A Braided Lambda Calculus

Masahito Hasegawa
Research Institute for Mathematical Sciences
Kyoto University
Kyoto, Japan
hassei@kurims.kyoto-u.ac.jp

We present an untyped linear lambda calculus with braids, the corresponding combinatory logic, and the semantic models given by crossed G-sets.

1 Introduction

A *braid with n-strands* [3, 4, 9] is n copies of the interval $[0,1]$ smoothly embedded in the cube $[-\frac{1}{2},\frac{1}{2}]\times[0,1]\times[0,1]$ (Figure 1) such that

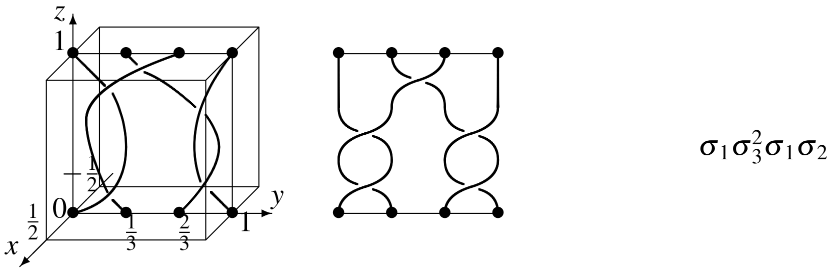
- each $t\in[0,1]$ is mapped to a point in the plane $\{(x,y,z)\mid z=t\}$
- the end points $0\in[0,1]$ are sent to the n points $\{(0,\frac{k}{n-1},0)\mid k=0,\dots,n-1\}$
- the end points $1\in[0,1]$ are sent to the n points $\{(0,\frac{k}{n-1},1)\mid k=0,\dots,n-1\}$

Two braids are identified if there is a continuous deformation between them preserving the boundaries (the ambient isotopy). It is well-known that braids (modulo ambient isotopy) can be identified with their projections to a plane modulo Reidemeister moves, and also with the elements of the braid group:

$$\begin{array}{cl} & \{\text{braids of } n\text{-strands}\}/\text{ambient isotopy} \\ \cong & \{\text{braid diagrams of } n\text{-strands}\}/\text{Reidemeister moves} \\ \cong & \text{Braid group } B_n \end{array}$$

In this paper, we introduce an *untyped linear lambda calculus with braids*, in which every permutation/exchange of variables is realized by a braid. Thus, for a term M with n (ordered) free variables and a braid s with n strands, we introduce a term $[s]M$ in which the free variables are permutated by s:

$$\frac{x_1,x_2,\dots,x_n\vdash M \quad s:\text{braid with } n \text{ strands}}{x_{s(1)},x_{s(2)},\dots,x_{s(n)}\vdash[s]M}\,\mathsf{braid}$$

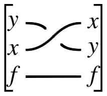
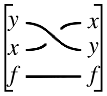


braid with 4 strands braid diagram element of the braid group B_4

Figure 1: Braids

Submitted to:
Linearity-TLLA 2020

For instance, we have braided **C**-combinators

$\mathbf{C}^{+} \equiv \lambda fxy.$  $(f\,y\,x)$ $\quad$ $\mathbf{C}^{-} \equiv \lambda fxy.$  $(f\,y\,x)$

which are "implementations" of the standard **C**-combinator $\lambda fxy.f\,y\,x$ using braids. The idea of realizing a braided calculus as a planar calculus enriched with explicit braids is not new, see for instance [5].

Braids do not play a serious role in most of the conventional computational models, and for the time being this work is largely a mathematical exercise with no real application. Nevertheless, let us say a little bit more on the motivation of this work and its potential applications. Extensionally, permutations (symmetry/exchange) are used for swapping two data. On the other hand, braids provide non-extensional information on how to implement permutations in three dimensions. If braids have some computational meaning, it should be something about low-level (intermediate) codes to be compiled in some 3D computational architectures. One such computational model allowing "braids for implementation" reading is *Topological Quantum Computation* [10], where the topological information of anyons in 3D space-time does matter; we hope that this work will find some usage in this context.

Our calculus is untyped. Compared to the typed case (including braided MLL [5] and tensorial logic [11]), we have a simpler syntax and subtler, more challenging semantics - while the simply typed braided lambda calculus can be modelled by any braided monoidal closed category, the untyped calculus requires a reflexive object, which is hard to find in the well-known braided categories in TQFT [13]. We overcome this difficulty by using a braided relational model constructed in our previous work [7].

Our contributions are summarized as follows.

