Semantic Analysis

By:
B.Senthil Kumar
Asst. Prof / CSE
Natural Language Processing

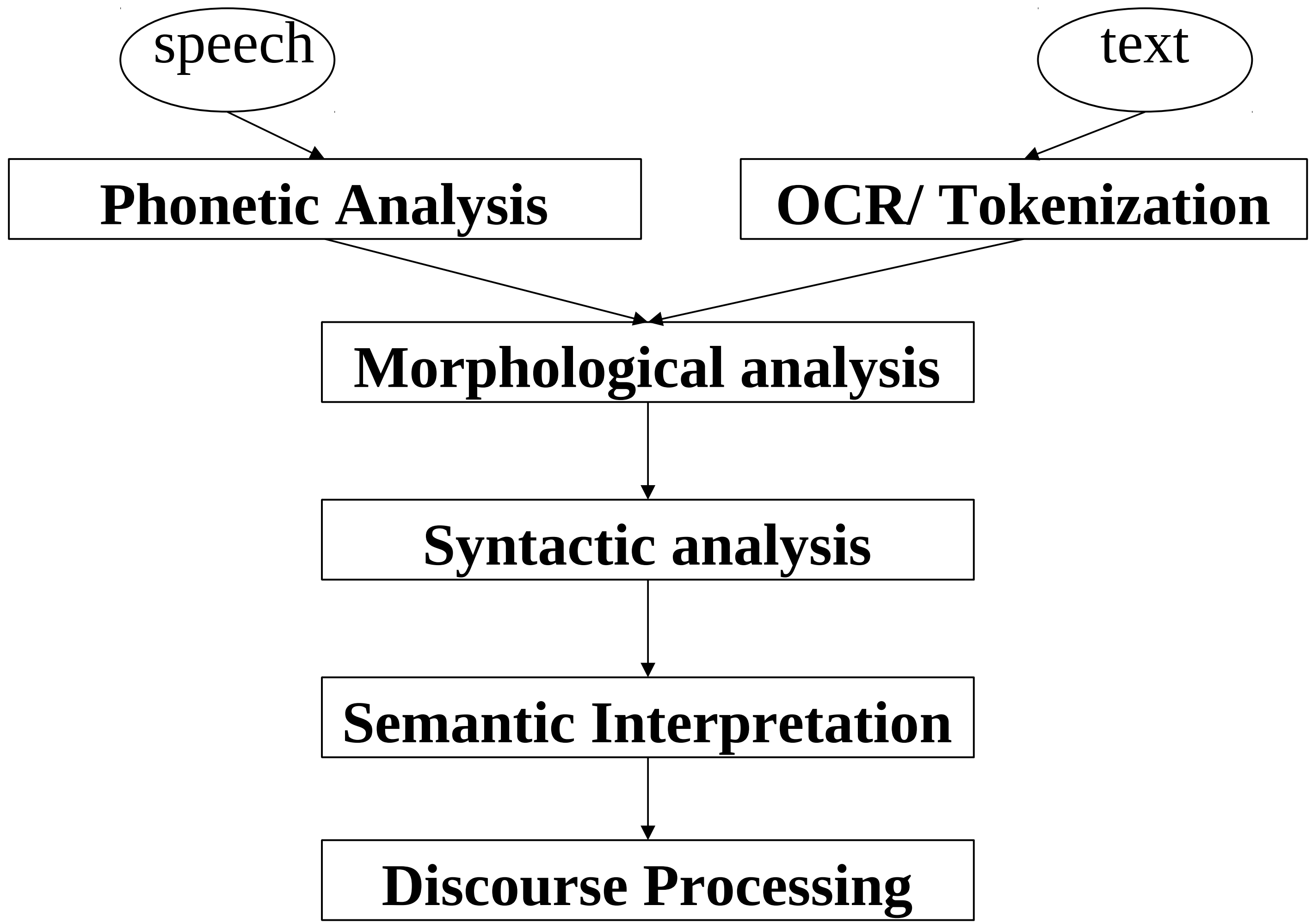
Slides from:

* Speech and Language Processing, Jurafsky and Martin
* Husni Al-Muhtaseb
* Ching-Long Yeh
* Heshaam Feili

Agenda

- Introduction
- Syntax-Driven Semantic Analysis
- Attachments for a Fragment of English
- Syntax-driven Analyser

