Semantics and Logical Form

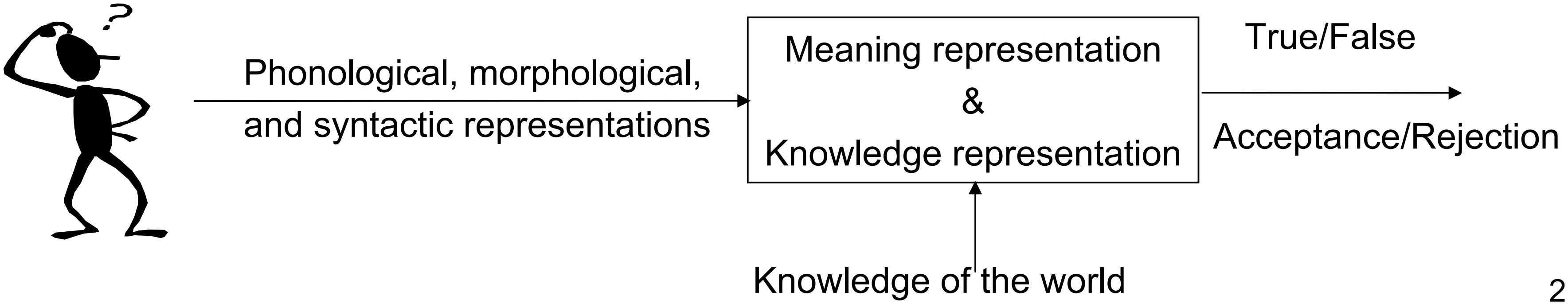
Berlin Chen 2003

References:

1. Speech and Language Processing, chapter 14
2. Natural Language Understanding, chapter 8
3. Jim Martin's lectures

Introduction

- Everyday language tasks
  - Answer an essay question on an exam
  - Decide what to order at a restaurant by reading a menu
  - Learn to use a new piece of software by reading the manual
  - Realize that you’ve been insulted
  - Follow a recipe



Phonological, morphological, and syntactic representations

Meaning representation & Knowledge representation

True/False

Acceptance/Rejection

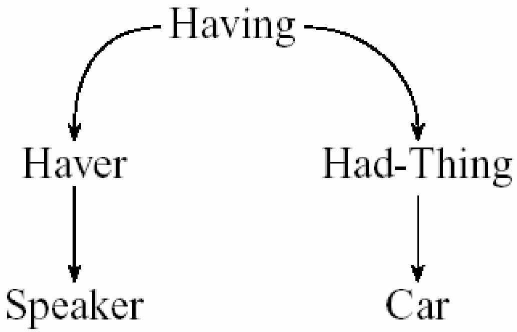
Knowledge of the world

Meaning Representations

- Example: “I have a car”

$$\exists x,y\, Having(x) \wedge Haver(Speaker,x) \wedge HadThing(y,x) \wedge Car(y)$$

First Order Predicate Calculus



Semantic Network

Conceptual Dependency Diagram

Car
⇑ POSS-BY
Speaker

Frame-based Representation

```
Having
 Haver:   Speaker
 HadThing:   Car
```

Semantics

- The study of the meaning of linguistic sentences
  - Meaning of morphemes
  - Meaning of words
  - Meaning of phrases

utterances

- Steps for determining the meaning of a sentence
  - Compute a context-independent notion of meaning in logical form (semantic interpretation)
  - Interpret the logical form in context to produce the final meaning representation (contextual interpretation)

The study of language in context is called pragmatics.

Issues

- Formal representations for capturing meaning
  - Meaning representation (languages)
  - E.g., First Order Predicate Calculus (FOPC), Semantic Network, Semantic Frames, …

- Algorithms for mapping from utterances to the meaning representations
  - E.g., compositional semantic analysis, semantic grammars, …

Desiderata for Meaning Representation

- **Verifiability**
  - Use meaning representation to determine the relationship between the meaning of a sentence and the world we know it
  - E.g., Query: “Does Maharani serve vegetarian food? ”

    Serves(*Maharani*, *VegetarianFood*)
  - The straightforward way
    - Make it possible for a system to compare, or match, the representation of meaning of an input against the representations (facts) in the KB

