Computing Quantum Mapping Class Group Representations with Haskell

Paul Gustafson

Motivation

Why should we care about quantum mapping class group representations?

- Topological quantum computation
- Understand mapping class groups (algebraic geometry)
- Understand tensor categories (representation theory, operator algebras)
- Intrinsic beauty

Mapping class group definition

- Let $\Sigma = \Sigma_{g,b}^{m}$ be the oriented compact surface of genus g with b boundary components and m marked points in its interior.

Mapping class group definition

- Let $\Sigma = \Sigma_{g,b}^{m}$ be the oriented compact surface of genus g with b boundary components and m marked points in its interior.
- The mapping class group of Σ, $MCG(\Sigma)$, is the group of isotopy classes of orientation-preserving homeomorphisms of Σ that preserve the boundary *pointwise* and preserve the marked points *setwise*.

Mapping class group definition

- Let $\Sigma = \Sigma_{g,b}^{m}$ be the oriented compact surface of genus g with b boundary components and m marked points in its interior.
- The mapping class group of Σ, $MCG(\Sigma)$, is the group of isotopy classes of orientation-preserving homeomorphisms of Σ that preserve the boundary *pointwise* and preserve the marked points *setwise*.
- Examples
  - $\mathrm{MCG}(\Sigma_{0,1}^{m}) = B_m$
  - $\mathrm{MCG}(\Sigma_{1,0}^{0}) = SL(2, \mathbb{Z})$

Mapping class group definition

- Let $\Sigma = \Sigma_{g,b}^{m}$ be the oriented compact surface of genus g with b boundary components and m marked points in its interior.
- The mapping class group of Σ, $MCG(\Sigma)$, is the group of isotopy classes of orientation-preserving homeomorphisms of Σ that preserve the boundary *pointwise* and preserve the marked points *setwise*.
- Examples
  - $\mathrm{MCG}(\Sigma_{0,1}^{m}) = B_m$
  - $\mathrm{MCG}(\Sigma_{1,0}^{0}) = SL(2,\mathbb{Z})$
- Birman (1969) found "nice" finite generating set for the mapping class group of any compact surface.

(2+1)-Topological Quantum Field Theories

- Quantum mapping class group representations come from (2+1)-TQFTs

(2+1)-Topological Quantum Field Theories

- Quantum mapping class group representations come from (2+1)-TQFTs
- A (2+1)-TQFT assigns a vector space to oriented 2-manifolds
- It also assigns a vector to oriented 3-manifolds. This vector inhabits the vector space corresponding to the 3-manifold's boundary.
- To get a mapping class group representation, apply the TQFT to mapping cylinders.

(2+1)-Topological Quantum Field Theories

- Quantum mapping class group representations come from (2+1)-TQFTs
- A (2+1)-TQFT assigns a vector space to oriented 2-manifolds
- It also assigns a vector to oriented 3-manifolds. This vector inhabits the vector space corresponding to the 3-manifold's boundary.
- To get a mapping class group representation, apply the TQFT to mapping cylinders.
- History: Witten and Atiyah (1980s)

(2+1)-Topological Quantum Field Theories

- Quantum mapping class group representations come from (2+1)-TQFTs
- A (2+1)-TQFT assigns a vector space to oriented 2-manifolds
- It also assigns a vector to oriented 3-manifolds. This vector inhabits the vector space corresponding to the 3-manifold's boundary.
- To get a mapping class group representation, apply the TQFT to mapping cylinders.
- History: Witten and Atiyah (1980s)
- Examples of mathematical (2+1)-TQFTs:
  - Reshitikhin-Turaev TQFT (input: modular category)
  - Turaev-Viro-Barret-Westbury TQFT (input: spherical fusion category)

Monoidal categories

A **monoidal category** is a category $\mathcal{C}$ equipped with

- a tensor product – a bifunctor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$
- an associativity isomorphism – a natural isomorphism $\alpha : (\cdot \otimes \cdot) \otimes \cdot \to \cdot \otimes (\cdot \otimes \cdot)$
- a unit object $1 \in \mathcal{C}$
- a left unitor – a natural isomorphism $\lambda_X : 1 \otimes X \to X$
- a right unitor – a natural isomorphism $\rho_X : X \otimes 1 \to X$,

satisfying certain coherence conditions (the triangle and pentagon axioms).

