Introduction to Semantic Theory
Extensions - lexical items and sentences

Dr. Sarah Zobel

Class: May 4, 2016

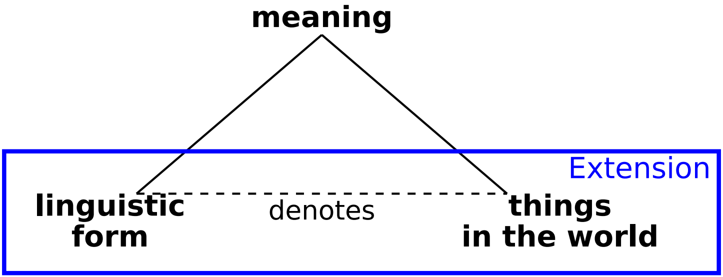
Connecting back to the previous lectures – I

The central points of the previous two lectures:

- Sentences can be **ambiguous** for various reasons; we distinguish lexical, structural (incl. scope), and referential ambiguities.
- Except for readings depending on lexical ambiguity, distinct readings are associated with distinct hierarchical structures.
- The level of structure that is relevant for semantic interpretation is the logical form of a sentence.
- The logical form is based on the full syntactic structure, but is further derived via the post-syntactic processes reconstruction and quantifier raising.

Connecting back to the previous lectures – II

Recapping the semiotic triangle:



Aim for today

The aim for today: to introduce a formalism to write down the results of the first class, and to extend on them with respect to the extension of sentences

⇒ set notation and function notation for the extensions of nouns, proper names, adjectives, and verbs

⇒ We will see: sentences can also be assigned an extension; the combinatoric system will, however, derive a different (but related) aspect of the meaning of sentences, their truth conditions

The extensions of nouns – I

What is the extension of the noun in (1)?

(1) *tree*

The extensions of nouns – I

What is the extension of the noun in (1)?

(1) *tree*

The extension of *tree* is the set of all trees in the world.

⇒ This sentence tells us all we need to know. But: it is hard (and cumbersome!) to use natural language descriptions of the conceptual ideas to model composition.

⇒ We need to find a way to write down “the extension of *tree*” and “the set containing all trees in the world” in a more concise manner.

The interpretation function

The first step is to introduce a concise notation for “the extension of *tree*”.

⇒ We use the **interpretation function**: $[\![\,.\,]\!]^{w}$

The superscript w of the interpretation function fixes for which world we determine the extension. Therefore, this superscript w is called the world of evaluation.

In the system that we build in this semester, the world of evaluation w is always the world in which the sentence is uttered – this is the world we live in, the **actual world**.

The extensions of nouns – II

Now that we have the interpretation function, we can substitute the first part of the sentence in (2) with the new notation.

(2) The extension of *tree* is the set of all trees in the world.

The extensions of nouns – II

Now that we have the interpretation function, we can substitute the first part of the sentence in (2) with the new notation.

(2) $[\![tree]\!]^w$ is the set of all trees in the world.

The extensions of nouns – II

Now that we have the interpretation function, we can substitute the first part of the sentence in (2) with the new notation.

(2) $[\![\textit{tree}]\!]^{w}$ is the set of all trees in the world.

At this point the copula *is* can also be substituted by an equality sign:

(3) $[\![\textit{tree}]\!]^{w} =$ the set of all trees in the world.

Set notation

The second step is to introduce a notation for “the set of all trees in the world”. Using mathematical set notation, there are two ways to do that.

Set notation

The second step is to introduce a notation for "the set of all trees in the world". Using mathematical set notation, there are two ways to do that.

- Listing all trees + assuming they can be named / numbered:

(4) $\{tree_1, tree_2, tree_3, tree_4, tree_5 \ldots\}$

Set notation

The second step is to introduce a notation for “the set of all trees in the world”. Using mathematical set notation, there are two ways to do that.

- Listing all trees + assuming they can be named / numbered:

(4) $\{\text{tree}_1, \text{tree}_2, \text{tree}_3, \text{tree}_4, \text{tree}_5 \ldots\}$

- Describing which kinds of things are elements of the set:

(5) $\{x : x \text{ is a tree in } w\}$

⇒ We will use option 2.

The extensions of nouns – III

Having introduced set notation, (6) can be further “mathematized” to give (7).

(6) $[\![tree]\!]^w =$ the set of all trees in the world.

(7) $[\![tree]\!]^w = \{x : x \text{ is a tree in } w\}$

Note: It is important that the superscripted world variable and the world variable in the set notation match!