- We formulate a braided lambda calculus whose syntax is a mild modification of the untyped linear lambda calculus with explicit braids (Section 2).
- We introduce the corresponding combinatory logic and show the combinatory completeness (Section 3).
- We give categorical semantics given by reflexive objects in braided monoidal closed categories, and present some concrete models using crossed G-sets (Section 4).

2 A braided lambda calculus

2.1 Syntax of the calculus

The untyped braided lambda calculus is an extension of the planar lambda calculus (the linear lambda calculus with no exchange)[1] with a rule for introducing braided terms.

$$\frac{}{x \vdash x}\ \text{variable} \qquad \frac{\Gamma, x \vdash M}{\Gamma \vdash \lambda x.M}\ \text{abstraction} \qquad \frac{\Gamma \vdash M \quad \Gamma' \vdash N}{\Gamma, \Gamma' \vdash MN}\ \text{application}$$

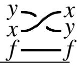
$$\frac{x_1, x_2, \ldots, x_n \vdash M \quad s : \text{braid with } n \text{ strands}}{x_{s(1)}, x_{s(2)}, \ldots, x_{s(n)} \vdash [\sigma]M}\ \text{braid}$$

[1] In the literature, there are (at least) two different notions of "planar lambda terms". Some authors emply the "left" abstraction rule (e.g. [15]) $\dfrac{x, \Gamma \vdash M}{\Gamma \vdash \lambda x.M}$ whereas others (e.g. [2]) use the same "right" abstraction rule as ours; see [15] for some comparison. Our choice has the advantage of preservation of planarity under the $\beta\eta$-conversions, and allows simpler semantics by reflexive objects in monoidal (right) closed categories.

where $s(i)$ denotes the outcome of applying the permutation on $\{1,2,\dots,n\}$ induced by s to i. Formally, a braid with n strands will be an element of the braid group B_n, and the braided term $[s]M$ is the result of the group action of B_n on terms with n free variables. However, for readability, we might present braids graphically, often with labels indicating the correspondence to variables.

Example 1 (braided C-combinator) *The derivation of the combinator* $\mathbf{C}^+$ *in the introduction is*

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{f \vdash f} \quad \cfrac{}{y \vdash y}}{f,y \vdash f\,y} \quad \cfrac{}{x \vdash x}}{f,y,x \vdash f\,y\,x} \quad s =}{f,x,y \vdash [s]\,(f\,y\,x)}}{f,x \vdash \lambda y.[s](f\,y\,x)}}{f \vdash \lambda xy.[s](f\,y\,x)}}{\vdash \lambda fxy.\,[s]\,(f\,y\,x)}$$



Remark 1 (Contexts are redundant) *In the braided lambda calculus, the context is always uniquely determined by the term, thus redundant. Given a braided lambda term M, we define the* list $\mathrm{cxt}(M)$ *of free variables in M as follows:* $\mathrm{cxt}(x) = x$, $\mathrm{cxt}(MN) = \mathrm{cxt}(M),\mathrm{cxt}(N)$, $\mathrm{cxt}(\lambda x.M) = \Gamma$ *where* $\mathrm{cxt}(M) = \Gamma, x$, *and* $\mathrm{cxt}([s]M) = s(\mathrm{cxt}(M))$ *where* $s(x_1,\dots,x_n) = x_{s(1)},\dots,x_{s(n)}$. *It follows that* $\Gamma \vdash M$ *iff* $\mathrm{cxt}(M) = \Gamma$. *Hence the context of a braided term is unique: if both* $\Gamma \vdash M$ *and* $\Gamma' \vdash M$ *are derivable, then* Γ *is identical to* Γ'.

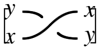
2.2 Equational theory

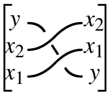
The $\beta\eta$-theory has the usual $\beta\eta$ axioms plus structural axioms for braids.

$$\begin{array}{lrcl}
\beta & (\lambda x.M)N & = & M[x := N] \\
\eta & \lambda x.Mx & = & M \\
str_{id} & [id_n]M & = & M \quad (M \text{ has } n \text{ free variables}) \\
str_{comp} & [s]([s']M) & = & [ss']M \\
str_{app} & ([s]M)\,([s']N) & = & [s \otimes s'](MN) \\
str_{abs} & [s](\lambda x.M) & = & \lambda x.[s \otimes id_1]M
\end{array}$$

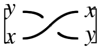
where id_n stands for the trivial braid with n strands (the unit element e of the braid group B_n), ss' is the composition of s and s' while $s \otimes s'$ the parallel composition (Figure 2). It might be worth pointing out that our calculus has some resemblance to the calculi with explicit substitutions [1]: braids can be thought as special substitutions (enriched with some extra information).