Rigid monoidal categories

- Let X be an object of a monoidal category $\mathcal{C}$. A **left dual** to X is an object X^* equipped with
  - an evaluation morphism, $\mathrm{ev}_X : X^* \otimes X \to 1$
  - a coevaluation morphism, $\mathrm{coev}_X : 1 \to X \otimes X^*$,

  satisfying certain coherence conditions (the snake equations).

Rigid monoidal categories

- Let X be an object of a monoidal category $\mathcal{C}$. A **left dual** to X is an object X^* equipped with
  - an evaluation morphism, $\mathrm{ev}_X : X^* \otimes X \to 1$
  - a coevaluation morphism, $\mathrm{coev}_X : 1 \to X \otimes X^*$,

  satisfying certain coherence conditions (the snake equations).
- Right duals are defined similarly.
- An object X is **rigid** if it has both a left and right dual.
- A monoidal category is rigid if all of its objects are rigid.

Fusion category

- A **fusion category** is a rigid semisimple linear monoidal category (tensor category), with only finitely many isomorphism classes of simple objects, such that the endomorphisms of the unit object form just the ground field k.

Fusion category

- A **fusion category** is a rigid semisimple linear monoidal category (tensor category), with only finitely many isomorphism classes of simple objects, such that the endomorphisms of the unit object form just the ground field k.
- Example: Representation category of a finite group

Spherical categories

- A **pivotal category** is a rigid monoidal category equipped with a pivotal structure, i.e. a tensor natural automorphism $j_X : X \to (X^*)^*$.
- A pivotal structure defines left and right traces $\mathrm{End}(X) \to \mathrm{End}(1)$ for every object X.

Spherical categories

- A **pivotal category** is a rigid monoidal category equipped with a pivotal structure, i.e. a tensor natural automorphism $j_X : X \to (X^*)^*$.
- A pivotal structure defines left and right traces $\mathrm{End}(X) \to \mathrm{End}(1)$ for every object X.
- A **spherical category** is a monoidal category such that all left and right traces coincide.

Example: Vect_G^ω

- Let G be a finite group, and $\omega : G \times G \times G \to \mathbb{C}$ be a 3-cocycle. The spherical fusion category Vect_G^ω is the skeletal category of G-graded finite-dimensional vector spaces with the following modified structural morphisms, where V_g is the simple object:
  - The associator $a_{g,h,k} : (V_g \otimes V_h) \otimes V_k \to V_g \otimes (V_h \otimes V_k)$

    $$a_{g,h,k} = \omega(g, h, k)$$

  - The evaluator $ev_g : V_g^* \otimes V_g \to 1$

    $$ev_g = \omega(g^{-1}, g, g^{-1})$$

  - The pivotal structure $j_g : V_g^{**} \to V_g$

    $$j_g = \omega(g^{-1}, g, g^{-1})$$

The TVBW space associated to a 2-manifold

- Let $\mathcal{A}$ be a spherical fusion category, and Σ an oriented compact surface with boundary.
- Using Kirillov's definitions, the representation space we consider is

$$H := \frac{\mathcal{A}\text{-colored graphs in } \Sigma}{\text{local relations}}$$

The TVBW space associated to a 2-manifold

- Let $\mathcal{A}$ be a spherical fusion category, and Σ an oriented compact surface with boundary.
- Using Kirillov's definitions, the representation space we consider is

$$H := \frac{\mathcal{A}\text{-colored graphs in } \Sigma}{\text{local relations}}$$

- The vector space H is canonically isomorphic to the Turaev-Viro state sum vector space associated to Σ. This isomorphism commutes with the mapping class group action.

Colored graphs in Σ

- Let $\Gamma \subset \Sigma$ be an undirected finite graph embedded in Σ.

