The dagger lambda calculus

Philip Atzemoglou

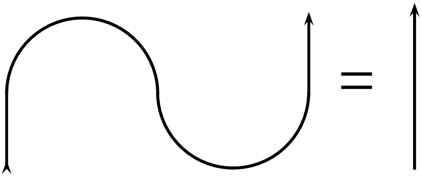
University of Oxford

abb.research@the-judges.com

Quantum Physics and Logic 2014

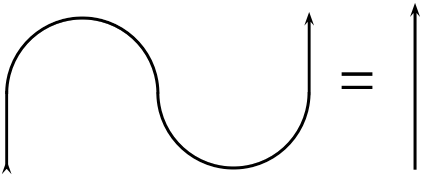
Why higher-order?

Teleportation should be the same:

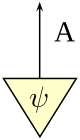


Why higher-order?

Teleportation should be the same:

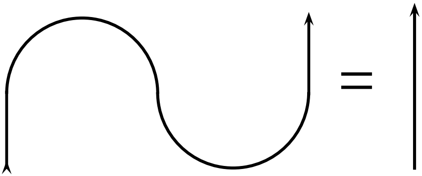


Regardless of whether you are teleporting

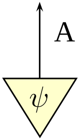


a state

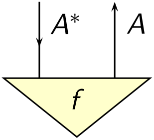
Teleportation should be the same:



Regardless of whether you are teleporting

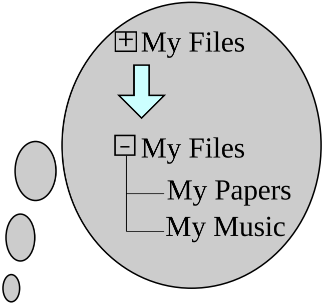


a state

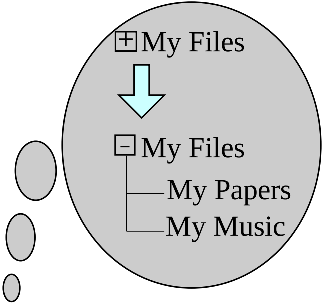


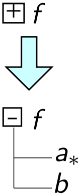
or a function

Why higher-order?



Why higher-order?





Terms

$$\langle \mathit{term} \rangle \quad ::= \quad \mathit{variable} \quad | \quad \mathit{constant} \quad | \quad \langle \mathit{term} \rangle \otimes \langle \mathit{term} \rangle \quad | \quad \langle \mathit{term} \rangle_*$$

Terms

$$\langle term \rangle \quad ::= \quad variable \quad | \quad constant \quad | \quad \langle term \rangle \otimes \langle term \rangle \quad | \quad \langle term \rangle_*$$

$$i.e. \quad x, y, z \quad | \quad c \quad | \quad t_1 \otimes t_2 \quad | \quad f_*, (t_1 \otimes t_2)_*$$

Types

$$\langle type \rangle \quad ::= \quad atomic \quad | \quad \langle type \rangle \otimes \langle type \rangle \quad | \qquad \langle type \rangle^{*}$$

Types

$$\langle type \rangle \quad ::= \quad atomic \quad | \quad \langle type \rangle \otimes \langle type \rangle \quad | \quad \langle type \rangle^*$$

$$i.e. \quad A, B, C \quad | \quad A \otimes B \quad | \quad (A \otimes B)^*, A^* \otimes A$$

Linear negation

- Involutive: $(a_*)_* \equiv a$ and $(A^*)^* \equiv A$

Linear negation

- Involutive: $(a_*)_* \equiv a$ and $(A^*)^* \equiv A$
- Is $\otimes$ equal to $\parr$?

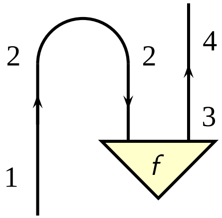
Linear negation

- Involutive: $(a_*)_* \equiv a$ and $(A^*)^* \equiv A$
- Is $\otimes$ equal to $\parr$? Almost:
  Planar negation: $(a \otimes b)_* := b_* \otimes a_*$ and $(A \otimes B)^* := B^* \otimes A^*$

