- If $\mu_i = (\lambda_i; \omega_i) \in M(T_i; H_i)$ for $i = 1, 2$ (and, of course, $T_1 \cap T_2 = \emptyset = H_1 \cap H_2$), set
$$\mu_1 * \mu_2 := (\lambda_1 * \lambda_2;\ \iota_1(\omega_1) + \iota_2(\omega_2)),$$
where ι_i are the obvious inclusions $\iota_i \colon CW(T_i) \to CW(T_1 \cup T_2)$.
- The only truly new definition is that of tha^{ux}:
$$(\lambda; \omega) \mathbin{/\!\!/} tha^{ux} := (\lambda;\ \omega + J_u(\lambda_x)) \mathbin{/\!\!/} RC_u^{\lambda_x}.$$
Thus the "new" tha^{ux} is just the "old" tha^{ux}, with an added term of $J_u(\lambda_x)$.

Theorem 4.2. *M, with the operations defined above, is a meta-group-action (MGA). Furthermore, if $\zeta \colon \mathcal{K}_0^{bh} \to M$ is defined on the generators in the same way as ζ_0, except extended by 0 to the CW factor,*
$$\zeta(\epsilon_u^t) := ((); \ 0), \qquad \zeta(\epsilon_x^h) := ((x \to 0);\ 0), \qquad \textit{and} \qquad \zeta(\rho_{ux}^{\pm}) := ((x \to \pm u);\ 0),$$
then it is well-defined; namely, the values above satisfy the relations in Definition def:Kbhz 2.5.

Proof. MORE.

Thus we have a tree-and-wheel valued invariant ζ defined on $\mathcal{K}_0^{bh}$, and thus $\delta \mathbin{/\!\!/} \zeta$ is a tree-and-wheel valued invariant of tangles and w-tangles.

5. Some Computational Examples

Part of the reason I am happy about the invariant ζ is that it is relatively easily computable. Cyclic words are easy to implement, and using the Lyndon basis (e.g. Reutenauer:FreeLie [Re, Chapter 5]), free Lie algebras are easy too. Hence I include here a demo-run of a rough implementation, written in *Mathematica*. The full source files are available at KBH [KBH].

First we load the package `FreeLie.m`, which contains a collection of programs to manipulate series in a completed Lie algebra and series of cyslic words. We tell `FreeLie.m` to show series by default only up to degree 3, and that if two (infinite) series are compared, they are to be compared by default only up to degree 5:

```
<< FreeLie.m
$SeriesShowDegree = 3; $SeriesCompareDegree = 5;
```

Merely as a test of `FreeLie.m`, we tell it to set `t1` to be $\mathrm{bch}(u, v)$. The computer's response is to print that series to degree 3:

```
t1 = BCH[⟨u⟩, ⟨v⟩]
```

$\mathrm{LS}\left[u + v,\ \frac{\overline{uv}}{2},\ \frac{1}{12}\,\overline{u\,\overline{uv}} + \frac{1}{12}\,\overline{\overline{uv}\,v}\right]$

Note that by default Lie series are printed in "top bracket form", which means that brackets are printed above their arguments, rather than around them. Hence $\overline{u\,\overline{uv}}$ means $[u, [u, v]]$. This practice is especially advantageous when it is used on highly-nested elements, when it becomes difficult for the eye to match left brackets with the their corresponding right brackets.

Note also that that `FreeLie.m` utilizes *lazy evaluation*, meaning that when a Lie series (or a series of cyclic words) is defined, its definition is stored but no computation take place until it is printed or until its value (at a certain degree) is exolicitly requested. Hence `t1` is a reference to the entire Lie series $\mathrm{bch}(u, v)$, and not merely to the degrees 1–3 parts of that series, which are printed above. Hence when we request the value of `t1` at degree 6, the computer complies:

```
t1@6 // TopBracketForm
```

$$-\frac{\overline{u\overline{u\overline{u\overline{\overline{uv}v}}}}}{1440}+\frac{1}{360}\overline{u\overline{u\overline{\overline{\overline{uv}v}v}}}+\frac{1}{240}\overline{u\overline{\overline{uv}\,\overline{\overline{uv}v}}}+\frac{1}{720}\overline{u\overline{u\overline{\overline{uv}\,\overline{uv}}}}-\frac{\overline{u\overline{\overline{\overline{\overline{uv}v}v}v}}}{1440}$$