Colored graphs in Σ

- Let $\Gamma \subset \Sigma$ be an undirected finite graph embedded in Σ.
- Define E^{or} to be the set of orientation edges of Γ, i.e. pairs $\mathbf{e} = (e, \text{orientation of } e)$; for such an oriented edge $\mathbf{e}$, we denote by $\bar{\mathbf{e}}$ the edge with opposite orientation.

Colored graphs in Σ

- Let $\Gamma \subset \Sigma$ be an undirected finite graph embedded in Σ.
- Define E^{or} to be the set of orientation edges of Γ, i.e. pairs $\mathbf{e} = (e, \text{orientation of } e)$; for such an oriented edge $\mathbf{e}$, we denote by $\bar{\mathbf{e}}$ the edge with opposite orientation.
- A *coloring* of Γ is the following data:
  - Choice of an object $V(\mathbf{e}) \in \operatorname{Obj} \mathcal{A}$ for every oriented edge $\mathbf{e} \in E^{or}$ so that $V(\bar{\mathbf{e}}) = V(\mathbf{e})^*$.

Colored graphs in Σ

- Let $\Gamma \subset \Sigma$ be an undirected finite graph embedded in Σ.
- Define E^{or} to be the set of orientation edges of Γ, i.e. pairs $\mathbf{e} = (e, \text{orientation of } e)$; for such an oriented edge $\mathbf{e}$, we denote by $\bar{\mathbf{e}}$ the edge with opposite orientation.
- A *coloring* of Γ is the following data:
  - Choice of an object $V(\mathbf{e}) \in \operatorname{Obj} \mathcal{A}$ for every oriented edge $\mathbf{e} \in E^{or}$ so that $V(\bar{\mathbf{e}}) = V(\mathbf{e})^*$.
  - Choice of a vector $\varphi(v) \in \operatorname{Hom}_{\mathcal{A}}(1, V_1 \otimes \cdots \otimes V_n)$ for every interior vertex v, where $\mathbf{e}_1, \ldots, \mathbf{e}_n$ are edges incident to v, taken in counterclockwise order and with outward orientation.

Local relations

- Isotopy of the graph embedding

Local relations

- Isotopy of the graph embedding
- Linearity in the vertex colorings

Local relations

- Isotopy of the graph embedding
- Linearity in the vertex colorings

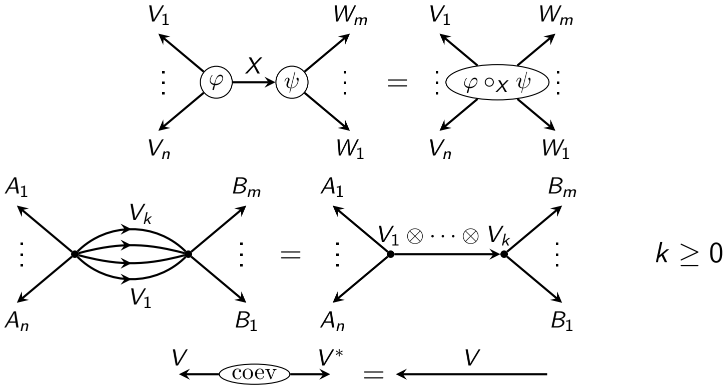

Figure : The remaining local relations.

Consequences of the local relations

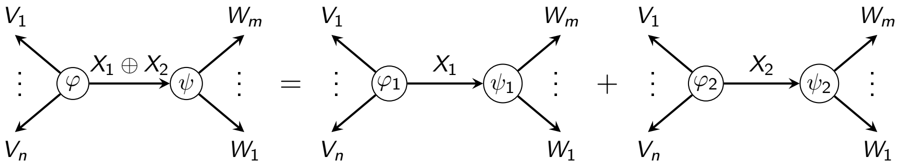

Figure : Additivity in edge colorings. Here φ_1, φ_2 are compositions of φ with projector $X_1 \oplus X_2 \to X_1$ (respectively, $X_1 \oplus X_2 \to X_2$), and similarly for ψ_1, ψ_2.

