Tutte polynomials and bialgebras

Iain Moffatt
(with Thomas Krajewski, Stephen Huggett, Adrian Tanasa)

Royal Holloway, University of London

CIRM, 27th September 2018

- Session on Tutte polynomials and bialgebras.
-
-
-
-

- Session on Tutte polynomials and bialgebras.
- You’ll see how a “Tutte polynomial” can be associated with a bialgebra.
- $$(\exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}))(G)$$
-
-

- Session on Tutte polynomials and bialgebras.
- You'll see how a "Tutte polynomial" can be associated with a bialgebra.
- "Wait. You're asking me swallow that

$$(\exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}))(G)$$

  is a sensible generalisation of the Tutte polynomial?"
-
-

- Session on Tutte polynomials and bialgebras.
- You'll see how a "Tutte polynomial" can be associated with a bialgebra.
- "Wait. You're asking me swallow that

$$(\exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}))(G)$$

is a sensible generalisation of the Tutte polynomial?"
- "What do you mean by a "Tutte polynomial?"
-

- Session on Tutte polynomials and bialgebras.
- You'll see how a "Tutte polynomial" can be associated with a bialgebra.
- "Wait. You're asking me swallow that

$$(\exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}))(G)$$

  is a sensible generalisation of the Tutte polynomial?"
- "What do you mean by a "Tutte polynomial?""
- Warning: I'll hide some details!

What is the Tutte polynomial?

- Many objects can be counted recursively:

k-colourings: $$P(G) = \begin{cases} P(G\backslash e) - P(G/e) & \text{ordinary} \\ (k-1)P(G/e) & \text{if e a bridge} \\ 0 & \text{if e a loop} \\ k^{|V|} & \text{if G edgeless} \end{cases}$$

k-flows: $$F(G) = \begin{cases} -F(G\backslash e) + F(G/e) & \text{ordinary} \\ 0 & \text{if e a bridge} \\ (k-1)F(G\backslash e) & \text{if e a loop} \\ 1 & \text{if G edgeless} \end{cases}$$

acyclic orients: $$a(G) = \begin{cases} a(G\backslash e) + a(G/e) & \text{ordinary} \\ 2a(G/e) & \text{if e a bridge} \\ 0 & \text{if e a loop} \\ 1 & \text{if G edgeless} \end{cases}$$

Unifying the (many similar) examples

- All counts of the form:

$$U(G) := \begin{cases} \alpha^{|V|} & \text{if G edgeless} \\ x\,U(G/e) & \text{if e a bridge} \\ y\,U(G\backslash e) & \text{if e a loop} \\ a\,U(G\backslash e) + b\,U(G/e) & \text{otherwise} \end{cases}$$

Unifying the (many similar) examples

- All counts of the form:

$$U(G) := \begin{cases} \alpha^{|V|} & \text{if G edgeless} \\ x\, U(G/e) & \text{if e a bridge} \\ y\, U(G\backslash e) & \text{if e a loop} \\ a\, U(G\backslash e) + b\, U(G/e) & \text{otherwise} \end{cases}$$

- Defines a 5-variable polynomial.....

Unifying the (many similar) examples

- All counts of the form:

$$U(G) := \begin{cases} \alpha^{|V|} & \text{if G edgeless} \\ x\, U(G/e) & \text{if e a bridge} \\ y\, U(G \backslash e) & \text{if e a loop} \\ a\, U(G \backslash e) + b\, U(G/e) & \text{otherwise} \end{cases}$$

- Defines a 5-variable polynomial.....
- ...actually it defines a 2-variable polynomial!

Unifying the (many similar) examples

- All counts of the form:

$$U(G) := \begin{cases} \alpha^{|V|} & \text{if G edgeless} \\ x\, U(G/e) & \text{if e a bridge} \\ y\, U(G\backslash e) & \text{if e a loop} \\ a\, U(G\backslash e) + b\, U(G/e) & \text{otherwise} \end{cases}$$

- Defines a 5-variable polynomial.....
- ...actually it defines a 2-variable polynomial!

