# Tournaments and slide rules for $\omega$ and $\psi$ class products on $\overline{M}_{0,n}$

Sean Griffin
UC Davis

FPSAC, IISc Bangalore
July 22, 2022

Joint work with Maria Gillespie (Colorado State Univ.) and Jake Levinson (Simon Fraser Univ.)

# Asymmetric multinomial coefficients

# Asymmetric multinomial coefficients

Recall the standard recursion for multinomial coefficients:

$$\binom{n}{k_1, \ldots, k_n} = \sum_{1 \le i \le n} \binom{n-1}{k_1, \ldots, k_{i-1}, k_i - 1, k_{i+1}, \ldots, k_n}.$$

Cavalieri–Gillespie–Monin introduced the *asymmetric multinomial coeffs*:

# Asymmetric multinomial coefficients

Recall the standard recursion for multinomial coefficients:

$$\binom{n}{k_1, \ldots, k_n} = \sum_{1 \le i \le n} \binom{n-1}{k_1, \ldots, k_{i-1}, k_i - 1, k_{i+1}, \ldots, k_n}.$$

Cavalieri–Gillespie–Monin introduced the *asymmetric multinomial coeffs*:

Fix $n$ and $\mathbf{k} = (k_1, \ldots, k_n) \vDash n$

Let $i(\mathbf{k}) \coloneqq$ index of rightmost 0 in $\mathbf{k}$

For $j > i(\mathbf{k})$, define $\widetilde{\mathbf{k}}_j$:

# Asymmetric multinomial coefficients

Recall the standard recursion for multinomial coefficients:

$$\binom{n}{k_1, \ldots, k_n} = \sum_{1 \le i \le n} \binom{n-1}{k_1, \ldots, k_{i-1}, k_i - 1, k_{i+1}, \ldots, k_n}.$$

Cavalieri–Gillespie–Monin introduced the *asymmetric multinomial coeffs*:

Fix $n$ and $\mathbf{k} = (k_1, \ldots, k_n) \vDash n$

Let $i(\mathbf{k}) \coloneqq$ index of rightmost 0 in $\mathbf{k}$

For $j > i(\mathbf{k})$, define $\widetilde{\mathbf{k}}_j$: Subtract $1$ from $k_j$, then
Delete the right-most 0 of the result.

**Example:**

$$n = 10,\ \mathbf{k} = (1, 0, 0, 1, 0, 1, 2, 1, 3, 1),\ i(\mathbf{k}) = 5$$

$j = 8$ $\qquad$ $j = 9$

$$\widetilde{\mathbf{k}}_8 = (1, 0, 0, 1, 0, 1, 2, 3, 1) \qquad \widetilde{\mathbf{k}}_9 = (1, 0, 0, 1, 1, 2, 1, 2, 1)$$

# Asymmetric multinomial coefficients

Let $i(\mathbf{k}) :=$ index of rightmost 0 in $\mathbf{k}$

For $j > i(\mathbf{k})$, define $\widetilde{\mathbf{k}}_j$: Subtract $1$ from $k_j$, then
Delete the right-most 0 of the result.

**Asymmetric string recursion:**

$$\left\langle {n \atop \mathbf{k}} \right\rangle := \sum_{j>i(\mathbf{k})} \left\langle {n-1 \atop \widetilde{\mathbf{k}}_j} \right\rangle, \qquad \left\langle {1 \atop 1} \right\rangle := 1.$$

**Example:**

$$\left\langle {6 \atop 1,0,1,2,1,1} \right\rangle = \left\langle {5 \atop 1,0,2,1,1} \right\rangle + \left\langle {5 \atop 1,1,1,1,1} \right\rangle + \left\langle {5 \atop 1,0,1,2,1} \right\rangle + \left\langle {5 \atop 1,0,1,2,1} \right\rangle.$$

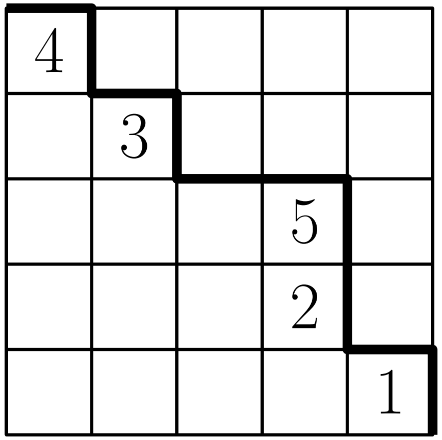
Cavalieri–Gillespie–Monin (2019): $\left\langle {n \atop \mathbf{k}} \right\rangle$ is the *multidegree* of a natural embedding of $\overline{M}_{0,n+3}$ into a product of projective spaces. Equivalently, it's a $0$-dimensional product of $\omega$ classes (more later).

# Column-restricted parking functions

$\mathbf{k}$ is called *reverse Catalan* if $k_n + \cdots + k_{n-i+1} \geq i$ for all $i$.

**Theorem (Cavalieri–Gillespie–Monin, 2019)**

$\left\langle {n \atop \mathbf{k}} \right\rangle \neq 0$ if and only if $\mathbf{k}$ is reverse Catalan.
In this case, $\left\langle {n \atop \mathbf{k}} \right\rangle$ is the number of *column-restricted* parking functions with $k_i$ labels in column $i$.



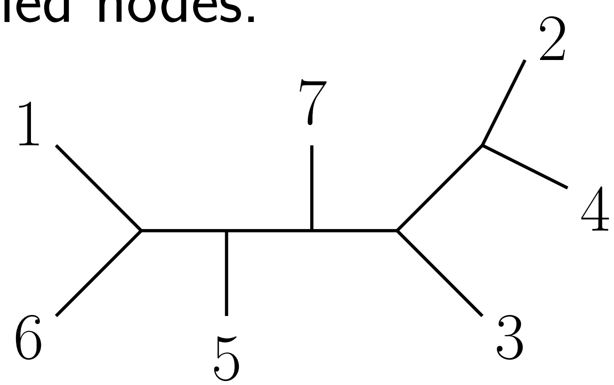

$\mathbf{k} = (1, 1, 0, 2, 1)$

**Theorem (Cavalieri–Gillespie–Monin, 2019)**

Total # column-restricted parking functions on $n$ is $(2n-1)!!$.

# Two more combinatorial interpretations

The geometry of $\overline{M}_{0,n}$ is intimately related to the combinatorics of *trivalent trees*: The *boundary points* ($0$-diml' strata) are indexed by trivalent trees on $n$ labeled nodes.




We give two new formulas for $\left\langle {n \atop \mathbf{k}} \right\rangle$:

(1) *Lazy tournaments*: An algorithm that partitions a set of trivalent trees into subsets that count $\left\langle {n \atop \mathbf{k}} \right\rangle$.

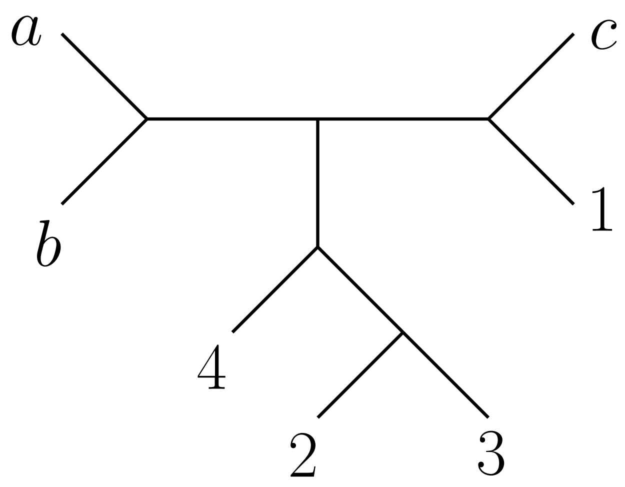
(2) *Slides:* An algorithm that directly generates a set of trivalent trees with cardinality $\left\langle {n \atop \mathbf{k}} \right\rangle$.