- Additivity in edge colorings

Consequences of the local relations

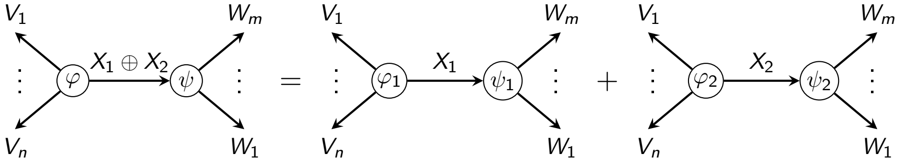

Figure : Additivity in edge colorings. Here φ_1, φ_2 are compositions of φ with projector $X_1 \oplus X_2 \to X_1$ (respectively, $X_1 \oplus X_2 \to X_2$), and similarly for ψ_1, ψ_2.

- Additivity in edge colorings
- A colored graph may be evaluated on any disk $D \subset S$, giving an equivalent colored graph Γ' such that Γ' is identical to Γ outside of D, has the same colored edges crossing ∂D, and contains at most one colored vertex within D.

Overall Strategy

- Find a basis of colored graphs for the representation space for a surface
- "Calculate" the representation of each mapping class group generator with respect to this basis
- Analyze the image of the representation (Is it finite? Can we do universal quantum computation with it (possibly adding extra measurements)?)

Modified Property F conjecture

Conjecture (Rowell)

A TVBW mapping class group representation associated to a spherical fusion category $\mathcal{A}$ has finite image iff $\mathcal{A}$ is weakly integral, i.e. the squared dimension of every simple object is an integer.

Related Work

Theorem (Ng–Schauenberg)

Every modular representation associated to a modular category has finite image.

Related Work

Theorem (Ng–Schauenberg)

Every modular representation associated to a modular category has finite image.

Theorem (Etingof–Rowell–Witherspoon)

The braid group representation associated to the modular category $\mathrm{Mod}(D^{\omega}(G))$ *has finite image.*

Related Work

Theorem (Ng–Schauenberg)

Every modular representation associated to a modular category has finite image.

Theorem (Etingof–Rowell–Witherspoon)

The braid group representation associated to the modular category $\mathrm{Mod}(D^\omega(G))$ *has finite image.*

ERW also asked if, more generally, all mapping class group representations associated to $\mathrm{Mod}(D^\omega(G))$ have finite image.

Related Work

Theorem (Ng–Schauenberg)

Every modular representation associated to a modular category has finite image.

Theorem (Etingof–Rowell–Witherspoon)

The braid group representation associated to the modular category $\mathrm{Mod}(D^\omega(G))$ *has finite image.*

ERW also asked if, more generally, all mapping class group representations associated to $\mathrm{Mod}(D^\omega(G))$ have finite image.

Theorem (Fjelstad–Fuchs)

Every mapping class group representation of a closed surface with at most one marked point associated to $\mathrm{Mod}(D(G))$ *has finite image.*

Answering ERW's question: First Dehn twist

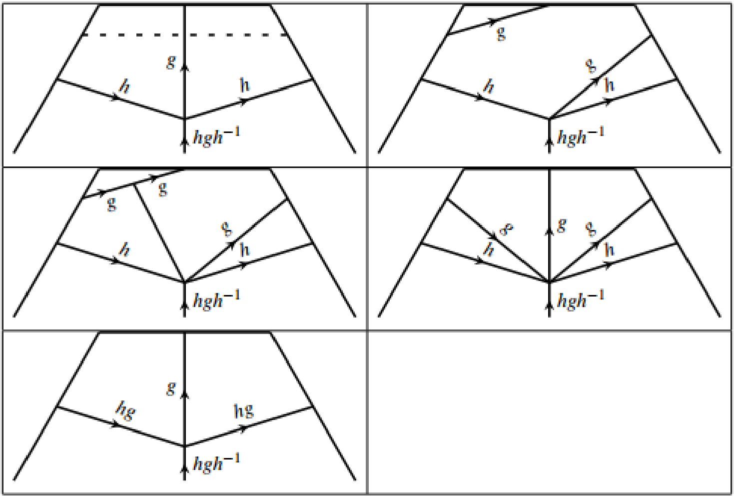

TABLE 1. First type of Dehn twist. Unlabeled interior edges are colored by the group identity element.