The Tutte polynomial, $T(G; x, y) \in \mathbb{Z}[x, y]$

$$T(G) = \begin{cases} 1 & \text{if G edgeless} \\ xT(G/e) & \text{if e a bridge} \\ yT(G\backslash e) & \text{if e a loop} \\ T(G\backslash e) + T(G/e) & \text{otherwise} \end{cases}$$

Unifying the (many similar) examples

- All counts of the form:

$$U(G) := \begin{cases} \alpha^{|V|} & \text{if G edgeless} \\ x\,U(G/e) & \text{if e a bridge} \\ y\,U(G\backslash e) & \text{if e a loop} \\ a\,U(G\backslash e) + b\,U(G/e) & \text{otherwise} \end{cases}$$

- Defines a 5-variable polynomial.....
- ...actually it defines a 2-variable polynomial!

The Tutte polynomial, $T(G;x,y) \in \mathbb{Z}[x,y]$

$$T(G) = \begin{cases} 1 & \text{if G edgeless} \\ xT(G/e) & \text{if e a bridge} \\ yT(G\backslash e) & \text{if e a loop} \\ T(G\backslash e) + T(G/e) & \text{otherwise} \end{cases}$$

- $U(G) = \alpha^k a^{|E|-r} b^r\, T\left(G; x/b, y/a\right).$

Extending the Tutte polynomial

- We are interested in extending the Tutte polynomial from graphs to different objects.

Extending the Tutte polynomial

- We are interested in extending the Tutte polynomial from graphs to different objects.
- What do we mean by a “Tutte polynomial”

Extending the Tutte polynomial

- We are interested in extending the Tutte polynomial from graphs to different objects.
- What do we mean by a "Tutte polynomial"

The Tutte polynomial of a graph

$$T(G) = \begin{cases} 1 & \text{if G edgeless} \\ xT(G/e) & \text{if e a bridge} \\ yT(G\backslash e) & \text{if e a loop} \\ T(G\backslash e) + T(G/e) & \text{otherwise} \end{cases}$$

- Here, by a "Tutte polynomial" we mean
  - '**Full**' recursive deletion-contraction definition.
  - Most general (universality).

Extending the Tutte polynomial

- We are interested in extending the Tutte polynomial from graphs to different objects.
- What do we mean by a "Tutte polynomial"

The Tutte polynomial of a graph

$$T(G) = \begin{cases} 1 & \text{if G edgeless} \\ xT(G/e) & \text{if e a bridge} \\ yT(G\backslash e) & \text{if e a loop} \\ T(G\backslash e) + T(G/e) & \text{otherwise} \end{cases}$$

- Here, by a "Tutte polynomial" we mean
  - '**Full**' recursive deletion-contraction definition.
  - Most general (universality).
- What's involved in defining this?
  - objects.
  - deletion & contraction
  - "loops" and "bridges" — cases

Objects, deletion, and contraction

To define a polynomial recursively we need:

- Ignore this:

1. A graded set $\mathcal{S} = \bigcup_{n \geq 0} \mathcal{S}_n$ of finite combinatorial objects such that each $S \in \mathcal{S}_n$ has a finite set $E(S)$ of exactly n sub-objects associated with it, and such that there is a unique element $1 \in \mathcal{S}_0$.
2. Two *minor operations*, $\backslash$ called *deletion* and $/$ called *contraction*, that associate elements $S \backslash e$ and S/e, respectively, to each pair $(S \in \mathcal{S}_n, e \in E(S))$, where $E(S \backslash e) = E(S/e) = E(S) \setminus e$, and such that for $e \neq f$
$$(S \backslash e) \backslash f = (S \backslash f) \backslash e, \quad (S/e) \backslash f = (S \backslash f)/e, \quad (S/e)/f = (S/f)/e.$$

- You need:
  - Objects with things like edges;
  - operations called “deletion” and “contraction” (need not be the usual ones),
  - that behave as you would expect.

