The Goussarov-Polyak-Viro (GPV) Theorem

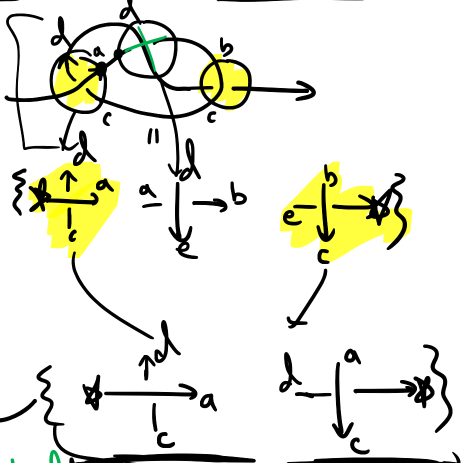
July 21, 2021

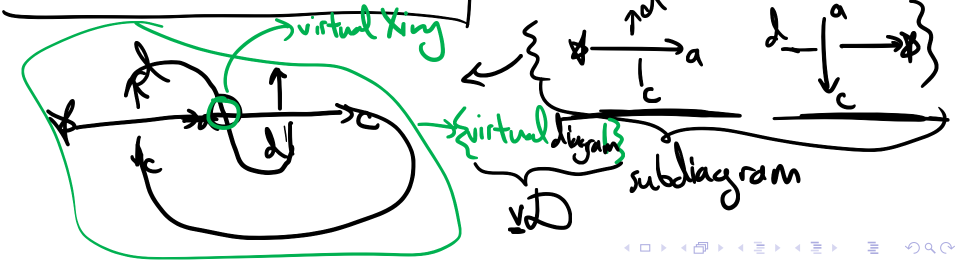
"Statement of the theorem"

"A finite type invariant of degree n factors through subdiagrams with $\leq n$ Xings"

$v : uD \longrightarrow$ Abelian gp

$v(\underbrace{\times \cdots \times}_{>n}) = 0$

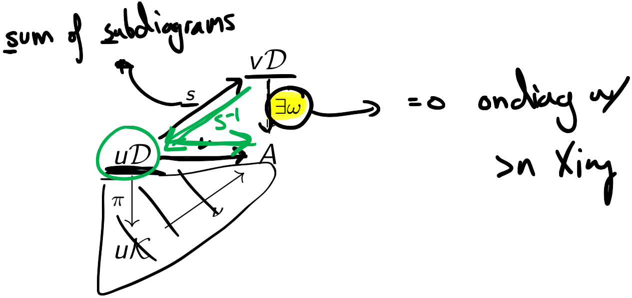




Statement of the theorem

Theorem (Goussarov, Polyak, Viro): Given a finite type invariant $\nu : u\mathcal{D} \to A$ of degree n, there exists $\omega : v\mathcal{D} \to A$ such that:

1. $\omega \circ s = \nu$ (the following diagram commutes)
2. $\omega = 0$ on diagrams with $> n$ (real or double point) Xings.

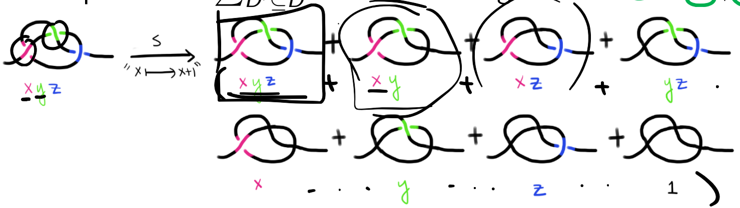


The map s and its inverse

sub contain x

The map $s: D \mapsto \sum_{D'\subseteq D} D'$ can be thought of as "$x \mapsto x + 1$":

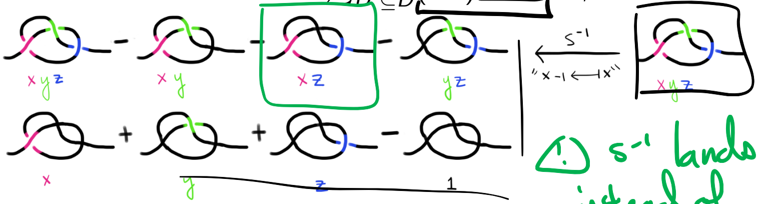
subdiag don't



$= (x+1)(y+1)(z+1)$

\# Xings in D but not D'

so its inverse is $s^{-1}: D \mapsto \sum_{D'\subseteq D} (-1)^{|D'-D|} D'$, or "$x \mapsto x - 1$":



