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

Braids 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

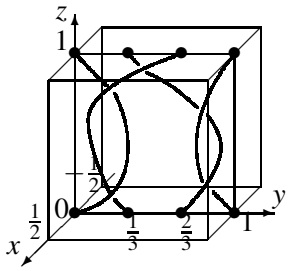
- each $t\in[0,1]$ is mapped to a point in the plane $\{(x,y,z)\mid z=t\}$
- the end points $0\in[0,1]$ are sent to the n points $\{(0,\frac{k}{n-1},0)\mid k=0,\ldots,n-1\}$
- the end points $1\in[0,1]$ are sent to the n points $\{(0,\frac{k}{n-1},1)\mid k=0,\ldots,n-1\}$

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:

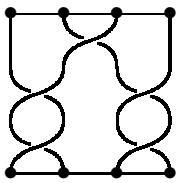
$$\begin{array}{ll} & \{\text{braids of } n\text{-strands}\}/\text{ambient isotopy}\\ \cong & \{\text{braid diagrams of } n\text{-strands}\}/\text{Reidemeister moves}\\ \cong & \text{Braid group } B_n\end{array}$$

A braided lambda calculus 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:

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



braid with 4 strands



braid diagram

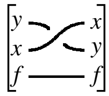
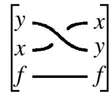
$\sigma_1\sigma_3^2\sigma_1\sigma_2$

element of the braid group B_4

Figure 1: Braids

U. Dal Lago and V. de Paiva (Eds.) Second Joint International Workshop on Linearity & Trends in Linear Logic and Applications.
EPTCS 353, 2021, pp. 94–108, doi:10.4204/EPTCS.353.5

For instance, we have two 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 $\lambda fxy.f\,yx$ using braids. The idea of realizing a braided calculus as a planar calculus enriched with explicit braids is not new, see for instance [5].

Why braids? 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 immediate 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. In a larger perspective, this work forms part of our research project on relating low-level codes and low-dimensional topology via categorical machineries.

From a more abstract point of view, our braided lambda calculus and the corresponding braided combinatory algebras are algebraic structures which can be described in terms of *PROBs* (*pro*ducts and *b*raids categories), the braided version of PROPs (products and permutations categories). It seems that the theory of PROBs is a sort of folklore and there are very few published works on it (cf. [12, 16]); we expect that the braided lambda calculus serves as a good test case of PROBs, e.g. the treatment of substitutions in braided algebras.

Why linear? Our calculus is linear, as there is no non-trivial braid in a non-linear setting. When the tensor product is cartesian, any braid $\sigma_{A,B} : A \times B \to B \times A$ is equal to the symmetry $\langle \pi'_{A,B}, \pi_{A,B} \rangle : A \times B \to B \times A$ because

$$\begin{array}{rcll} \sigma_{A,B} & = & \langle \pi_{B,A}, \pi'_{B,A} \rangle \circ \sigma_{A,B} & \langle \pi, \pi' \rangle = id \\ & = & \langle \pi_{B,A} \circ \sigma_{A,B}, \pi'_{B,A} \circ \sigma_{A,B} \rangle & \\ & = & \langle \pi'_{A,B}, \pi_{A,B} \rangle & \end{array}$$

where $\pi_{B,A} \circ \sigma_{A,B} = \pi'_{A,B}$ because

$$\begin{array}{rcll} \pi_{B,A} \circ \sigma_{A,B} & = & (id_B \times !_A) \circ \sigma_{A,B} & \\ & = & \sigma_{1,B} \circ (!_A \times id_B) & \text{naturality of } \sigma \\ & = & id_B \circ \pi'_{A,B} & \sigma_{1,B} = id_B \\ & = & \pi'_{A,B} & \end{array}$$

Similarly $\pi'_{B,A} \circ \sigma_{A,B} = \pi_{A,B}$ holds. Hence linearity is essential for studying a braided calculus in a meaningful way.

Why untyped? 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

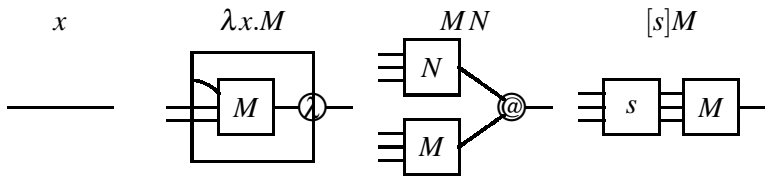


Figure 2: Term graphs

calculus requires a reflexive object, which is hard to find in the well-known braided categories in TQFT [15]. We overcome this difficulty by using a braided relational model constructed in our previous work [7]. As far as we know, this is the first non-trivial example of a reflexive object in a non-symmetric ribbon category.

Contributions 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 which ensures that our combinatory logic is as expressive as the braided lambda calculus (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.

