



# Compositional Semantics: Montagovian Semantics and Lambda Calculus

**Instructor:** Jackie CK Cheung

COMP-550

J&M Ch. 14 – 14.3 (1<sup>st</sup>); J&M Ch. 17 – 17.4  
(2<sup>nd</sup>); J&M Ch. 19 (3<sup>rd</sup>)

# Reminders

---

Project proposal due Oct 27, 2023

- If you still don't have a project team yet, please drop me an e-mail or come to my office hours.

RA2 due Oct 31, 2023

Midterm on Nov 9

- More info on that on Thursday (format, practice questions, old exams)

# Outline

---

Principle of compositionality

Semantic inference

First-order logic

Lambda calculus

# The Principle of Compositionality

**Compositionality:** The meaning of a phrase depends on the meanings of its parts.

*COMP-550 is a fantastically awesome class.*

Lexical semantics gives us the meanings and behaviours of each of the words:

- *COMP-550, is, a, fantastically, awesome, class*

We build up the meaning of the entire sentence through composition.

# Goal of Compositional Semantics

Derive a meaning representation of a phrase/sentence from its parts

What is a good meaning representation?

Relates the linguistic expression to the world:

- Asserts a proposition that is either true or false relative to the world

*Pandas are purple and yellow.*

- Conveys information about the world

*It will snow tomorrow.*

- Is a query about the state of the world

*What is the weather like in Montreal next week?*

# Idioms – Violation of Compositionality

Idioms are expressions whose meanings cannot be predicted from their parts.

*kick the bucket*

*the last straw*

*piece of cake*

*hit the sack*

# Co-Compositionality

---

Consider the meaning of *red* when modifying each of the following nouns

- *rose*
- *wine*
- *cheeks*
- *hair*

Is red really combining compositionally with each of these nouns?

- **Co-compositionality** (Pustejovsky, 1995) – the meanings of words depend also on the other words that they are composed with

# Montagovian Semantics

---

Montague (1970) started a tradition of using a logical formalism to represent the meaning of a sentence, with a tight connection to syntax.

*There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed I consider it possible to comprehend the syntax and semantics of both kinds of languages with a single natural and mathematically precise theory. (Montague 1970, 222)*

Natural language inference then can be seen as applying logical rules of inference.

- So, what is inference?



# Semantic Inference

---

Making *explicit* something that is *implicit* in language  
(Blackburn and Bos, 2003)

*I want to visit the capital of Italy.*

*The capital of Italy is Rome.*

---

*∴ I want to visit Rome.*

*All wugs are borks.*

*All borks are cute.*

---

*∴ All wugs are cute.*

# First-Order Predicate Calculus

---

## Domain of discourse

A set of entities that we care about

e.g., the students in the class, the topics we study, classrooms, courses, etc.

## Variables

Typically lower-case

Stands for potential elements of the domain

e.g.,  $x, y, z$

# First-Order Predicate Calculus

---

## Predicates

Maps elements of the domain to truth values

Can be of different valences

e.g. *inCourse*( $x, y$ ): takes in two elements of the domain, returns true if  $x$  is a student in course  $y$ , false otherwise

## Functions

Maps elements to other elements

Can be of different valences

e.g. *instructorOf*( $x$ ): takes  $x$ , returns an element corresponding to  $x$ 's instructor

What is a valence 0 function?

# First-Order Predicate Calculus

---

## Logical connectives

- All the standard ones

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

## Quantifiers

- Existential  $\exists$
- Universal  $\forall$

# Example Sentences

---

*The capital of Italy is Rome.*

$$\textit{capitalOf}(\textit{ITALY}) = \textit{ROME}$$

*All wugs are borks.*

$$\forall x. \textit{wug}(x) \rightarrow \textit{bork}(x)$$

# Interpreting FOL

---

A particular instance of a FOL consists of:

- Predicate and function names and arity
- A set of sentences in FOL using those predicates and functions

An **interpretation** or **model** of a FOL consists of:

Domain of discourse,  $D$

Mapping for the functions to elements of  $D$

Mapping for the predicates to True or False

# Exercise

---

Come up with a FOL characterization of the following:

- Students who study AND do homework will get an A
- Students who only do one of them get a B
- Students who do neither get a C

List the predicates and functions that are necessary. Make constants for the grades (A, B, C).

Come up with an interpretation of this FOL, where you and two of your friends are the elements in the domain of discourse, such that the above FOL formulas are true.