In the β rule, the substitution $M[x := N]$ means replacing the (unique) free variable x in M by N and also x-labelled strings occuring in braids in M by Γ-strings where $\Gamma \vdash N$. (When N contains no free variable, all x-strings are removed.) For instance:

$(\lambda y.$ 

$(yx))\,[x := (x_1\,x_2)] \;\equiv\; \lambda y.$ 

$(y\,(x_1\,x_2))$

$(\lambda y.$ 

$(yx))\,[x := \lambda z.z] \;\equiv\; \lambda y.\,[y \text{———} y]\,(y\,(\lambda z.z)) \;=_{\beta\eta}\; \lambda y.(y\,(\lambda z.z))$

Thus substitution is much subtler than one might first guess. Below we discuss the formal definition of substitution, in which braids are algebraically handled as elements of the braid group.

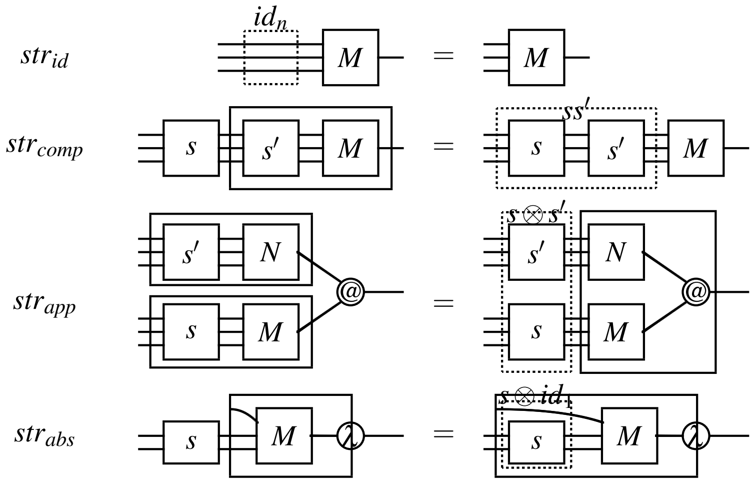


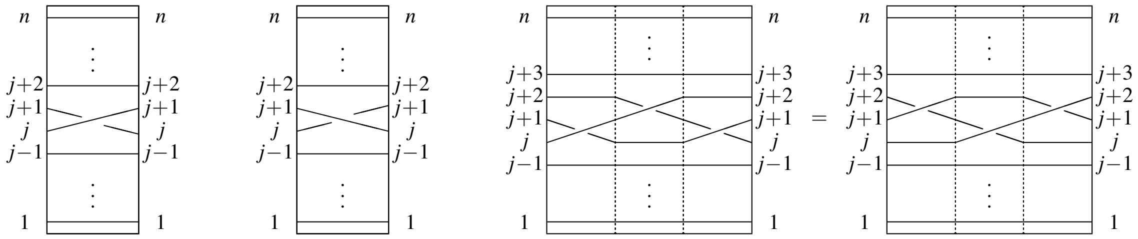
Figure 2: Structural axioms

2.3 Formal treatment of braids and substitution

The braid group Let B_n be the Artin braid group [3, 4, 9] generated by $n-1$ generators $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ with relations

- $\sigma_i \sigma_j = \sigma_j \sigma_i$ for $1 \leq i, j \leq n-1$ with $|i-j| \geq 2$, and
- $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$ for $1 \leq i \leq n-1$.

The following geometric reading in terms of braid diagrams may be useful for understanding the behaviour of the generators σ_i and σ_i^{-1}:

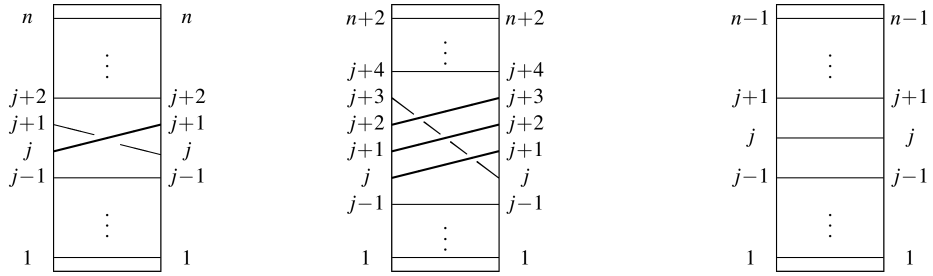


σ_j σ_j^{-1} $\sigma_j \sigma_{j+1} \sigma_j = \sigma_{j+1} \sigma_j \sigma_{j+1}$

In the sequel we will denote the unit element (id_n) of the braid group by e.