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

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

where $s(i)$ denotes the outcome of applying the permutation on $\{1, 2, \ldots, n\}$ 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.

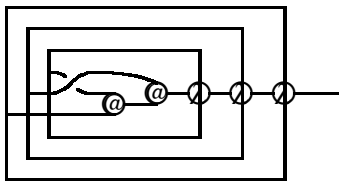
It will be helpful to look at *term graphs* corresponding to terms (Figure 2), especially when discussing the equational theory of the braided lambda calculus.

[1] In the literature, there are (at least) two different notions of "planar lambda terms". Some authors employ the "left" abstraction rule (e.g. [18]) $\dfrac{x, \Gamma \vdash M}{\Gamma \vdash \lambda x.M}$ whereas others (e.g. [2, 14]) use the "right" abstraction rule as we do in the present paper; see [18] 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.

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{f\vdash f}\quad\overline{y\vdash y}}{f,y\vdash fy}\quad\overline{x\vdash x}}{f,y,x\vdash fyx}\quad s=\;\begin{smallmatrix}y&x\\x&y\end{smallmatrix}\!\!\times\!\!\begin{smallmatrix}f&f\end{smallmatrix}}{f,x,y\vdash[s](fyx)}}{f,x\vdash\lambda y.[s](fyx)}}{f\vdash\lambda xy.[s](fyx)}}{\vdash\lambda fxy.[s](fyx)}$$

and the term graph corresponding to $\mathbf{C}^+$ *is*



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* $s(x_1,\ldots,x_n)=x_{s(1)},\ldots,x_{s(n)}$. *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 of the braided lambda calculus has the usual $\beta\eta$ axioms plus structural axioms for braids:

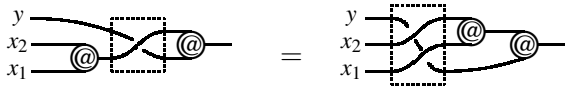
$$\begin{array}{lrcl}
\beta & (\lambda x.M)N & = & M[x:=N]\\
\eta & \lambda x.Mx & = & M\\
str_{id} & [id_n]M & = & M \quad (M \text{ has } n \text{ free variables})\\
str_{comp} & [s]([s']M) & = & [ss']M\\
str_{app} & ([s]M)([s']N) & = & [s\otimes s'](MN)\\
str_{abs} & [s](\lambda x.M) & = & \lambda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

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 3). The structural axioms identify two terms when they have the same underlying term graph (Figure 5). 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.) This informal definition can be justified if we look at the corresponding term graphs: intuitively,

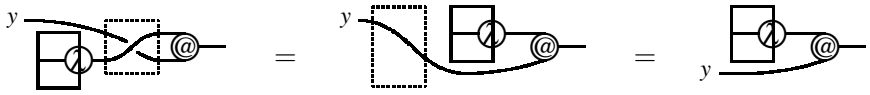
$$\left(\begin{bmatrix}y\\x\end{bmatrix}\!\!\times\!\!\begin{bmatrix}x\\y\end{bmatrix}(yx)\right)[x:=(x_1\,x_2)] \quad \text{should be} \quad \begin{bmatrix}y\\x_2\\x_1\end{bmatrix}\!\!\times\!\!\begin{bmatrix}x_2\\x_1\\y\end{bmatrix}(y(x_1\,x_2))$$

because they express the same term graph (modulo continuous deformation):



Similarly,

$$\left(\begin{bmatrix} y \\ x \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (yx)\right)[x := \lambda z.z] \quad \text{should be} \quad [y \text{ —— } y]\,(y(\lambda z.z)) =_{str_{id}} y(\lambda z.z).$$



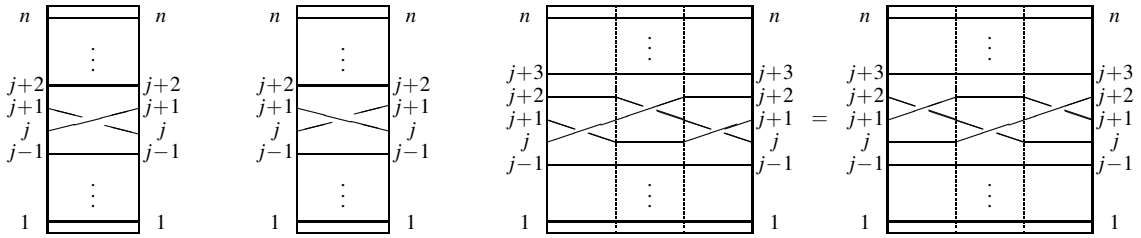
Thus substitution is much subtler than one might first guess. Below we discuss the formal definition of substitution, in which braids are algebraically handled as elements of the braid group.