$= (x-1)(y-1)(z-1)$

⚠ s^{-1} lands in vD instead of uD

The map P: properties

We want a map $P : v\mathcal{D} \to u\mathcal{D}$ which satisfies:

1. $\nu \circ P = \nu$ on real knot diagrams (commutes)
2. $\nu \circ P \circ \underbrace{s^{-1}}_{g} = 0$ on diagrams with $> n$ (real or singular) Xings

$$\begin{tikzcd}
v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\
u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"] \arrow[d, "\pi"'] & A \\
u\mathcal{K} \arrow[ur, "\nu"'] &
\end{tikzcd}$$

Then we can define

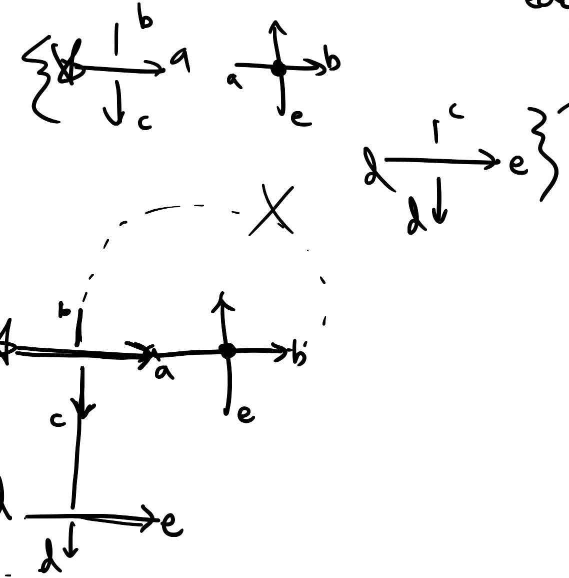
$$\boxed{\omega = \nu \circ P \circ s^{-1}}$$

The map P: what it actually "is" (Roukema's version)

1. Make a first-come-first-serve tree

vertices = Xings
edges = strands b/w Xings

eg:



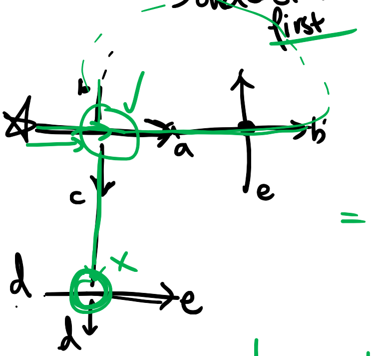
$$\begin{tikzcd}
v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\
u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"] \arrow[d, "\pi"'] & A \\
u\mathcal{K} \arrow[ur, "\nu"'] &
\end{tikzcd}$$

The map P: what it actually "is" (Roukema's version)

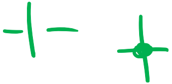
1. Make a first-come-first-serve tree
2. Change all bad Xings to good Xings (using the double point relation)

understrand first

overstrand first



=



$$\begin{tikzcd}
v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\
u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"] \arrow[d, "\pi"'] & A \\
u\mathcal{K} \arrow[ur, "\nu"'] &
\end{tikzcd}$$

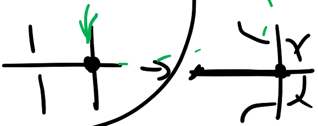
The map P: what it actually "is" (Roukema's version)

1. Make a first-come-first-serve tree
2. Change all bad Xings to good Xings (using the double point relation)
3. Sweep the first real crossing on the double point tree

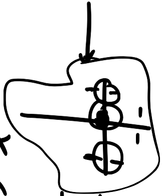
$$\begin{tikzcd} v\mathcal{D} & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\ u\mathcal{D} \arrow[u, leftarrow, "P"] \arrow[ur, "s"] \arrow[r, "\nu"] \arrow[d, "\pi"'] & A \\ u\mathcal{K} \arrow[ur, "\nu"'] & \end{tikzcd}$$

smallest subtree containg all double pts and root

so all vertices have deg 1 or 4



order by distance from root and then counterclockwise



eg.