NLP Pipeline



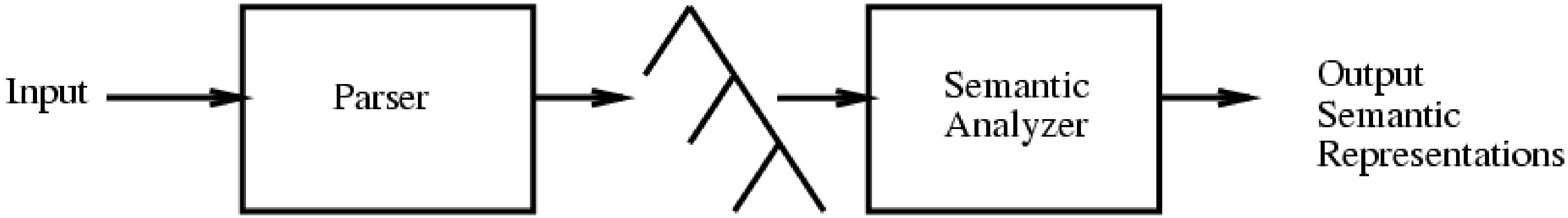
Introduction

- **Semantic analysis**
  - The process whereby meaning representations are composed and assigned to linguistic inputs.
- Among the source of knowledge typically used are
  - the meanings of words,
  - the meanings associated with grammatical structures,
  - knowledge about the structure of the discourse,
  - Knowledge about the context in which the discourse is occurring, and
  - Common-sense knowledge about the topic at hand.
- **Syntax-driven semantic analysis**
  - Assigning meaning representations to input based solely on static knowledge from the lexicon and the grammar.

Syntax-Driven Semantic Analysis

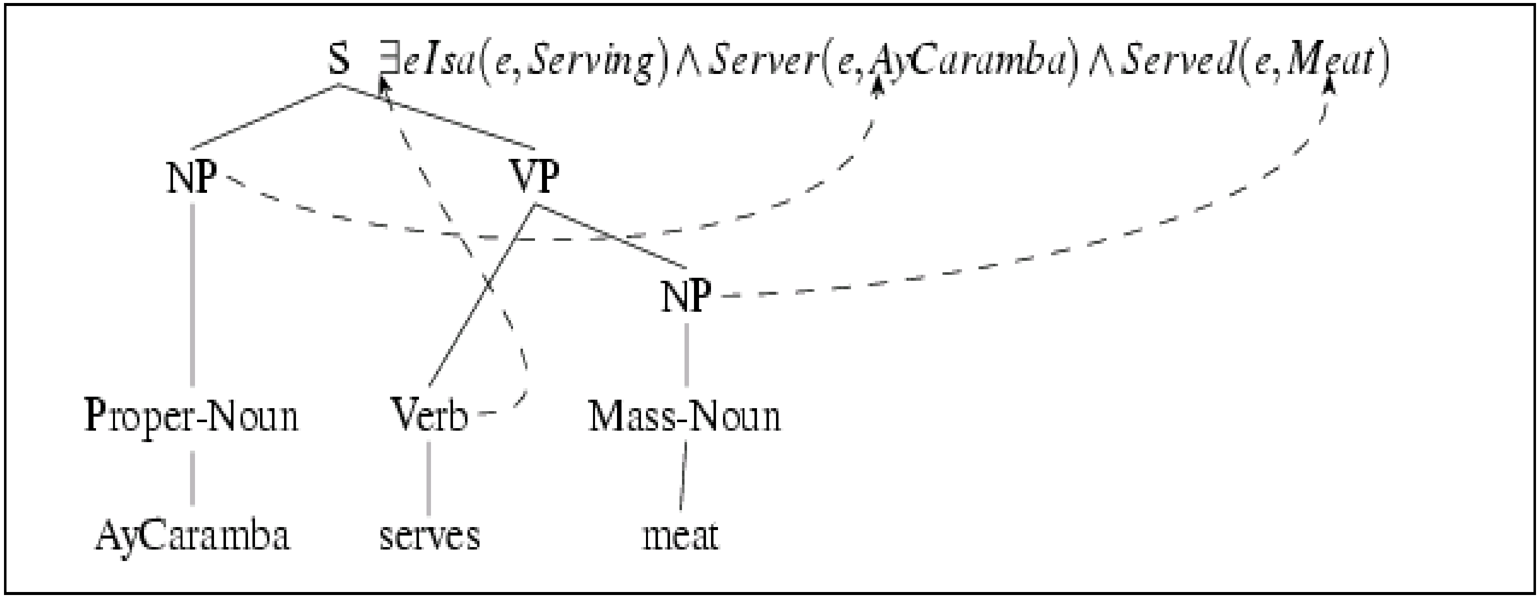
- **Principle of compositionality**
  - The meaning of a sentence can be composed from the meanings of its parts.
- Meaning of a sentence is not based solely on the words, but also on:
  - the ordering and grouping of words, and
  - the relations among the words in the sentence
- Hence the meaning of a sentence is partially based on its syntactic structure
- The composition of meaning representations is guided by the syntactic *components* and *relations* provided by the grammars discussed previously.