Defining substitutions Define the substitution map $(-)[i := m] : B_n \to B_{n+m-1}$ for $1 \leq i \leq n$ and $m \geq 0$ as follows.

- $e[i := m] \equiv e$.
- $(\sigma_j s)[i := m] \equiv \sigma_{j+m-1}(\sigma[i := m])$ when $i \leq j-1$.
- $(\sigma_j s)[i := m] \equiv \sigma_j(\sigma[i := m])$ when $i \geq j+2$.
- $(\sigma_j s)[j := m] \equiv \begin{cases} s[j+1 := 0] & m = 0 \\ \sigma_{j+m-1} \cdots \sigma_{j+1} \sigma_j (s[j+1 := m]) & m \geq 1 \end{cases}$



σ_j $\quad\quad$ $\sigma_j[j := 3] = \sigma_{j+2}\sigma_{j+1}\sigma_j$ $\quad\quad$ $\sigma_j[j := 0] = e(= id_{n-1})$

Figure 3: Substitution map

- $(\sigma_j s)[j+1 := m] \equiv \begin{cases} s[j := 0] & m = 0 \\ \sigma_j\sigma_{j+1}\cdots\sigma_{j+m-1}(s[j := m]) & m \geq 1 \end{cases}$
- Similarly for $\sigma_j^{-1}s$.

The substitution map is well-defined: $s[i := m]$ does not depend on the choice of $g_1,\ldots,g_k \in \{\sigma_1^{\pm},\ldots,\sigma_n^{\pm}\}$ such that $s = g_1\cdots g_k$. Note that $s[i := 1] \equiv s$ holds for any $s \in B_n$ and i. We give some examples of the substitution map in Figure 3.

In the sequel we identify an element of B_n with a braid with n strands. For a braided term $[s]M$ with

$$\frac{x_1, x_2, \ldots, x_n \vdash M \quad s \in B_n}{x_{s(1)}, x_{s(2)}, \ldots, x_{s(n)} \vdash [s]M}$$

and a term $y_1,\ldots,y_m \vdash N$, we define the substitution $([s]M)[x_i := N]$ as

$$([s]M)[x_i := N] \;\equiv\; \left[s[s^{-1}(i) := m]\right](M[x_i := N])$$

2.4 Rewriting and decidability

Let $\equiv_{str}$ be the smallest congruence on braided lambda terms containing the equational theory of braid groups and structural axioms. We say that a term M (1-step) $\beta\eta$-reduces to N modulo $\equiv_{str}$ when there exists M_1 such that $M \equiv_{str} M_1$ and M_1 reduces to N via a single $\beta\eta$-reduction.

Theorem 1 *The $\beta\eta$-reduction modulo $\equiv_{str}$ is strong normalizing, and Church-Rosser modulo $\equiv_{str}$.*

Note that a normal form of β-reduction modulo $\equiv_{str}$ is just a β-normal linear lambda term decorated by braids (while it is not the case for the η-reduction), and a normal form of a braided term can be easily obtained by tracing the normalization of the corresponding linear lambda term (with all braids dropped). Since the word problem for braid groups is decidable [3, 9] and so is the equational theory of structural axioms, we conclude:

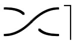
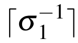
Theorem 2 *The $\beta\eta$-theory is decidable.*

3 Combinatory logic

3.1 Representing braids by $\mathbf{C}^{\pm}$

For a braid s with n strands, let $\lceil s \rceil$ be the combinator

$$\lambda f x_{s(1)} \ldots x_{s(n)}.[id_1 \otimes s](f x_1 \ldots x_n)$$

In particular, when $n = 2$ $\lceil \sigma_1 \rceil = \lceil$  $\rceil = \mathbf{C}^+$ and $\lceil \sigma_1^{-1} \rceil = \lceil$  $\rceil = \mathbf{C}^-$.
As usual, we have the combinators $\mathbf{I} \equiv \lambda x.x$ and $\mathbf{B} \equiv \lambda xyz.x\,(yz)$.

Lemma 1 *1.* $\lceil id_n \rceil \;=_{\beta\eta}\; \mathbf{I}$.

2. $\lceil ss' \rceil \;=_{\beta\eta}\; \mathbf{B}\,\lceil s \rceil\,\lceil s' \rceil$.