Second Dehn twist

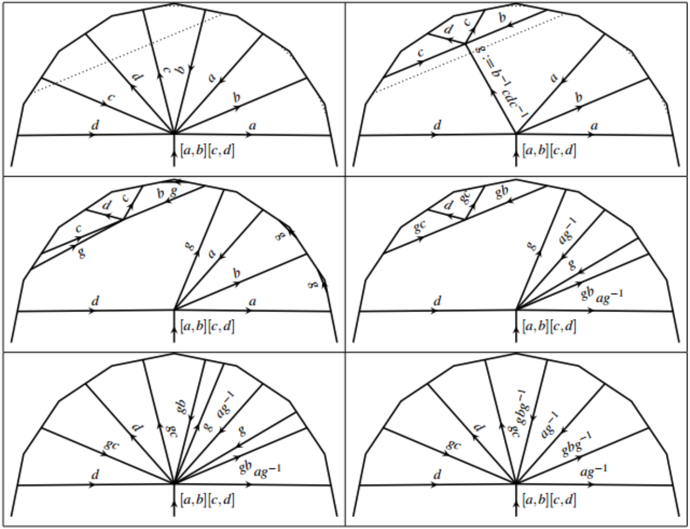

TABLE 2. Second type of Dehn twist.

Braid generator

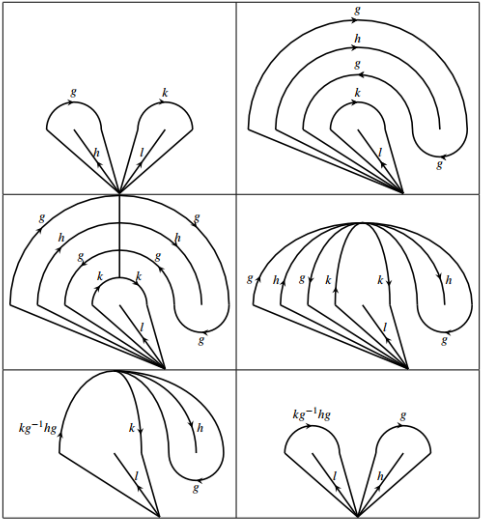

TABLE 3. A braid generator. Unlabeled interior edges are colored by the group identity element.

Dragging a point

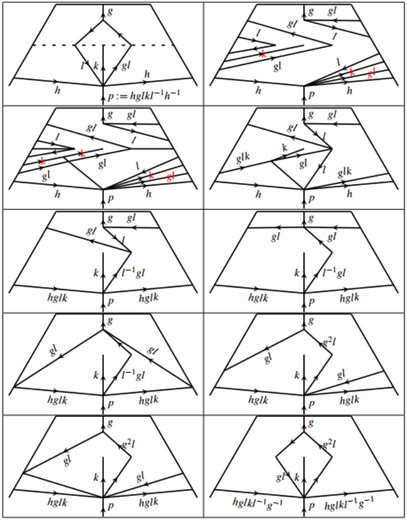

TABLE 4. Pulling a boundary component along a generator for the fundamental group of the corresponding closed surface. Unlabeled interior edges are colored by the group identity element.

First result

Theorem

The image of any Vect_G^ω *TVBW representation* ρ *of a mapping class group of an orientable, compact surface* Σ *with boundary is finite.*

Sketch of proof.

- For any k, let $\mu_{|G|}$ denote the set of $|G|$-th roots of unity. Then ω is cohomologous to a cocycle taking values in $\mu_{|G|}$. Since cohomologous cocycles give rise to equivalent spherical categories Vect_G^ω, WLOG ω takes values in $\mu_{|G|}$.