Syntax-Driven Semantic Analysis

- 

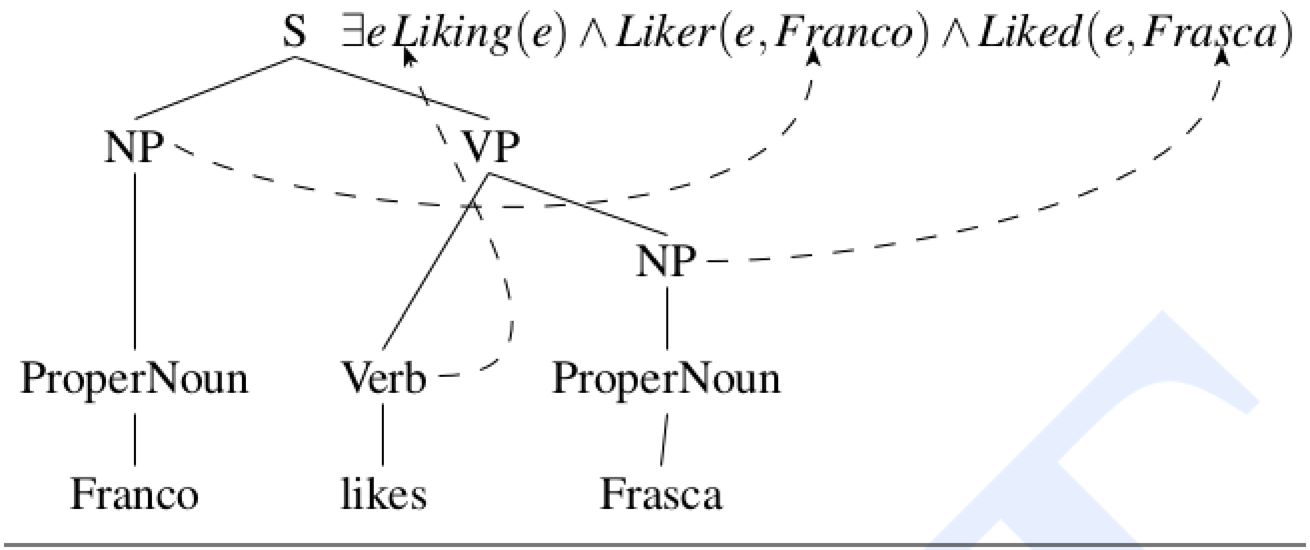
- Assumption:
    - Do not resolve the ambiguities arising from the previous stages.

Syntax-Driven Semantic Analysis



- What does it mean for syntactic constituents to have meaning?
- What do these meanings have to be like so that they can be composed into larger meanings?

Syntax-Driven Semantic Analysis



8.2 Parse tree for the sentence *Franco likes Frasca.*

- What does it mean for syntactic constituents to have meaning?
- What do these meanings have to be like so that they can be composed into larger meanings?

Semantic Augmentation to CFG Rules

- The concrete entities are represented by FOPC constants:

$ProperNoun \rightarrow AyCaramba$ $\{AyCaramba\}$

$MassNoun \rightarrow meat$ $\{Meat\}$

- These two specify that *the meanings associated* with these rules consist of the constants *AyCaramba* and *Meat.*
- Upper *NP*s obtain their meaning representations from the meanings of their children (*principle of compositionality*)

$NP \rightarrow ProperNoun$ $\{ProperNoun.sem\}$

$NP \rightarrow MassNoun$ $\{MassNoun.sem\}$

- The meaning representation of *NP* are the same as the meaning representations of their individual components, *ProperNoun.sem MassNoun.sem*

Semantic Augmentation to CFG Rules

- A generic *Serving* event involves a *Server* and something *Served*, as captured in following logical formula:

  $$Verb \rightarrow serves \; \{ \exists e,\mathrm{x},y \; Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, y) \}$$

- Moving up parse tree, *VP* dominates both *serves* and *meat.*
- *Incorporate* the meaning of *NP* into the meaning of the *Verb* and assign the resulting representation to the *VP.sem.*
- Replacing variable *y* with logical term *Meat* as the second argument of the *Served* role of the *Serves* event.