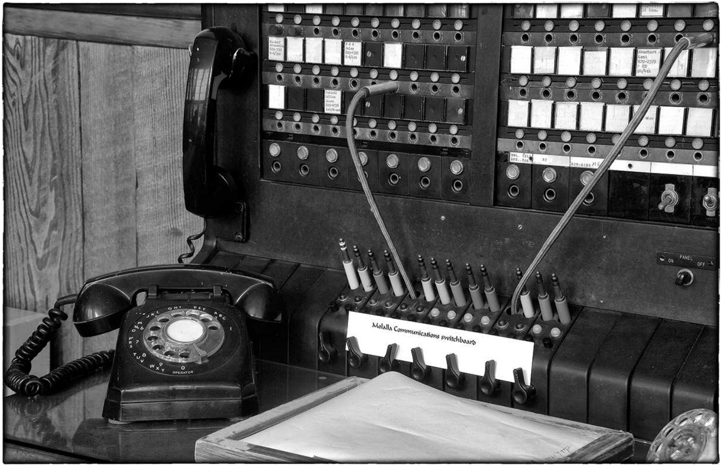
The soup

A set of connections between equityped terms. These connections correspond to connecting wires in a categorical diagram:

$$S = \{x_1 : x_2, f : x_{2*} \otimes x_3, x_3 : x_4\}$$



Switchboard



Used in compliance with the CC-Attribution license of the original from: https://www.flickr.com/photos/glenbledsoe/6245440290/

Sequents

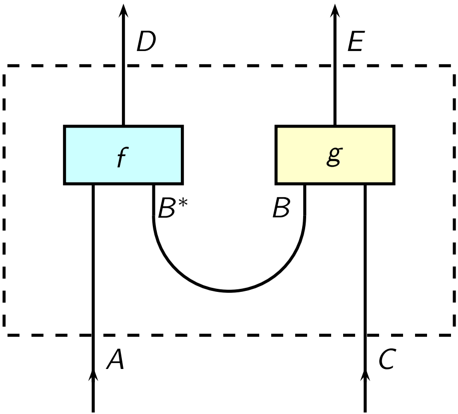
$$t_1 : A_1,\ t_2 : A_2,\ \ldots,\ t_n : A_n \vdash_S t : B$$

Reconstructing λ

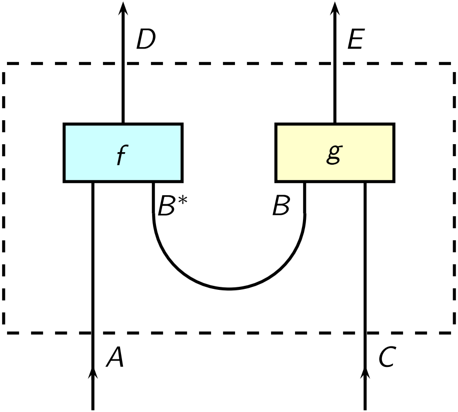
$$\lambda a.b := a_* \otimes b$$

$$A \multimap B := A^* \otimes B$$

Representing connections between wires



Representing connections between wires



$$a : A, c : C \vdash_S d \otimes e : D \otimes E$$

where

$$S = \left\{ \begin{array}{ll} f & : \lambda\,(a \otimes b_*)\,.d, \\ g & : \lambda\,(b \otimes c)\,.e \end{array} \right\}$$

Sequent rules

$$\frac{}{x : A \vdash x : A}\ \text{Id},$$

$$\frac{\Gamma, a : A, b : B \vdash_S c : C}{\Gamma, a \otimes b : A \otimes B \vdash_S c : C}\ \otimes L,$$

$$\frac{\Gamma \vdash_{S_1} a : A \qquad \Delta \vdash_{S_2} b : B}{\Gamma, \bigotimes \Delta \vdash_{S_1 \cup S_2} a \otimes b : A \otimes B}\ \otimes R,$$

$$\frac{\Gamma \vdash_{S_1} a : A \qquad a' : A, \Delta \vdash_{S_2} b : B}{\Gamma, \Delta \vdash_{S_1 \cup S_2 \cup \{a : a'\}} b : B}\ \text{Cut},$$

$$\frac{a : A, \Gamma \vdash_S b : B}{\Gamma \vdash_S a_* \otimes b : A^* \otimes B}\ \text{Curry},$$

$$\frac{a : A \vdash_S b : B}{a_* : A^* \vdash_{S_*} b_* : B^*}\ \text{Negation},$$

$$\frac{\Gamma, a : A, b : B, \Delta \vdash c : C}{\Gamma, b : B, a : A, \Delta \vdash c : C}\ \text{Exchange},$$

$$\frac{\Gamma \vdash_{S \cup \{i_* : 1\}} b : B}{i : I, \Gamma \vdash_S b : B}\ \lambda_\Gamma,$$

$$\frac{\Gamma \vdash_{S \cup \{i_* : 1\}} b : B}{\Gamma, i : I \vdash_S b : B}\ \rho_\Gamma.$$

