# CSCI 5832
# Natural Language Processing

**Lecture 21**
**Jim Martin**



# Today: 4/10

- **Compositional Semantics**
  - **Syntax-driven methods of assigning semantics to sentences**

## Meaning Representations

- We're going to take the same basic approach to meaning that we took to syntax and morphology
- We're going to create representations of linguistic inputs that capture the meanings of those inputs.
- But unlike parse trees and the like these representations aren't primarily descriptions of the structure of the inputs…



## Semantic Processing

- We're going to discuss 2 ways to attack this problem (just as we did with parsing)
  - There's the theoretically motivated correct and complete approach…
    - Computational/Compositional Semantics
  - And there are practical approaches that have some hope of being useful and successful.
    - Information extraction

## Semantic Analysis

- **Compositional Analysis**
  - **Create a FOL representation that accounts for all the entities, roles and relations present in a sentence.**
- **Information Extraction**
  - **Do a superficial analysis that pulls out only the entities, relations and roles that are of interest to the consuming application.**



## Representational Schemes

- **We're going to make use of First Order Predicate Calculus (FOPC) as our representational framework**
  - **Not because we think it's perfect**
  - **All the alternatives turn out to be either too limiting or**
  - **They turn out to be notational variants**

## FOPC

- Allows for…
  - The analysis of truth conditions
    - Allows us to answer yes/no questions
  - Supports the use of variables
    - Allows us to answer questions through the use of variable binding
  - Supports inference
    - Allows us to answer questions that go beyond what we know explicitly



## FOPC

- This choice isn't completely arbitrary or driven by the needs of practical applications
- FOPC reflects the semantics of natural languages because it was designed that way by human beings
- In particular…

## Meaning Structure of Language

- The semantics of human languages…
  - Display a basic predicate-argument structure
  - Make use of variables
  - Make use of quantifiers
  - Use a partially compositional semantics



## Predicate-Argument Structure

- Events, actions and relationships can be captured with representations that consist of predicates and arguments to those predicates.
- Languages display a division of labor where some words and constituents function as predicates and some as arguments.

## Predicate-Argument Structure

- Predicates
  - Primarily Verbs, VPs, PPs, Sentences
  - Sometimes Nouns and NPs
- Arguments
  - Primarily Nouns, Nominals, NPs, PPs
  - But also everything else; as we'll see it depends on the context



## Example

- *Mary gave a list to John.*
- Giving(Mary, John, List)
- More precisely
  - Gave conveys a three-argument predicate
  - The first arg is the subject
  - The second is the recipient, which is conveyed by the NP in the PP
  - The third argument is the thing given, conveyed by the direct object

## Not exactly

- When we say that
  - The first arg is the subject
- We really mean that the meaning underlying the subject phrase plays the role of the giver.



## Better

- Turns out this representation isn't quite as useful as it could be.
  - Giving(Mary, John, List)
- Better would be

$$\exists x, y\ Giving(x) \wedge Giver(Mary, x) \wedge Given(y, x)$$
$$\wedge Givee(John, x) \wedge Isa(y, List)$$

## Predicates

- The notion of a predicate just got more complicated…
- In this example, think of the verb/VP providing a template like the following

$$\exists w,x,y,z Giving(x) \wedge Giver(w,x) \wedge Given(y,x) \wedge Givee(z,x)$$

- The semantics of the NPs and the PPs in the sentence plug into the slots provided in the template



## Semantic Analysis

- Semantic analysis is the process of taking in some linguistic input and assigning a meaning representation to it.
  - There a lot of different ways to do this that make more or less (or zero) use of syntax
  - We're going to start with the idea that syntax does matter
    - The compositional rule-to-rule approach

## Compositional Analysis

- Principle of Compositionality
  - The meaning of a whole is derived from the meanings of the parts
- What parts?
  - The constituents of the syntactic parse of the input
- What could it mean for a part to have a meaning?