  $$Verb \rightarrow serves \; \{ \exists e,\mathrm{x}, \; Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, Meat) \}$$

Semantic Augmentation to CFG Rules

- To move *Verb* upwards to *VP*, the *VP* semantic attachment must have two capabilities:
    - The means to *know* exactly which variables within the *Verb*'s semantic attachment are to be replaced by the semantics of the *Verb*'s arguments, and
    - The ability to perform such a replacement.
- FOPC does not provide any advice about when and how such things are done.
    - **Lambda notation** extends the syntax of FOPC to include expression:

      $$\lambda x P(x)$$

      x: variable, $P(x)$: FOPC expression using the variable x
    - λ-reduction – Textual replacement of the λ variables with the specified FOPC terms, accompanied by the subsequent removal of the λ.

Semantic Augmentation to CFG Rules

$\lambda x P(x)(A)$ (λ-reduction)

$\Rightarrow P(A)$ - variable x is replaced with constant A in FOPC term

$\lambda x \lambda y Near(x, y)$

$\lambda x \lambda y Near(x, y)(ICSI)$

$\Rightarrow \lambda y Near(ICSI, y)$ by λ-reduction

$\lambda y Near(ICSI, y)(AyCaramba)$

$\Rightarrow Near(ICSI, AyCaramba)$ by λ-reduction

- **Currying**
  - A way of converting a predicate with multiple arguments into a sequence of single argument predicates.

Semantic Augmentation to CFG Rules

– With λ-notation

$Verb \rightarrow serves\ \{\lambda x\ \lambda y\ \exists e\ Isa(e, Serving) \wedge Server(e, y) \wedge Served(e, x)\}$

– *VP* rule specifies, λ-expression is provided by *Verb.sem* and the argument is provided by *NP.sem*.

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

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

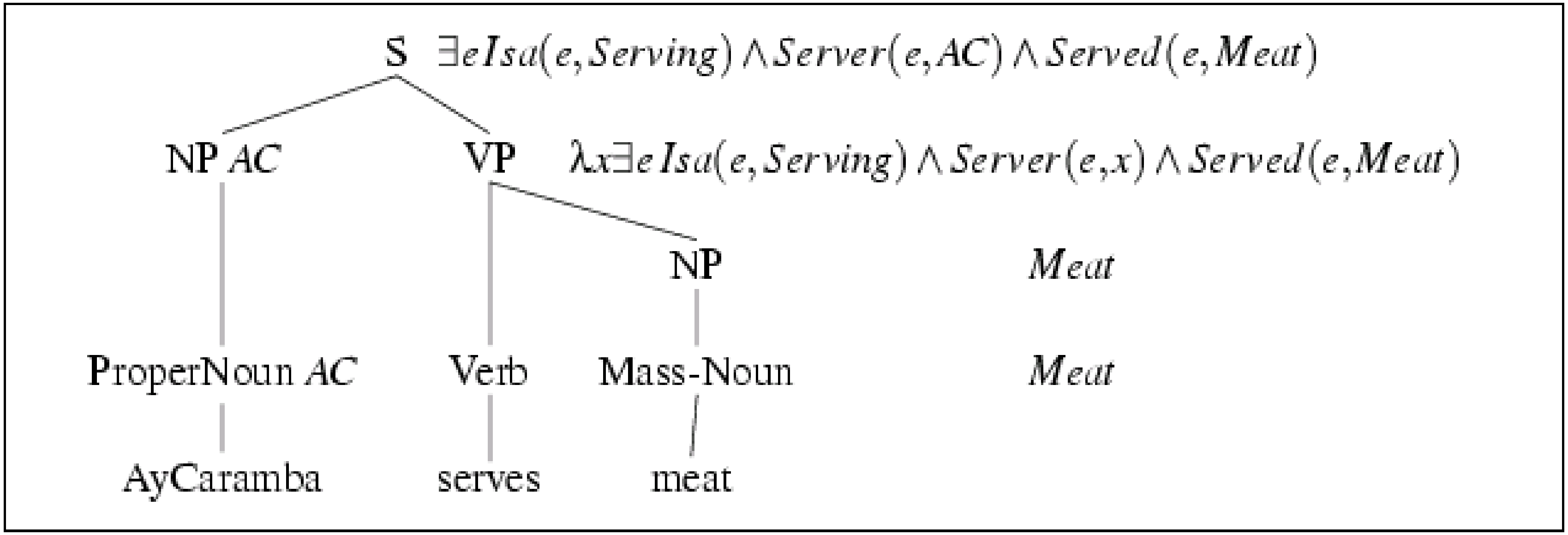
$\Rightarrow \lambda y\ \exists e\ Isa(e, Serving) \wedge Server(e, y) \wedge Served(e, Meat)$