The two interpretations have different geometric properties.

# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

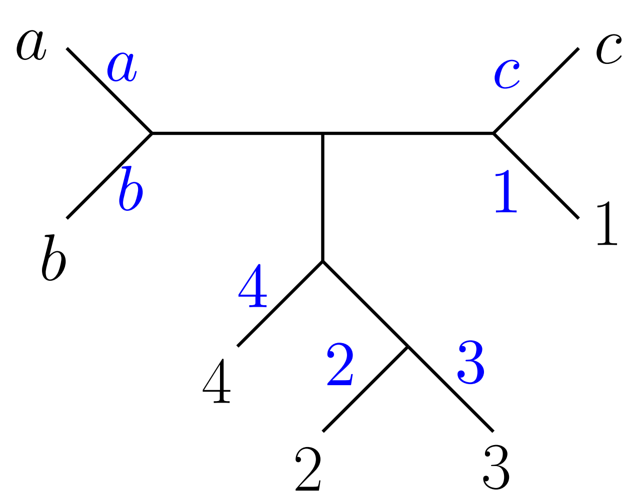


# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.

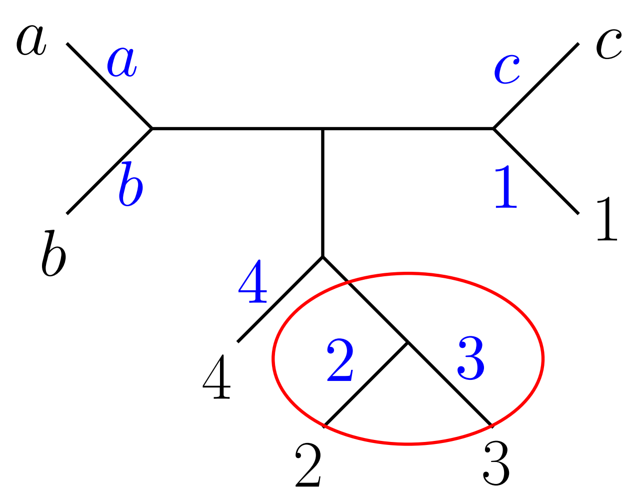


# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.



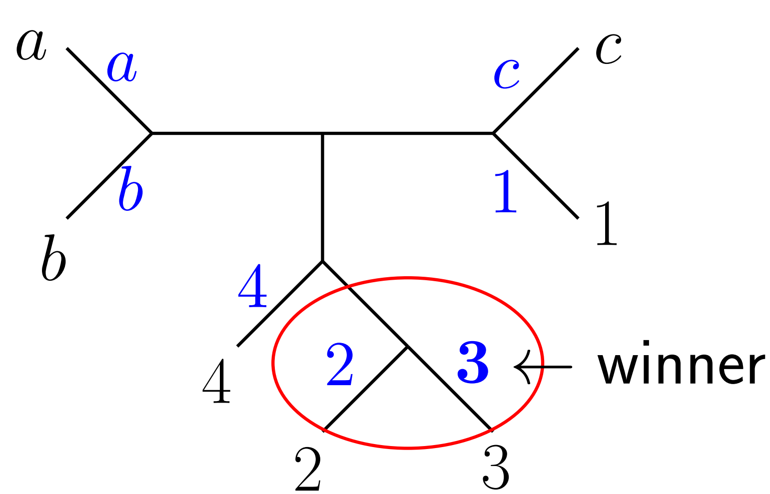

$i = 2, j = 3$

# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.
- (**Determine winner**) Larger label $j$ **wins** the round.



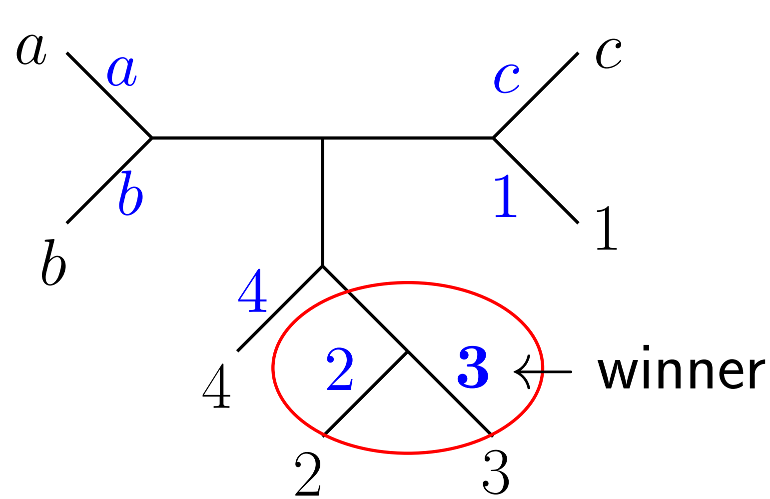

$i = 2, j = 3$

# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.
- (**Determine winner**) Larger label $j$ **wins** the round.
- (**Determine which advances to next round**) $j$ advances, unless it can be **lazy**: if the next edge is adjacent to some label $u > i$, then $i$ advances despite losing the round.



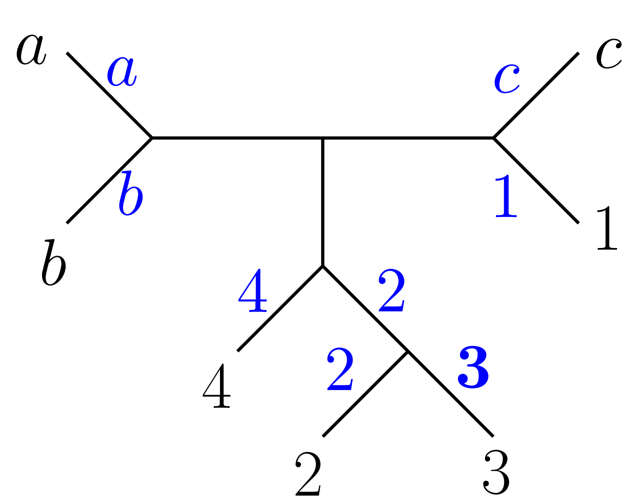

$i = 2, j = 3$
$u = 4$

# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.
- (**Determine winner**) Larger label $j$ **wins** the round.
- (**Determine which advances to next round**) $j$ advances, unless it can be **lazy**: if the next edge is adjacent to some label $u > i$, then $i$ advances despite losing the round.

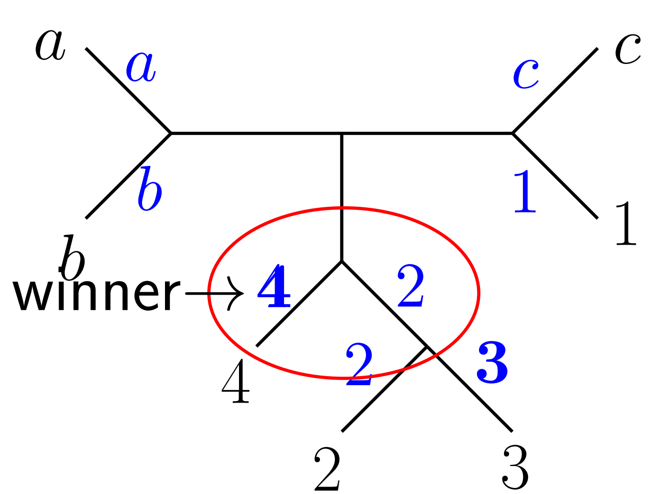


# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.
- (**Determine winner**) Larger label $j$ **wins** the round.
- (**Determine which advances to next round**) $j$ advances, unless it can be **lazy**: if the next edge is adjacent to some label $u > i$, then $i$ advances despite losing the round.

