Keywords
higher category theory, concurrency, message-passing, types, Curry-Howard

ABSTRACT
We present an approach to logics and types in terms of category theory.

Logics as distributive laws

Michael Stay
Google
stay@google.com

L.G. Meredith
Biosimilarity, LLC
lgreg.meredith@biosimilarity.com

1. INTRODUCTION

TBD

1.0.1 Related work

TBD

1.0.2 Organization of the rest of the paper

TBD

2. THE CALCULUS

TBD

2.1 Our running process calculus

2.1.1 Syntax

TBD

Due to space limitations we do not treat replication, $!P$.

2.1.2 Free and bound names

TBD

2.1.3 Structural congruence

TBD

2.1.4 Operational Semantics

TBD

2.1.5 Bisimulation

TBD

3. CATEGORICAL MACHINERY

Here's how our construction would work with the S,K combinator basis to model head normal form.

Monoidal 2-category.

4. THE INTERPRETATION

One object $\mathcal{T}$ of terms. Generating morphisms

- $\mathsf{S} : \mathsf{I} \to \mathcal{T}$
- $\mathsf{K} : \mathsf{I} \to \mathcal{T}$
- $() : \mathcal{T} \otimes \mathcal{T} \to \mathcal{T}$ // application
- $\delta : \mathcal{T} \to \mathsf{I}$ // delete a subterm
- $\Delta : \mathcal{T} \to \mathcal{T} \otimes \mathcal{T}$ // duplicate a subterm
- $\mathsf{R} : \mathcal{T} \to \mathcal{T}$ // Reduction context

Generating rewrites

- $\forall\, x, y, z\ \mathsf{R}((xy)z) \to \mathsf{R}(\mathsf{R}(xy)z)$
- $\forall\, x, y\ \mathsf{R}(\mathsf{R}(Kx)y) \to x \otimes \delta(y)$
- $\forall\ x, y, z\ \mathsf{R}(\mathsf{R}(\mathsf{R}(Sx)y)z) \to \mathsf{R}((-_1-_2)(-_3-_4)) \circ (T \otimes swap \otimes T) \circ (T \otimes T \otimes \Delta)(x, y, z)$ i.e. $\mathsf{R}((xz)(yz))$

We can take the 2-category to be Poset as before.

This is a completely untyped combinator calculus.

4.1 Semantics

To get types, we can consider sets of terms that satisfy a proposition:

- $[\![\top]\!] = \mathcal{T}$
- $[\![\bot]\!] =$
- $[\![S]\!] = \{S\}$

- $[\![K]\!] = \{K\}$
- $[\![A1B]\!] = \{t \in \mathcal{T} | \exists u \in [\![A]\!], v \in B, t' \in \mathcal{T}.t \rightarrow t', (vu) \rightarrow t'\}$
- $[\![A2B]\!] = \{t \in \mathcal{T} | \exists u \in [\![A]\!], v \in [\![B]\!], t' \in \mathcal{T}.v \rightarrow t', (tu) \rightarrow t'\}$
- $[\![A3B]\!] = \{t \in \mathcal{T} | \exists u \in [\![A]\!], v \in [\![B]\!], t' \in \mathcal{T}.v \rightarrow t', (ut) \rightarrow t'\}$ etc.

We often use $\multimap$ to mean 2.

4.1.1 Bisimulation again

TBD

5. CONCLUSIONS AND FUTURE WORK

TBD

Acknowledgments. TBD