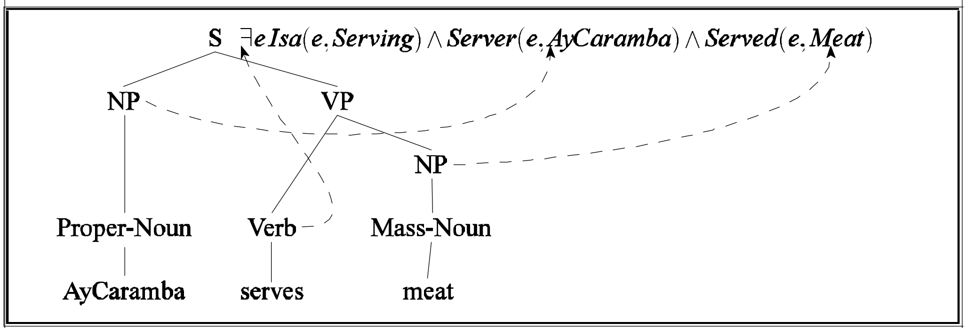


## Example

- AyCaramba serves meat

$$\exists e\ Serving(e) \wedge Server(e, AyCaramba) \wedge Served(e, Meat)$$

# Compositional Analysis






# Augmented Rules

- **We'll accomplish this by attaching semantic formation rules to our syntactic CFG rules**
- **Abstractly**

$$A \rightarrow \alpha_1 \ldots \alpha_n \quad \{f(\alpha_1.sem, \ldots \alpha_n.sem)\}$$

- **This should be read as the semantics we attach to A can be computed from some function applied to the semantics of A's parts.**

## Example

- Easy parts…
  - NP -> PropNoun
  - NP -> MassNoun
  - PropNoun -> AyCaramba
  - MassMoun -> meat
- Attachments
  {PropNoun.sem}
  {MassNoun.sem}
  {AyCaramba}
  {MEAT}



## Example

- S -> NP VP
- VP -> Verb NP
- Verb -> serves
- {VP.sem(NP.sem)}
- {Verb.sem(NP.sem)
- ???

$$\lambda x \lambda y \; \exists e \; Serving(e) \wedge Server(e,y) \wedge Served(e,x)$$

# Lambda Forms

- **A simple addition to FOPC**
  - **Take a FOPC sentence with variables in it that are to be bound.**
  - **Allow those variables to be bound by treating the lambda form as a function with formal arguments**