The map P: what it actually "is" (Roukema's version)

1. Make a first-come-first-serve tree
2. Change all bad Xings to good Xings (using the double point relation)
3. Sweep the first real crossing on the double point tree
4. Repeat steps 1, 2 and 3 until... all terms:

$$\begin{tikzcd}
v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\
u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"] \arrow[d, "\pi"'] & A \\
u\mathcal{K} \arrow[ur, "\nu"'] &
\end{tikzcd}$$

• $> n$ X

OR • don't need to do #2 or #3 anymore (no bad Xings, no more real Xings on dpt)

terminate?

↳ if we sweep enough times

↳ we will get lots of clean branches on dpt.

The map P: what it actually "is" (Roukema's version)

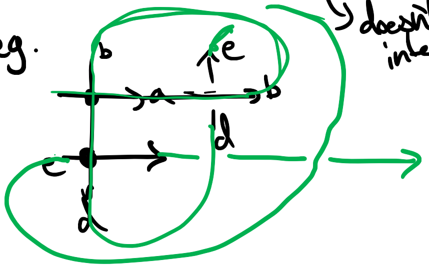
1. Make a first-come-first-serve tree
2. Change all bad Xings to good Xings (using the double point relation)
3. Sweep the first real crossing on the double point tree
4. Repeat steps 1, 2 and 3 until...
5. Set to 0 if $> n$ double points, otherwise, connect strands off the tree in a good way to make a knot

doesn't intersect tree

overstrand comes first.

$$\begin{tikzcd}
v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\
u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"] \arrow[d, "\pi"'] & A \\
u\mathcal{K} \arrow[ur, "\nu"'] &
\end{tikzcd}$$

eg.



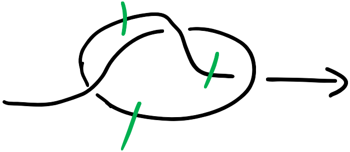
The map P: why it works

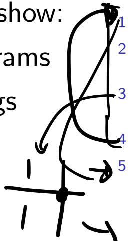
The map P:

$$\begin{tikzcd} v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\ u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"] & A \end{tikzcd}$$

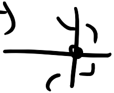
To show that P works, we must show:

1. $\nu \circ P = \nu$ on real knot diagrams
2. $\nu \circ P \circ s^{-1} = 0$ on $> n$ Xings





1. Make a first-come-first-serve tree
2. Change all bad Xings to good Xings (using the double point relation) =
3. Sweep the first real crossing on the double point tree
4. Repeat steps 1, 2 and 3 until...
5. Set to 0 if $> n$ double points, otherwise, connect strands off the tree in a good way to make a knot



just Reid.

The map P: why it works

The map P:

$$\begin{tikzcd} v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\ u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"'] & A \end{tikzcd}$$

1. Make a first-come-first-serve tree
2. ~~2.~~ Change all bad Xings to good Xings (using the double point relation)
3. ~~3.~~ Sweep the first real crossing on the double point tree
4. ~~4.~~ Repeat steps 1, 2 and 3 until...
5. Set to 0 if $> n$ double points, otherwise, connect strands off the tree in a good way to make a knot

To show that P works, we must show:

1. $\nu \circ P = \nu$ on real knot diagrams
2. $\underline{\nu \circ P \circ s^{-1}} = 0$ on $> n$ Xings

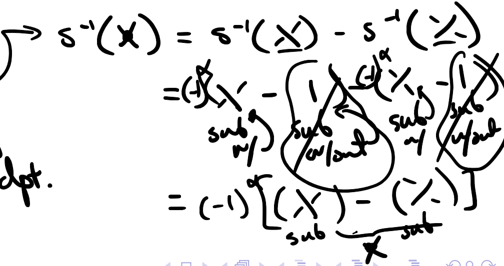
a) first show it's true on "good" diag

don't have to do #2, 3, 4.

first: s^{-1}(good) = good diag.

$s^{-1}(\times)$ = "$\times$"

s^{-1} preserves the dpt.