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}:



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

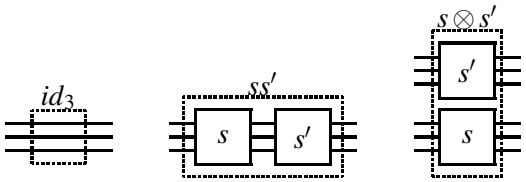


Figure 3: Operations on braids

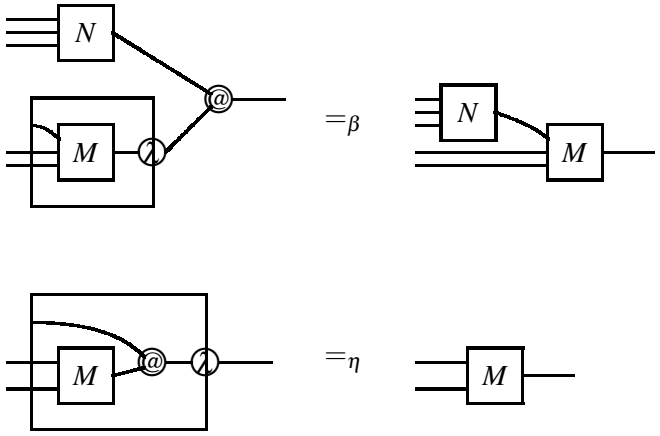


Figure 4: $\beta\eta$ axioms

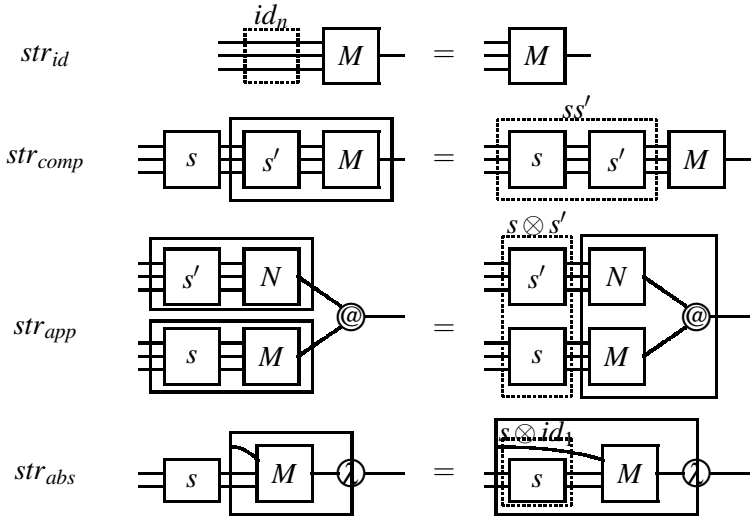
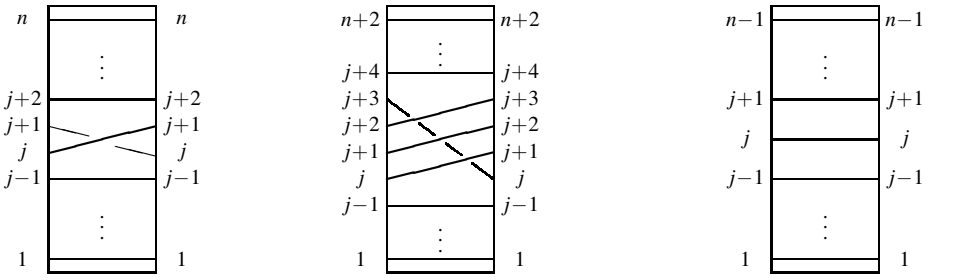


Figure 5: Structural axioms



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 6: Substitution map

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.

- $e[i := m] \equiv e$.
- $(\sigma_j s)[i := m] \equiv \sigma_{j+m-1}(s[i := m])$ when $i \leq j-1$.
- $(\sigma_j s)[i := m] \equiv \sigma_j(s[i := m])$ when $i \geq j+2$.
- $(\sigma_j s)[j := m] \equiv \begin{cases} s[j+1 := 0] & m = 0 \\ \sigma_{j+m-1} \cdots \sigma_{j+1}\sigma_j(s[j+1 := m]) & m \geq 1 \end{cases}$
- $(\sigma_j s)[j+1 := m] \equiv \begin{cases} s[j := 0] & m = 0 \\ \sigma_j\sigma_{j+1} \cdots \sigma_{j+m-1}(s[j := m]) & m \geq 1 \end{cases}$
- 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 6.

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

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

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 $\equiv_{str}$.*

Proof (outline) For a braided lambda term M, let $u(M)$ be the linear lambda term obtained by deleting all braids in M; thus $u(x) = x$, $u(\lambda x.M) = \lambda x.u(M)$, $u(MN) = u(M)\,u(N)$ and $u([s]M) = u(M)$. Then we have that

- If M 1-step $\beta\eta$-reduces to N, then $u(M)$ 1-step $\beta\eta$-reduces to $u(N)$ in the linear lambda calculus.
- If $M \equiv_{str} M_1$, then $u(M) \equiv u(M_1)$.