Desiderata for Meaning Representation

- **Unambiguous Representations**
  - Single linguistic inputs may have different meaning representations assigned to them based on the circumstances in which they occur
  - ambiguity cf. vagueness
    - It's not always easy to distinguish ambiguity from vagueness
    - E.g.,

child or goat

ambiguity I have two kids and George has three

vagueness I have one horse and George has two

mare, colt, trotter

Desiderata for Meaning Representation

- **Unambiguous Representations**
  - Ambiguity
    - Lexical (word sense) ambiguity
    - Syntactic (structural) ambiguity
    - Disambiguation
      - Structural information of the sentences
      - Word co-occurrence constraints
  - Vagueness
    - Make it difficult to determine what to do with a particular input based on it’s meaning representations
    - Some word senses are more specific than others

Desiderata for Meaning Representation

- **Canonical Form**
  - Inputs talking the same thing should have the same meaning representation
  - Dilemma in internal knowledge representations
    - If the knowledge based contain all possible alternative representations of the same fact
    - How to maintain consistence between various representations is a crucial problem
  - Example

Overheads on
KB maintenance or
semantic analysis

The input query
Using various
propositions

Does Maharani have vegetarian dish?
Does they have vegetarian food at Maharani?
Are vegetarian dishes served at Maharani?
Does Maharani serve vegetarian fare?

Desiderata for Meaning Representation

- **Canonical Form**
  - Assign the same meaning representation to various propositions for a query
    - Simplify the matching/reasoning tasks
    - But complicate the semantic analysis because of different words and syntax used in the propositions
      - vegetarian fare/dishes/food
      - having/serving
  - We can exploit the underlying systematic meaning relationships among word senses and among grammatical constructions to make this task tractable
    - E.g., choosing the shared sense among words

Desiderata for Meaning Representation

- **Inference and Variables**
  - Simple matching of knowledge base will not always give the appropriate answer to the request
    - E.g.: "*Can vegetarians eat at Maharani?*"
  - The system should has the ability to draw valid conclusions based on the meaning representation of inputs and the stored background knowledge
    - Determine the TRUE or FALSE of the input propositions
  - Such a process is called **inference**

Desiderata for Meaning Representation

- **Inference and Variables**
  - For the request without making reference to any particular object, involving the use of **variable** is needed, e.g.,

    I'd like to find a restaurant where I can get vegetarian food.

    Restaurant(*x*) ^ *Serves*(*x*, *VegetarianFood*)

  - Matching is successful only if the variable can be replaced by some known object in the KB such that the entire proposition is satisfied

Desiderata for Meaning Representation

- **Expressiveness**
  - The meaning representation scheme must be expressive enough to handle an extremely wide range of subject matter
  - That’s a ideal situation!

Predicate-Argument Structure

- All languages have a form of predicate-argument arrangement at the core of their semantic structure
- Predicate
  - Constants that describe events, actions, relationships and properties
- Argument
  - An appropriate number of terms serve as the arguments

Predicate-Argument Structure

- As we have seen before
  - In natural languages, some words and constituents function as predicates and some as arguments

    Verbs, VPs, PPs, …   Nouns, NPs, …

- **Example**

  I want Italian food. ⇨ *want*(*I*, *ItalianFood*)

  - "want" conveys a two-argument predicate
  - There are two arguments to this predicate
  - Both arguments must be NPs
  - The first argument ("I") is pre-verbal and plays the role of the subject
  - The second argument ("Italian food") is post-verbal and plays the role of direct object

Predicate-Argument Structure

- Verbs by no means the only objects in a grammar that can carry a predicate-argument structure
  - **Example1**: “prepositions”

    an Italian restaurant under fifteen dollars

    ⇨ *Under*(*ItalianRestaurant, $15*)
  - **Example2**: “Nouns”

    Make a reservation for this evening at 8

    ⇨ *Reservation*(*Hearer, Today, 8PM*)

First Order Predicate Calculus (FOPC)

- Also called First Order Logic (FOL)

- Make use of FOPC as the representational framework, because it is
  - Fexible, well-understood, and computational tractable
  - Produced directly from the syntactic structure of a sentence
  - Specify the sentence meaning without having to refer back natural language itself
  - Context-independency: does not contain the results of any analysis that requires interpretation of the sentences in context

Facilitate concise representations and semantics for sound reasoning procedures.