First result

Theorem

The image of any Vect_G^ω *TVBW representation* ρ *of a mapping class group of an orientable, compact surface* Σ *with boundary is finite.*

Sketch of proof.

- For any k, let $\mu_{|G|}$ denote the set of $|G|$-th roots of unity. Then ω is cohomologous to a cocycle taking values in $\mu_{|G|}$. Since cohomologous cocycles give rise to equivalent spherical categories Vect_G^ω, WLOG ω takes values in $\mu_{|G|}$.
- Let $L \subset \mathrm{MCG}(\Sigma)$ be the Birman generating set. From the figures and the definitions of structural morphisms of Vect_G^ω, $\rho(L)$ acts by permutations on $\mu_{|G|}S$ for a spanning set S.

First result

Theorem

The image of any Vect_G^ω *TVBW representation* ρ *of a mapping class group of an orientable, compact surface* Σ *with boundary is finite.*

Sketch of proof.

- For any k, let $\mu_{|G|}$ denote the set of $|G|$-th roots of unity. Then ω is cohomologous to a cocycle taking values in $\mu_{|G|}$. Since cohomologous cocycles give rise to equivalent spherical categories Vect_G^ω, WLOG ω takes values in $\mu_{|G|}$.
- Let $L \subset \mathrm{MCG}(\Sigma)$ be the Birman generating set. From the figures and the definitions of structural morphisms of Vect_G^ω, $\rho(L)$ acts by permutations on $\mu_{|G|}S$ for a spanning set S.
- Let $B \subset S$ be a basis for H. Then $\rho(\mathrm{MCG}(\Sigma))B \subset \rho(\mathrm{MCG}(\Sigma))S \subset \mu_{|G|}S$.

First result

Theorem

The image of any Vect_G^ω *TVBW representation* ρ *of a mapping class group of an orientable, compact surface* Σ *with boundary is finite.*

Sketch of proof.

- For any k, let $\mu_{|G|}$ denote the set of $|G|$-th roots of unity. Then ω is cohomologous to a cocycle taking values in $\mu_{|G|}$. Since cohomologous cocycles give rise to equivalent spherical categories Vect_G^ω, WLOG ω takes values in $\mu_{|G|}$.
- Let $L \subset \mathrm{MCG}(\Sigma)$ be the Birman generating set. From the figures and the definitions of structural morphisms of Vect_G^ω, $\rho(L)$ acts by permutations on $\mu_{|G|}S$ for a spanning set S.
- Let $B \subset S$ be a basis for H. Then $\rho(\mathrm{MCG}(\Sigma))B \subset \rho(\mathrm{MCG}(\Sigma))S \subset \mu_{|G|}S$.
- Thus, $|\mathrm{Im}(\rho)| < \infty$.

Next steps

- Tambara-Yamagami categories
  - Simple class of categories with “multi-fusion channels”
  - Simple examples of gauging (with respect to group inversion action)
- Problem: Want to calculate actual matrices

Easiest example (first Dehn twist, Vect_G^{ω}):