(8) $[\![tree]\!]^{w} = \{x : x \text{ is a tree in } w\}$

The extension of proper names

What do proper names denote? What is their extension?

(9) $[\![Peter]\!]^w =$

The extension of proper names

What do proper names denote? What is their extension?

(9) $[\![\textit{Peter}]\!]^{w} = \text{Peter}$

In words: The extension of the proper name *Peter* is the individual who bears the name “Peter”.

The extensions of adjectives – I

How do we translate the extensions of adjectives into the new notation?

(10) The extension of *blue* is the set containing all blue things in the world.

The extensions of adjectives – I

How do we translate the extensions of adjectives into the new notation?

(10) The extension of *blue* is the set containing all blue things in the world.

(11) $[\![blue]\!]^{w} = \{x : x \text{ is blue in } w\}$

The extensions of verbs – I

For verbs, we have already seen that the number of arguments that a verb takes has an effect on the type of extension we assign to the verb.

- *to snore*:
- *to like*:
- *to give*:

The extensions of verbs – I

For verbs, we have already seen that the number of arguments that a verb takes has an effect on the type of extension we assign to the verb.

- *to snore*: the set containing all individuals in the world that are snoring
- *to like*:
- *to give*:

The extensions of verbs – I

For verbs, we have already seen that the number of arguments that a verb takes has an effect on the type of extension we assign to the verb.

- *to snore*: the set containing all individuals in the world that are snoring
- *to like*: the set containing all pairs of individuals in the world where the first individual in the pair likes the second individual in the pair
- *to give*:

The extensions of verbs – I

For verbs, we have already seen that the number of arguments that a verb takes has an effect on the type of extension we assign to the verb.

- *to snore*: the set containing all individuals in the world that are snoring
- *to like*: the set containing all pairs of individuals in the world where the first individual in the pair likes the second individual in the pair
- *to give*: the set containing all triples of individuals in the world where the first individual in the triple gives the second individual in the triple to the third individual in the triple.

The extensions of verbs – II

The extensions of intransitive verbs match the extensions of nouns and adjectives. Hence, the extension of the intransitive verb *snore* is not surprisingly as in (12).

(12) $[\![\textit{snore}]\!]^w = \{x : x \text{ is snoring in } w\}$

Ordered pairs and triples

The extensions of transitive and ditransitive verbs *cannot* be captured in the same manner. We still need a formal way to write down that two individuals form a pair or that three individuals form a triple.

Ordered pairs and triples

The extensions of transitive and ditransitive verbs *cannot* be captured in the same manner. We still need a formal way to write down that two individuals form a pair or that three individuals form a triple.

(13) Ordered pairs: $\langle x, y \rangle$ (the pair of x and y)

(14) Ordered triples: $\langle x, y, z \rangle$ (the triple of x, y, and z)

Why are they called "ordered pairs" and "ordered triples"?

Ordered pairs and triples

The extensions of transitive and ditransitive verbs *cannot* be captured in the same manner. We still need a formal way to write down that two individuals form a pair or that three individuals form a triple.

(13) Ordered pairs: $\langle x, y \rangle$ (the pair of x and y)

(14) Ordered triples: $\langle x, y, z \rangle$ (the triple of x, y, and z)

Why are they called "ordered pairs" and "ordered triples"?

They are ordered because
⟨ Peter, the book, Mary ⟩ ≠ ⟨ Peter, Mary, the book ⟩!

The extensions of verbs – III

Turning to transitive verbs:

How can the second part of the description of the extension of *like* be translated into set notation?

(15) $[\![like]\!]^w$ = the set containing all pairs of individuals in the world where the first individual in the pair likes the second individual in the pair

The extensions of verbs – III

Turning to transitive verbs:

How can the second part of the description of the extension of *like* be translated into set notation?

(15) $[\![like]\!]^w$ = the set containing all pairs of individuals in the world where the first individual in the pair likes the second individual in the pair

(16) $[\![like]\!]^w = \{\langle x, y\rangle : x \text{ likes } y \text{ in } w\}$

The extensions of verbs – IV

And analogously, the description of the extension of *give* in (17) can be translated into (18).

