String Diagrams: Studies on an effect-based approach to semantic parsing

Matthieu Boyer

École Normale Supérieure | Yale University

26th June 2025

Plan

General Introduction

This work, based on [BC25] aims to provide a categorical formalization of a type and effects system for semantic interpretation of the natural language.

We will develop a graphical formalism for semantic type-driven parsing and prove it is equivalent to a miinmalistic coloured merge interpretation of syntax.

Types in Semantics of Natural Languages

Expression	Type	λ-Term
planet	$e \to t$	$\lambda x.\mathbf{planet}\, x$
	Generalizes to **common nouns**	
carnivorous	$(e \to t)$	$\lambda x.\mathbf{carnivorous}\, x$
	Generalizes to **predicative adjectives**	
skillful	$(e \to t) \to (e \to t)$	$\lambda p.\lambda x.px \wedge \mathbf{skillful}\, x$
	Generalizes to **predicate modifier adjectives**	
Jupiter	e	$\mathbf{j} \in \mathrm{Var}$
	Generalizes to **proper nouns**	
sleep	$e \to t$	$\lambda x.\mathbf{sleep}\, x$
	Generalizes to **intransitive verbs**	

Including Non-Determinisms and Anaphoras

What should be the type of expressions such as **a cat** or **Jupiter, a planet**?

Including Non-Determinisms and Anaphoras

What should be the type of expressions such as **a cat** or **Jupiter, a planet**?

Since we should be able to use **a cat** and **the cat** interexchangebly - from a syntax point of view - they should have the same type. We use effects to do the difference between:

$$\textbf{a cat} = \{c \mid \textbf{cat}\, c\}$$

$$\textbf{the cat} = x \text{ if } \textbf{cat}^{-1}(\top) = \{x\} \text{ else } \#$$

Plan

A Type and Effect System

Let $\mathcal{L}$ be our language: a typed lexicon and syntactic rules. We only suppose that our words can be applied to one another in their denotation system.

A Type and Effect System

Let $\mathcal{L}$ be our language: a typed lexicon and syntactic rules. We only suppose that our words can be applied to one another in their denotation system.

Let $\mathcal{C}$ be a cartesian closed category used for typing the lexicon. Let $\mathcal{F}(\mathcal{L})$ be a set of functors used for representing the words that add an effect to our language.

A Type and Effect System

Let $\mathcal{L}$ be our language: a typed lexicon and syntactic rules. We only suppose that our words can be applied to one another in their denotation system.

Let $\mathcal{C}$ be a cartesian closed category used for typing the lexicon. Let $\mathcal{F}(\mathcal{L})$ be a set of functors used for representing the words that add an effect to our language.

We consider $\bar{\mathcal{C}}$ the categorical closure of $\mathcal{C}$ under the action of $\mathcal{F}(\mathcal{L})^*$. We close it for the cartesian product and exponential of $\mathcal{C}$.

A Type and Effect System

Let $\mathcal{L}$ be our language: a typed lexicon and syntactic rules. We only suppose that our words can be applied to one another in their denotation system.

Let $\mathcal{C}$ be a cartesian closed category used for typing the lexicon. Let $\mathcal{F}(\mathcal{L})$ be a set of functors used for representing the words that add an effect to our language.

We consider $\bar{\mathcal{C}}$ the categorical closure of $\mathcal{C}$ under the action of $\mathcal{F}(\mathcal{L})^*$. We close it for the cartesian product and exponential of $\mathcal{C}$.

We also introduce the notions of applicatives and monads, as they grant more flexibility with semantic combinations.

Plan

Handlers

A handler for an effect F is a natural transformation $F \Rightarrow \mathrm{Id}$.

Handlers are not generally language-defined and are speaker-dependent. Adjunctions and comonads provide handlers intrinsic to their effects. Handlers should also be exact inverses to monadic and applicative units: this justifies semantically why we can remove the usage of the unit rule out of certain situations.

A large question we have to solve before parsing is whether two denotations will always yield the same result, considering effect handling.

String Diagrams in Denotations - 1

We will represent our denotations as string diagrams to make computations easier and to better understand what happens to each effect when handling.

This allows us to look at equality of string diagrams instead of complex commutative diagrams.

What happens when combining diagrams and why they have such a shape will be detailed in the following section.

String Diagram Isotopy

Theorem 3.1 — **Theorem 3.1 [Sel10], Theorem 1.2 [JS91]** A well-formed equation between morphism terms in the language of monoidal categories follows from the axioms of monoidal categories if and only if it holds, up to planar isotopy, in the graphical language.

Reduction Scheme

[DV22] proposed a combinatorial description to check in linear time for equality under Theorem 3.1.

Reduction Scheme

[DV22] proposed a combinatorial description to check in linear time for equality under Theorem 3.1.

Theorem 3.3 — Confluence Our reduction system is confluent and therefore defines normal forms:

1. Right reductions are confluent and therefore define *right* normal forms for diagrams under the equivalence relation induced by exchange.
2. Equational reductions are confluent and therefore define *equational* normal forms for diagrams under the equivalence relation induced by exchange.

Polynomial Time Reductions

Theorem 3.4 — Normalization Complexity Reducing a diagram to its normal form is done in quadratic time in the number of natural transformations in it.

This is accomplished using a formalism based on [DV22].

Plan

String Diagram Combination

Given our grammar, we could build parsing trees, but that would blur the actual usefulness of our grammar and our string diagrammatic representation of sentences.

We thus consider diagrams whose 1-cells are objects in $\bar{\mathcal{C}}$, i.e. types and effects and whose natural transformations are the combinators of our grammar.

Bibliography I

[BC25] Dylan Bumford and Simon Charlow. *Effect-Driven Interpretation: Functors for Natural Language Composition*. Mar. 2025. (Visited on 24/04/2025).

[DV22] Antonin Delpeuch and Jamie Vicary. *Normalization for Planar String Diagrams and a Quadratic Equivalence Algorithm*. Jan. 2022. DOI: 10.48550/arXiv.1804.07832. arXiv: 1804.07832. (Visited on 31/03/2025).

[JS91] André Joyal and Ross Street. "The Geometry of Tensor Calculus, I". In: *Advances in Mathematics* 88.1 (July 1991), pp. 55–112. ISSN: 00018708. DOI: 10.1016/0001-8708(91)90003-P. (Visited on 28/03/2025).

Bibliography II

[Sel10] Peter Selinger. "A Survey of Graphical Languages for Monoidal Categories". In: vol. 813. 2010, pp. 289–355. DOI: 10.1007/978-3-642-12821-9_4. arXiv: 0908.3347 [math]. (Visited on 03/04/2025).