Cases

Classical case

$$T(G) = \begin{cases} 1 & \text{if G edgeless} \\ xT(G/e) & \text{if e a bridge} \\ yT(G\backslash e) & \text{if e a loop} \\ T(G\backslash e) + T(G/e) & \text{otherwise} \end{cases}$$

For the general case

- We have:
  - deletion and contraction
  - objects closed under the operations
- We need:
  - to identify the cases for the recursive definition
  - i.e., find analogues of loops and bridges
- Strategy — “de-graph” the Tutte polynomial’s cases

“De-graphing” bridges and loops

- 2 connected graphs on 1 edge: 

“De-graphing” bridges and loops

- 2 connected graphs on 1 edge:  , 
- For $e^c := E \backslash e$, look at pair

$$(G \backslash e^c, G/e^c)$$

“De-graphing” bridges and loops

- 2 connected graphs on 1 edge: 
- For $e^c := E \backslash e$, look at pair

$$(G \backslash e^c, G / e^c)$$

-  $\iff$ e bridge  $\iff$ e loop

   $\iff$ e ordinary  is impossible

“De-graphing” bridges and loops

- 2 connected graphs on 1 edge: 
- For $e^c := E \backslash e$, look at pair

$$(G \backslash e^c, G/e^c)$$

-  $\iff$ e bridge  $\iff$ e loop

   $\iff$ e ordinary  is impossible
- Define

$$U(G) = \begin{cases} 1 & \text{if G edgeless} \\ a_i U(G \backslash e) + b_j U(G/e) & \text{if } e \text{ is } (i,j) \end{cases}$$

“De-graphing” bridges and loops

- 2 connected graphs on 1 edge: 
- For $e^c := E \backslash e$, look at pair

$$(G \backslash e^c, G/e^c)$$

-  $\iff$ e bridge  $\iff$ e loop

   $\iff$ e ordinary  is impossible

- Define

$$U(G) = \begin{cases} 1 & \text{if } G \text{ edgeless} \\ a_i U(G \backslash e) + b_j U(G/e) & \text{if } e \text{ is } (i,j) \end{cases}$$

- Then $U(G)$ is Tutte polynomial:

$$U(G) = a_l^{|E|-r(G)} b_b^{r(G)} T(G; \tfrac{a_b}{b_b} + 1, \tfrac{b_l}{a_l} + 1)$$

An extended Tutte polynomial

- $S_1 = \{s_1, s_2, \ldots, s_n\}$ objects of grading 1 (e.g., graphs on one edge)
- e of *type*-(i,j), for $i,j \in S_1$ if

$$(G\backslash e^c, G/e^c) = (i,j),$$

Definition (*Krajewski–M.–Tanasa; Huggett–M.*)

The "Tutte polynomial" for the class of objects is

$$U(G) = \begin{cases} f(S_0) & \text{if } G \in S_0 \\ a_i U(G\backslash e) + b_j U(G/e) & \text{if } e \text{ is } (i,j) \end{cases}$$

So far

Main point

- There is a **canonical** way to associate a "Tutte polynomial" of
  - combinatorial objects equipped
  - with **some** notion of deletion and contraction.
- Incorporates/extends many known polys.

- Classical Tutte polynomial
- Las Vergnas' Tutte poly. of morphism of a matroid
- Ribbon graph polynomial
- Bollobás-Riordan polynomial (extended version)
- Kruskal polynomial (extended version)
- Penrose polynomial
- ?????

Bialgebras

Bialgebras

1. Graded vector space: $\mathcal{H} := \bigoplus_{i \geq 0} \mathcal{H}_i$

Example — matroids

1. Matroids
(formal lin. combs. of, $\otimes$ a formal symbol)

Bialgebras

Bialgebras

1. Graded vector space: $\mathcal{H} := \bigoplus_{i \geq 0} \mathcal{H}_i$
2. product $m : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H}$

Example — matroids

1. Matroids (formal lin. combs. of, $\otimes$ a formal symbol)
2. direct sum

Bialgebras

Bialgebras

1. Graded vector space: $\mathcal{H} := \bigoplus_{i\geq 0} \mathcal{H}_i$
2. product $m : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H}$
3. coproduct $\Delta : \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$

Example — matroids

1. Matroids (formal lin. combs. of, $\otimes$ a formal symbol)
2. direct sum
3. $\Delta(M) = \sum_{A\subseteq E} M|A \otimes M/A$

Bialgebras

Bialgebras

1. Graded vector space: $\mathcal{H} := \bigoplus_{i\geq 0} \mathcal{H}_i$
2. product $m : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H}$
3. coproduct $\Delta : \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$
4. both nice (graded, (co)assoc., ...)