The strong normalization follows immediately from these observations as an infinite $\beta\eta$-reduction sequence modulo $\equiv_{str}$ would give rise to an infnite $\beta\eta$-reduction sequence in the linear lambda calculus (which of course is strongly normalizing).

The confluence of β-reduction modulo $\equiv_{str}$ follows from the observation that when $u(M)$ β-reduces to N in the linear lambda calculus, there exists a braided term N_0 such that M β-reduces to N_0 modulo $\equiv_{str}$ and $u(N_0) = N$ holds. This does not hold for the η-reduction: for $M = \lambda y.\, [{}^{y}_{x}\!\times\!{}^{y}_{x}]\,(xy)$, observe that $u(M) = \lambda y.xy$ η-reduces to x while M is η-normal in the braided calculus. Fortunately the η-postponement holds in this setting and we obtain the confluence of $\beta\eta$-reduction 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, and a normal form of a braided term can be easily obtained by tracing the normalization of the corresponding linear lambda term. 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 of the braided lambda calculus (as given in Section 2.2) 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 $n = 2$ $\lceil \sigma_1 \rceil = \lceil \times \rceil = \mathbf{C}^+$ and $\lceil \sigma_1^{-1} \rceil = \lceil \times \rceil = \mathbf{C}^-$.
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. $\lceil s \otimes id_1 \rceil =_{\beta\eta} \lceil s \rceil$.

Below let us write $M^{n+1}N$ for$M\,(M^n N)$ and $M^0 N$ for N.

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* $\mathbf{B}$, $\mathbf{I}$, $\mathbf{C}^+$ *and* $\mathbf{C}^-$.

3.2 Combinatory Completeness of $\mathbf{BC}^{\pm}\mathbf{I}$

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

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

because

$$\begin{aligned}
\lceil s \rceil\,(\lambda x_1 \ldots x_n.M)\,x_{s(1)} \ldots x_{s(n)} &= (\lambda f x_{s(1)} \ldots x_{s(n)}.[id_1 \otimes s](f\,x_1\,\ldots\,x_n))\,(\lambda x_1 \ldots x_n.M)\,x_{s(1)} \ldots x_{s(n)} \\
&= (\lambda x_{s(1)} \ldots x_{s(n)}.[s]((\lambda x_1 \ldots x_n.M)\,x_1\,\ldots\,x_n))\,x_{s(1)} \ldots x_{s(n)} \\
&= [s]((\lambda x_1 \ldots x_n.M)\,x_1\,\ldots\,x_n) \\
&= [s]M
\end{aligned}$$

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* $\mathbf{B}$, $\mathbf{I}$, $\mathbf{C}^+$ *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 $\mathbf{BC}^{\pm}\mathbf{I}$-terms.

$$x^\flat \equiv x \qquad (MN)^\flat \equiv M^\flat N^\flat \qquad (\lambda x.M)^\flat \equiv \lambda^* x.M^\flat$$

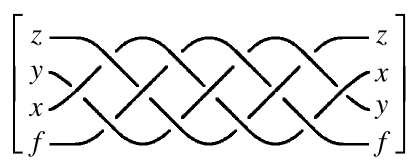
$$([s]M)^\flat \equiv \lceil s \rceil\,(\lambda^* x_1 \ldots x_n.M^\flat)\,x_{s(1)} \ldots x_{s(n)} \quad (\mathrm{cxt}(M) = x_1, \ldots, x_n)$$

$$\lambda^* x.x \equiv \mathbf{I} \qquad \lambda^* x.PQ \equiv \begin{cases} \mathbf{C}^+\,(\lambda^* x.P)\,Q & (x \in \mathrm{fv}(P)) \\ \mathbf{B}\,P\,(\lambda^* x.Q) & (x \in \mathrm{fv}(Q)) \end{cases}$$

(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 $\lceil \sigma_1\sigma_1^{-1} \rceil = \mathbf{BC}^+\mathbf{C}^-$, and they are $\beta\eta$-equal.)