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

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

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

Semantic Augmentation to CFG Rules



Semantic Augmentation to CFG Rules

- (15.1) *A restaurant serves meat.*
- Meaning representation for the phrase *a restaurant:*

$Det \rightarrow a \quad \{\exists\}$

$Noun \rightarrow restaurant \ \{Restaurant\}$

$Nominal \rightarrow Noun \ \{\lambda x \ Isa(x, Noun.sem)\}$

$NP \rightarrow Det \ Nominal \ \{<Det.sem \ x \ Nominal.sem(x)>\}$

$Noun{:} \quad Restaurant$

$Nominal{:} \ \lambda x \ Isa(x, Restaurant)$

NP: $<\exists \ x \ Isa(x, Restaurant)>$

- $VP \rightarrow Verb \ NP \ \{Verb.sem(NP.sem)\}$

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

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

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

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

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

Semantic Augmentation to CFG Rules

- (15.1) *A restaurant serves meat*.
- $\exists e\ Isa(e, Serving) \wedge Server(e, <\exists x\ \mathbf{Isa(x, Restaurant)}>) \wedge Served(e, Meat)$

  Not a valid FOPC
- Solve this problem by introducing the notion of a **complex-term**
- A complex term: < *Quantifier variable body* >
- Rewriting a predicate using a complex-terms

  $P(<Quantifier\ variable\ body>) \Rightarrow$

  $Quantifier\ variable\ body\ Connective\ P(variable)$

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

Semantic Augmentation to CFG Rules

– We have introduced five concrete mechanisms that instantiate the abstract functional characterization of semantic attachments
  - The association of normal FOPC expressions with lexical items
  - The association of function-like λ-expressions with lexical items
  - The copying of semantic values from children to parents
  - The function-like application of λ-expressions to the semantics of one or more children of a constituent
  - The use of complex-terms to allow quantified expressions to be temporarily treated as terms

– **Quasi-logical forms** or **intermediate representations**
  - The use of λ-expressions and complex-terms motivated by the gap between the syntax of FOPC and the syntax of English

Semantic Augmentation to CFG Rules

Grammar Rule	Semantic Attachment
$S \rightarrow NP\ VP$	$\{NP.sem(VP.sem)\}$
$NP \rightarrow Det\ Nominal$	$\{Det.sem(Nominal.sem)\}$
$NP \rightarrow ProperNoun$	$\{ProperNoun.sem\}$
$Nominal \rightarrow Noun$	$\{Noun.sem\}$
$VP \rightarrow Verb$	$\{Verb.sem\}$
$VP \rightarrow Verb\ NP$	$\{Verb.sem(NP.sem)\}$
$Det \rightarrow every$	$\{\lambda P.\lambda Q.\forall x P(x) \Rightarrow Q(x)\}$
$Det \rightarrow a$	$\{\lambda P.\lambda Q.\exists x P(x) \wedge Q(x)\}$
$Noun \rightarrow restaurant$	$\{\lambda r.Restaurant(r)\}$
$ProperNoun \rightarrow Matthew$	$\{\lambda m.m(Matthew)\}$
$ProperNoun \rightarrow Franco$	$\{\lambda f.f(Franco)\}$
$ProperNoun \rightarrow Franco$	$\{\lambda f.f(Frasca)\}$
$Verb \rightarrow closed$	$\{\lambda x.\exists e Closing(e) \wedge Closed(e,x)\}$
$Verb \rightarrow opened$	$\{\lambda w.\lambda z.w(\lambda x.\exists e Opening(e) \wedge Opener(e,z)$ $\wedge Opened(e,x))$

Figure 18.3 Semantic attachments for a fragment of our English grammar and lexicon.