Example — matroids

1. Matroids (formal lin. combs. of, $\otimes$ a formal symbol)
2. direct sum
3. $\Delta(M) = \sum_{A\subseteq E} M|A \otimes M/A$
4. ✓

Bialgebras

Bialgebras

1. Graded vector space: $\mathcal{H} := \bigoplus_{i \geq 0} \mathcal{H}_i$
2. product $m : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H}$
3. coproduct $\Delta : \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$
4. both nice (graded, (co)assoc., ...)
5. play nicely together (compatibility conds.)

Example — matroids

1. Matroids (formal lin. combs. of, $\otimes$ a formal symbol)
2. direct sum
3. $\Delta(M) = \sum_{A \subseteq E} M|A \otimes M/A$
4. ✓
5. ✓

Bialgebras

Bialgebras

1. Graded vector space: $\mathcal{H} := \bigoplus_{i\geq 0} \mathcal{H}_i$
2. product $m : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H}$
3. coproduct $\Delta : \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$
4. both nice (graded, (co)assoc., ...)
5. play nicely together (compatibility conds.)
6. Need here $\dim(\mathcal{H}_0) = 1$

Example — matroids

1. Matroids (formal lin. combs. of, $\otimes$ a formal symbol)
2. direct sum
3. $\Delta(M) = \sum_{A\subseteq E} M|A \otimes M/A$
4. ✓
5. ✓
6. ✓

Bialgebras

Bialgebras

1. $\mathcal{H}_1 = \{S_1, \ldots, S_n\}$

Example — matroids

1. $\mathcal{H}_1 = \{$  ,  $\}$

Bialgebras

Bialgebras

1. $\mathcal{H}_1 = \{S_1, \ldots, S_n\}$
2. $\delta_i(S) := \begin{cases} 1 & \text{if } S = S_i \\ 0 & \text{otherwise} \end{cases}$

Example — matroids

1. $\mathcal{H}_1 = \{$ ,  $\}$
2. $\delta_b\Big($  $\Big) = 1$, $\delta_l\Big($  $\Big) = 1$

Bialgebras

Bialgebras

1. $\mathcal{H}_1 = \{S_1, \ldots, S_n\}$
2. $\delta_i(S) := \begin{cases} 1 & \text{if } S = S_i \\ 0 & \text{otherwise} \end{cases}$
3. $\delta_{\mathbf{a}} := \sum_{i=1}^{n} a_i \delta_i$

Example — matroids

1. $\mathcal{H}_1 = \{$  ,  $\}$
2. $\delta_b\Big($  $\Big) = 1$, $\delta_l\Big($  $\Big) = 1$
3. $\delta_{\mathbf{x}} = x_1\delta_b + x_2\delta_l$
   $\delta_{\mathbf{x}}\Big(5$  $- 2$  $\Big) = 5x_1 - 2x_2$

Bialgebras

Bialgebras

1. $\mathcal{H}_1 = \{S_1, \ldots, S_n\}$
2. $\delta_i(S) := \begin{cases} 1 & \text{if } S = S_i \\ 0 & \text{otherwise} \end{cases}$
3. $\delta_{\mathbf{a}} := \sum_{i=1}^{n} a_i \delta_i$

Example — matroids

1. $\mathcal{H}_1 = \{$  ,  $\}$
2. $\delta_b \Big($  $\Big) = 1$, $\delta_l \Big($  $\Big) = 1$
3. $\delta_{\mathbf{x}} = x_1\delta_b + x_2\delta_l$
   $\delta_{\mathbf{x}} \Big(5$  $- 2$  $\Big) = 5x_1 - 2x_2$