Example 2 *As an example involving a fairly complex braid, let us consider a Celtic C-combinator (inspired by the traditional Celtic braid):*

$$\lambda fxyz.\,\left[(\sigma_2\sigma_1^{-1}\sigma_3^{-1})^4\sigma_2\right](f\,y\,x\,z) \quad = \quad \lambda fxyz.$$



$$(f\,y\,x\,z)$$

Thanks to the combinatory completeness and the translation above, we have that this combinator is $\beta\eta$-equal to

$$\begin{aligned}
&\mathbf{B}\,\mathbf{C}^+\,(\mathbf{C}^-\,(\mathbf{B}\,(\mathbf{B}\,\mathbf{C}^-)) \\
&\quad (\mathbf{B}\,\mathbf{C}^+\,(\mathbf{C}^-\,(\mathbf{B}\,(\mathbf{B}\,\mathbf{C}^-)) \\
&\quad\quad (\mathbf{B}\,\mathbf{C}^+\,(\mathbf{C}^-\,(\mathbf{B}\,(\mathbf{B}\,\mathbf{C}^-)) \\
&\quad\quad\quad (\mathbf{B}\,\mathbf{C}^+\,(\mathbf{C}^-\,(\mathbf{B}\,(\mathbf{B}\,\mathbf{C}^-))\,\mathbf{C}^+)))))))
\end{aligned}$$

built from $\mathbf{B}$, $\mathbf{I}$, $\mathbf{C}^+$ *and* $\mathbf{C}^-$.

Therefore it is possible to formulate a *braided combinatory logic* with constants $\mathbf{B}$, $\mathbf{C}^{\pm}$, $\mathbf{I}$ 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 $\lceil s \rceil =_{\mathscr{A}} \lceil s' \rceil$. Finding a complete (hopefully finite) axiomatization (which should satisfy (i) and (ii) above) is left as future work.

For comparison, in Figure 7 we give an axiomatization of the linear combinatory logic **BCI** which is sound and complete for the $\beta\eta$-theory of the linear lambda calculus.[2] We expect that a complete

[2]This axiomatization is our own version (and might contain some redundancies); we were unable to find such a complete axiomatization of **BCI** in the literature, though we think that an axiomatization like ours should be known to specialists. For reference, we include an outline of the proof of completeness in Appendix A.

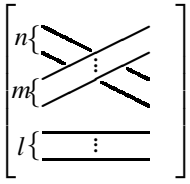
$$
\begin{array}{rcll}
\mathbf{B}LMN & = & L(MN) & \text{(B)} \\
\mathbf{C}LMN & = & LNM & \text{(C)} \\
\mathbf{I}M & = & M & \text{(I)} \\
\mathbf{BI} & = & \mathbf{I} & \\
\mathbf{CBI} & = & \mathbf{I} & \\
\mathbf{B}(\mathbf{BB})\mathbf{B} & = & \mathbf{B}(\mathbf{CBB})(\mathbf{BBB}) & \\
\mathbf{B}(\mathbf{BC})(\mathbf{BBB}) & = & \mathbf{B}(\mathbf{CBC})(\mathbf{BBB}) & \\
\mathbf{B}(\mathbf{BB})\mathbf{C} & = & \mathbf{BC}(\mathbf{B}(\mathbf{BC})\mathbf{B}) & \\
\mathbf{BCC} & = & \mathbf{I} & \textit{Reidemeister II} \\
\mathbf{B}(\mathbf{BC})(\mathbf{BC}(\mathbf{BC})) & = & \mathbf{BC}(\mathbf{B}(\mathbf{BC})\mathbf{C}) & \textit{Reidemeister III}
\end{array}
$$

Figure 7: A complete axiomatization of **BCI**

axiomatization of $\mathbf{BC}^{\pm}\mathbf{I}$ can be given like this axiomatization of **BCI**, with some needed modifications. For instance, *Reidemeister II* and *Reidemeister III* should be replaced by the braided versions

$$
\begin{array}{rcll}
\mathbf{B}\mathbf{C}^{\pm}\mathbf{C}^{\mp} & = & \mathbf{I} & \textit{Reidemeister II} \\
\mathbf{B}(\mathbf{BC}^{+})(\mathbf{BC}^{+}(\mathbf{BC}^{+})) & = & \mathbf{BC}^{+}(\mathbf{B}(\mathbf{BC}^{+})\mathbf{C}^{+}) & \textit{Reidemeister III}
\end{array}
$$

which amount to $\lceil \sigma_1 \sigma_1^{-1} \rceil = \lceil \sigma_1^{-1} \sigma_1 \rceil = \mathbf{I}$ and $\lceil \sigma_1 \sigma_2 \sigma_1 \rceil = \lceil \sigma_2 \sigma_1 \sigma_2 \rceil$. It seems much more difficult to find a braided variant of the axiom (C): when the terms L, M, N have l, m, n free variables respectively, we have

$\mathbf{C}^{+}LMN =$  (LNM)

where the braid in the right hand side of the equation is not an identity unless m or n is zero, and the corresponding $\mathbf{BC}^{\pm}\mathbf{I}$-term contains mn $\mathbf{C}^{+}$s. This suggests that finding a finite axiomatization is not an obvious task.