(17) $[\![give]\!]^w$ = the set containing all triples of individuals in the world where the first individual in the triple gives the second individual in the triple to the third individual in the triple

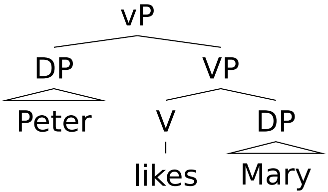
(18) $[\![give]\!]^w = \{\langle x, y, z\rangle : x \text{ is giving } y \text{ to } z \text{ in } w\}$

Interim summary

- $[\![Peter]\!]^w$ = Peter
- $[\![tree]\!]^w = \{x : x \text{ is a tree in } w\}$
- $[\![blue]\!]^w = \{x : x \text{ is blue in } w\}$
- $[\![snore]\!]^w = \{x : x \text{ is snoring in } w\}$
- $[\![like]\!]^w = \{\langle x, y\rangle : x \text{ likes } y \text{ in } w\}$
- $[\![give]\!]^w = \{\langle x, y, z\rangle : x \text{ is giving } y \text{ to } z \text{ in } w\}$

The problem with set notation – I

Consider the LF for *Peter likes Mary* and the extension of the transitive verb *like* in set notation:

(19)


(20) $[\![like]\!]^w = \{\langle x, y\rangle : x \text{ likes } y \text{ in } w\}$

Why is (20) problematic if we want to design a model in which composition proceeds in a step-wise fashion?

The problem with set notation – II

Assuming that composition proceeds in a step-wise fashion, the LF of *Peter likes Mary* "tells us" that *like* first combines with the extension of the internal argument *Mary* and in a second step combines with the extension of the external argument *Peter*.

In contrast, the set notation for the extension of *like* suggests that the verb only combines with one complex structure, a pair of the extensions of the external and the internal argument.

⇒ We need to find another way to express the same content that is straightforwardly compatible with stepwise composition.

Characteristic functions – I

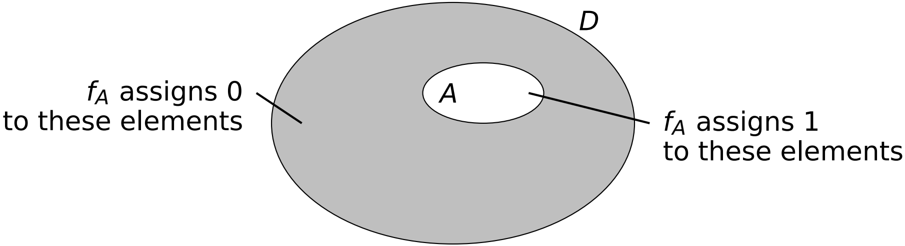
Before we can get rid of sets as extensions, we need to define one special set: **the domain of entities**

The domain of individuals D is a set that contains all individuals/entities/things in the world.

If we have at our disposition the set containing all individuals/entities in the world, it is possible to assign to a set A of individuals/entities a function that encodes which individuals/entities in D are members of A. This function is called the **characteristic function** of A.

Characteristic functions – II

The characteristic function f_A assigns “true” (= 1) to all elements of D that are in A and “false” (= 0) to all elements of D that are not in A.



Characteristic functions – III

Like for set notation, there are various different ways to write down that something is a characteristic function. In this course, we use λ**-notation**.

(21) set: A $\rightsquigarrow$ characteristic function: $\lambda x.\ x \in A$

The λ**-term**, λx, signals that the function takes one argument. The function returns the value 1 if an argument satisfies the condition after the dot, and returns 0 if the value does not satisfy the condition after the dot. The variable in the λ-term, x, determines which spot in the condition the argument gets assigned.

Characteristic functions – IV

A mathematical example: $D = \{1, 2, 3, 4\}$

(22) $\lambda x.\ x + 1 > 2$

For which numbers in *D* does the function in (22) return 1, and for which 0?

Characteristic functions – IV

A mathematical example: $D = \{1, 2, 3, 4\}$

(22) $\lambda x.\ x + 1 > 2$

For which numbers in D does the function in (22) return 1, and for which 0?

- For 1: Is $1 + 1 > 2$? No! The function returns 0.
- For 2: Is $2 + 1 > 2$? Yes! The function returns 1.
- For 3: Is $3 + 1 > 2$? Yes! The function returns 1.
- For 4: Is $4 + 1 > 2$? Yes! The function returns 1.

$\Rightarrow$ For D above, the function describes the set $\{2, 3, 4\}$.

Application to extensions – I

For the purpose of natural language, we assume D contains all individuals/entities/things in the world. Hence, we can substitute the set with its characteristic function.

(23) $[\![tree]\!]^w = \{x : x \text{ is a tree in } w\}$

(24) $[\![tree]\!]^w = \lambda y.\ y \in \{x : x \text{ is a tree in } w\}$

The version in (24) can be further simplified!