First Order Predicate Calculus (FOPC)

- FOPC allows
  - 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
      - Determine the truth of propositions that do not literally (exactly) present in the KB

Adopted From Jim Martin

Elements of FOPC

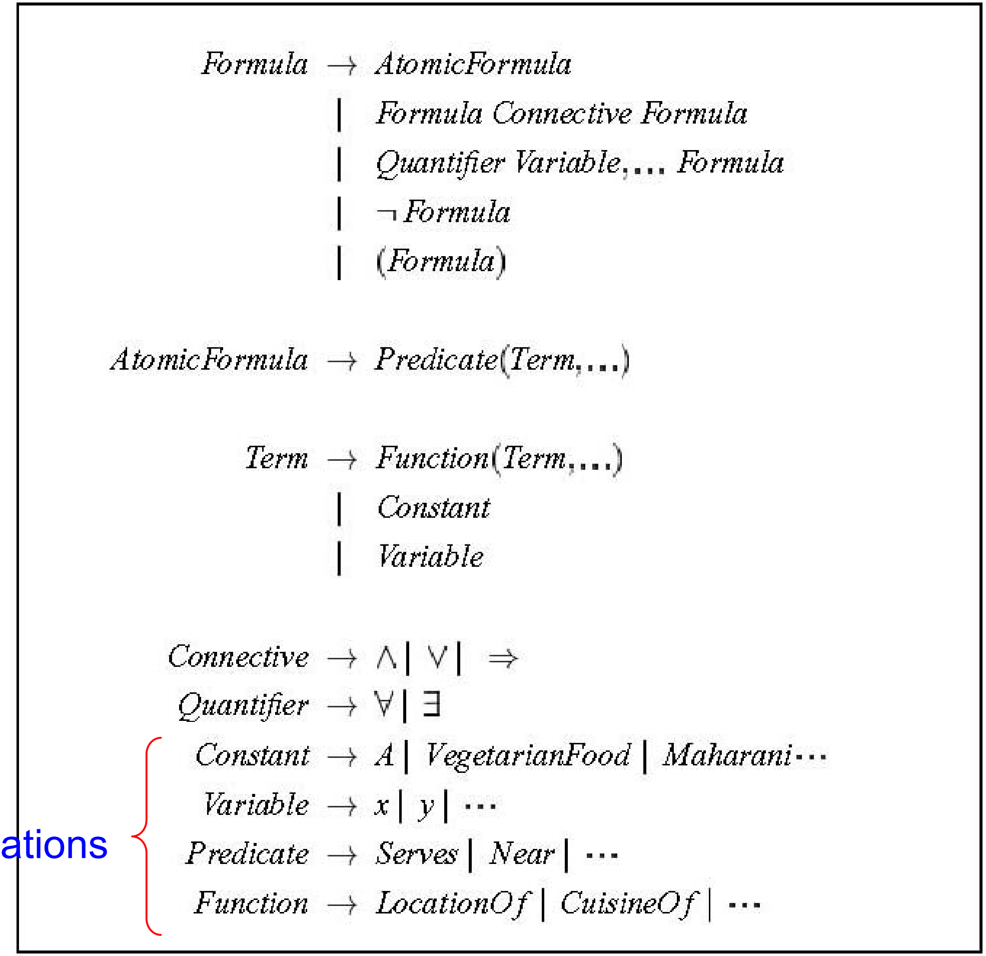
- **Terms**: the device for representing objects
  - **Variables**
    - Make assertions and draw references about objects without having to make reference to any particular named object (anonymous objects)
    - Depicted as single lower-case letters
  - **Constants**
    - Refer to specific objects in the world being described
    - Depicted as single capitalized letters or single capitalized words