3. $\lceil id_1 \otimes s \rceil \;=_{\beta\eta}\; \mathbf{B}\,\lceil s \rceil$.

4. $\lceil s \otimes id_1 \rceil \;=_{\beta\eta}\; \lceil s \rceil$.

Proposition 1 $\lceil \sigma_i \rceil =_{\beta\eta} \mathbf{B}^{i-1}\mathbf{C}^+$ *and* $\lceil \sigma_i^{-1} \rceil =_{\beta\eta} \mathbf{B}^{i-1}\mathbf{C}^-$.

Since any braid is given by composing e, σ_i and σ_i^{-1}, we conclude:

Theorem 3 *For any braid s, $\lceil s \rceil$ is $\beta\eta$-equal to a combinator generated by* **B**, **I**, $\mathbf{C}^+$ *and* $\mathbf{C}^-$.

3.2 Combinatory completeness of BC$^{\pm}$I

For the braided term $x_{s(1)}, x_{s(2)}, \ldots, x_{s(n)} \vdash [s]M$, we have

$$[s]M \quad =_{\beta\eta} \quad \lceil s \rceil\,(\lambda x_1 \ldots x_n.M)\,x_{s(1)} \ldots x_{s(n)}$$

Thus any braided lambda term is equal to a planar lambda term (a term which does not involve the braid rule) enriched with $\mathbf{C}^+$ and $\mathbf{C}^-$. In particular, for combinators we have

Theorem 4 *Any closed term of the braided lambda calculus is $\beta\eta$-equal to a combinator generated by* **B**, **I**, $\mathbf{C}^+$ *and* $\mathbf{C}^-$.

This, in the context of combinatory logic, can be thought as a *combinatory completeness*. Indeed, we have the following translation $(-)^\flat$ from the braided lambda calculus to $\mathbf{BC^{\pm}I}$-terms.

$$\begin{array}{c}
x^\flat \;\equiv\; x \qquad (MN)^\flat \;\equiv\; M^\flat N^\flat \qquad (\lambda x.M)^\flat \;\equiv\; \lambda^* x.M^\flat \\
([s]M)^\flat \;\equiv\; \lceil s \rceil\,(\lambda^* x_1 \ldots x_n.M^\flat)\,x_{s(1)} \ldots x_{s(n)} \quad (\mathrm{cxt}(M) = x_1, \ldots, x_n) \\
\lambda^* x.x \;\equiv\; \mathbf{I} \qquad \lambda^* x.PQ \;\equiv\; \left\{ \begin{array}{ll} \mathbf{C}^+\,(\lambda^* x.P)\,Q & (x \in \mathrm{fv}(P)) \\ \mathbf{B}\,P\,(\lambda^* x.Q) & (x \in \mathrm{fv}(Q)) \end{array} \right.
\end{array}$$

(To be precise, this determines a translation on terms modulo $\beta\eta$-equality, because Lemma 1 and Proposition 1 define $\lceil s \rceil$ only up to $\beta\eta$-equality. For instance, $e = \sigma_1\sigma_1^{-1}$ in B_2 and $\lceil e \rceil = \lambda fxy.fxy$ while $\lceil \sigma_1\sigma_1^{-1} \rceil = \mathbf{B}\mathbf{C}^+\mathbf{C}^-$, and they are $\beta\eta$-equal.)

Therefore it is possible to formulate a *braided combinatory logic* with constants **B**, $\mathbf{C}^{\pm}$, **I** and an appropriate set of axioms (say $\mathscr{A}$) ensuring (i) $M =_{\mathscr{A}} M'$ implies $\lambda^* x.M =_{\mathscr{A}} \lambda^* x.M'$ and (ii) $s = s'$ in B_n implies $\lceil s \rceil =_{\mathscr{A}} \lceil s' \rceil$. Below we present axioms which are sound for the $\beta\eta$-equality; among them, the last two axioms correspond to the Reidemeister moves. Finding a complete (hopefully finite) axiomatization (which should contain or imply these axioms and satisfy (i) and (ii) above) is left as future work.