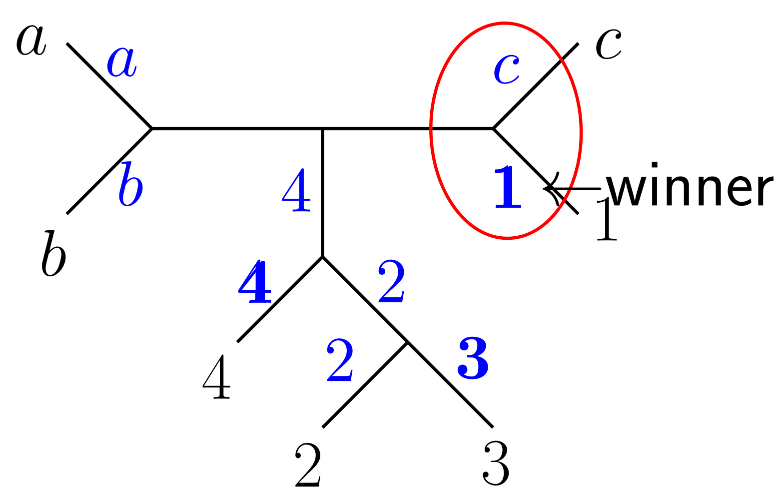


# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.
- (**Determine winner**) Larger label $j$ **wins** the round.
- (**Determine which advances to next round**) $j$ advances, unless it can be **lazy**: if the next edge is adjacent to some label $u > i$, then $i$ advances despite losing the round.

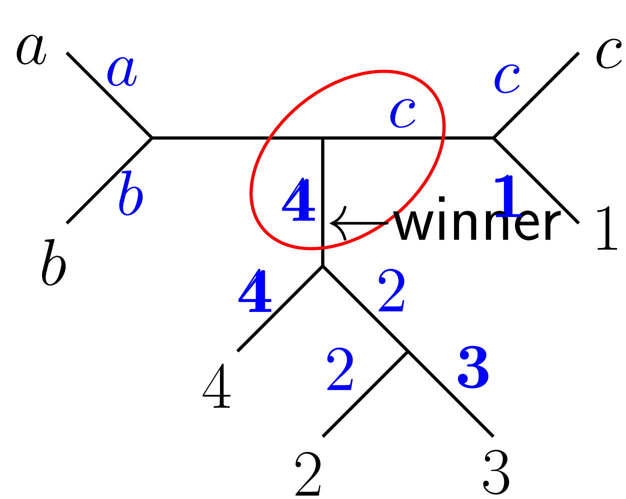


# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.
- (**Determine winner**) Larger label $j$ **wins** the round.
- (**Determine which advances to next round**) $j$ advances, unless it can be **lazy**: if the next edge is adjacent to some label $u > i$, then $i$ advances despite losing the round.

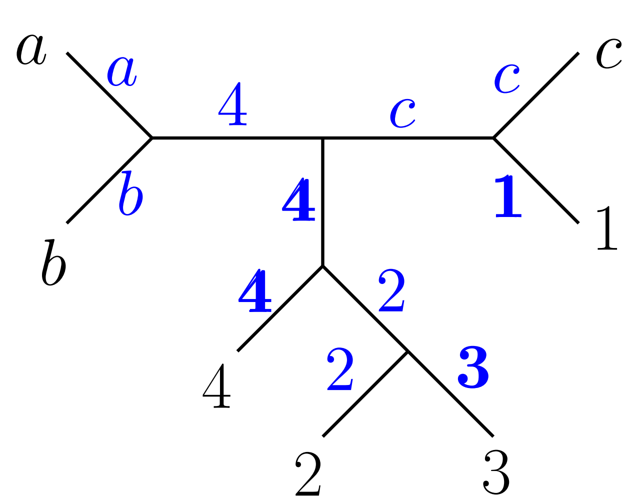


# Lazy tournaments

Let the labels be: $a < b < c < 1 < \cdots < n$

Suppose $T$ is a trivalent tree such that $a$ and $b$ are adjacent.
The *lazy tournament* of $T$ is the following edge labeling:

- (**Identify which pair "face off"**) The adjacent pair $i < j$ with largest smaller element $i$ "face off" first.
- (**Determine winner**) Larger label $j$ **wins** the round.
- (**Determine which advances to next round**) $j$ advances, unless it can be **lazy**: if the next edge is adjacent to some label $u > i$, then $i$ advances despite losing the round.



# Lazy tournaments

**Theorem (Gillespie-G.-Levinson, 2021)**

$\left\langle {n \atop \mathbf{k}} \right\rangle = \#$ trivalent trees $T$ such that $a$ and $b$ are adjacent and each $i$ wins $k_i$ many rounds of the lazy tournament of $T$.

The lazy tournament trees satisfy the asymmetric string recursion for $\left\langle {n \atop \mathbf{k}} \right\rangle$. The proof uses the "forgetting map" on $\overline{M}_{0,n+3}$ that forgets the label $i$ and contracts its edge.

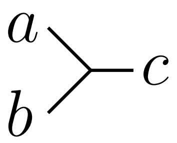
The $(2n-1)!!$ formula is an easy corollary:

$$\sum_{\mathbf{k}} \left\langle {n \atop \mathbf{k}} \right\rangle = \#\text{ trivalent trees such that } a \text{ and } b \text{ are adjacent}$$

$$= (2n-1)!!.$$

# Slide trees

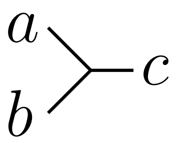
- Fix $\mathbf{k}$, start with unique trivalent tree on $a, b, c$:



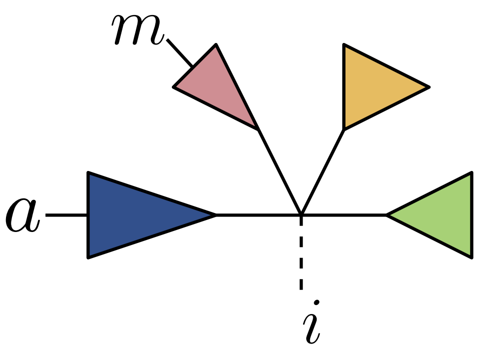

- At $i$th step, insert $i$ and perform $k_i$ *slides* at $i$:

# Slide trees

- Fix $\mathbf{k}$, start with unique trivalent tree on $a, b, c$:

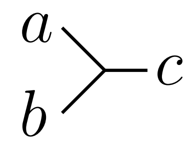



- At $i$th step, insert $i$ and perform $k_i$ *slides* at $i$:
  Let $m =$ min label not in the $a$ branch



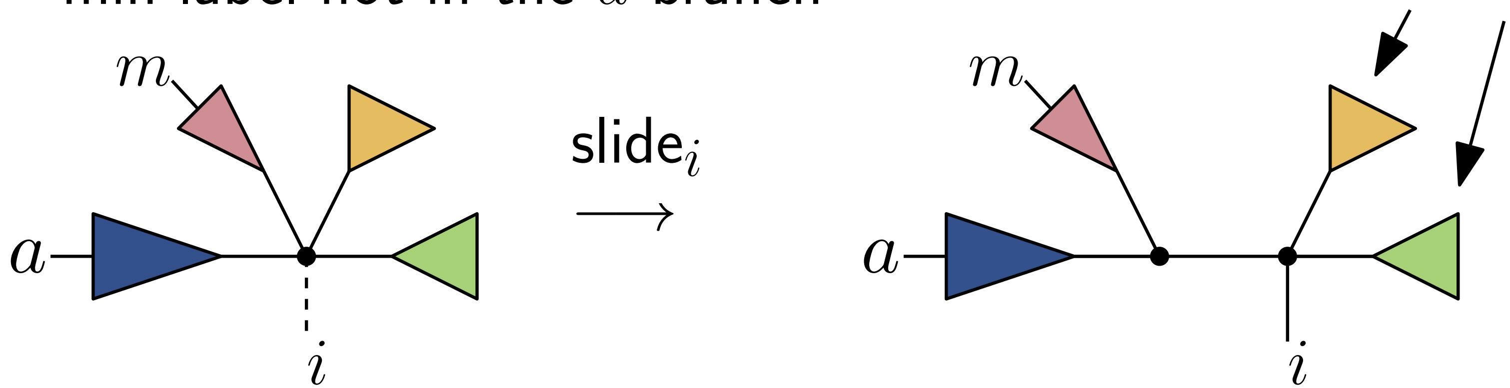
# Slide trees

- Fix $\mathbf{k}$, start with unique trivalent tree on $a, b, c$:



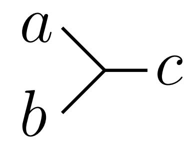

- At $i$th step, insert $i$ and perform $k_i$ *slides* at $i$:
  Let $m$ = min label not in the $a$ branch

distribute remaining branches

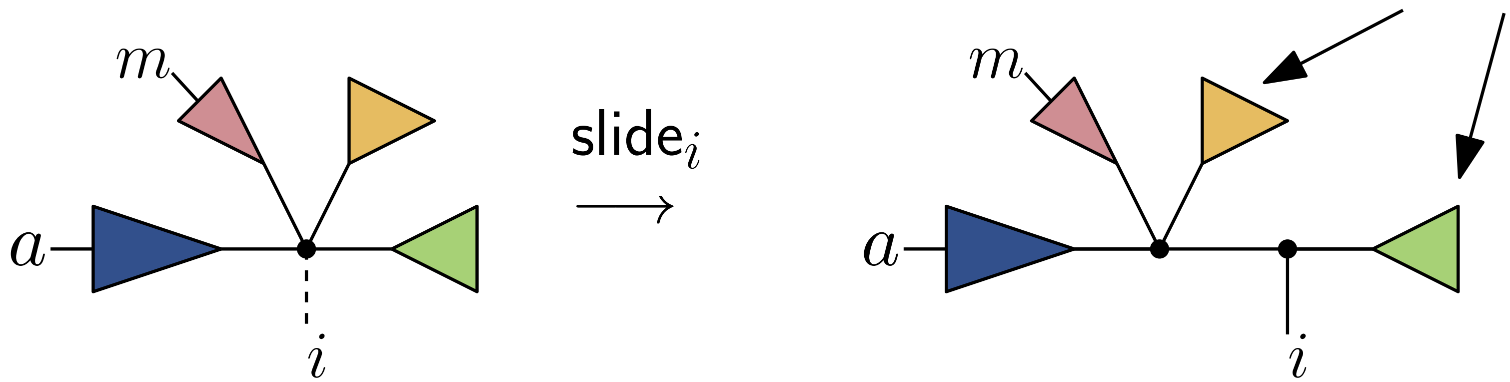


# Slide trees

- Fix $\mathbf{k}$, start with unique trivalent tree on $a, b, c$:

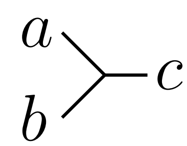



- At $i$th step, insert $i$ and perform $k_i$ *slides* at $i$:
  Let $m$ = min label not in the $a$ branch



# Slide trees

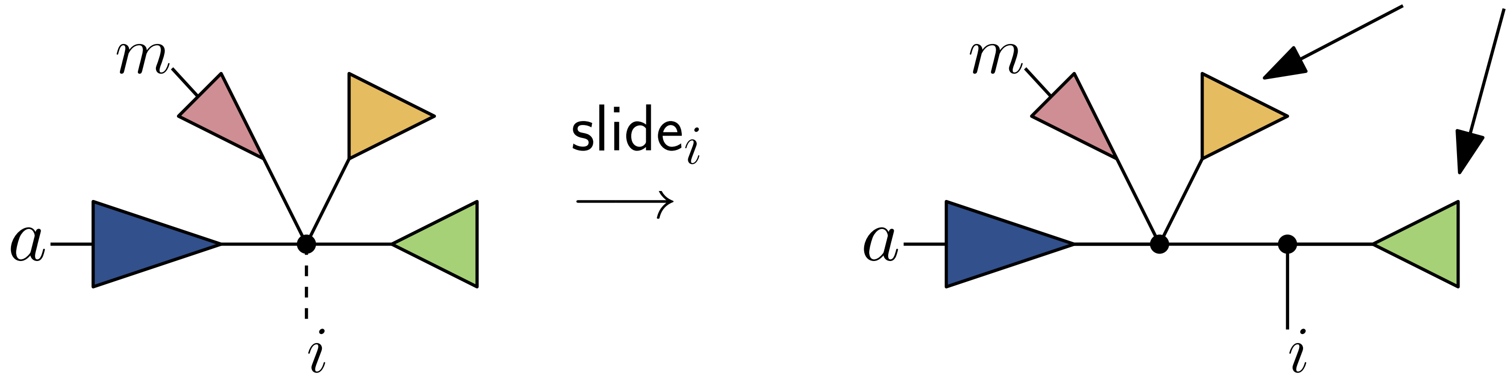
- Fix $\mathbf{k}$, start with unique trivalent tree on $a, b, c$:




- At $i$th step, insert $i$ and perform $k_i$ *slides* at $i$:

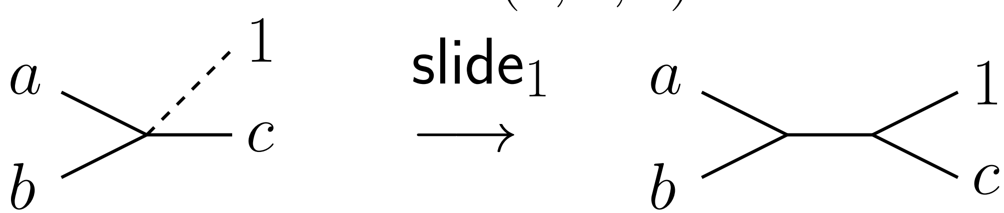
  Let $m$ = min label not in the $a$ branch

distribute remaining branches



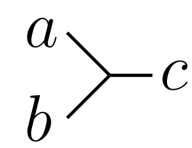

- The resulting trivalent trees are denoted by $\mathrm{Slide}^{\omega}(k_1, \ldots, k_n)$.

**Example:** Let $n = 3$ and $\mathbf{k} = (1, 0, 2)$.

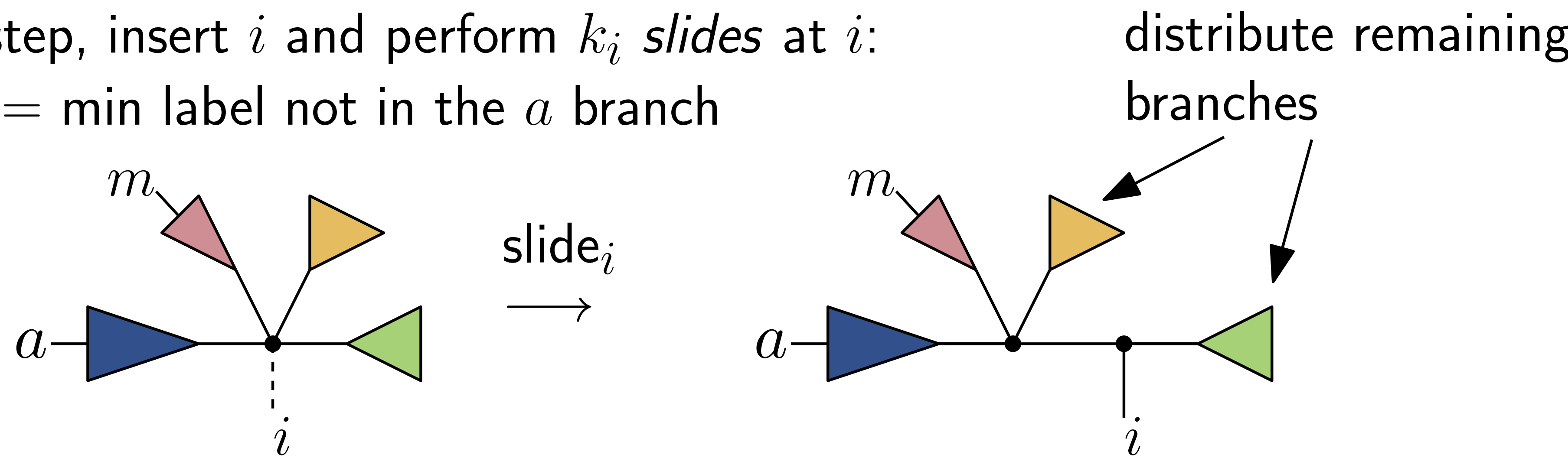


# Slide trees

- Fix $\mathbf{k}$, start with unique trivalent tree on $a, b, c$:

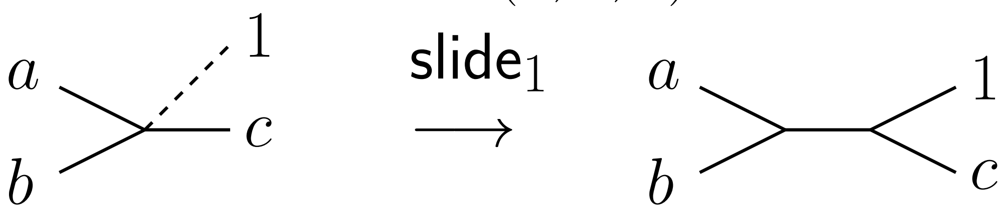



- At $i$th step, insert $i$ and perform $k_i$ *slides* at $i$:
  Let $m =$ min label not in the $a$ branch

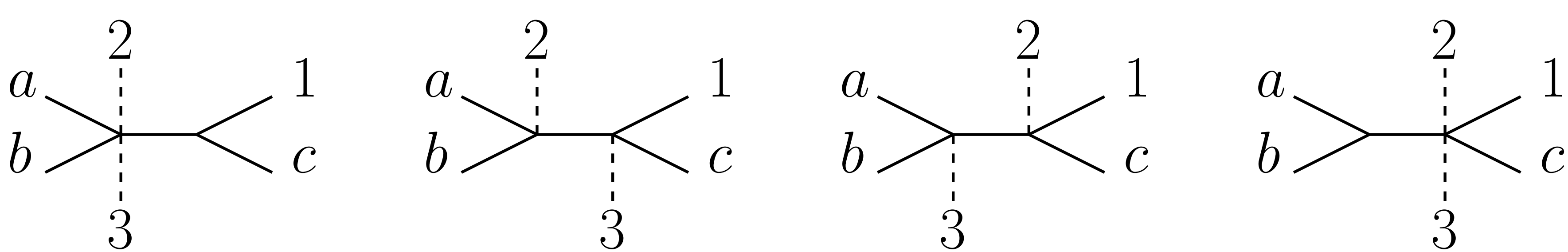



- The resulting trivalent trees are denoted by $\mathrm{Slide}^\omega(k_1, \ldots, k_n)$.

**Example:** Let $n = 3$ and $\mathbf{k} = (1, 0, 2)$.



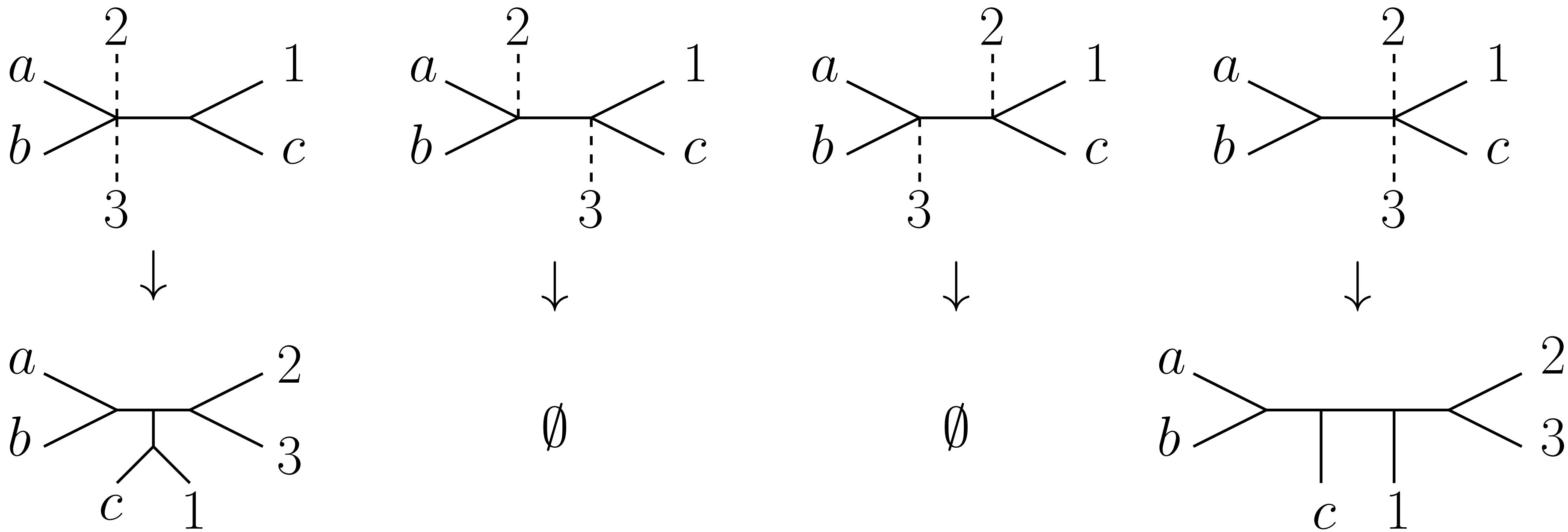

Insert $2$ and $3$, and then perform slide$_3$ twice:



# Slide trees

**Example:** Let $n = 3$ and $\mathbf{k} = (1, 0, 2)$.

Perform slide$_3$ twice:




**Theorem (Gillespie-G.-Levinson, 2021)**

$$\left\langle {n \atop \mathbf{k}} \right\rangle = \#\mathrm{Slide}^{\omega}(k_1, \ldots, k_n).$$