The package `FreeLie.m` know about various free Lie algebra operations, but not about our specific circumstances. Hence we have to make some further definitions. The first few are set-theoretic in nature. We define the "domain" of a function stored as a list of *key*→*value* pairs to be the set of "first elements" of these pair; meaning, the set of keys. We define what it means to remove a key (and its corresponding value), and likewise for a list of keys. We define what it means for two functions to be equal (their domains must be equal, and for every key $\#$, we are to have $\# \mathbin{/\!/} f_1 = \# \mathbin{/\!/} f_2$). We also define how to apply a Lie morphism `mor` to a function (apply it to each value), and how to compare (λ, ω) pairs (in $FL(T)^H \times CW(T)$):

```
Domain[f_List] := First /@ f;
f_ \ key_ := DeleteCases[f, key → _];
f_ \ keys_List := Fold[#1 \ #2 &, f, keys];
f1_List ≡ f2_List := Domain[f1] === Domain[f2] && (And @@ (
      ((# /. f1) ≡ (# /. f2)) & /@ Domain[f1]
    ));
LieMorphism[mor_][f_List] := MapAt[LieMorphism[mor], f, {All, 2}];
M[λ1_, ω1_] ≡ M[λ2_, ω2_] := (λ1 ≡ λ2) && (ω1 ≡ ω2);
```

Next we enter some free-Lie definitions that are not a part of `FreeLie.m`. Namely we define $R^{\lambda_x}_{u,\bar{u}}(s)$ to be the result of "stable application" of the morphism $u \to e^{\mathrm{ad}(\lambda_x)}(\bar{u})$ to s (namely, apply the morphism repeatedly until things stop changing; at any fixed degree this happens after a finite number of iterations). We define $R^{\lambda_x}_u$ to be $R^{\lambda_x}_{u,\bar{u}} \mathbin{/\!/} (\bar{u} \to u)$. Finally, we define J as in Equation (13):

```
RC[u_, λx_LieSeries, ub_][s_] :=
  StableApply[LieMorphism[⟨u⟩ → Ad[λx][⟨ub⟩]], s];
RC[u_, λx_LieSeries][s_] := s // RC[u, λx, ⟨υ⟩] // LieMorphism[⟨υ⟩ → ⟨u⟩];
J[u_, λx_] := Module[{s},
   IntegrateCWSeries[
    div[u, λx // RC[u, s λx]] // LieMorphism[u → Ad[-s λx][u]],
    {s, 0, 1}]];
```

Next is a series of definitions that implement the definitions of $*$, tm, hm, and tha following Sections 3.2 and 4.2:

Commuting missing

Same

```
M /: M[λ1_, ω1_] ⋃ M[λ2_, ω2_] := M[λ1 ⋃ λ2, ω1 + ω2];
tm[u_, v_, w_][λ_List] := λ // LieMorphism[⟨u⟩ → ⟨w⟩, ⟨v⟩ → ⟨w⟩];
tm[u_, v_, w_][M[λ_, ω_]] := LieMorphism[⟨u⟩ → ⟨w⟩, ⟨v⟩ → ⟨w⟩] /@ M[λ, ω];
hm[x_, y_, z_][λ_List] := Union[λ \ {x, y}, {z → BCH[x /. λ, y /. λ]}];
hm[x_, y_, z_][M[λ_, ω_]] := M[λ // hm[x, y, z], ω];
tha[u_, x_][λ_List] := MapAt[RC[u, x /. λ], λ, {All, 2}];
tha[u_, x_][M[λ_, ω_]] :=
  M[λ // tha[u, x], (ω + J[u, x /. λ]) // RC[u, x /. λ]];

ρ⁺[u_, x_] := M[{x → MakeLieSeries[⟨u⟩]}, MakeCWSeries[0]];
ρ⁻[u_, x_] := M[{x → MakeLieSeries[-⟨u⟩]}, MakeCWSeries[0]];

Print /@ {{u = ⟨"u"⟩, v = ⟨"v"⟩, w = ⟨"w"⟩};
   1 → (t1 = M[{
         x → MakeLieSeries[u + v + w],
         y → MakeLieSeries[b[u, v] + b[v, w]]
        }, MakeCWSeries[CW["uvw"]]]),
   2 → (t1 // tm[u, v, u]),
   3 → (t2 = t1 // tm[u, v, u] // tm[u, w, u]),
   4 → (t1 // tm[v, w, v]),
   5 → (t3 = t1 // tm[v, w, v] // tm[u, v, u]),
   6 → (t2 ≡ t3)
  };

1 → M[{x → LS[u + v + w, 0, 0], y → LS[0, ū̄v̄ + v̄w̄, 0]}, CWS[0, 0, CW[uvw]]]
2 → M[{x → LS[2 u + w, 0, 0], y → LS[0, ūw̄, 0]}, CWS[0, 0, CW[uuw]]]
3 → M[{x → LS[3 u, 0, 0], y → LS[0, 0, 0]}, CWS[0, 0, CW[uuu]]]
4 → M[{x → LS[u + 2 v, 0, 0], y → LS[0, ūv̄, 0]}, CWS[0, 0, CW[uvv]]]
5 → M[{x → LS[3 u, 0, 0], y → LS[0, 0, 0]}, CWS[0, 0, CW[uuu]]]
6 → True
```

