Nominal PROPs

Samuel Balco – Alexander Kurz
University of Leicester – Chapman University

23rd of May 2019

Overview

1. Partially Monoidal Categories

2. A Calculus of Simultaneous Substitutions

3. Internal Monoidal Categories

5. Nominal PROPs

5. Equivalence of PROPs and nominal PROPs

6. Conclusion

Partially Monoidal Categories

Monoidal categories are models of resources

In some models partiality arises naturally

Example: Memory allocation

Example: Simultaneous substitutions

2-Dimensional Calculus of Simultaneous Substitutions

horizontal/sequential composition: $[a{\mapsto}b]\,;[b{\mapsto}c] = [a{\mapsto}c]$

vertical/parallel composition: $[a{\mapsto}b] \oplus [c{\mapsto}d] = [a{\mapsto}b, c{\mapsto}d]$

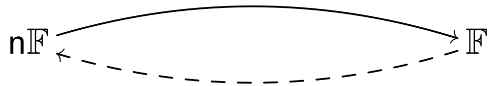
$\oplus$ is partial since the following is not allowed: $[a{\mapsto}b] \oplus [a{\mapsto}c]$

semantics: functions $f : \{a, c\} \to \{b, d\}$

Semantics of Simultaneous Substitutions

The category $\mathsf{n}\mathbb{F}$ of finite subsets of a countably infinite set $\mathcal{N}$ of ‘names’ or ‘variables’.

$\mathsf{n}\mathbb{F}$ is equivalent to the category $\mathbb{F}$ of finite cardinals with all functions.



So why do we care of representing $\mathsf{n}\mathbb{F}$ as opposed to $\mathbb{F}$?

- Syntax is not invariant under isomorphism, see variables vs de Bruin indices in λ-calculus.

- $\mathsf{n}\mathbb{F}$ has more structure, namely that of a **nominal category**, and this structure is not preserved by the equivalence.

- in other words: $\mathbb{F} \dashrightarrow \mathsf{n}\mathbb{F}$ is not an internal functor in the category Nom of **nominal sets**.

Internal Monoidal Categories

What is the relevant structure of $\mathsf{n}\mathbb{F}$?

It is an **internal monoidal category** in $(\mathsf{Nom}, 1, *)$ where $*$ is the so-called separated product of nominal sets.

To make this precise we need to show that we can extend the monoidal operation

$$* : \mathsf{Nom} \times \mathsf{Nom} \to \mathsf{Nom}$$

to an operation

$$* : \mathsf{Cat}(\mathsf{Nom}) \times \mathsf{Cat}(\mathsf{Nom}) \to \mathsf{Cat}(\mathsf{Nom})$$

on internal categories in Nom.

In the following we generalise from Nom to $\mathcal{V}$ and only assume that $(\mathcal{V}, I, \otimes)$ is a monoidal category with finite limits in which I is the terminal object.

Internal Monoidal Categories

pull back the internal category

$$(\mathbb{C}_1 \times \mathbb{D}_1, \mathbb{C}_0 \times \mathbb{D}_0)$$

along

$$\mathbb{C}_0 \otimes \mathbb{D}_0 \to \mathbb{C}_0 \times \mathbb{D}_0$$

$$\begin{array}{ccc} (\mathbb{C} \otimes \mathbb{D})_1 & \dashrightarrow & \mathbb{C}_1 \times \mathbb{D}_1 \\ {\scriptstyle dom} \downarrow \downarrow {\scriptstyle cod} & & {\scriptstyle dom} \downarrow \downarrow {\scriptstyle cod} \\ C_0 \otimes \mathbb{D}_0 & \longrightarrow & \mathbb{C}_0 \times \mathbb{D}_0 \end{array} \tag{1}$$

Lifting $\mathbb{C}_0 \otimes \mathbb{D}_0 \to \mathbb{C}_0 \times \mathbb{D}_0$ to $\mathbb{C} \otimes \mathbb{D} \to \mathbb{C} \times \mathbb{D}$ has a universal property

Lemma 1: The forgetful functor $\mathsf{Cat}(\mathcal{V}) \to \mathcal{V}$ is a fibration.

Where: $\mathsf{Cat}(\mathcal{V})$ is the category of internal catgories in $\mathcal{V}$.

Internal Monoidal Categories, cont'd

But we need more, namely that

$$\mathbb{C}_0 \otimes \mathbb{D}_0 \to \mathbb{C}_0 \times \mathbb{D}_0 \qquad \text{and} \qquad \mathbb{C} \otimes \mathbb{D} \to \mathbb{C} \times \mathbb{D}$$

are *natural transformations*.

Hence, we extend the previous lemma to functor categories:

Lemma 2: If $P : \mathcal{E} \to \mathcal{B}$ is a fibration, then $P^{\mathcal{A}} : \mathcal{E}^{\mathcal{A}} \to \mathcal{B}^{\mathcal{A}}$ is a fibration.