Elements of FOPC

- **Terms**: (cont.)
  - **Functions**
    - Refer to unique objects without having to associate a name constant with them
    - Syntactically the same as single predicates
- **Predicates**:
  - Symbols refer to the **relations** holding among some fixed number of objects in a given domain
  - Or symbols refer to the **properties** of a single object
    - Encode the category membership
  - The arguments to a predicates must be terms, not other predicates

Elements of FOPC

- A CFG specification of the syntax of FOPC

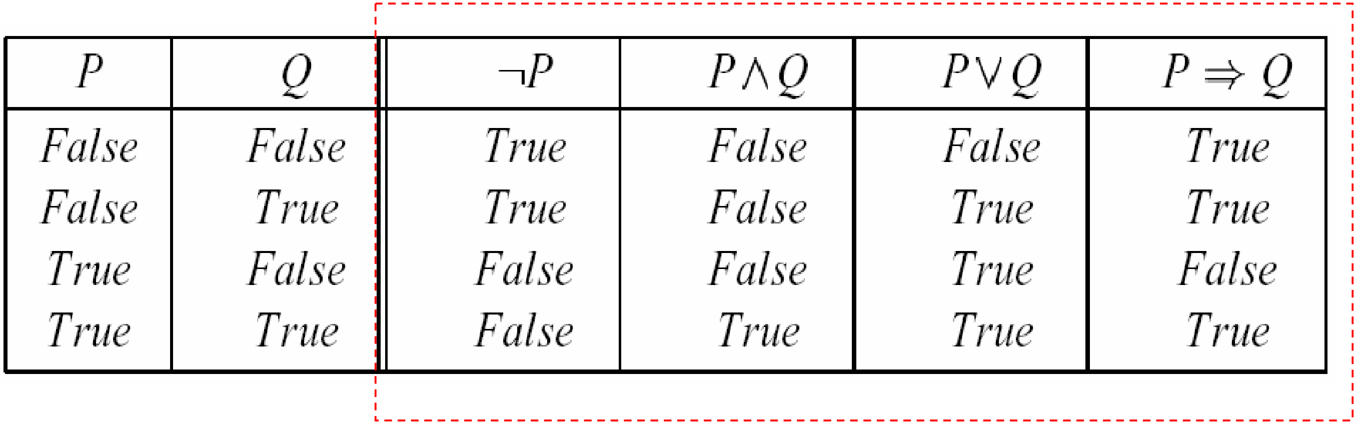


$$
\begin{aligned}
\mathit{Formula} &\rightarrow \mathit{AtomicFormula}\\
&\mid \mathit{Formula\ Connective\ Formula}\\
&\mid \mathit{Quantifier\ Variable}, \ldots\ \mathit{Formula}\\
&\mid \neg\,\mathit{Formula}\\
&\mid (\mathit{Formula})\\[1em]
\mathit{AtomicFormula} &\rightarrow \mathit{Predicate}(\mathit{Term}, \ldots)\\[1em]
\mathit{Term} &\rightarrow \mathit{Function}(\mathit{Term}, \ldots)\\
&\mid \mathit{Constant}\\
&\mid \mathit{Variable}\\[1em]
\mathit{Connective} &\rightarrow \wedge \mid \vee \mid\ \Rightarrow\\
\mathit{Quantifier} &\rightarrow \forall \mid \exists
\end{aligned}
$$

$$
\text{atomic representations}\left\{
\begin{aligned}
\mathit{Constant} &\rightarrow A \mid \mathit{VegetarianFood} \mid \mathit{Maharani} \cdots\\
\mathit{Variable} &\rightarrow x \mid y \mid \cdots\\
\mathit{Predicate} &\rightarrow \mathit{Serves} \mid \mathit{Near} \mid \cdots\\
\mathit{Function} &\rightarrow \mathit{LocationOf} \mid \mathit{CuisineOf} \mid \cdots
\end{aligned}
\right.
$$

Elements of FOPC

- **Logical Connectives**
  - The $\wedge$ (and), $\neg$ (or), $\vee$ (not), $\Rightarrow$ (imply) operators
  - 16 possible truth functional binary values



P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$
False	*False*	*True*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*	*True*
True	*False*	*False*	*False*	*True*	*False*
True	*True*	*False*	*True*	*True*	*True*