Proof: Hands-on intersection theory calculation on $\overline{M}_{0,n}$

# The moduli space $\overline{M}_{0,n}$

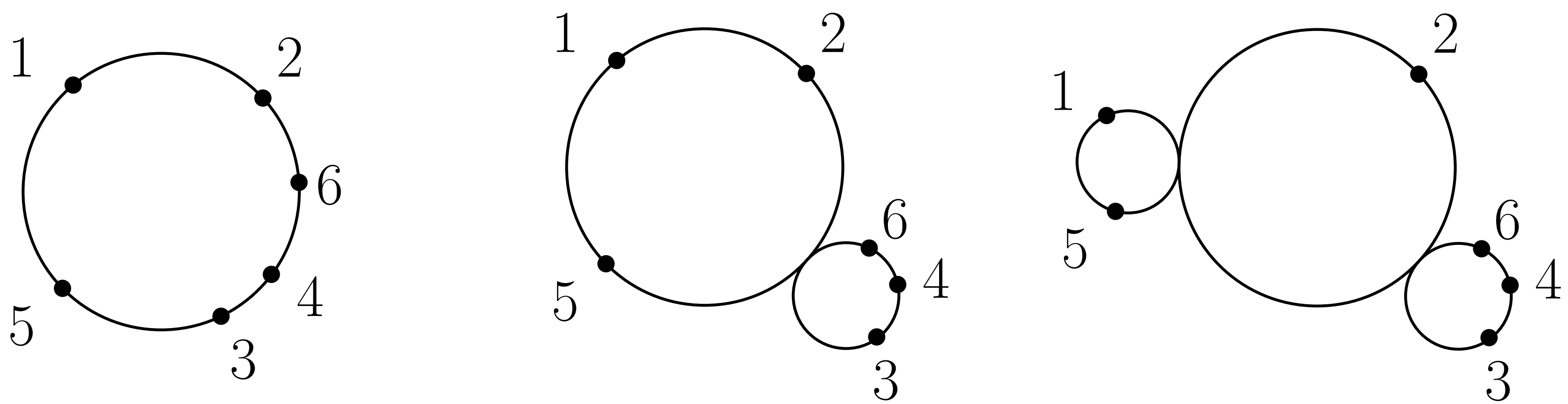
# Moduli space $\overline{M}_{0,n}$

A *moduli space* is (informally) a space that parametrizes geometric objects (e.g. the Grassmannian of $k$ planes in $\mathbb{C}^n$).

$M_{0,n}$ is the moduli space of isomorphism classes of $n$ ordered distinct marked points on $\mathbb{CP}^1$.

$\overline{M}_{0,n}$ (the Deligne-Mumford compactification) is the moduli space parametrizing genus zero **stable** curves with $n$ marked points:

- Stable curves can have multiple irreducible components, but each component has a total of at least $3$ marked points and nodes.

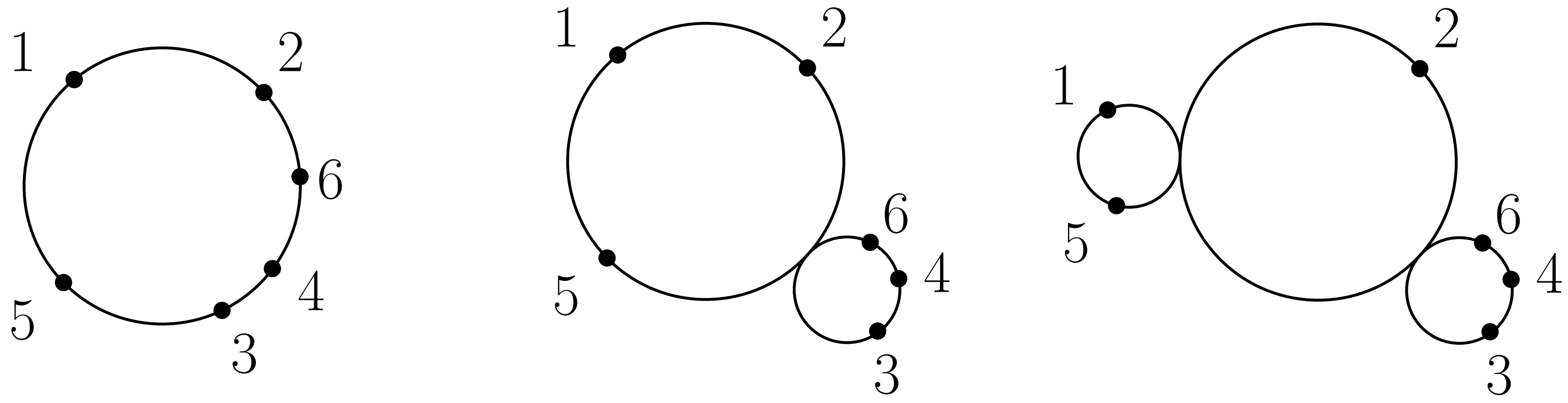


# Dual tree

Associate a *dual tree* to each stable curve in $\overline{M}_{0,n}$:

- Vertices: Components and marked points

- Edges: Adjacent components, and marked points on their components

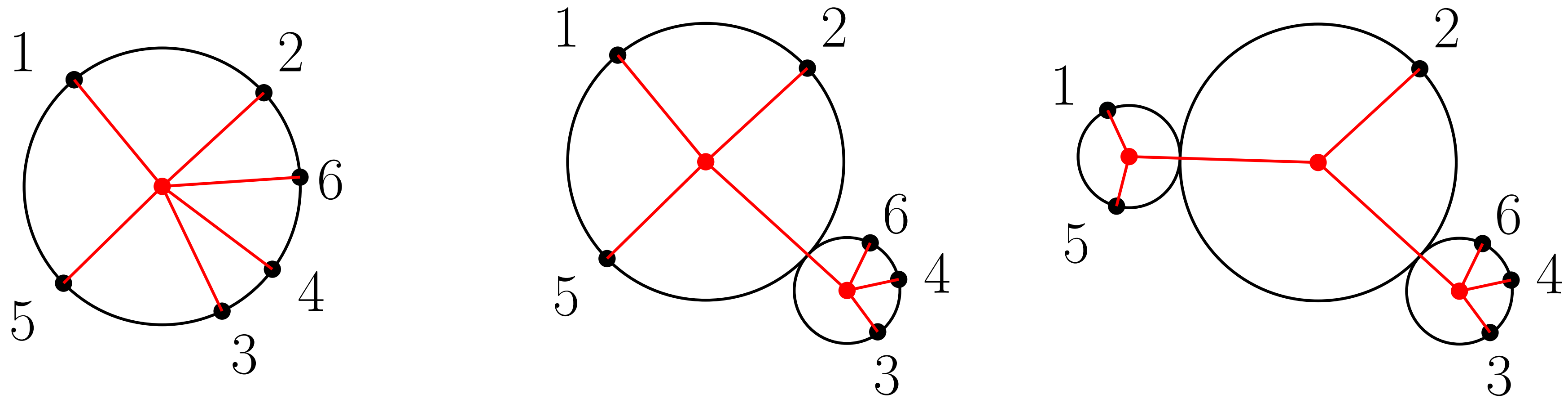


# Dual tree

Associate a *dual tree* to each stable curve in $\overline{M}_{0,n}$:

- Vertices: Components and marked points

- Edges: Adjacent components, and marked points on their components

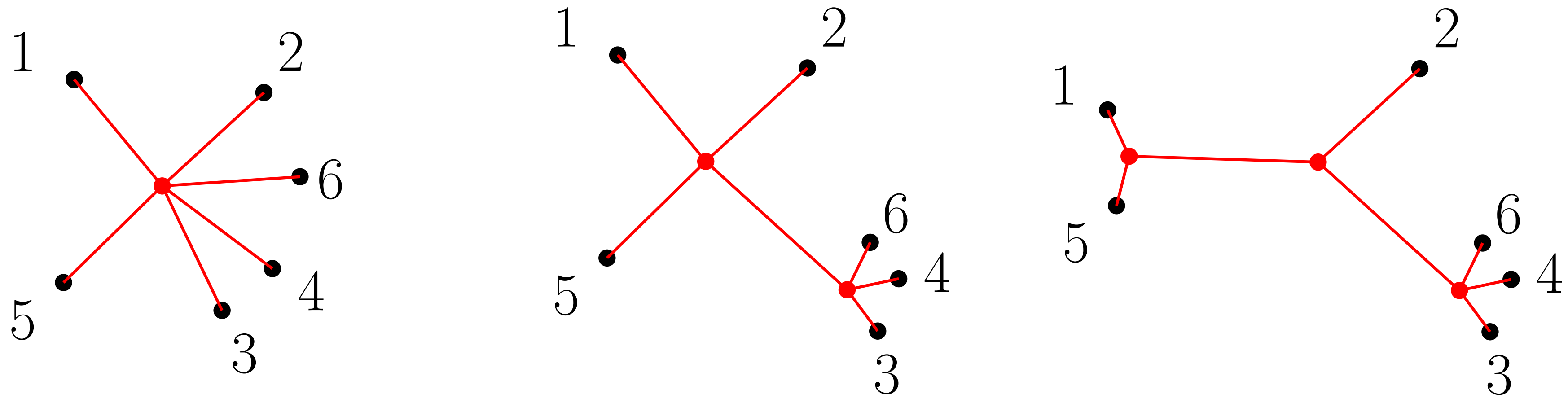


# Dual tree

Associate a *dual tree* to each stable curve in $\overline{M}_{0,n}$:

- Vertices: Components and marked points

- Edges: Adjacent components, and marked points on their components




Stable curve $\rightarrow$ Every non-leaf vertex has degree $\geq 3$.