- **convolution product**: $f * g := m \circ (f \otimes g) \circ \Delta$

Bialgebras

1. $\mathcal{H}_1 = \{S_1, \ldots, S_n\}$
2. $\delta_i(S) := \begin{cases} 1 & \text{if } S = S_i \\ 0 & \text{otherwise} \end{cases}$
3. $\delta_{\mathbf{a}} := \sum_{i=1}^{n} a_i \delta_i$

Example — matroids

1. $\mathcal{H}_1 = \{$  ,  $\}$
2. $\delta_b \Big($  $\Big) = 1$, $\delta_l \Big($  $\Big) = 1$
3. $\delta_{\mathbf{x}} = x_1 \delta_b + x_2 \delta_l$

   $\delta_{\mathbf{x}} \Big(5$  $- 2$  $\Big) = 5x_1 - 2x_2$

- **convolution product**: $f * g := m \circ (f \otimes g) \circ \Delta$
- **exponential:** $\exp_*(\delta) := \sum_{n=0}^{\infty} \frac{\delta^{*n}}{n!}$

Bialgebras

Bialgebras

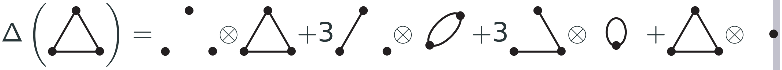
1. $\mathcal{H}_1 = \{S_1, \ldots, S_n\}$
2. $\delta_i(S) := \begin{cases} 1 & \text{if } S = S_i \\ 0 & \text{otherwise} \end{cases}$
3. $\delta_{\mathbf{a}} := \sum_{i=1}^{n} a_i \delta_i$

Example — matroids

1. $\mathcal{H}_1 = \{$  ,  $\}$
2. $\delta_b($  $) = 1$, $\delta_l($  $) = 1$
3. $\delta_{\mathbf{x}} = x_1 \delta_b + x_2 \delta_l$

   $\delta_{\mathbf{x}}(5$  $- 2$  $) = 5x_1 - 2x_2$

- **convolution product**: $f * g := m \circ (f \otimes g) \circ \Delta$
- **exponential:** $\exp_*(\delta) := \sum_{n=0}^{\infty} \frac{\delta^{*n}}{n!}$

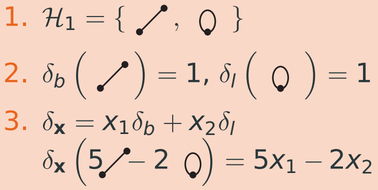


Bialgebras

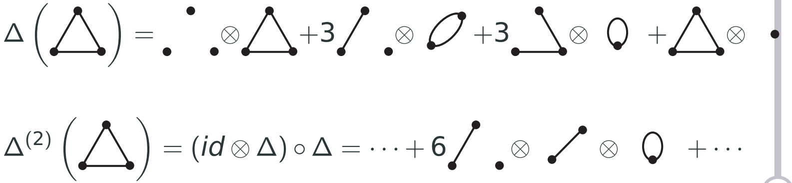
Bialgebras

1. $\mathcal{H}_1 = \{S_1, \ldots, S_n\}$
2. $\delta_i(S) := \begin{cases} 1 & \text{if } S = S_i \\ 0 & \text{otherwise} \end{cases}$
3. $\delta_{\mathbf{a}} := \sum_{i=1}^{n} a_i \delta_i$

Example — matroids



- **convolution product**: $f * g := m \circ (f \otimes g) \circ \Delta$
- **exponential:** $\exp_*(\delta) := \sum_{n=0}^{\infty} \frac{\delta^{*n}}{n!}$