  - Used to form larger composite representations
  - Example

I only have five dollars and I don't have a lot of time

Have(*Speaker*, *FiveDollars*) $\wedge$ $\neg$ *Have*(*Speaker*, *LotOfTime*)

Elements of FOPC

- **Quantifiers**
  - The **existential** quantifier $\exists$
    - Pronounced as "there exists"
    - Example:

a restaurant that serves Mexican food near ICSI.

$$\exists x\ Restaurant(x) \wedge Serve(x, MexicanFood)$$
$$\wedge\ Near(LocationOf(x), LocationOf(ICSI))$$

  - The **universal** quantifier $\forall$
    - Pronounced as "for all"
    - Example: $\forall x\ VegetarianRestaurant(x) \wedge Serve(x, MexicanFood)$?

To satisfy the condition,
at least one substitution must result in truth

All vegetarian restaurant serve vegetarian food.

$$\forall x\ VegetarianRestaurant(x) \Rightarrow Serve(x, MexicanFood)$$

To satisfy the condition, all substitutions must result in truth

Inference

- The ability to add valid new propositions to a KB, or to determine the truth of propositions that are not literally (exactly) contained in the KB

- **modus ponens**
  - The most important inference method in FOPC
  - Known as "if-then"

$$\begin{array}{c} \alpha \\ \alpha \Rightarrow \beta \\ \hline \beta \end{array}$$

antecedent

consequent

The formula below the line can be inferred from the formulas above the line by some form of inference.

  - If the left-hand side of an implication rule is present in the KM, then the right-hand side can be inferred

Inference

- Example

$$\frac{\begin{array}{l} \mathit{Vegetarian\ Restaurant}(\mathit{Rudys}) \\ \forall x\ \mathit{VegetarianRestaurant}(x) \Rightarrow \mathit{Serve}(x, \mathit{MexicanFood}) \end{array}}{\mathit{Serve}(\mathit{Rudys}, \mathit{MexicanFood})}$$

a new fact

Inference

- Two ways of use
  - Forward chaining
    - Just as described previously
    - As individual facts are added into KB, modus ponens is used to fire all applicable implication rules
    - All inference is performed in advance
      - Advantage: answer subsequent queries using simple table lookup (*fast!*)
      - Disadvantage: store too much facts that will never be needed
    - Example: "**production systems**" in cognitive modeling work

Inference

- Two ways of use (cont.)
  - Backward chaining
    - Run in reverse to prove specific propositions, call the queries
    - First see if the queries is present in the KB
    - If not, search for applicable implications in KB, whose consequent matches the query formula
      - If there are such a rule, then the query can be proved if the antecedent of any one of them can be shown to be true
    - Recursively performed by backward chaining on the antecedent as a new query
    - Example: the **Prolog** is a backward chaining system

Inference

- Backward chaining (cont.)
  - Should distinguish between
    - Reasoning via backward chaining from queries to known facts
    - Reasoning backwards from known consequent to unknown antecedents

Representations of Important Topics

- Several issues should be considered in meaning representation of a few important topics
  - Categories
  - Events
  - Time
  - Aspect
  - Beliefs

Categories

- **Old representations**
  - Categories are commonly presented using unary predicates

$$VegetarianRestaurant(Maharani)$$

  - However, categories are relations, rather than objects
  - Difficult to make assertion about categories themselves

$$MostPopular(Maharani, VegetarianRestaurant)$$

is a predicate, not a term

  - Solution → **reification**
    - Represent categories as objects

Categories

- **New representations**
  - The new notation of membership in a category, or relation held **between objects and the categories** , e.g.,

$$ISA(Maharani, VegetarianRestaurant)$$

(is a)

  - Relation held **between categories**, e.g.,

$$AKO(VegetarianRestaurant, Restaurant)$$

(a kind of)

    - A category inclusion relationship

Events

- **Old representations**
  - Events are represented as single predicates with as many arguments as are needed, e.g.