21

Commentary

add a wheels part

```
Print /@ {
   1 → (t1 = ρ⁺[u, x] ⋃ ρ⁺[v, y] ⋃ ρ⁺[w, z]),
   2 → (t1 // hm[x, y, x]),
   3 → (t2 = t1 // hm[x, y, x] // hm[x, z, x]),
   4 → (t1 // hm[y, z, y]),
   5 → (t3 = t1 // hm[y, z, y] // hm[x, y, x]),
   6 → (t2 ≡ t3)
  };
```

$1 \to M[\{x \to LS[u, 0, 0], y \to LS[v, 0, 0], z \to LS[w, 0, 0]\}, CWS[0, 0, 0]]$

$2 \to M\left[\left\{x \to LS\left[u + v, \frac{\overline{uv}}{2}, \frac{1}{12}\overline{u\overline{uv}} + \frac{1}{12}\overline{\overline{uv}v}\right], z \to LS[w, 0, 0]\right\}, CWS[0, 0, 0]\right]$

$3 \to M\left[\left\{x \to LS\left[u + v + w, \frac{\overline{uv}}{2} + \frac{\overline{uw}}{2} + \frac{\overline{vw}}{2}, \frac{1}{12}\overline{u\overline{uv}} + \frac{1}{12}\overline{u\overline{uw}} + \frac{1}{3}\overline{u\overline{vw}} + \frac{1}{12}\overline{v\overline{vw}} + \frac{1}{12}\overline{\overline{uv}v} + \frac{1}{6}\overline{\overline{uw}v} + \frac{1}{12}\overline{\overline{uw}w} + \frac{1}{12}\overline{\overline{vw}w}\right]\right\}, CWS[0, 0, 0]\right]$

$4 \to M\left[\left\{x \to LS[u, 0, 0], y \to LS\left[v + w, \frac{\overline{vw}}{2}, \frac{1}{12}\overline{v\overline{vw}} + \frac{1}{12}\overline{\overline{vw}w}\right]\right\}, CWS[0, 0, 0]\right]$

$5 \to M\left[\left\{x \to LS\left[u + v + w, \frac{\overline{uv}}{2} + \frac{\overline{uw}}{2} + \frac{\overline{vw}}{2}, \frac{1}{12}\overline{u\overline{uv}} + \frac{1}{12}\overline{u\overline{uw}} + \frac{1}{3}\overline{u\overline{vw}} + \frac{1}{12}\overline{v\overline{vw}} + \frac{1}{12}\overline{\overline{uv}v} + \frac{1}{6}\overline{\overline{uw}v} + \frac{1}{12}\overline{\overline{uw}w} + \frac{1}{12}\overline{\overline{vw}w}\right]\right\}, CWS[0, 0, 0]\right]$

6 → True

test taction, haction

```
Print /@ {
   1 → (t1 = ρ⁺[u, x] ⋃ ρ⁺[v, y] ⋃ ρ⁺[w, z]),
   2 → (t2 = t1 // tm[v, w, v] // hm[x, y, x] // tha[u, z]),
   3 → (t3 = ρ⁺[v, x] ⋃ ρ⁺[w, z] ⋃ ρ⁺[u, y]),
   4 → (t4 = t3 // tm[v, w, v] // hm[x, y, x]),
   5 → (t2 ≡ t4)
  };
```

$1 \to M[\{x \to LS[u, 0, 0], y \to LS[v, 0, 0], z \to LS[w, 0, 0]\}, CWS[0, 0, 0]]$

$2 \to M\left[\left\{x \to LS\left[u + v, -\frac{\overline{uv}}{2}, \frac{1}{12}\overline{u\overline{uv}} + \frac{1}{12}\overline{\overline{uv}v}\right], z \to LS[v, 0, 0]\right\}, CWS[0, 0, 0]\right]$

$3 \to M[\{x \to LS[v, 0, 0], y \to LS[u, 0, 0], z \to LS[w, 0, 0]\}, CWS[0, 0, 0]]$

$4 \to M\left[\left\{x \to LS\left[u + v, -\frac{\overline{uv}}{2}, \frac{1}{12}\overline{u\overline{uv}} + \frac{1}{12}\overline{\overline{uv}v}\right], z \to LS[v, 0, 0]\right\}, CWS[0, 0, 0]\right]$

5 → True

8/17

Borromean

6. The Relation with the BF Topological Quantum Field Theory

7. The Simplest Non-Commutative Reduction and an Ultimate Alexander Invariant

8. The Relation with Alekseev-Torossian and with [BND]

WKO

9. Odds and Ends

ubsec:Hopf

9.1. Linking Numbers and Signs. If x is an oriented S^1 and u is an oriented S^2 in an oriented S^4 (or $\mathbb{R}^4$) and the two are disjoint, their linking number l_{ux} is defined as follows. Pick a ball B whose oriented boundary is u (using the "outward pointing normal" convention