# Boundary strata/points

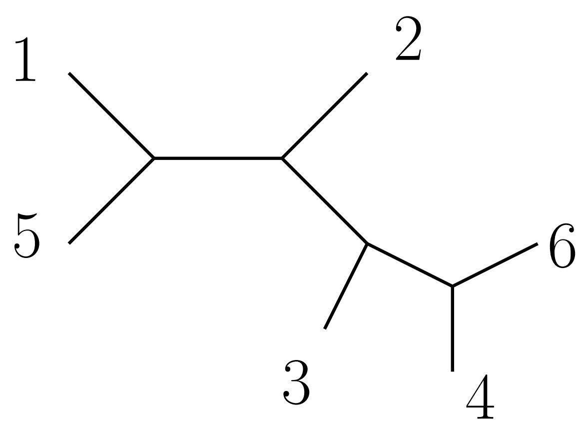
$X_T^\circ = \{$stable curves with dual tree $T\}$

$X_T = \overline{X_T^\circ}$

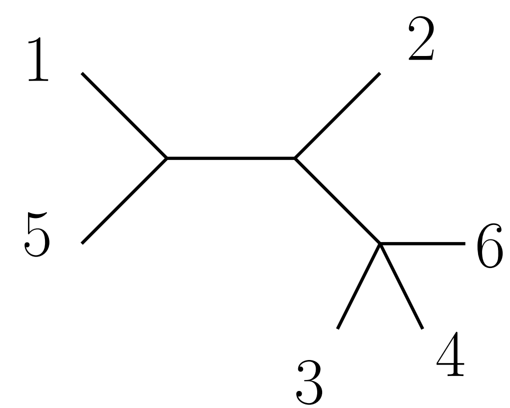
- $\dim(X_T) = \sum_{\text{internal } v} (\deg(v) - 3)$
- $T$ is *trivalent* iff $X_T$ is a point.

**Examples:**

$T =$




$T' =$




$X_T$ is a *boundary point* of $\overline{M}_{0,6}$

$X_{T'}$ is a curve in $\overline{M}_{0,6}$

# Kapranov map

Consider $\overline{M}_{0,n+3}$ with marked points $a < b < c < 1 < 2 < \cdots < n$.

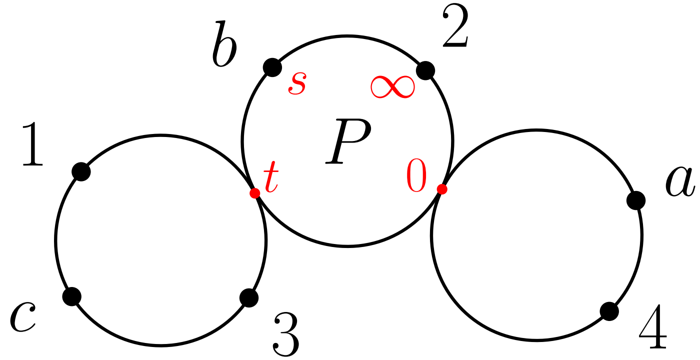
For each $i$, let $\psi_i \in A^*(\overline{M}_{0,n}) \cong H^*(\overline{M}_{0,n})$ be the divisor on $\overline{M}_{0,n}$ corresponding to the $i$*th cotangent line bundle* $\mathbb{L}_i \to \overline{M}_{0,n+3}$.

$\psi_i$ corresponds to the $i$**th Kapranov map**:

$$|\psi_i| : \overline{M}_{0,n+3} \to \mathbb{P}^n$$

If $H$ is the class of a hyperplane in $\mathbb{P}^n$, then $\psi_i = |\psi_i|^*(H)$.

$|\psi_i|$ can be computed in coordinates:




$$\overset{\psi_2}{\mapsto} \quad [x_b : x_c : x_1 : x_3 : x_4] = [s : t : t : t : 0]$$

# Combined Kapranov maps

Let $\pi_n : \overline{M}_{0,n+3} \to \overline{M}_{0,n+2}$ be the "forgetting map" that forgets the marked point $n$ and stabilizes the curve.

$$\overline{M}_{0,n+3} \hookrightarrow \overline{M}_{0,n+2} \times \mathbb{P}^n$$
$$C \mapsto (\pi_n(C), |\psi_n|(C))$$

Iterating this, we get an embedding:

**Kapranov embedding:**

$$\Omega_n : \overline{M}_{0,n+3} \hookrightarrow \mathbb{P}^1 \times \mathbb{P}^2 \times \cdots \times \mathbb{P}^n$$

Define $\omega_i = \Omega_n^*(H_i)$.

$\Omega_n$ first appeared in Keel and Tevelev's work on the *log-canonical embedding* of $\overline{M}_{0,n+3}$.

# $\psi$ and $\omega$ class products

The $\psi$ and $\omega$ classes can be multiplied in $A^*(\overline{M}_{0,n+3})$:

- When $k_1 + k_2 + \cdots + k_n = n$:

$$\int_{\overline{M}_{0,n+3}} \psi_1^{k_1} \cdots \psi_n^{k_n} = \binom{n}{k_1, k_2, \ldots, k_n}.$$

**Aside:** (Kontsevich) Higher genus intersection numbers $\to$ A solution to KdV eqn

- Asymmetric multinomial coefficients are the *multidegrees* of $\Omega_n$:

$$\int_{\overline{M}_{0,n+3}} \omega_1^{k_1} \cdots \omega_n^{k_n} \underset{\text{CGM}}{=} \left\langle {n \atop k_1, \ldots, k_n} \right\rangle.$$

The asymmetric string equation takes the form:

$$\int_{\overline{M}_{0,n+3}} \omega^{\mathbf{k}} = \sum_{j > i(\mathbf{k})} \int_{\overline{M}_{0,n+2}} \omega^{\widetilde{\mathbf{k}}_j}$$

# Explicit hyperplane intersections

Multiplying $\omega_i$ classes $\leftrightarrow$ Intersecting (generic) pulled-back hyperplanes ($k_i$ from the $i$th projective space factor)

**Question:** Can $\left\langle {n \atop \mathbf{k}} \right\rangle$ be realized as counting subsets of boundary points that are obtained from explicit hyperplane intersections?

Let $[x_b : x_c : x_1 : \cdots : x_{i-1}]$ be the projective coordinates for $\mathbb{P}^i$.

Define $H_i(t) = x_b + t x_c + t^2 x_1 + \cdots + t^i x_{i-1}$.

**Theorem (Gillespie-G.-Levinson, 2021)**

$$\lim_{\vec{t} \to \vec{0}} \bigcap_{i=1}^{n} \bigcap_{j=1}^{k_i} \Omega_n^{-1}(H_i(t_{i,j})) = \mathrm{Slide}^{\omega}(k_1, \ldots, k_n)$$

# Slide rule for $\omega$ products

(Keel) $A^*(\overline{M}_{0,n})$ is generated by the $[X_T]$ (satisfying certain relations).