Bialgebras

Bialgebras

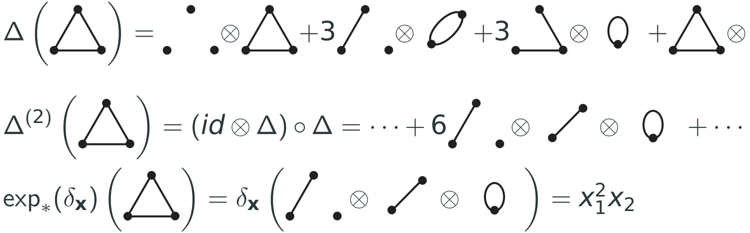
1. $\mathcal{H}_1 = \{S_1, \ldots, S_n\}$
2. $\delta_i(S) := \begin{cases} 1 & \text{if } S = S_i \\ 0 & \text{otherwise} \end{cases}$
3. $\delta_{\mathbf{a}} := \sum_{i=1}^{n} a_i \delta_i$

Example — matroids

1. $\mathcal{H}_1 = \{$  ,  $\}$
2. $\delta_b($  $) = 1$, $\delta_l($  $) = 1$
3. $\delta_{\mathbf{x}} = x_1 \delta_b + x_2 \delta_l$

   $\delta_{\mathbf{x}}\big(5$  $- 2$  $\big) = 5x_1 - 2x_2$

- **convolution product**: $f * g := m \circ (f \otimes g) \circ \Delta$
- **exponential:** $\exp_*(\delta) := \sum_{n=0}^{\infty} \frac{\delta^{*n}}{n!}$



Tutte polynomial of a bialgebra

Definition (*Krajewski–M.–Tanasa*)

*The **Tutte polynomial** of a bialgebra $\mathcal{H}$ is*

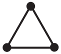
$$\alpha(\mathbf{a}, \mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}).$$

Tutte polynomial of a bialgebra

Definition (*Krajewski–M.–Tanasa*)

*The **Tutte polynomial** of a bialgebra $\mathcal{H}$ is*

$$\alpha(\mathbf{a}, \mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}).$$

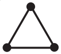
$\alpha(\mathbf{x}, \mathbf{y}) \Bigg($  $\Bigg)$

Tutte polynomial of a bialgebra

Definition (*Krajewski–M.–Tanasa*)

*The **Tutte polynomial** of a bialgebra $\mathcal{H}$ is*

$$\alpha(\mathbf{a}, \mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}).$$

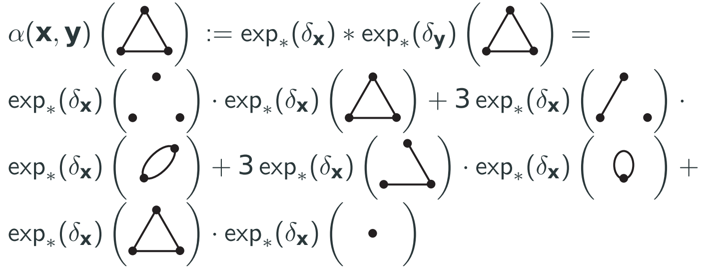
$\alpha(\mathbf{x}, \mathbf{y})\Big($  $\Big) := \exp_*(\delta_{\mathbf{x}}) * \exp_*(\delta_{\mathbf{y}})\Big($  $\Big)$

Tutte polynomial of a bialgebra

Definition (*Krajewski–M.–Tanasa*)

*The **Tutte polynomial** of a bialgebra $\mathcal{H}$ is*

$$\alpha(\mathbf{a}, \mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}).$$

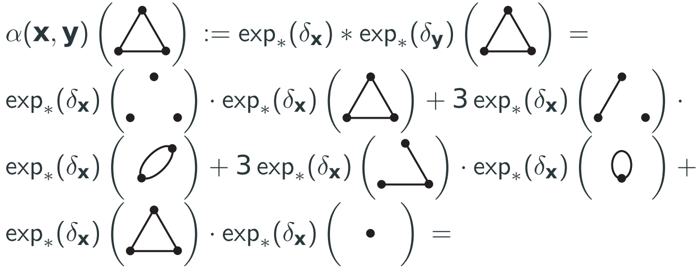


Tutte polynomial of a bialgebra

Definition (*Krajewski–M.–Tanasa*)

*The **Tutte polynomial** of a bialgebra $\mathcal{H}$ is*

$$\alpha(\mathbf{a}, \mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}).$$