1	I ate.	$Eating_1(Speaker)$
2	I ate a turkey sandwich.	$Eating_2(Speaker, TurkeySand\ wich)$
3	I ate a turkey sandwich at my desk.	$Eating_3(Speaker, TurkeySand\ wich, Desk)$
4	I ate at my desk.	$Eating_4(Speaker, Desk)$
5	I ate lunch.	$Eating_5(Speaker, Lunch)$
6	I ate a turkey sandwich for lunch.	$Eating_6(Speaker, TurkeySand\ wich, Lunch)$
7	I ate a turkey sandwich for lunch at my desk.	$Eating_7(Speaker, TurkeySand\ wich, Lunch, Desk)$

  - How can we make logic connections between these predicates

5
2
6 → 1

Events

- **New representations**
  - Solution → **reification**
    - **Represent events as objects** which can be quantified and related to other objects

① $$\exists w\ ISA(w, Eating) \wedge Eater(w, Speaker)$$

② $$\exists w\ ISA(w, Eating) \wedge Eater(w, Speaker) \wedge Eaten(w, TurkeySandwich)$$

⑥ $$\exists w\ ISA(w, Eating) \wedge Eater(w, Speaker) \\ \wedge Eaten(w, TurkeySandwich) \wedge MealEaten(w, Lunch)$$

  - Features
    - No need to specify a fixed number of arguments for a given surface predicate

Time

- Events are associated with either points or intervals in time, as on a time line
  - An ordering can be imposed on distinct events by situating them on the time line
  - **Ordering relationship**: past, present, future
- Representations **without temporal information**

$$\left.\begin{array}{l}\text{I arrived in New York.}\\ \text{I am arriving in New York.}\\ \text{I will arrive in New York.}\end{array}\right\} \begin{array}{l}\exists w\ ISA(w, Arriving) \wedge Arriver(w, Speaker)\\ \qquad \wedge Destinatio\ n(w, NewYork)\end{array}$$

Time

- Representations **with temporal information**

I arrived in New York.

$$\exists w\, ISA(w, Arriving) \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork) \\ \wedge IntervalOf(w, i) \wedge EndPoint(i, e) \wedge Precedes(e, Now)$$

I am arriving in New York.

$$\exists w\, ISA(w, Arriving) \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork) \\ \wedge IntervalOf(w, i) \wedge MemberOf(i, Now)$$

I will arrive in New York.

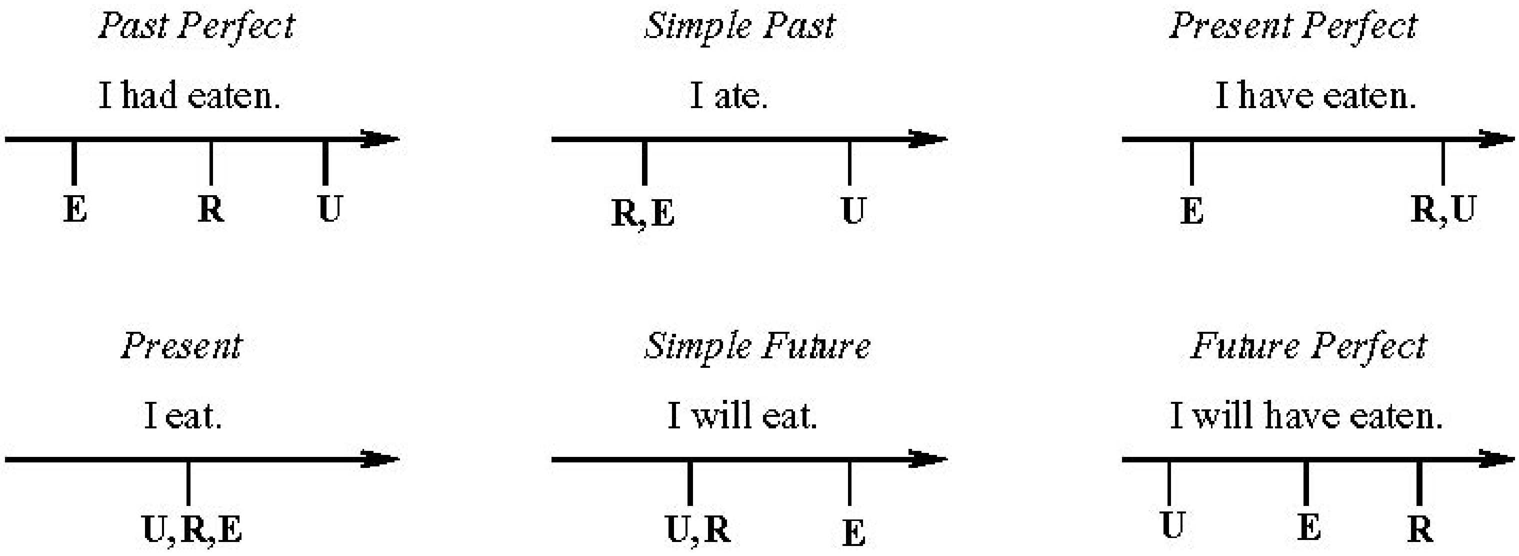
$$\exists w\, ISA(w, Arriving) \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork) \\ \wedge IntervalOf(w, i) \wedge EndPoint(i, e) \wedge Precedes(Now, e)$$