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. The situation is largely the same as that of the models of the untyped lambda calculus given by reflexive objects in cartesian closed categories. Let us sketch how it works. Let $ev : [X,X] \otimes X \to X$ be the evaluation map given by the monoidal closure, and write $\Lambda(f) : \Gamma \to [X,X]$ for the currying of an arrow $f : \Gamma \otimes X \to X$. Assume an arrow $\varphi : [X,X] \to X$ with a right inverse $\psi : X \to [X,X]$. Then we can interpret a braided lambda term

$x_1,\ldots,x_n \vdash M$ as an arrow $[\![x_1,\ldots,x_n \vdash M]\!]$ from $X^{\otimes n} = \overbrace{X \otimes \cdots \otimes X}^{n}$ to X as follows.

$$\begin{array}{rcl} [\![x \vdash x]\!] & = & id_X \\ [\![\Gamma \vdash \lambda x.M]\!] & = & \Lambda([\![\Gamma, x \vdash M]\!]);\varphi \\ [\![\Gamma,\Delta \vdash MN]\!] & = & ([\![\Gamma \vdash M]\!];\psi \otimes [\![\Delta \vdash N]\!]);ev \\ [\![x_{s(1)},\ldots,x_{s(n)} \vdash [s]M]\!] & = & [\![s]\!];[\![x_1,\ldots,x_n \vdash M]\!] \end{array}$$

where ; denotes the relational composition, and the interpretation $[\![s]\!]$ is the interpretation of the braid s on $X^{\otimes n}$. The β-equality is validated because $\varphi;\psi = id_{[X,X]}$ hold. An extensional model (i.e., validating η) is given by an X such that $[X,X]$ is isomorphic to X, i.e., $\psi = \varphi^{-1}$.

There are plenty of braided monoidal closed categories in the literature — many of them are found in the context of representation theory of quantum groups [15]. 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 disappears.

4.2 A Crossed G-Set Model of Finite Binary Trees

Fix a group $G = (G, e, \cdot, (-)^{-1})$. Recall that a *crossed G-set* [17] 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* [13, 15] $\mathbf{XRel}(G)$ 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 $X \otimes Y = (X \times Y, (g,(x,y)) \mapsto (g \bullet x, g \bullet y), (x,y) \mapsto |x||y|)$. For this monoidal structure we have a braiding $\sigma_{X,Y} : X \otimes Y \stackrel{\cong}{\to} Y \otimes X$ as

$$\sigma_{X,Y} = \{((x,y),(|x| \bullet y, x)) \mid x \in X, y \in Y\}.$$

See [7] for further details of $\mathbf{XRel}(G)$.

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 leaves are labelled by elements of G (or the implicational formulas generated from G):

$$t ::= g \mid t \circ\!\!- t \quad (g \in G)$$

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

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

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

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

in $\mathbf{XRel}(G)$, with a right inverse $\psi = \{(x \circ\!\!- y, (x,y)) \mid x,y \in \mathscr{T}\} : \mathscr{T} \to \mathscr{T} \otimes \mathscr{T}^*$. 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 $((g \bullet u_1, \ldots, g \bullet u_n), g \bullet a) \in r$ 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.

$$\begin{array}{rcl} [\![x \vdash x]\!] & = & \{(a,a) \mid a \in \mathscr{T}\} \\ [\![\Gamma \vdash \lambda x.M]\!] & = & \{(\vec{u}, b \circ\!\!- a) \mid ((\vec{u},a),b) \in [\![\Gamma, x \vdash M]\!]\} \\ [\![\Gamma,\Delta \vdash MN]\!] & = & \{((\vec{u},\vec{v}),b) \mid \exists a\, (\vec{u}, b \circ\!\!- a) \in [\![\Gamma \vdash M]\!] \;\&\; (\vec{v},a) \in [\![\Delta \vdash N]\!]\} \\ [\![x_{s(1)}, \ldots, x_{s(n)} \vdash [s]M]\!] & = & [\![s]\!]; [\![x_1, \ldots, x_n \vdash M]\!] \end{array}$$

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

$$\begin{array}{rcl} [\![\text{  }]\!] & = & \{((a,b),(|a| \bullet b, a)) \mid a,b \in \mathscr{T}\} \\ [\![\text{  }]\!] & = & \{((a,b),(b, |b|^{-1} \bullet a)) \mid a,b \in \mathscr{T}\} \end{array}$$

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

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

This model does not validate the η-equality:

$$[\![x \vdash \lambda y.xy]\!] = \{(b \circ\!\!- a, b \circ\!\!- a) \mid a,b \in \mathscr{T}\} \neq [\![x \vdash x]\!].$$

This is because φ is not an isomorphism; the right inverse ψ cannot map leaves of $\mathscr{T}$ to elements of $\mathscr{T} \otimes \mathscr{T}^*$. (It might be tempting to remedy this by taking a quotient of $\mathscr{T}$ by identifying $x \circ\!\!- e$ with x, as suggested by an anonymous reviewer. This certainly makes $\psi;\varphi = id_X$ and the η-equality becomes valid. Unfortunately, on this quotient, $\varphi;\psi$ is no longer the identity, and the β-equality becomes invalid.)