†-flip

$$\frac{a : A \vdash_S b : B}{b : B \vdash_{S_*} a : A} \text{ †-flip}$$

†-flip

$$\frac{a : A \vdash_S b : B}{b : B \vdash_{S_*} a : A}\ \dagger\text{-flip}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{a : A \vdash_S b : B}{a_* : A^* \vdash_{S_*} b_* : B^*}\ \text{Negation}}{b : B, a_* : A^* \vdash_{S_*}}\ \text{Uncurry}}{a_* : A^*, b : B \vdash_{S_*}}\ \text{Exchange}}{b : B \vdash_{S_*} a : A}\ \text{Curry}$$

Soup reduction

The soup propagation rules are *bifunctoriality*, *trace* and *cancellation*:

$$S \cup \{a \otimes b : c \otimes d\} \longrightarrow S \cup \{a : c, b : d\}$$
$$S \cup \{x :_A x\} \longrightarrow S \cup \{D_A : 1\}$$
$$S \cup \{1 : 1\} \longrightarrow S$$

Soup reduction

The soup propagation rules are *bifunctoriality*, *trace* and *cancellation*:

$$\begin{aligned} S \cup \{a \otimes b : c \otimes d\} &\longrightarrow S \cup \{a : c, b : d\} \\ S \cup \{x :_A x\} &\longrightarrow S \cup \{D_A : 1\} \\ S \cup \{1 : 1\} &\longrightarrow S \end{aligned}$$

Our soup rules also contain a *consumption rule*:

$$\Gamma \vdash_{S \cup \{t:u\}} b : B \longrightarrow \left(\Gamma \vdash_S b : B\right) \begin{cases} [t/u], & \text{if } u \text{ has no constants} \\ [u/t], & \text{if } t \text{ has no constants} \end{cases}$$

Application

Application is defined as a notational shorthand, representing a variable and a connection in the soup. The origins of the application affect the structure of its corresponding soup connection:

$$\begin{aligned} ft : B, \Gamma \vdash c : C \quad &:= \quad x : B, \Gamma \vdash_{\{f : t_* \otimes x\}_*} c : C \text{ and} \\ \Gamma \vdash ft : B \quad &:= \quad \Gamma \vdash_{\{f : t_* \otimes x\}} x : B \end{aligned}$$

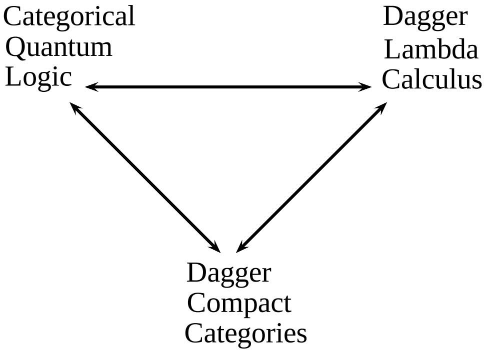
For an application originating inside our soup, we have:

$$\{ft : c\} := \{x : c\} \cup \{f : t_* \otimes x\} \text{ and}$$
$$\{c : ft\} := \{c : x\} \cup \{f : t_* \otimes x\}_*$$

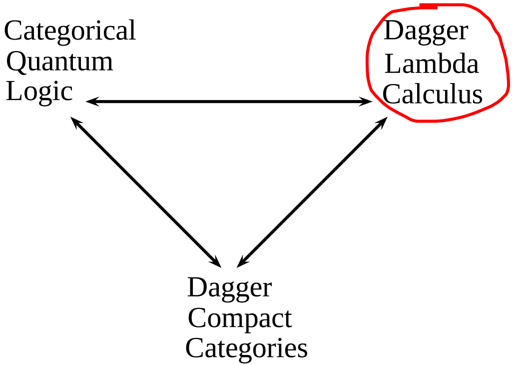
Properties

- Subject reduction
- Consistency
- Strong normalisation
- Confluence

Curry-Howard-Lambek correspondence



Curry-Howard-Lambek correspondence



- Internal language for *dagger compact categories*

Conclusion

Future work:

- Extend to cover complementary classical structures and dualisers
- Support for the non-determinacy of measurements
- Higher-order representation for MBQC

Thanks are due to:

- Samson Abramsky,
- Bob Coecke,
- Prakash Panangaden,
- Jonathan Barrett,
- ... and many others ...