$$\begin{array}{rclrcl}
\mathbf{I}M & = & M & \mathbf{B}LMN & = & L(MN) \\
\mathbf{B}\,(\mathbf{B}MN) & = & \mathbf{B}\,(\mathbf{B}M)\,(\mathbf{B}N) & \mathbf{B}\mathbf{B}\,(\mathbf{B}M) & = & \mathbf{B}\,(\mathbf{B}\,(\mathbf{B}M))\,\mathbf{B} \\
\mathbf{B}\mathbf{I} & = & \mathbf{I} & \mathbf{B}M\mathbf{I} & = & M
\end{array}$$

$$\begin{array}{rcl}
\mathbf{C}^{\pm}LMN & = & LNM \quad (M \text{ or } N \text{ closed}) \\
\mathbf{C}^{\pm}\,(\mathbf{B}\mathbf{B}M)\,\mathbf{I} & = & M \\
\mathbf{C}^{\alpha}\,(\mathbf{B}\,\mathbf{C}^{\beta}\,(\mathbf{B}\,(\mathbf{B}\mathbf{B})M))\,\mathbf{I} & = & M \quad (\alpha, \beta \in \{+,-\}) \\
\mathbf{B}\,\mathbf{C}^{\pm}\,\mathbf{C}^{\mp} & = & \mathbf{I} \\
\mathbf{B}\,\mathbf{C}^{\pm}\,(\mathbf{B}\,(\mathbf{B}\,\mathbf{C}^{\pm})\,\mathbf{C}^{\pm}) & = & \mathbf{B}\,(\mathbf{B}\,\mathbf{C}^{\pm})\,(\mathbf{B}\,\mathbf{C}^{\pm}\,(\mathbf{B}\,\mathbf{C}^{\pm}))
\end{array}$$

4 Semantics

4.1 Categorical models

A model of the braided lambda calculus (without η) can be given by an object X in a braided monoidal closed category [8] such that the internal hom $[X,X]$ is a retract of X. An extensional model (i.e., validating η) is given by an X such that $[X,X]$ is isomorphic to X.

There are plenty of braided monoidal closed categories in the literature — many of them found in the context of representation theory of quantum groups [13]. However, finding a braided monoidal closed category with a non-trivial reflexive object is not easy — impossible if we stick to finite dimensional linear representations, as the dimension of $[X,X]$ is strictly higher than that of X unless X is one-dimensional. Below we present models using braided relational semantics [7] where the problem of dimensions disappear.

4.2 A crossed G-set model

Fix a group $G = (G, e, \cdot, (-)^{-1})$. Recall that a *crossed G-set* [14] is a set X equipped with a G-action $\bullet : G \times X \to X$ and a valuation map $|\_| : X \to G$ satisfying $|g \bullet x| = g|x|g^{-1}$ for $g \in G$ and $x \in X$. There is a *ribbon category* [12, 13] $\mathbf{XRel}(G)$ whose objects are crossed G-sets and a morphism from $(X, \bullet, |\_|)$ to $(Y, \bullet, |\_|)$ is a binary relation $r \subseteq X \times Y$ between X and Y such that $(x,y) \in r$ implies $|x| = |y|$ as well as $(g \bullet x, g \bullet y) \in r$ for any $g \in G$ [7]. The dual of a crossed G-set $X = (X, \bullet, |\_|)$ is $X^* = (X, \bullet, |\_|^{-1})$. The tensor of $X = (X, \bullet, |\_|)$ and $Y = (Y, \bullet, |\_|)$ is $X \otimes Y = (X \times Y, (g,(x,y)) \mapsto (g \bullet x, g \bullet y), (x,y) \mapsto |x||y|)$. Below we will give a crossed G-set $\mathscr{T}$ such that the internal hom $[\mathscr{T}, \mathscr{T}] = \mathscr{T} \otimes \mathscr{T}^*$ is a retract of $\mathscr{T}$, which forms a model of the braided lambda calculus.

Let $\mathscr{T}$ be the set of binary trees whose leafs are labelled by G's elements (or the implicational formulas generated from G):

$$t \ ::= \ g \mid t \multimapinv t \qquad (g \in G)$$

$\mathscr{T}$ is a crossed G-set with the valuation $|\_| : \mathscr{T} \to G$ given by $|g| = g$ and $|x \multimapinv y| = |x||y|^{-1}$ and the G-action $\bullet : G \times \mathscr{T} \to \mathscr{T}$ given by

$$g \bullet h = ghg^{-1} \ \ (h \in G), \quad g \bullet (x \multimapinv y) = (g \bullet x) \multimapinv (g \bullet y)$$

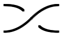
Moreover the map $\varphi : \mathscr{T} \times \mathscr{T} \to \mathscr{T}$ sending (x,y) to $x \multimapinv y$ gives a morphism