$$\frac{\omega(h,g,h^{-1})\omega(h,gh^{-1},hg^{-1}h^{-1})\omega(g,h^{-1},hg^{-1}h^{-1})\omega(g,g^{-1}h^{-1},h)}{\omega(g^{-1},g^{-1},g)\omega(g^{-1},g^{-1}h^{-1},h)\omega(g^{-1},h^{-1},hg^{-1}h^{-1})\omega(g,g^{-2}h^{-1},hg)}\cdot$$
$$\frac{\omega(g,g^{-1},h^{-1})\omega(g,g^{-1}h^{-1},h)\omega(g^{-1}h^{-1},h,g)\omega(g^{-2}h^{-1},h,g)}{\omega(g,g^{-1}h^{-1},hg^{-1}h^{-1})\omega(hg,h^{-1},hg^{-1}h^{-1})\omega(hg,g,g^{-1}h^{-1})}$$
$$=\frac{\omega(h,g,g^{-1}h^{-1})\omega(g,g^{-1}h^{-1},h)}{\omega(g^{-1},g^{-1},g)\omega(g^{-1},g^{-1}h^{-1},h)\omega(g^{-1},h^{-1},hg^{-1}h^{-1})\omega(g,g^{-2}h^{-1},hg)}\cdot$$
$$\frac{\omega(g,g^{-1},h^{-1})\omega(g,g^{-1}h^{-1},h)\omega(g^{-1}h^{-1},h,g)\omega(g^{-2}h^{-1},h,g)}{\omega(g,g^{-1}h^{-1},hg^{-1}h^{-1})\omega(hg,g,g^{-1}h^{-1})}$$
$$=\frac{\omega(h,g,g^{-1}h^{-1})\omega(g,g^{-1}h^{-1},h)}{\omega(g^{-1},g^{-1},g)\omega(g^{-1},g^{-1}h^{-1},h)\omega(g,g^{-2}h^{-1},hg)}\cdot$$
$$\frac{\omega^2(g,g^{-1},h^{-1})\omega(g,g^{-1}h^{-1},h)\omega(g^{-1}h^{-1},h,g)\omega(g^{-2}h^{-1},h,g)}{\omega(g,g^{-1},g^{-1}h^{-1})\omega(hg,g,g^{-1}h^{-1})}$$
$$=\frac{\omega(h,g,g^{-1}h^{-1})\omega^2(g,g^{-1}h^{-1},h)\omega^2(g,g^{-1},h^{-1})\omega^2(g^{-1}h^{-1},h,g)}{\omega(g,g^{-2}h^{-1},hg)\omega(g,g^{-1},g^{-1}h^{-1})\omega(hg,g,g^{-1}h^{-1})\omega(g^{-1},g^{-1}h^{-1},hg)}$$
$$=\frac{\omega(h,g,g^{-1}h^{-1})\omega^2(g,g^{-1}h^{-1},h)\omega^2(g,g^{-1},h^{-1})\omega^2(g^{-1}h^{-1},h,g)}{\omega(hg,g,g^{-1}h^{-1})}$$
$$=\frac{\omega(h,g,g^{-1}h^{-1})\omega^2(g^{-1}h^{-1},h,g)}{\omega^2(g^{-1},h^{-1},h)\omega(hg,g,g^{-1}h^{-1})}$$

Why is it hard?

- Computationally intensive
- High level of abstraction

Solution: Haskell

```
data Stringnet = Stringnet
                 { vertices       :: [InteriorVertex]
                 , edges          :: [Edge]
                 , disks          :: [Disk]
                 , perimeter      :: Disk -> [Edge]

                 -- image under contractions
                 , imageVertex     :: Vertex -> Vertex

                 , edgeTree       :: Vertex -> Tree Edge
                 , morphismLabel :: InteriorVertex
                                     -> Morphism
                 , objectLabel    :: Edge -> Object
                 }
```

Two-Complex Datatypes

```
data Puncture = LeftPuncture | RightPuncture
data InteriorVertex = Main | Midpoint Edge | Contraction Edge
data Vertex = Punc Puncture | IV InteriorVertex
data InitialEdge = LeftLoop | RightLoop | LeftLeg | RightLeg
data Edge
  = IE InitialEdge
  | FirstHalf Edge
  | SecondHalf Edge
  | Connector Edge Edge Disk
  | TensorE Edge Edge
  | Reverse Edge
data Disk = Outside | LeftDisk | RightDisk | Cut Edge
```

Objects

```
data Object
  = OVar InitialEdge
  | One
  | Star Object
  | TensorO Object Object
```

Morphisms

```
data Morphism
  = Phi
  | Id Object
  | Lambda Object
  | LambdaI Object
  | Rho Object
  | RhoI Object
  | Alpha Object Object Object
  | AlphaI Object Object Object
  | Coev Object
  | Ev Object
  | TensorM Morphism Morphism
  | PivotalJ Object
  | PivotalJI Object
  | Compose Morphism Morphism
```