What has to be the case so that an entity y is a member of the set $\{x : x \text{ is a tree in } w\}$?

Application to extensions – I

For the purpose of natural language, we assume D contains all individuals/entities/things in the world. Hence, we can substitute the set with its characteristic function.

(23) $[\![tree]\!]^w = \{x : x \text{ is a tree in } w\}$

(24) $[\![tree]\!]^w = \lambda y.\ y \in \{x : x \text{ is a tree in } w\}$

The version in (24) can be further simplified!

What has to be the case so that an entity y is a member of the set $\{x : x \text{ is a tree in } w\}$?

(25) $[\![tree]\!]^w = \lambda y.\ y \text{ is a tree in } w = \lambda y.\ \text{tree'}(y)(w)$

Application to extensions – II

For adjectives and intransitive verbs, the same result obtains:

(26) $[\![blue]\!]^w = \lambda y.\ y$ is blue in $w = \lambda y.$ blue'$(y)(w)$

(27) $[\![snore]\!]^w = \lambda y.\ y$ is snoring in $w = \lambda y.$ snore'$(y)(w)$

Application to extensions – II

For adjectives and intransitive verbs, the same result obtains:

(26) $[\![\textit{blue}]\!]^w = \lambda y.\ y$ is blue in $w = \lambda y.$ blue'$(y)(w)$

(27) $[\![\textit{snore}]\!]^w = \lambda y.\ y$ is snoring in $w = \lambda y.$ snore'$(y)(w)$

Transitive and ditransitive verbs: the formal representation that we obtain via the idea of the characteristic function can be generalized.

(28) Generalization: For each individual that is involved in the verb meaning, one λ-term is needed. The descriptive content of the function gives a condition for all of these individuals.

Application to extensions – III

How many λ-terms do we need for transitive verbs? How many for ditransitive verbs? What is the descriptive content?

(29) $[\![like]\!]^{w} =$

(30) $[\![give]\!]^{w} =$

Application to extensions – III

How many λ-terms do we need for transitive verbs? How many for ditransitive verbs? What is the descriptive content?

(29) $[\![like]\!]^w = \lambda y.\lambda x.$

(30) $[\![give]\!]^w = \lambda z.\lambda y.\lambda x.$

Application to extensions – III

How many λ-terms do we need for transitive verbs? How many for ditransitive verbs? What is the descriptive content?

(29) $[\![like]\!]^w = \lambda y.\lambda x.$ x likes y in w
$= \lambda y.\lambda x.\text{like'}(y)(x)(w)$

(30) $[\![give]\!]^w = \lambda z.\lambda y.\lambda x.$

Application to extensions – III

How many λ-terms do we need for transitive verbs? How many for ditransitive verbs? What is the descriptive content?

(29) $[\![\textit{like}]\!]^w = \lambda y.\lambda x.\ x \text{ likes } y \text{ in } w$
$= \lambda y.\lambda x.\text{like'}(y)(x)(w)$

(30) $[\![\textit{give}]\!]^w = \lambda z.\lambda y.\lambda x.\ x \text{ is giving } y \text{ to } z \text{ in } w$
$= \lambda z.\lambda y.\lambda x.\text{give'}(z)(y)(x)(w)$

Interim summary 2

- $[\![\textit{Peter}]\!]^w = \text{Peter}$
- $[\![\textit{tree}]\!]^w = \lambda x.\ \text{tree'}(x)(w)$
- $[\![\textit{blue}]\!]^w = \lambda x.\ \text{blue'}(x)(w)$
- $[\![\textit{snore}]\!]^w = \lambda x.\ \text{snore'}(x)(w)$
- $[\![\textit{like}]\!]^w = \lambda y.\lambda x.\text{like'}(y)(x)(w)$
- $[\![\textit{give}]\!]^w = \lambda z.\lambda y.\lambda x.\text{give'}(z)(y)(x)(w)$

What about sentences?

To build a model for semantic composition, we not only need to have an assumption about the "starting point" (i.e., the extensions of lexical items), but also about the "end point":

What is the extension of a sentence? And is the extension of a sentence what the formal model should derive?

Since the model should capture the semantic ability of speakers: take a closer look at what speakers know and are able to determine about sentences.

Semantic ability of competent speakers

- Competent speakers of a language can understand infinitely many sentences, and also sentences that they have never seen or heard before.

  ⇒ Consequence of the combinatoric nature of meaning

Semantic ability of competent speakers

- Competent speakers of a language can understand infinitely many sentences, and also sentences that they have never seen or heard before.