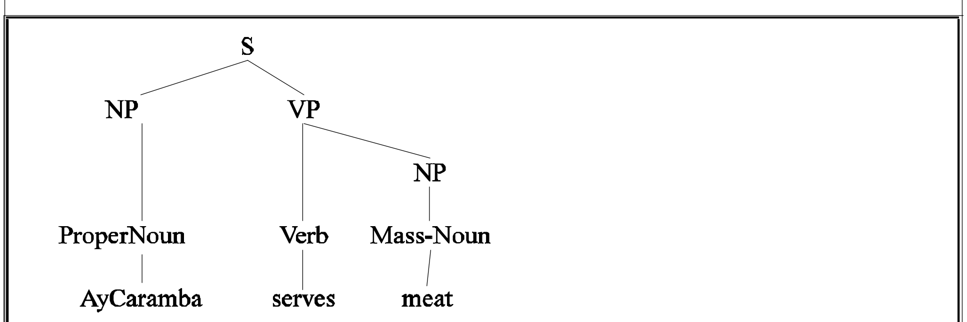
$$\lambda x P(x)$$

$$\lambda x P(x)(Sally)$$

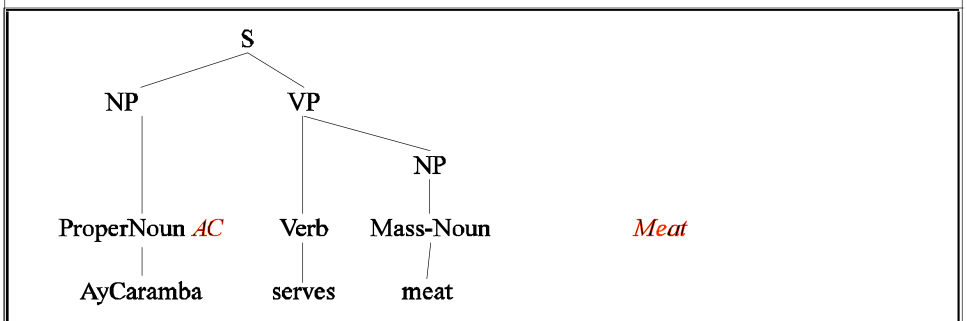
$$P(Sally)$$



# Example

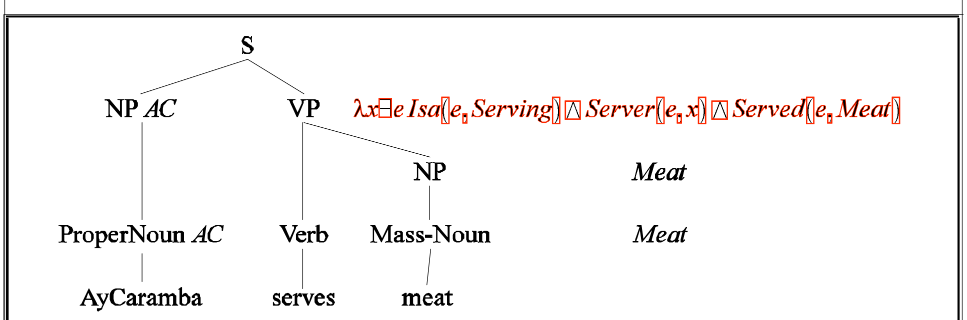


# Example

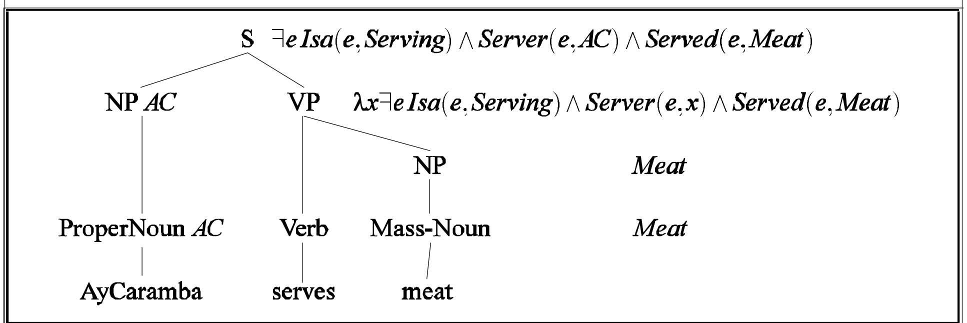





# Example



# Example






# Break

- **Read Chapters 16 and 17 (to be posted real soon now).**
- **Schedule**
  - **Next time lexical semantics**
  - **Then we'll cover information extraction, discourse, IR/QA and then MT.**

## Syntax/Semantics Interface: Two Philosophies

1. Let the syntax do what syntax does well and don't expect it to know much about meaning
   - In this approach, the lexical entry's semantic attachments do all the work
2. Assume the syntax does know something about meaning
   - Here the grammar gets complicated and the lexicon simpler (constructional approach)



## Example

- Mary freebled John the nim.
  - Who has it?
  - Where did he get it from?
  - Why?

# Example

- Consider the attachments for the VPs

  VP -> Verb NP NP rule (gave Mary a book)

  VP -> Verb NP PP (gave a book to Mary)

  Assume the meaning representations should be the same for both. Under the lexicon-heavy scheme, the VP attachments are:

  VP.Sem(NP.Sem, NP.Sem)

  VP.Sem(NP.Sem, PP.Sem)



# Example

- Under a syntax-heavy scheme we might want to do something like
- VP -> V NP NP

  V.sem ^ Recip(NP1.sem) ^ Object(NP2.sem)
- VP -> V NP PP

  V.Sem ^ Recip(PP.Sem) ^ Object(NP1.sem)
- I.e the verb only contributes the predicate, the grammar "knows" the roles.

## Integration

- **Two basic approaches**
  - **Integrate semantic analysis into the parser (assign meaning representations as constituents are completed)**
  - **Pipeline… assign meaning representations to complete trees only after they're completed**



## Example

- **From BERP**
  - **I want to eat someplace near campus**
- **Two parse trees, two meanings**

## Pros and Cons

- If you integrate semantic analysis into the parser as it is running…
  - You can use semantic constraints to cut off parses that make no sense
  - But you assign meaning representations to constituents that don't take part in the correct (most probable) parse



## Mismatches

- There are unfortunately some annoying mismatches between the syntax of FOPC and the syntax provided by our grammars…
- So we'll accept that we can't always directly create valid logical forms in a strictly compositional way
  - We'll get as close as we can and patch things up after the fact.