Local Moves

```
tensor :: Disk -> State Stringnet ()
contract :: Edge -> State Stringnet InteriorVertex
connect :: Edge -> Edge -> Disk -> State Stringnet Edge
addCoev :: Edge
  -> State Stringnet (InteriorVertex, Edge, Edge)
```

Vertex Hom-Space Moves

```
associateL ::
  InteriorVertex -> Tree Edge -> State Stringnet (Tree Edge)
associateR ::
  InteriorVertex -> Tree Edge -> State Stringnet (Tree Edge)
isolateR :: InteriorVertex -> State Stringnet ()
isolateL :: InteriorVertex -> State Stringnet ()
zMorphism :: Object -> Object -> Morphism -> Morphism
zRotate :: InteriorVertex -> State Stringnet ()
isolate2 :: Edge -> Edge -> InteriorVertex
  -> State Stringnet ()
```

Braid move

```
(_,l1,r1) <- addCoev $ IE LeftLoop
(_,l2,r2) <- addCoev $ IE LeftLeg
(_,r13,l3) <- addCoev r1
(_,_,r4) <- addCoev $ IE RightLoop
e1 <- connect (rev l1) r2 LeftDisk
e2 <- connect (rev l2) (rev r13) (Cut $ e1)
e3 <- connect l3 r4 Outside
contract e1
contract e2
contract e3
tensor (Cut $ rev e1)
tensor (Cut $ rev e2)
tensor (Cut $ rev e3)
v <- contract r4
```

+ some reassociating

Tambara-Yamagami categories

Let A be a finite abelian group, χ a bicharacter on A, and $\nu \in \{\pm 1\}$. The **Tamabara-Yamagami category** $\mathcal{TY}(A, \chi, \nu)$ is the skeletal spherical category with simple objects $\{a : a \in A\} \cup \{m\}$, fusion rules given by

$$a \otimes b = ab \text{ for } a, b \in A \qquad a \otimes m = m \qquad m \otimes m = \bigoplus_{a \in A} a,$$

and the following nontrivial structural morphisms

$$\alpha_{a,m,b} = \chi(a,b)\, \mathrm{id}_m \qquad \alpha_{m,a,m} = \bigoplus_{b \in A} \chi(a,b)\, \mathrm{id}_b$$

$$\alpha_{m,m,m} = (\nu |A|^{-1/2} \chi^{-1}(a,b)\, \mathrm{id}_m)_{a,b \in A},$$

$$j_m = \nu\, \mathrm{id}_m \qquad ev_m = \nu |A|^{1/2} \pi_1$$

TambaraYamagami types

```
newtype AElement = AElement Int

newtype RootOfUnity = RootOfUnity AElement

data Scalar =  Scalar
  { coeff :: [Int]
  , tauExp :: Sum Int
  }
```

A scalar is represented as $\tau^k \sum_{i=0}^{n-1} a_i \zeta_n^i$

TambaraYamagami types

```
data SimpleObject =
  -- Group-element-indexed simple objects
  AE !AElement

  -- non-group simple object
  | M

newtype Object = Object
  { multiplicity_ :: [Int]
  }
```

TambaraYamagami types

```
data Morphism = Morphism
  { domain   :: Object
  , codomain :: Object
  , subMatrix_ :: [M.Matrix Scalar]
  }
```

TambaraYamagami types

```
data BasisElement = BasisElement
  { initialLabel :: S.InitialEdge -> SimpleObject
  , oneIndex :: Int
  }
```

Output Example

```
*Test> (zipWith (,) answer finalET) !! 81
([( Scalar {coeff = [4,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,4], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [4,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,4], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [4,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,4], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [4,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,4], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
( Scalar {coeff = [0,0], tauExp = Sum {getSum = 0}} )
,,],[[0,0,1],[0,0,1],[0,0,1],[2,2,0],[0,0,1],[2,2,0]])
```

Next steps

- Distribute tensor products using dictionary ordering
- Double check coev and ev definitions
- Verify snake equations
- Verify braid relations
- Compare with Ising R-matrices
- Optimize composition to be local wrt tensor products

Thanks

Thanks for listening!
Any questions?