  ⇒ Consequence of the combinatoric nature of meaning

- Competent speakers can determine whether one sentence entails another sentence, or not.

  ⇒ Remember lab class 1

Semantic ability of competent speakers

- Competent speakers of a language can understand infinitely many sentences, and also sentences that they have never seen or heard before.

  ⇒ Consequence of the combinatoric nature of meaning

- Competent speakers can determine whether one sentence entails another sentence, or not.

  ⇒ Remember lab class 1

- Competent speakers are able to detect ambiguities.

  ⇒ We have seen and practiced that in the last two lectures.

Semantic ability of competent speakers

- Competent speakers of a language can understand infinitely many sentences, and also sentences that they have never seen or heard before.

  ⇒ Consequence of the combinatoric nature of meaning
- Competent speakers can determine whether one sentence entails another sentence, or not.

  ⇒ Remember lab class 1
- Competent speakers are able to detect ambiguities.

  ⇒ We have seen and practiced that in the last two lectures.
- Competent speakers know what has to be the case in the world so that a given sentence describes it correctly. They know **the truth conditions of a sentence**.

Truth conditions – I

The truth conditions of a sentence have been famously linked to its meaning by Ludwig Wittgenstein.

> To understand a sentence means to know what is the case if it is true.
>
> (Wittgenstein 1922)

The methodological problem with this idea is, however, that to make clear what the truth conditions of a sentence are, we need to use language. But if we use language to talk about language, we need to distinguish two levels: **object language** and **meta language**.

Truth conditions – II

To illustrate the necessity for the object vs. meta language distinction, consider the truth conditions for '*Peter is snoring*' in (31).

(31) Peter is snoring is true in a world if and only if Peter is snoring in that world.

Truth conditions – II

To illustrate the necessity for the object vs. meta language distinction, consider the truth conditions for '*Peter is snoring*' in (31).

(31) Peter is snoring is true in a world if and only if Peter is snoring in that world.

In the course of the truth conditions, we change from object language to meta language:

The expression/sentence that we want to describe (the object of our investigations) is given in object language. The language we use to describe the object of our investigations is the meta language.

Truth conditions – III

Wittgenstein's connection between truth conditions and meaning also states that to grasp the meaning of a sentence, we do not need to know whether it is true, or not! This is important to realize.

⇒ The truth conditions of a sentence will be the "end point", i.e., what the formal model should ultimately derive.

But are the truth conditions of a sentence its extension?

The extension of sentences

Since the extension of an expression is supposed to be a “thing”/a set of “things” in the world, the truth conditions cannot be the extension of a sentence.

What would be a good candidate for the extension of a sentence?

The extension of sentences

Since the extension of an expression is supposed to be a “thing”/a set of “things” in the world, the truth conditions cannot be the extension of a sentence.

What would be a good candidate for the extension of a sentence?

The standard assumption, which we will adopt, is that the extension of a sentence is its truth value (true/false) in the world.

Truth-conditions and truth value of a sentence

Connection between the truth conditions of a sentence and its truth value in a world:
Given the truth conditions—if we know what the world is like, we can determine whether the sentence is true (= 1) or false (= 0) in the world.

(32) *Peter is happy*

Truth-conditions and truth value of a sentence

Connection between the truth conditions of a sentence and its truth value in a world:

Given the truth conditions—if we know what the world is like, we can determine whether the sentence is true (= 1) or false (= 0) in the world.

(32) *Peter is happy*

⇒ *Peter is happy* is true in *w* iff Peter is happy in *w*

Truth-conditions and truth value of a sentence

Connection between the truth conditions of a sentence and its truth value in a world:

Given the truth conditions—if we know what the world is like, we can determine whether the sentence is true (= 1) or false (= 0) in the world.

(32) *Peter is happy*

⇒ *Peter is happy* is true in *w* iff Peter is happy in *w*

World$_1$: ☺

$[\![\textit{Peter is happy}]\!]^{World_1} = 1$

World$_2$: ☹

$[\![\textit{Peter is happy}]\!]^{World_2} = 0$

Summary

- We have introduced two types of notation for extensions of linguistic expressions: set notation and function notation.
- The set notation and the function notation are connected via the notion of characteristic function.
- For sentences, we need to distinguish truth conditions from truth values.
- The truth conditions of a sentence state what has to be the case in the world for that sentence to be true.
- The truth value of a sentence is either true (= 1) or false (= 0) depending on whether the world satisfies the sentence's truth conditions.