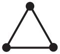
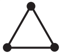
$$y_1^2 y_2 + 3x_1 y_1 y_2 + 3x_2^2 y_2 + x_1^2 x_2.$$

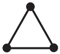
Tutte polynomial of a bialgebra

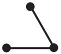
Definition (*Krajewski–M.–Tanasa*)

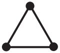
*The **Tutte polynomial** of a bialgebra $\mathcal{H}$ is*

$$\alpha(\mathbf{a},\mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}).$$

$\alpha(\mathbf{x},\mathbf{y})\Big($  $\Big) := \exp_*(\delta_{\mathbf{x}}) * \exp_*(\delta_{\mathbf{y}})\Big($  $\Big) =$

$\exp_*(\delta_{\mathbf{x}})\Big($  $\Big) \cdot \exp_*(\delta_{\mathbf{x}})\Big($  $\Big) + 3\exp_*(\delta_{\mathbf{x}})\Big($  $\Big) \cdot$

$\exp_*(\delta_{\mathbf{x}})\Big($  $\Big) + 3\exp_*(\delta_{\mathbf{x}})\Big($  $\Big) \cdot \exp_*(\delta_{\mathbf{x}})\Big($  $\Big) +$

$\exp_*(\delta_{\mathbf{x}})\Big($  $\Big) \cdot \exp_*(\delta_{\mathbf{x}})\Big($  $\Big) =$

$y_1^2y_2 + 3x_1y_1y_2 + 3x_2^2y_2 + x_1^2x_2.$

Tutte polynomial (universal form)

$$\alpha(\mathbf{x},\mathbf{y})(G) = x_1^{r(G)}y_2^{|E(G)|-r(G)}T_G\left(\frac{y_1}{x_1}+1, \frac{x_2}{y_2}+1\right).$$

Tutte polynomial of a bialgebra

Definition (*Krajewski–M.–Tanasa*)

*The **Tutte polynomial** of a bialgebra $\mathcal{H}$ is*

$$\alpha(\mathbf{a}, \mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}).$$

- We can recognise our previous construction of the Tutte polynomials
  - Objects with suitable deletion and contraction
  - $\Delta(S) = \sum_{A \subseteq E} S|A \otimes S/A$
- then we have:

deletion-contraction

$$\alpha(\mathbf{a}, \mathbf{b})(S) = \delta_{\mathbf{b}}(S/e^c) \cdot \alpha(S \backslash e) + \delta_{\mathbf{a}}(S \backslash e^c) \cdot \alpha(S/e)$$

- We've seen this before!

Tutte polynomial of a bialgebra

Definition (*Krajewski–M.–Tanasa*)

*The **Tutte polynomial** of a bialgebra $\mathcal{H}$ is*

$$\alpha(\mathbf{a},\mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}}).$$

- deletion-contraction

$$\alpha(\mathbf{a},\mathbf{b})(S) = \delta_{\mathbf{b}}(S/e^c) \cdot \alpha(S \backslash e) + \delta_{\mathbf{a}}(S \backslash e^c) \cdot \alpha(S/e)$$

- Universality
  (set a_i or b_i to 1)
- Convolution
  $(\exp(x)\exp(y) = \exp(x+y))$
- State sums

$$\alpha(\mathbf{a},\mathbf{b})(S) = \prod_{j \in J} y_j^{r_j(S)} \sum_{A \subseteq E(S)} \prod_{j \in J} \left(\frac{x_j}{y_j}\right)^{r_j(A)}$$

- Specialisation $\leftrightarrow$ bialgebra maps
- duality

Tutte polynomial of a bialgebra

We've seen the two approaches

Definition (*elementary approach*)

$$U(G) = a_i U(G \backslash e) + b_j U(G/e) \quad \textit{if } e \textit{ is } (i,j)$$

Definition (*algebraic approach*)

$$\alpha(\mathbf{a}, \mathbf{b}) := \exp_*(\delta_{\mathbf{a}}) * \exp_*(\delta_{\mathbf{b}})$$

are the same except for

- bialgebra approach requires **one** object of grading 1
- elementary approach does not.
- but...

Thank You!