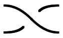
$$\varphi = \{((x,y), x \multimapinv y) \mid x, y \in \mathscr{T}\} : \mathscr{T} \otimes \mathscr{T}^* \to \mathscr{T}$$

in $\mathbf{XRel}(G)$, with a right inverse $\{(x \multimapinv y, (x,y)) \mid x,y \in \mathscr{T}\} : \mathscr{T} \to \mathscr{T} \otimes \mathscr{T}^*$. It follows that we can model the untyped braided lambda calculus (without η) using $\mathscr{T}$ as follows. A term $x_1, \ldots, x_n \vdash M$ is interpreted as a relation r from $\mathscr{T}^n$ to $\mathscr{T}$ such that $((u_1, \ldots, u_n), a) \in r$ implies $|u_1| \cdots |u_n| = |a|$ as well as $((g \bullet u_1, \ldots, g \bullet g_n), g \bullet a) \in r$ for any $g \in G$. In particular, a closed term is interpreted as a subset of $\{x \in \mathscr{T} \mid |x| = e\}$ closed under the G-action.

$$\begin{array}{rcl}
\llbracket x \vdash x \rrbracket & = & \{(a,a) \mid a \in \mathscr{T}\} \\
\llbracket \Gamma \vdash \lambda x.M \rrbracket & = & \{(\vec{u}, b \multimapinv a) \mid ((\vec{u}, a), b) \in \llbracket \Gamma, x \vdash M \rrbracket\} \\
\llbracket \Gamma, \Delta \vdash MN \rrbracket & = & \{((\vec{u}, \vec{v}), b) \mid \exists a \ (\vec{u}, b \multimapinv a) \in \llbracket \Gamma \vdash M \rrbracket \ \& \ (\vec{v}, a) \in \llbracket \Delta \vdash N \rrbracket\} \\
\llbracket \Gamma' \vdash [s]M \rrbracket & = & \llbracket s \rrbracket ; \llbracket \Gamma \vdash M \rrbracket
\end{array}$$

where the interpretation $[\![\sigma]\!]$ of a braid σ is built from

$[\![$  $]\!] = \{((a,b),(|a|\bullet b,a)) \mid a,b\in\mathscr{T}\}$

$[\![$  $]\!] = \{((a,b),(b,|b|^{-1}\bullet a)) \mid a,b\in\mathscr{T}\}$

For instance, the braided **C** combinators are interpreted as

$$\begin{aligned}[\![\mathbf{C}^{+}]\!] &= \{((z \multimapinv |x|^{-1}\bullet y)\multimapinv x)\multimapinv((z\multimapinv x)\multimapinv y) \mid x,y,z\in\mathscr{T}\}\\ [\![\mathbf{C}^{-}]\!] &= \{((z \multimapinv y)\multimapinv |y|\bullet x)\multimapinv((z\multimapinv x)\multimapinv y) \mid x,y,z\in\mathscr{T}\}\end{aligned}$$

4.3 An extensional crossed G-set model

Now we expand $\mathscr{T}$ to a crossed G-set of infinite binary trees. Let

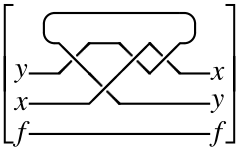
$$\mathscr{D} = \{f : \{0,1\}^{*} \to G \mid f(w) = f(w0)\cdot f(w1)^{-1}\}$$

$\mathscr{D}$ is a crossed G-set with $|f| = f(\varepsilon)$ and $(g\bullet f)(w) = g\cdot f(w)\cdot g^{-1}$. Its dual $\mathscr{D}^{*}$ is identical to $\mathscr{D}$ except the valuation $|f| = f(\varepsilon)^{-1}$. There is an isomorphism $\varphi : \mathscr{D}\otimes\mathscr{D}^{*} \xrightarrow{\cong} \mathscr{D}$ induced by the bijective map $\varphi : \mathscr{D}^{2}\to\mathscr{D}$ given by

$$\begin{cases}\varphi(f_0,f_1)(\varepsilon) &= f_0(\varepsilon)f_1(\varepsilon)^{-1}\\ \varphi(f_0,f_1)(0w) &= f_0(w)\\ \varphi(f_0,f_1)(1w) &= f_1(w)\end{cases}$$

Note that $\varphi^{-1}(f) = (\lambda w.f(0w),\lambda w.f(1w))$ holds. Also $\mathscr{D}\cong\mathscr{D}^{*}$ with $f\mapsto f^{*} = \varphi(\lambda w.f(1w),\lambda w.f(0w))$ (thus $f^{*}(\varepsilon) = f(\varepsilon)^{-1}$, $f^{*}(0w) = f(1w)$ and $f^{*}(1w) = f(0w)$). $\mathscr{D}$ is a model of the braided lambda calculus validating the η equality. The interpretation of terms is essentially the same as the case of $\mathscr{T}$, with $x\multimapinv y$ replaced by $\varphi(x,y)$.