- However, the relation between verb tenses and points in time is by no means straightforward

Flight 1902 arrived late.

Flight 1902 had arrived late.

Time



E: the time of event

R: the reference time

U: the time of utterance

Aspects

- Aspect concerns a cluster of relative topics about events
  - **Stative**
    - The event participant has a particular property, or is in a state, at a given point in time
    - E.g.,

      I know my departure gate.
  - **Activity**
    - The event undertaken by a participant that has no particular end point
    - E.g.,

      John is flying.

      I live in Brooklyn **for** a month.

Aspects

- **Accomplishment**
  - The event has a natural end point and result in a particular state
  - E.g.,

    He booked me a reservation.

    She booked a flight **in** a minute.

    ..stopping booking ..
- **Achievement**
  - Though of as happening in an instant, also results in a state
  - E.g.,

    ..stopping reaching ..?

    She found her gate.

    I reached New York.

Beliefs

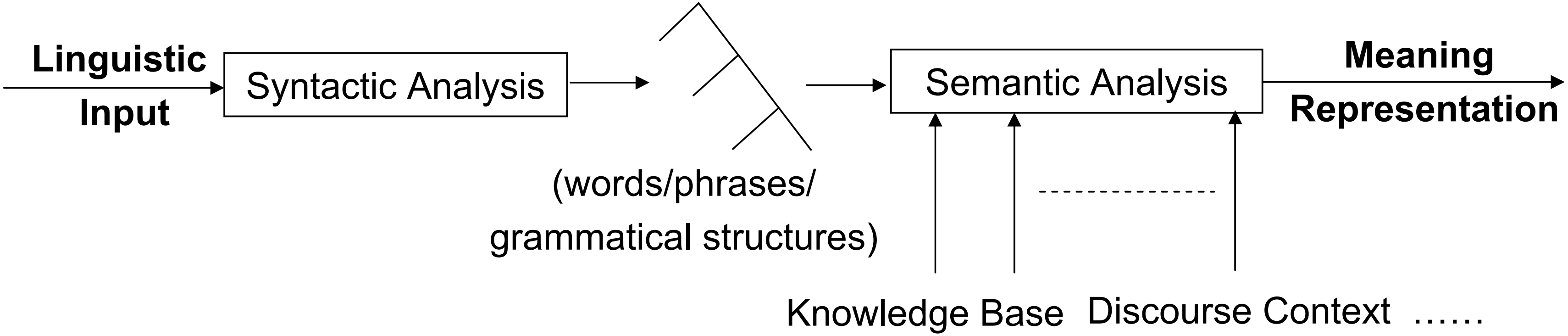
- Representations for some kind of hypothetical world
  - Denote a relation from the speaker, or some other entry, to this hypothetical world
  - Words have such an ability: believe, want, image, know… (take various sentence-like constituents as arguments)
    - E.g.,

      I believe that Mary ate British food.

$$\underset{\substack{\uparrow \\ \text{modal operator}}}{Believes}\left(Speaker, \exists v\ ISA(v, Eating) \wedge Eater(v, Marry) \wedge Eaten(v, BritishFood)\right)$$