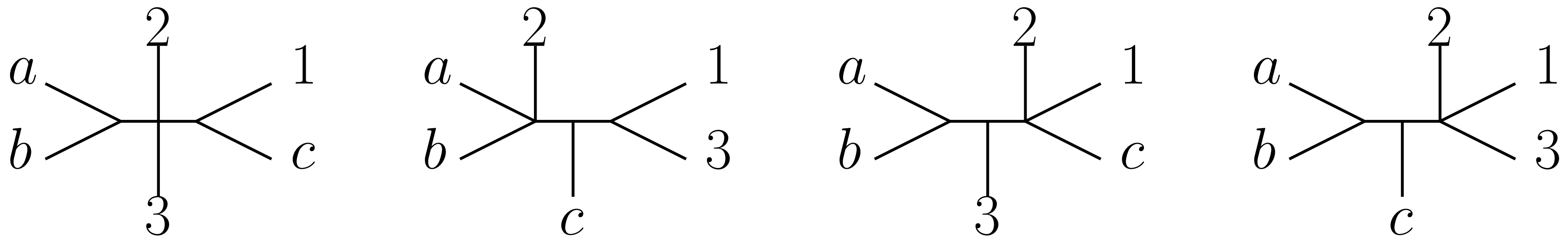
**Bonus:** The slide rule works for $k_1 + \cdots + k_n < n$ to expand positive-dim'l $\omega_i$ products as a positive multiplicity-free sum of $[X_T]$.

**Theorem (Gillespie-G.-Levinson, 2021)**

For any $\mathbf{k}$ with $k_1 + \cdots + k_n \leq n$, we have

$$\omega_1^{k_1}\omega_2^{k_2}\cdots\omega_n^{k_n} = \sum_{T \in \mathrm{Slide}^{\omega}(k_1,\ldots,k_n)} [X_T].$$

**Example:** $\omega_1\omega_3$ expands as the sum of classes of:



# Slide rule for $\psi$ products

**Double bonus:** The same kind of limiting hyperplanes work for $\psi_i$ products, and more general products of pulled-back $\psi$ classes.

**Theorem (Gillespie-G.-Levinson, 2021)**

For any $\mathbf{k}$ with $k_1 + \cdots + k_n \leq n$, we have

$$\psi_1^{k_1}\psi_2^{k_2}\cdots\psi_n^{k_n} = \sum_{T \in \mathrm{Slide}^{\psi}(k_1,\ldots,k_n)} [X_T].$$

# Open questions

- The trees $\mathrm{Slide}^{\omega}(1, 1, \ldots, 1)$ can be partially described using $23$-$1$ pattern avoidance. Pattern avoidance criteria for slide trees in general?

- Direct bijection between $\mathrm{Slide}^{\omega}(\mathbf{k})$ and tournament trees for the case when $\sum_i k_i = n$?

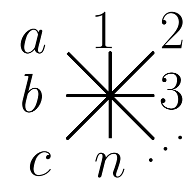
- Generalization to $\psi$ class products in higher genus $A^*(\overline{M}_{g,n})$? Hassett spaces? Stable maps?

Thanks for your attention!

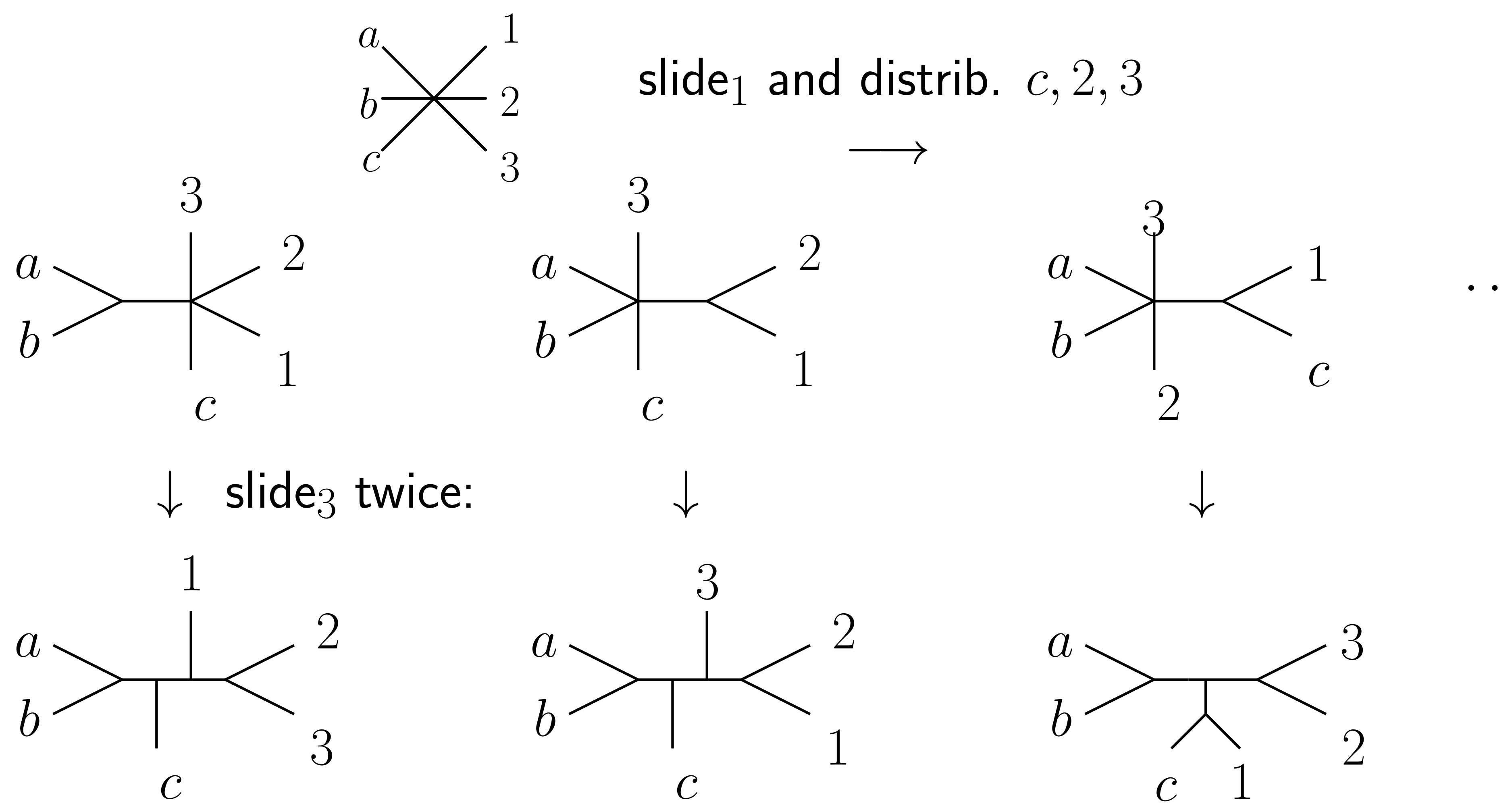
# Slide rule for $\psi$ products

**Double bonus:** The same kinds of limiting hyperplanes and slide rules work for $\psi$ products as well!

Perform the same algorithm, except start with:




**Example:** $n = 3$ and $\mathbf{k} = (1, 0, 2)$.



# Slide rule for $\psi$ products

Let $\mathrm{Slide}^{\psi}(k_1, \ldots, k_n)$ be the set of stable trees obtained.

**Theorem (Gillespie-G.-Levinson, 2021)**

For any $(k_1, \ldots, k_n)$ with $k_1 + \cdots + k_n \leq n$, we have

$$\psi_1^{k_1}\psi_2^{k_2}\cdots\psi_n^{k_n} = \sum_{T \in \mathrm{Slide}^{\psi}(k_1,\ldots,k_n)} [X_T].$$

Proof: Again using limiting hyperplane intersections.

A slight variation of the slide rule also give formulas for any mixed product of $\omega$ and $\psi$ classes: First compute the product of the $\omega$'s, then multiply by the $\psi$'s.