Remark 2 (a two-objects ribbon category, and the tangled lambda calculus) *Since $\mathscr{D}\cong\mathscr{D}^{*}\cong\mathscr{D}\otimes\mathscr{D}$, the full subcategory of* **XRel**(G) *with just $\mathscr{D}$ and the tensor unit I is a ribbon category. This also means that, with $\mathscr{D}$, we can interpret not just braids but also framed tangles (ribbons). Thus $\mathscr{D}$ is a model of "tangled lambda calculus" in which we should be able to express a term involving tangles like*

$\lambda fxy.\ \Big[$  $\Big]\ (f\,y\,x)$

Such a tangled lambda calculus is yet to be studied; defining a substitution already seems to be much harder than the braided case. Also it might be more appropriate to use traced monoidal closed categories [6] as semantic models rather than ribbon categories.

5 Conclusion

We introduced the syntax and semantics of an untyped braided lambda calculus. This work is part of our on-going research on relating low-level codes and low-dimensional topology via categorical machineries. Future work will include the typed variants, extension to the tangled lambda calculus, and applications to novel computational models making use of braids, most notably topological quantum computation.

Acknowledgements I thank Haruka Tomita for stimulating discussions related to this work. This work was supported by JSPS KAKENHI Grant Numbers JP18K11165 and JST ERATO Grant Number JPMJER1603, Japan.

References

[1] M. Abadi, L. Cardelli, P.-L. Curien & J.-J. Lévy (1991): *Explicit substitutions*. J. Funct. Programming 1(4), pp. 375–416, doi:10.1017/S0956796800000186.

[2] S. Abramsky (2007): *Temperley-Lieb algebra: from knot theory to logic and computation via quantum mechanics*. In L. Kauffman & S.J. Lomonaco, editors: Mathematics of Quantum Computing and Technology, Taylor&Francis, pp. 415–458, doi:10.1201/9781584889007.

[3] E. Artin (1925): *Theorie der Zöpfe*. Abh. Math. Sem. Univ. Hamburg 4, pp. 47–72, doi:10.1007/BF02950718.

[4] E. Artin (1947): *Theory of braids*. Ann. of Math. 48, pp. 101–126, doi:10.2307/1969218.

[5] A. Fleury (2003): *Ribbon braided multiplicative linear logic*. Mat. Contemp. 24, pp. 39–70.

[6] M. Hasegawa (2009): *On traced monoidal closed categories*. Mathematical Structures in Computer Science 19(2), pp. 217–244, doi:10.1017/S0960129508007184.

[7] M. Hasegawa (2012): *A quantum double construction in Rel*. Mathematical Structures in Computer Science 22(4), pp. 618–650, doi:10.1017/S0960129511000703.

[8] A. Joyal & R.H. Street (1993): *Braided tensor categories*. Adv. Math. 102(1), pp. 20–78, doi:10.1006/aima.1993.1055.

[9] C. Kassel & V.G. Turaev (2008): *Braid Groups*. Graduate Texts in Mathemtics 247, Springer-Verlag, doi:10.1007/978-0-387-68548-9.

[10] A. Kitaev (2003): *Fault-tolerant quantum computation by anyons*. Annals of Physics 303, pp. 3–20, doi:10.1016/S0003-4916(02)00018-0.

[11] P.-A. Melliès (2018): *Ribbon tensorial logic*. In: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS2018), ACM, pp. 689–698, doi:10.1145/3209108.3209129.

[12] M.C. Shum (1994): *Tortile tensor categories*. J. Pure Appl. Algebra 93(1), pp. 57–110, doi:10.1016/0022-4049(92)00039-T.

[13] V.G. Turaev (1994): *Quantum Invariants of Knots and 3-Manifolds*. Studies in Mathematics 18, De Gruyter, doi:10.1515/9783110435221.

[14] J.H.C. Whitehead (1949): *Combinatorial homotopy, II*. Bulletin of the American Mathematical Society 55, pp. 453–496, doi:10.1090/S0002-9904-1949-09213-3.

[15] N. Zeilberger & A. Giorgetti (2015): *A correspondence between rooted planar maps and normal plain lambda terms*. Logical Methods in Computer Science 11(3), pp. 1–39, doi:10.2168/LMCS-11(3:22)2015.