4.3 An Extensional Crossed G-Set Model of Infinite Binary Trees

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 $\varphi : \mathscr{D} \otimes \mathscr{D}^* \xrightarrow{\simeq} \mathscr{D}$ induced by the bijective map $\varphi : \mathscr{D}^2 \to \mathscr{D}$ given by (see Figure 8)

$$\left\{ \begin{array}{lcl} \varphi(f_0,f_1)(\varepsilon) & = & f_0(\varepsilon) f_1(\varepsilon)^{-1} \\ \varphi(f_0,f_1)(0w) & = & f_0(w) \\ \varphi(f_0,f_1)(1w) & = & f_1(w) \end{array} \right.$$

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 $x \circ\!\!- y$ 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*

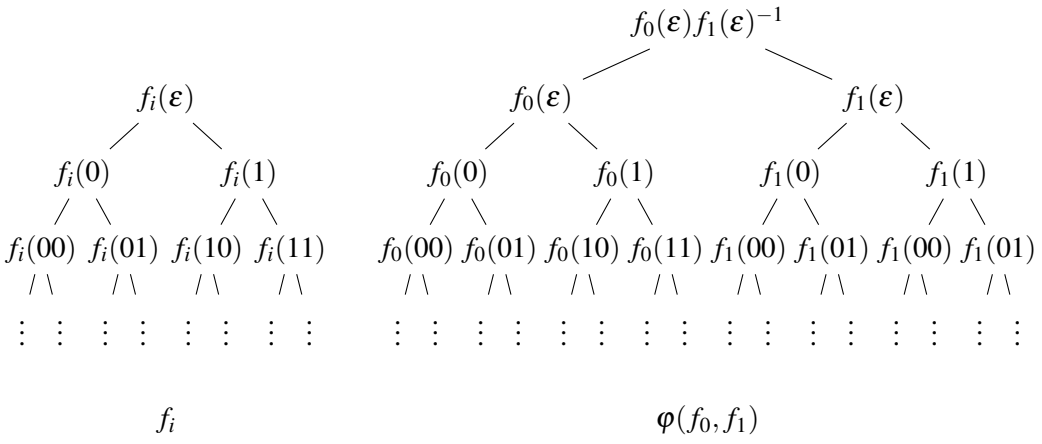
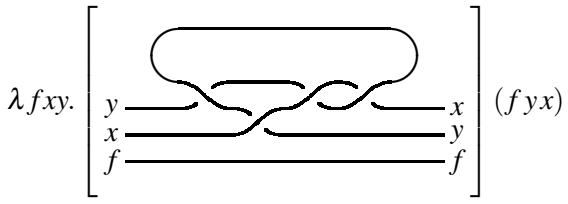


Figure 8: $\varphi(f_0, f_1)$

model of a "tangled lambda calculus" in which we should be able to express a term involving tangles like



Such a tangled lambda calculus is yet to be studied; defining 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. Future work will include the typed variants, complete axiomatization of the braided combinatory logic, 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, and the anonymous reviewers for their helpful comments. This work was supported by JSPS KAKENHI Grant Numbers JP18K11165, JP21K11753 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] *PRO in nLab*. https://ncatlab.org/nlab/show/PRO.

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

[14] H. Tomita (2021): *Realizability without symmetry*. In: *Proceedings of the 29th EACSL Annual Conference on Computer Science Logic (CSL2021)*, *LIPIcs* 183, pp. 38:1–38:16, doi:10.4230/LIPIcs.CSL.2021.38.

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

[16] D. Verdon (2017): *Coherence for braided and symmetric pseudomonoids*. Available at https://arxiv.org/abs/1705.09354.

[17] 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.

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

A Axiomatizing BCI

Let λ_{lin} be the set of linear lambda terms and **BCI** be the set of terms generated by variables (each occurring just once), **B**, **C**, **I** and application. Let $=_{\mathbf{BCI}}$ be the smallest congruence on **BCI** satisfying the axioms in Figure 7. Define translations $(-)^\sharp : \mathbf{BCI} \to \lambda_{lin}$ and $(-)^\flat : \lambda_{lin} \to \mathbf{BCI}$ by

$$\mathbf{B}^\sharp \equiv \lambda xyz.x(yz) \quad \mathbf{C}^\sharp \equiv \lambda xyz.xzy \quad \mathbf{I}^\sharp \equiv \lambda x.x$$
$$(PQ)^\sharp \equiv P^\sharp Q^\sharp \quad x^\sharp \equiv x$$

$$(\lambda x.M)^\flat \equiv \lambda^* x.M^\flat \quad (MN)^\flat \equiv M^\flat N^\flat \quad x^\flat \equiv x$$
$$\lambda^* x.x \equiv \mathbf{I} \quad \lambda^* x.MN \equiv \begin{cases} \mathbf{C}(\lambda^* x.M)N & (x \in \mathrm{fv}(M)) \\ \mathbf{B}M(\lambda^* x.N) & (x \in \mathrm{fv}(N)) \end{cases}$$