# Building Meaning Representations

---

Target MR: a logical formula in FOL

Still needed:

A procedure to map sentences to a FOL formula  
compositionally

Tool: **Lambda calculus**



# Lambda Calculus

---

Basically a way to describe computation using mathematical functions

- The computation we will be doing is to build up a FOL sentence as the meaning representation of a sentence.

Terms in Lambda calculus can be defined recursively:

- A variable (e.g.,  $x$ )
- $\lambda x. t$ , where  $t$  is a lambda term
- $ts$ , where  $t$  and  $s$  are lambda terms

# Functional Application

---

Function application (or **beta reduction**) of term  $(\lambda x. t)s$

- Replace all instances of  $x$  in  $t$  with the expression  $s$

e.g.,  $(\lambda x. x + y)2$  simplifies to  $2 + y$

$$(\lambda x. xx)(\lambda x. x) = (\lambda x. x)(\lambda x. x) = (\lambda x. x)$$

Function application is **left-associative**:

$$abcd = ((ab)c)d$$

I define this notion intuitively here, and gloss over some details, but these definitions can (should) be formalized, in order to be precise:

<http://arxiv.org/pdf/1503.09060.pdf>

# Power of Lambda Calculus

---

They allow us to store partial computations of the MR, as we are composing the meaning of the sentence constituent by constituent.

*Whiskers disdained catnip.*

*disdained*  $\lambda x. \lambda y. disdained(y, x)$

*disdained catnip*  $(\lambda x. \lambda y. disdained(y, x)) catnip$   
 $= \lambda y. disdained(y, catnip)$

*Whiskers disdained catnip*

$(\lambda y. disdained(y, catnip)) Whiskers$   
 $= disdained(Whiskers, catnip)$

# Exercises

---

What is the result of simplifying the following expressions in lambda calculus through beta reduction?

$(\lambda z. z)(\lambda y. y y)(\lambda x. x a)$

$((\lambda x. \lambda y. (x y))(\lambda y. y)) w$

$(\lambda x. x x) (\lambda y. y x) z$

# Syntax-Driven Semantic Composition

---

Augment CFG trees with lambda expressions

- Syntactic composition = function application

Semantic attachments:

$$A \rightarrow \alpha_1 \dots \alpha_n \qquad \{f(\alpha_j.sem, \dots, \alpha_k.sem)\}$$

syntactic composition

semantic attachment

# Proper Nouns

---

Proper nouns are FOL constants

$$PN \rightarrow COMP550 \quad \{COMP550\}$$

Actually, we will **type-raise** proper nouns

$$PN \rightarrow COMP550 \quad \{\lambda x. x(COMP550)\}$$

- It is now a function rather than an argument.
- We will see why we do this next class.

NP rule:

$$NP \rightarrow PN \quad \{PN.sem\}$$

# Common Nouns

---

Common nouns are predicates inside a lambda expression of type  $\langle e, t \rangle$

- Takes an entity, tells you whether the entity is a member of that class

$N \rightarrow student \quad \{\lambda x. Student(x)\}$

Let's talk more about common nouns next class when we also talk about quantifiers.

# Intransitive Verbs

---

We introduce an *event variable*  $e$ , and assert that there exists a certain event associated with this verb, with arguments.

$$V \rightarrow rules \quad \{\lambda x. \exists e. Rules(e) \wedge Ruler(e, x)\}$$

Then, composition is

$$S \rightarrow NP VP \quad \{NP.sem(VP.sem)\}$$

Let's derive the representation of the sentence "*COMP-550 rules*"



# Neo-Davidsonian Event Semantics

---

Notice that we have changed how we represent events

**Method 1:** multi-place predicate

*Rules(x)*

**Method 2:** Neo-Davidsonian version with event variable

$\exists e. Rules(e) \wedge Ruler(e, x)$

Reifying the event variable makes things more flexible

- Optional elements such as location and time, passives
- Add information to the event variable about tense, modality

# Transitive Verbs

---

## Transitive verbs

$V \rightarrow enjoys$

$\{\lambda w. \lambda z. w(\lambda x. \exists e. Enjoys(e) \wedge Enjoyer(e, z) \wedge Enjoyee(e, x))\}$

$VP \rightarrow V NP$

$\{V.sem(NP.sem)\}$

$S \rightarrow NP VP$

$\{NP.sem(VP.sem)\}$

**Exercise:** verify that this works with the sentence “*Jackie enjoys COMP-550*”

# Next Class

---

Quantifiers and common nouns

Adjectives, adverbs, and modifiers

Underspecification