Semantic Augmentation to CFG Rules

- *Every restaurant closed.*
- λ–Calculus permit λ–variables to range over FOL predicates as well as terms

  $\lambda Q \forall .x Restaurant(x) \Rightarrow Q(x)$
- $NP \rightarrow Det\ Nominal \quad \{Det.Sem(Nominal.sem)\}$

  $\lambda P.\lambda Q.\forall x P(x) \Rightarrow Q(x)\ (\lambda x.Restaurant(x))$

  $\lambda Q.\forall x \lambda x.Restaurant(x)(x) \Rightarrow Q(x)$

  $\lambda Q.\forall x\ Restaurant(x) \Rightarrow Q(x)$
- $S \rightarrow NP\ VP\ \{NP.sem(VP.sem)\}$

  $Verb \rightarrow close\ \{\lambda x.\ \exists e\ Closing(e) \wedge Closed(e, x)\}$

Semantic Augmentation to CFG Rules

- *Every restaurant closed.*
- $S \rightarrow NP\ VP$ {NP.*sem*(VP.*sem*)}

  $Verb \rightarrow close\ \{\lambda x.\ \exists e\ Closing(e) \wedge Closed(e, x)\}$

  $\lambda Q.\forall x\ Restaurant(x) \Rightarrow Q(x)\ (\lambda y.\exists e Closing(e) \wedge Closed(e, y))$

  $\forall x\ Restaurant(x) \Rightarrow \lambda y.\exists e Closing(e) \wedge Closed(e, y)\ (x)$

  $\forall x\ Restaurant(x) \Rightarrow \exists e Closing(e) \wedge Closed(e, x)$

Try out λ-reduction for the following sentence:

Matthew opened a restaurant.

Answer:

$\exists x Restaurant(x) \wedge \exists e Opening(e) \wedge Opener(e, Matthew) \wedge Opened(e, \mathrm{x})$

Quantifier Scoping and the Translation of Complex-Terms

- (15.3) Every restaurant has a menu.

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

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

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

- The problem of ambiguous **quantifier scoping** — a single logical formula with two complex-terms give rise to two distinct and incompatible FOPC representations.

Attachments for a Fragment of English

- Semantic attachments for a small fragment of English:
  - Sentences
  - Noun Phrases
  - Adjective Phrases
  - Verb Phrases
  - Prepositional Phrases

Sentences

- (15.4) Flight 487 serves lunch.

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

- (15.5) Serve lunch. [begin with a verb phrase and missing subject]

  $S \rightarrow VP \quad \{IMP(VP.sem(DummyYou))$

  $IMP(\exists e Serving(e) \wedge Server(e, DummyYou) \wedge Served(e, Lunch)$

  Imperatives can be viewed as a kind of **speech act**.

- (15.6) Does Flight 207 serve lunch?

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

  $YNQ(\exists e Serving(e) \wedge Server(e, Flt207) \wedge Served(e, Lunch)$

Sentences

- (15.7) Which flights serve lunch?

  $S \rightarrow WhWord\ NP\ VP \quad \{WHQ(NP.sem.var, VP.sem(NP.sem))\}$

  Representation consists of variable corresponding to the subject of the sentence and the body of the preposition.

  $$WHQ(x, \exists e,x\ Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, Lunch) \wedge Isa(x, Flight))$$

- (15.8) How can I go from Minneapolis to Long Beach?

  $S \rightarrow WhWord\ Aux\ NP\ VP\ \{WHQ(WhWord.sem, VP.sem(NP.sem))\}$

  Indicate the question in representation:

  $$WHQ(How, \exists e\ Isa(e, Going) \wedge Goer(e, User) \wedge Origin(e, Minn) \wedge Destination(e, LongBeach))$$

Noun Phrases

- The meaning representations for NPs can be either normal FOPC terms or complex-terms.
- Compound nominals or noun-noun sequences: final noun is the head of the phrase and denotes an object. It is semantically related to other nouns.
- (15.9) Flight schedule
- (15.10) Summer flight schedule

$Nominal \rightarrow Noun$

$Nominal \rightarrow Nominal\ Noun\ \{\lambda x\ Nominal.sem(x) \wedge NN(Noun.sem, x)\}$

NN – relation between elements of compound nominal and head noun.

$\lambda x\ Isa(x, Schedule) \wedge NN(x, Flight)$

$\lambda x\ Isa(x, Schedule) \wedge NN(x, Flight) \wedge NN(x, Summer)$

Noun Phrases

- Genitive noun phrases – *NP* with possessive markers.
- (Ex.) Atlanta’s airport
- (Ex.) Maharani’s menu
- With compound nominals, best to simply state an abstract semantic relation between various constituents
- $NP \rightarrow ComplexDet\ Nominal$

$$\{<\exists x Nominal.sem(x) \wedge GN(x, ComplexDet.sem)>\}$$

$ComplexDet \rightarrow NP\text{'}s\ \{NP.sem\}$

$$<\exists x\ Isa(x, Airport) \wedge GN(x, Atlanta)>$$

Adjective Phrases

- (15.11) I don’t mind a cheap restaurant.
- (15.12) This restaurant is cheap.
- For pre-nominal case, an obvious and *often incorrect* proposal is:

  $Nominal \rightarrow Adj\ Nominal$

  $$\{\lambda x\ Nominal.sem(x) \wedge Isa(x, Adj.sem)\}$$

  $Adj \rightarrow cheap\ \{Cheap\}$

  $\lambda x\ Isa(x, Restaurant) \wedge Isa(x, Cheap)$ **intersective semantics**
- ***Wrong***
  - *small elephant* $\Rightarrow \lambda x\ Isa(x, Elephant) \wedge Isa(x, Small)$
  - *former friend* $\Rightarrow \lambda x\ Isa(x, Friend) \wedge Isa(x, Former)$
  - *fake gun* $\Rightarrow \lambda x\ Isa(x, Gun) \wedge Isa(x, Fake)$

Incorrect interactions

Adjective Phrases

– The best approach:

- Simply note the status of a special kind of modification relation and
- Assume that some further procedure with access to additional relevant knowledge can replace this vague relation with an appropriate representation.

$$Nominal \rightarrow Adj\ Nominal \quad \{\lambda x\ Nominal.sem(x) \wedge AM(x, Adj.sem)\}$$

$$Adj \rightarrow cheap \quad \{Cheap\}$$

$$\exists x\ Isa(x, Restaurant) \wedge AM(x, Cheap)$$

Verb Phrases

- In general, the λ-expression attached to the verb is simply applied to the semantic attachments of the verb's arguments.
- However, some special cases, for example, infinite VP.
- (15.13) I *told* Harry to go to Maharani.
  - The meaning representation could be:

    $\exists e, f, x\ Isa(e, Telling) \wedge Isa(f, Going)$

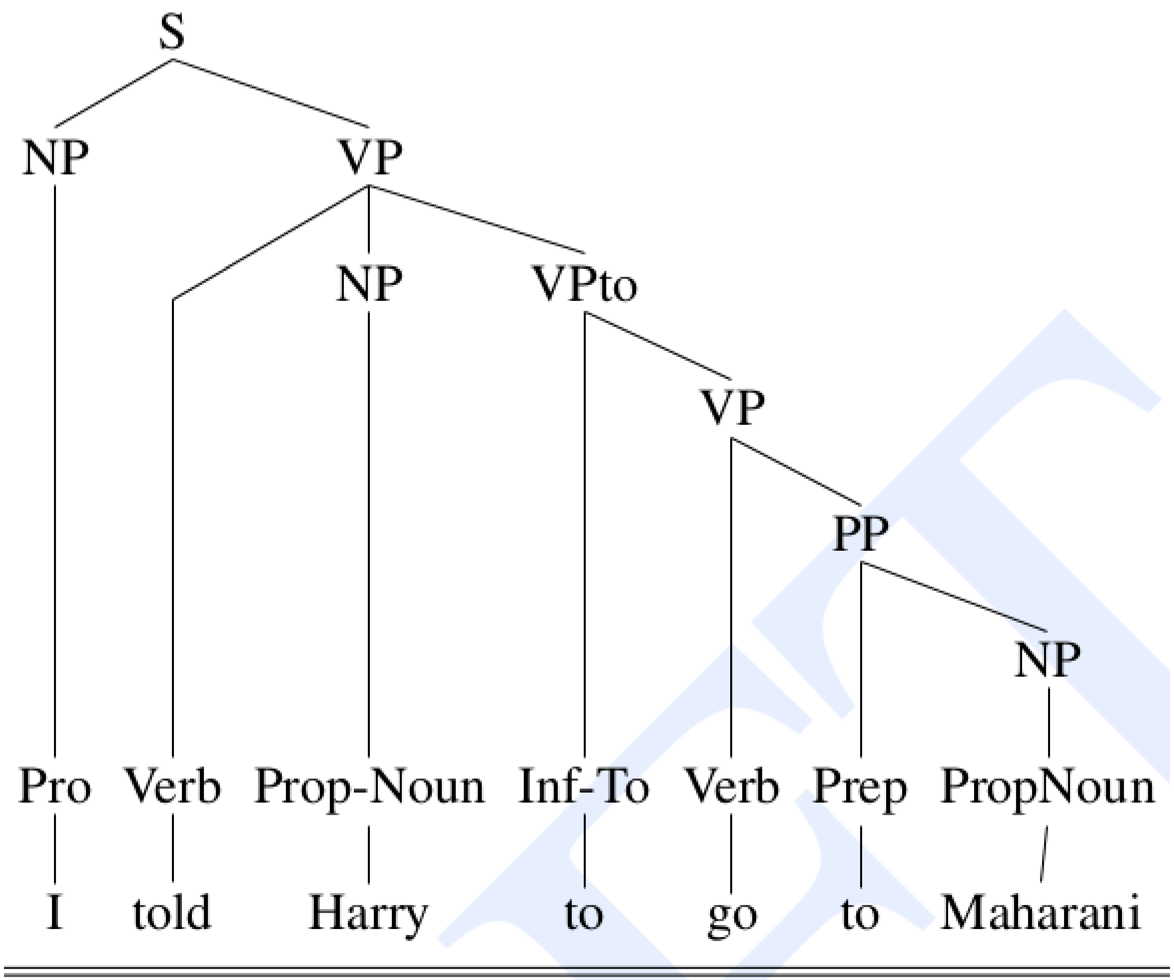
    $\wedge Teller(e, Speaker) \wedge Tellee(e, Harry) \wedge ToldThing(e, f)$