We show that these translations give isomorphisms between the equational theories. It is routine to see:

Lemma 2 *$P =_{\mathbf{BCI}} Q$ implies $P^\sharp =_{\beta\eta} Q^\sharp$.*

The following lemma is crucial and the most difficult:

Lemma 3 *$P =_{\mathbf{BCI}} Q$ implies $\lambda^* x.P =_{\mathbf{BCI}} \lambda^* x.Q$.*

Proof For each axiom $P = Q$ with free x we show $\lambda^* x.P = \lambda^* x.Q$. The relevant cases are (B), (C) and (I). For the case of (I), we are to show $\lambda^* x.\mathbf{I}M = \lambda^* x.M$ with free x in M, which follows from

$$\lambda^* x.\mathbf{I}M \equiv \mathbf{B}\mathbf{I}(\lambda^* x.M) = \mathbf{I}(\lambda^* x.M) = \lambda^* x.M$$

The case of (B) contains three sub-cases depending on where the free x occurs. For instance, showing $\lambda^* x.\mathbf{B}LMN = \lambda^* x.L(MN)$ with free x in N amounts to showing $\mathbf{B}L(\mathbf{B}M(\lambda^* x.N)) = \mathbf{B}(\mathbf{B}LM)(\lambda^* x.N)$ for which it suffices to show the associativity $\mathbf{B}L(\mathbf{B}MN) = \mathbf{B}(\mathbf{B}LM)N$.

$$\begin{array}{rcll}
\mathbf{B}L(\mathbf{B}MN) & = & \mathbf{B}(\mathbf{B}L)(\mathbf{B}M)N & (B) \\
& = & \mathbf{B}\mathbf{B}\mathbf{B}L(\mathbf{B}M)N & (B) \\
& = & \mathbf{B}(\mathbf{B}\mathbf{B}\mathbf{B}L)\mathbf{B}MN & (B) \\
& = & \mathbf{C}\mathbf{B}\mathbf{B}(\mathbf{B}\mathbf{B}\mathbf{B}L)MN & (C) \\
& = & \mathbf{B}(\mathbf{C}\mathbf{B}\mathbf{B})(\mathbf{B}\mathbf{B}\mathbf{B})LMN & (B) \\
& = & \mathbf{B}(\mathbf{B}\mathbf{B})\mathbf{B}LMN & (\mathbf{B}(\mathbf{B}\mathbf{B})\mathbf{B} = \mathbf{B}(\mathbf{C}\mathbf{B}\mathbf{B})(\mathbf{B}\mathbf{B}\mathbf{B})) \\
& = & \mathbf{B}\mathbf{B}(\mathbf{B}L)MN & (B) \\
& = & \mathbf{B}(\mathbf{B}LM)N & (B)
\end{array}$$

Other two sub-cases of (B) and three sub-cases of (C) are similar (and more lengthy).

Lemma 4 *$M =_{\beta\eta} N$ implies $M^\flat =_{\mathbf{BCI}} N^\flat$.*

Proof The most nontrivial part is to show that $M =_{\beta\eta} N$ implies $(\lambda x.M)^\flat =_{\mathbf{BCI}} (\lambda x.N)^\flat$, which follows from Lemma 3.

The following two lemmas are fairly straightforward.

Lemma 5 *$(P^\sharp)^\flat =_{\mathbf{BCI}} P$.*

Lemma 6 *$(\lambda^* x.P)^\sharp =_\beta \lambda x.P^\sharp$.*

Lemma 7 *$(M^\flat)^\sharp =_{\beta\eta} M$.*

Proof Induction on M. Only the case of lambda abstraction is nontrivial, in which we use Lemma 6.

Proposition 2 *$P =_{\mathbf{BCI}} Q$ iff $P^\sharp =_{\beta\eta} Q^\sharp$.*

Proof $P^\sharp =_{\beta\eta} Q^\sharp$ implies $P =_{\mathbf{BCI}} (P^\sharp)^\flat =_{\mathbf{BCI}} (Q^\sharp)^\flat =_{\mathbf{BCI}} Q$ by Lemma 5 and 4.

Proposition 3 *$M =_{\beta\eta} N$ iff $M^\flat =_{\mathbf{BCI}} N^\flat$.*

Proof $M^\flat =_{\mathbf{BCI}} N^\flat$ implies $M =_{\beta\eta} (M^\flat)^\sharp =_{\beta\eta} (N^\flat)^\sharp =_{\beta\eta} N$ by Lemma 7 and 2.