{q},u) = (u p_1 + [D_\alpha, p_1]) B(\mathbf{p},\mathbf{q},u)$$

# How to construct PDE's for $\tau_b^{(k)}$? $(m = 1, k = 1$, Schur case).

- Hook content formula:

$$\tilde{s}_\lambda(\underline{u}) = \prod_{\square \in \lambda} (u + c(\square))$$



$c(\square) := x(\square) - y(\square) = -1$

$$\longrightarrow B(\mathbf{p}, \mathbf{q}, u) = \sum_\lambda \Big( \prod_{\square \in \lambda} (u + c(\square)) \Big) s_\lambda(\mathbf{p}) s_\lambda(\mathbf{q})$$

- Pieri rule: multiply by $p_1$ adds a box to a Schur function: $p_1 s_\lambda(\mathbf{p}) = \sum_{\square} s_{\lambda \uplus \square}(\mathbf{p})$.
  $\partial / \partial p_1$ removes a box.

- Another rule $D_\alpha s_\lambda = \Big( \sum_{\square \in \lambda} c(\square) \Big) s_\lambda$

$$\longrightarrow [D_\alpha, p_1] s_\lambda = \sum_{\square} c(\square) s_{\lambda \uplus \square}(\mathbf{p})$$

- From this we obtain a differential equation for our function!!!

$$\boxed{\frac{\partial}{\partial q_1} B(\mathbf{p}, \mathbf{q}, u) = (u p_1 + [D_\alpha, p_1]) B(\mathbf{p}, \mathbf{q}, u)}$$

The miracle is that (by computing the commutator explicitly) this is the same equation as the one we wrote for maps in the prevous slides!

La preuve que le miracle a lieu et que les opérateurs sont les mêmes pour tout $k$ et $m$ est une partie de l'histoire que j'omets...

Merci!