    $\wedge Goer(f, Harry) \wedge Destination(f, x)$
  - Two things to note:
    - It consists of two events [*Telling* and *Going*], and
    - One of the participants, *Harry*, plays a role in both of the two events

Verb Phrases

– Semantic attachment of the verb, *tell*

$\lambda x,y\ \lambda z\ \exists e\ Isa(e, Telling) \wedge Teller(e, z) \wedge Tellee(e, x) \wedge ToldThing(e,y)$



Parse tree for *I told Harry to go to Maharani.*

Verb Phrases

– Semantic attachment of the verb, *tell*

$\lambda x,y\ \lambda z\ \exists e\ Isa(e, Telling)$
$\wedge Teller(e, z) \wedge Tellee(e, x) \wedge ToldThing(e,y)$

- Providing three semantic roles:
  - – a person doing the telling,
  - – a recipient of the telling, and
  - – the proposition being conveyed
- **Problem:**
  - – *Harry* plays the role in both *Telling* and *Going* event
  - – But *Harry* is not available when the *Going* event is created within the infinite verb phrase.

Verb Phrases

– Solution:

$VP \rightarrow Verb\ \ NP\ \ VPto \quad \{Verb.sem(NP.sem, VPto.sem)\}$

$VPto \rightarrow to\ VP\ \{VP.sem\}$

$Verb \rightarrow tell \quad \{\lambda x,y\ \ \lambda z$

$\exists e,\ y.variable\ Isa(e, Telling) \wedge Teller(e, z) \wedge Tellee(e, x)$

$\wedge ToldThing(e, y.variable) \wedge y(x)\}$

- The λ-variable *x* plays the role of the *Tellee* of the telling and the argument to the semantics of the infinitive.
- The expression *y*(*x*) represents a λ-reduction that inserts *Harry* into the *Going* event as the *Goer*.
- The notation *y.variable*, gives us access to the event variable representing *Going* event within the infinitive's meaning representation.

Prepositional Phrases

- At an abstract level, PPs serve two functions:
  - They assert binary relations between their heads and the constituents to which they attached
  - They signal arguments to constituents that have an argument structure
- Consider three places in the grammar where PP serve these roles:
  - Modifiers of NPs
  - Modifiers of VPs, and
  - Arguments to VPs

Prepositional Phrases

self-study

Syntax-Driven Analyser

- Semantic analysis can also be performed in parallel with syntactic processing
- The integration of semantic analysis into an Earley parser:
  - The grammar rules contain their semantic attachments
  - The chart states includes a new field to hold the meaning representation of constituent
  - ENQUEUE is altered – when a complete state is entered into the chart its semantics are computed and stored in state's semantic field

Integrating Semantic Analysis into Earley Parser

procedure ENQUEUE(*state, chart-entry*)
  if INCOMPLETE?(*state*) **then**
      if *state* is not already in *chart-entry* **then**
          PUSH(*state, chart-entry*)
  else if UNIFY-STATE(*state*) succeeds **then**
      if APPLY-SEMANTICS(*state*) succeeds **then**
        if *state* is not already in *chart-entry* **then**
          PUSH(*state, chart-entry*)

procedure APPLY-SEMANTICS(*state*)
  meaning-rep ← APPLY(*state.semantic-attachment, state*)
  if *meaning-rep* does not equal **failure then**
    state.meaning-rep ← *meaning-rep*

If the state is complete and unification succeeds then call APPLY-SEMANTICS to compute and store meaning representation of completed states. Semantic analysis will be performed only on valid trees.

References

- Speech and Language Processing, *Jurafsky and H.Martin* [Chapter 15. Semantic Analysis]