Theorem: Let $(\mathcal{V}, 1, \otimes)$ be a (symmetric) monoidal category with finite limits in which the monoidal unit is the terminal object. $(\mathsf{Cat}(\mathcal{V}), 1, \otimes)$ inherits from $(\mathcal{V}, 1, \otimes)$ the structure of a (symmetric) monoidal category with finite limits in which the monoidal unit is the terminal object,

Internal Monoidal Categories, cont'd

Definition: A strict internal monoidal category $\mathbb{C}$ is a monoid $(\mathbb{C}, \varnothing, \odot)$ in $(\mathsf{Cat}(\mathcal{V}), 1, \otimes)$.

Example: The category $\mathsf{n}\mathbb{F}$ of finite subsets of a set $\mathcal{N}$ of names is an internal monoidal category in $(\mathsf{Nom}, 1, *)$, where

$$* : \mathsf{Cat}(\mathsf{Nom}) \times \mathsf{Cat}(\mathsf{Nom}) \to \mathsf{Cat}(\mathsf{Nom})$$

$$\uplus : \mathsf{n}\mathbb{F} * \mathsf{n}\mathbb{F} \to \mathsf{n}\mathbb{F}$$

$\mathsf{n}\mathbb{F} * \mathsf{n}\mathbb{F}$ has objects: pairs of disjoint sets

arrows: pairs of functions with disjoint domains and disjoint codomains

$\uplus$ is disjoint union, partial wrt to $\mathsf{n}\mathbb{F} \times \mathsf{n}\mathbb{F} \to \mathsf{n}\mathbb{F}$ but total wrt $\mathsf{n}\mathbb{F} * \mathsf{n}\mathbb{F} \to \mathsf{n}\mathbb{F}$

Nominal PROPs

Definition: A nominal PROP is strict internal monoidal category in $(\mathtt{Nom}, 1, *)$ which has finite subsets of $\mathcal{N}$ as objects (supported by themselves) and all bijections as arrows. A morphism of nominal PROPs is an internal strict monoidal functor that preserves bijections.

Equivalence of PROPs and nominal PROPs

Definition/Proposition: For any PROP $\mathcal{S}$, there is an nPROP

$$NOM(\mathcal{S})$$

that has for all arrows $f : \underline{n} \to \underline{m}$ of $\mathcal{S}$, and for all lists $\boldsymbol{a} = [a_1, \ldots a_n]$ and $\boldsymbol{b} = [b_1, \ldots b_m]$ arrows $[\boldsymbol{a}\rangle f \langle \boldsymbol{b}]$. These arrows are subject to equations

$$[\boldsymbol{a}\rangle f \,; g\langle \boldsymbol{c}] = [\boldsymbol{a}\rangle f \langle \boldsymbol{b}] \,; [\boldsymbol{b}\rangle g\langle \boldsymbol{c}] \tag{NOM-1}$$

$$[\boldsymbol{a} +\!\!\!+\, \boldsymbol{c}\rangle f \oplus g\langle \boldsymbol{b} +\!\!\!+\, \boldsymbol{d}] = [\boldsymbol{a}\rangle f \langle \boldsymbol{b}] \uplus [\boldsymbol{c}\rangle g\langle \boldsymbol{d}] \tag{NOM-2}$$

$$[\boldsymbol{a}\rangle id\langle \boldsymbol{b}] = [\boldsymbol{a}|\boldsymbol{b}] \tag{NOM-3}$$

$$[\boldsymbol{a}\rangle \, \langle \boldsymbol{b}|\boldsymbol{b}'\rangle \,; f \, \langle \boldsymbol{c}] \; = [\boldsymbol{a}|\boldsymbol{b}] \,; [\boldsymbol{b}'\rangle f \langle \boldsymbol{c}] \tag{NOM-4}$$

$$[\boldsymbol{a}\rangle \, f \,; \langle \boldsymbol{b}|\boldsymbol{b}'\rangle \, \langle \boldsymbol{c}] \; = [\boldsymbol{a}\rangle f \langle \boldsymbol{b}] \,; [\boldsymbol{b}'|\boldsymbol{c}] \tag{NOM-5}$$