## Quantified Phrases

- Consider
  - A restaurant serves meat.
- Assume that A restaurant looks like

$$\exists x\, Isa(x, Restaurant)$$

- If we do the normal lambda thing we get

$$\exists e Serving(e) \wedge Server(e, \exists x Isa(x, Restaurant)) \wedge Served(e, Meat))$$



## Complex Terms

- Allow the compositional system to pass around representations like the following as objects with parts:

`Complex-Term` → `<Quantifier var body>`

$$< \exists\, x\ Isa(x, \text{Restaurant}) >$$

## Example

- **Our restaurant example winds up looking like**

$$\exists e Serving(e) \wedge Server(e, <\exists x Isa(x, Restaurant)>) \wedge Served(e, Meat)$$

- **Big improvement…**



## Conversion

- **So… complex terms wind up being embedded inside predicates. So pull them out and redistribute the parts in the right way…**

**P(<quantifier, var, body>)**
**turns into**
**Quantifier var body connective P(var)**

# Example

$$Server(e, < \exists x\ Isa(x, \text{Restaurant}) >)$$
$$\Rightarrow$$
$$\exists x\ Isa(x, \text{Restaurant}) \wedge Server(e, x)$$



# Quantifiers and Connectives

- **If the quantifier is an existential, then the connective is an ^ (and)**
- **If the quantifier is a universal, then the connective is an -> (implies)**

# Multiple Complex Terms

- Note that the conversion technique pulls the quantifiers out to the front of the logical form…
- That leads to ambiguity if there's more than one complex term in a sentence.



# Quantifier Ambiguity

- Consider
  - Every restaurant has a menu

  - That could mean that
    every restaurant has a menu
  - Or that
    There's some uber-menu out there and all restaurants have that menu

## Quantifier Scope Ambiguity

$$\forall x Restaurant(x) \Rightarrow$$
$$\exists e, y Having(e) \wedge Haver(e,x) \wedge Had(e,y) \wedge Isa(y, Menu)$$

$$\exists y Isa(y, Menu) \wedge \forall x Isa(x, Restaurant) \Rightarrow$$
$$\exists e Having(e) \wedge Haver(e,x) \wedge Had(e,y)$$



## Ambiguity

- **This turns out to be a lot like the prepositional phrase attachment problem**
- **The number of possible interpretations goes up exponentially with the number of complex terms in the sentence**
- **The best we can do is to come up with weak methods to prefer one interpretation over another**

## Non-Compositionality

- **Unfortunately, there are lots of examples where the meaning (loosely defined) can't be derived from the meanings of the parts**
  - **Idioms, jokes, irony, sarcasm, metaphor, metonymy, indirect requests, etc**



## English Idioms

- **Kick the bucket, buy the farm, bite the bullet, run the show, bury the hatchet, etc…**
- **Lots of these… constructions where the meaning of the whole is either**
  - **Totally unrelated to the meanings of the parts (kick the bucket)**
  - **Related in some opaque way (run the show)**

## The Tip of the Iceberg

- **Describe this construction**
  1. **A fixed phrase with a particular meaning**
  2. **A syntactically and lexically flexible phrase with a particular meaning**
  3. **A syntactically and lexically flexible phrase with a partially compositional meaning**
  4. **…**



## Example

- **Enron is the tip of the iceberg.**
  **NP -> "the tip of the iceberg"**
- **Not so good… attested examples…**
  - **the tip of Mrs. Ford's iceberg**
  - **the tip of a 1000-page iceberg**
  - **the merest tip of the iceberg**
- **How about**
  - **That's just the iceberg's tip.**

# Example

- **What we seem to need is something like**
- **NP ->**

  **An initial NP with tip as its head followed by a subsequent PP with of as its head and that has iceberg as the head of its NP**

  **And that allows modifiers like merest, Mrs. Ford, and 1000-page to modify the relevant semantic forms**