The map P: why it works

The map P:

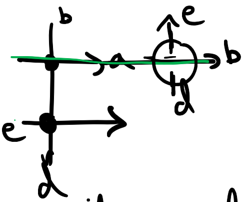
$$\begin{tikzcd} v\mathcal{D} & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\ u\mathcal{D} \arrow[u, leftarrow, "P"] \arrow[ur, "s"] \arrow[r, "\nu"] & A \end{tikzcd}$$

1. Make a first-come-first-serve tree
2. Change all bad Xings to good Xings (using the double point relation)
3. Sweep the first real crossing on the double point tree
4. Repeat steps 1, 2 and 3 until...
5. Set to 0 if $> n$ double points, otherwise, connect strands off the tree in a good way to make a knot

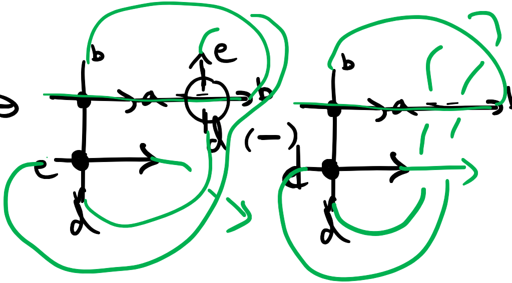
To show that P works, we must show:

1. $\nu \circ P = \nu$ on real knot diagrams
2. $\nu \circ P \circ s^{-1} = 0$ on $> n$ Xings

why is $= 0$ on good diag?



$\xrightarrow{s^{-1}}$



if no real Xing

$> n$ X

The map P: why it works

To show that P works, we must show:

1. $\nu \circ P = \nu$ on real knot diagrams
2. $\nu \circ P \circ s^{-1} = 0$ on $> n$ Xings

The map P:

$$\begin{tikzcd} v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\ u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"'] & A \end{tikzcd}$$

1. Make a first-come-first-serve tree
2. Change all bad Xings to good Xings (using the double point relation)
3. Sweep the first real crossing on the double point tree
4. Repeat steps 1, 2 and 3 until...
5. Set to 0 if $>$ n double points, otherwise, connect strands off the tree in a good way to make a knot

$x - (\bar{x}) \pm \dot{x}$

ok. Let $x y \dot{z} \in v\mathcal{D}$

is the first bad Xing

$\boxed{w(xy\dot{z})}$

$= w((x+1)y\dot{z}) - \cancel{w(y\dot{z})}$

$= \nu P s^{-1}((x+1)y\dot{z})$

$= \nu P(x(y-1)\dot{z})$

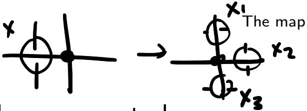
$= \nu P((\bar{x} \pm \dot{x})(y-1)\dot{z}) = \underbrace{\nu P s^{-1}}_{w}((\bar{x} \pm \dot{x} \cancel{+1)y)} \dot{z})$

"good version" of x

$= \boxed{w((\bar{x} \pm \dot{x})y\dot{z})}$

$\bar{x}y\dot{z}$ $\quad$ $\dot{x}y\dot{z}$

The map P: why it works



The map P:

$$\begin{tikzcd} v\mathcal{D} \arrow[d, "P"'] & v\mathcal{D} \arrow[l, "s^{-1}"'] \arrow[d, dashed, "\exists\omega"] \\ u\mathcal{D} \arrow[ur, "s"] \arrow[r, "\nu"] & A \end{tikzcd}$$

1. Make a first-come-first-serve tree
2. Change all bad Xings to good Xings (using the double point relation)
3. Sweep the first real crossing on the double point tree
4. Repeat steps 1, 2 and 3 until...
5. Set to 0 if $>n$ double points, otherwise, connect strands off the tree in a good way to make a knot

To show that P works, we must show:

1. $\nu \circ P = \nu$ on real knot diagrams
2. $\nu \circ P \circ s^{-1} = 0$ on $> n$ Xings

Let $x y \dot{z} \in v\mathcal{D}$

x be first real Xing on dpt.

$$w(xy\dot{z}) = w((x+1)y\dot{z}) - w(y\dot{z})$$

$$= vP(x(y-1)\dot{z})$$

$$= vP(x_1x_2x_3(y-1)\dot{z}) = vPs^{-1}((x_1+1)(x_2+1)(x_3+1)(y)\dot{z})$$

$$= w((x_1+1)(x_2+1)(x_3+1)y\dot{z} - y\dot{z}) \longrightarrow$$ so all terms have $>n$ Xngs

So we are done! But...

Confusions

- Is it necessary to repeat step 1?
- How important is a first-come-first-serve tree? Can we build other trees?
- How can we change the ordering of the issues?
- Is there a more “direct” description of ω?

Generalizations

- To (not-long) knots, tangles, links?

u

v

long knots = knots

long knot diag $\neq$? knot diag

virtual long knot $\neq$ virtual knots