Equivalence of PROPs and nominal PROPs, cont'd

Definition/Proposition: For any nPROP $\mathcal{T}$ there is a PROP

$$ORD(\mathcal{T})$$

that has for all arrows $f : A \to B$ of $\mathcal{T}$, and for all lists $\boldsymbol{a} = [a_1, \ldots a_n]$ and $\boldsymbol{b} = [b_1, \ldots b_m]$ arrows $\langle \boldsymbol{a}] f [\boldsymbol{b}\rangle$. These arrows are subject to equations

$$\langle \boldsymbol{a}]\, f \,;\, g\, [\boldsymbol{c}\rangle = \langle \boldsymbol{a}]\, f\, [\boldsymbol{b}\rangle \,;\, \langle \boldsymbol{b}]\, g\, [\boldsymbol{c}\rangle \tag{ORD-1}$$

$$\langle \boldsymbol{a}_f +\!\!+\, \boldsymbol{a}_g]\, f \uplus g\, [\boldsymbol{b}_f +\!\!+\, \boldsymbol{b}_g\rangle = \langle \boldsymbol{a}_f]\, f\, [\boldsymbol{b}_f\rangle \oplus \langle \boldsymbol{a}_g]\, g\, [\boldsymbol{b}_g\rangle \tag{ORD-2}$$

$$\langle \boldsymbol{a}]\, id\, [\boldsymbol{a}\rangle = id \tag{ORD-3}$$

$$\langle \boldsymbol{a}]\, [\boldsymbol{a}'|\boldsymbol{b}] \,;\, f\, [\boldsymbol{c}\rangle = \langle \boldsymbol{a}|\boldsymbol{a}'\rangle \,;\, \langle \boldsymbol{b}]\, f\, [\boldsymbol{c}\rangle \tag{ORD-4}$$

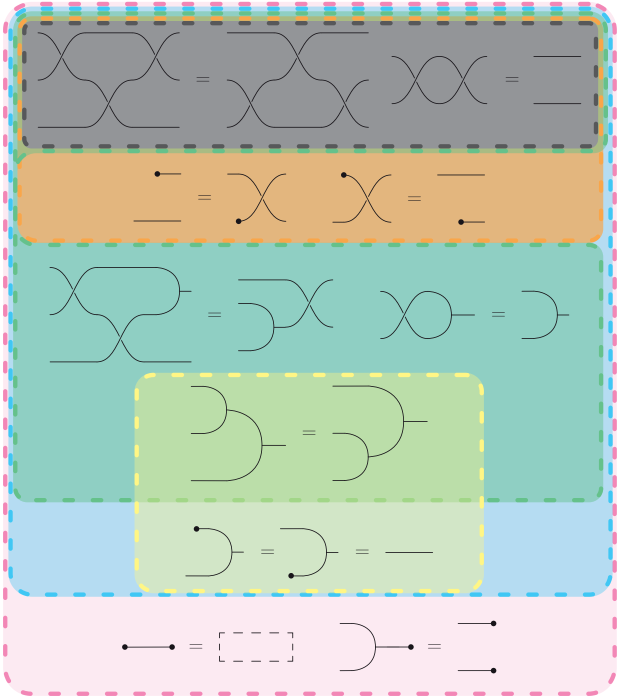
$$\langle \boldsymbol{a}]\, f \,;\, [\boldsymbol{b}|\boldsymbol{c}]\, [\boldsymbol{c}'\rangle = \langle \boldsymbol{a}]\, f\, [\boldsymbol{b}\rangle \,;\, \langle \boldsymbol{c}|\boldsymbol{c}'\rangle \tag{ORD-5}$$

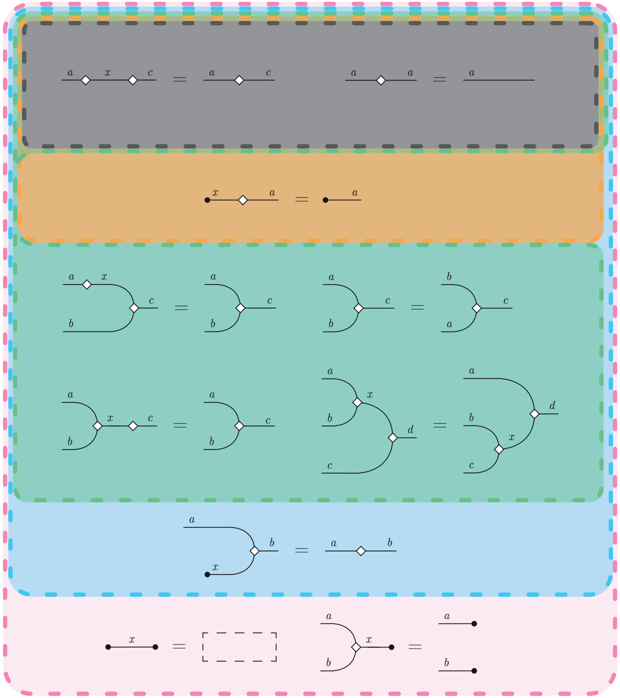
Equivalence of PROPs and nominal PROPs, cont'd

Theorem: The categories PROP and nPROP are equivalent.

Remark: The interesting part of the proof is to show how commutativity of $\uplus$ in nPROPs and naturality of symmetries in PROPs correspond to each other.

Equivalence of PROPs and nominal PROPs, cont'd





Equivalence of PROPs and nominal PROPs, cont'd