Semantic Analysis

- The process of assigning a meaning representation to a linguistic input
  - A lot of ways to deal with it
  - Make more or less use of syntax



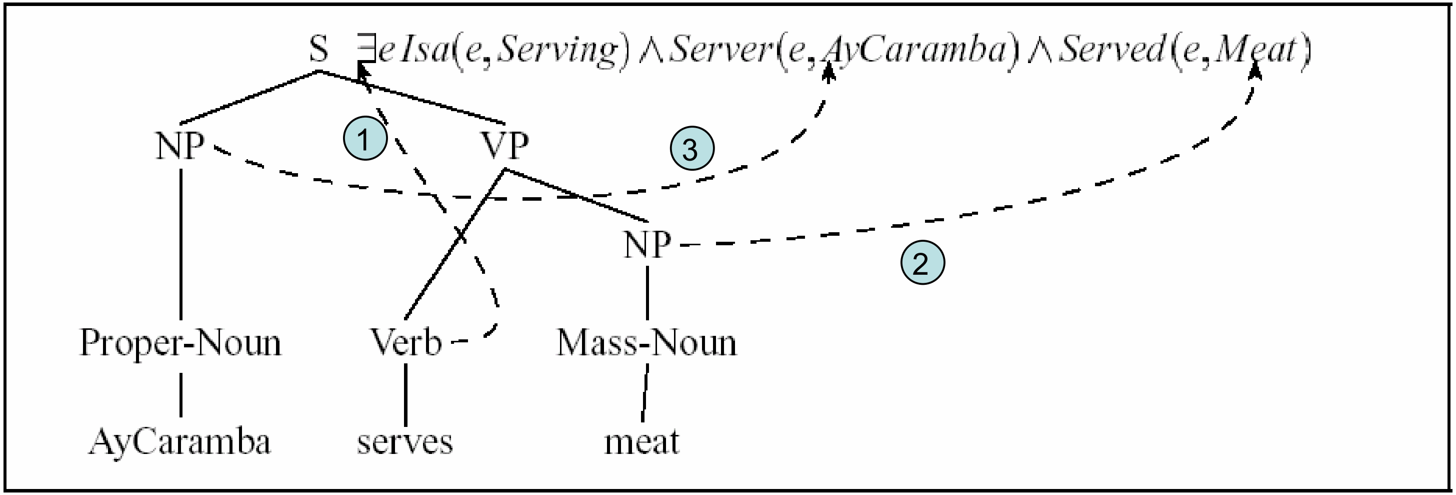
Compositional Analysis

- Principle of Compositionality
  - The meaning of a sentence/construction can be composed (derived) from the meanings of its parts
  - What parts?
    - The constituents of the syntactic parse of the linguistic input
    - Words → Phrases → Clauses ….

- Non-compositionality
  - There are lots of constructions whose meanings can't be derived from the meanings of their parts
  - E.g., idioms, metaphors, …

Syntax-Driven Semantic Analysis

- The meaning representation to the input utterance is solely based on static knowledge from the lexicon and the syntactic grammar


S $\exists e\, Isa(e, Serving) \wedge Server(e, AyCaramba) \wedge Served(e, Meat)$
NP
1
VP
3
NP
2
Proper-Noun
Verb
Mass-Noun
AyCaramba
serves
meat

Semantic Argumentations to CFG Rules

- A set of instructions to specify how to compute the meaning representation of a construction from the meaning of its constituent parts

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

$$A.sem = f(\alpha_j.sem, \ldots \alpha_k.sem)$$

  - The semantics attached to *A* can be computed from some function applied to the semantics of *A*'s parts

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

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

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

$$MassNoun \rightarrow Meat \quad \{Meat\}$$

Semantic Argumentations to CFG Rules

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

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

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

lambda notation

- Take the semantics attached to one daughter and applying it as a function to the semantics of the other daughters

Semantic Argumentations to CFG Rules

- The operations permitted in the semantic rules fall into two classes
  - Pass the semantics of a daughter up unchanged to the mother

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

  - Apply (as a function) the semantics of one of the daughters of a node to the semantics of the other daughters

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