1 The Compositionality Principle

Recall the important fact that there are infinitely many grammatical declarative sentences in English, but native speakers of English know their truth-conditions. How is this possible?

You might remember that the smallest building blocks for syntax/semantics are called **morphemes**. They are expressions that cannot be divided into meaningful sub-expressions. For instance, 'cat' is a morpheme, because it has no meaningful subparts, while 'cats' is not a morpheme, because it can be decomposed into 'cat' and '-s'. In this sense, morphemes are the primitives in syntax and semantics. We call the set of all morphemes of a language the **lexicon**.

The syntax combines morphemes in the lexicon to produce complex expressions, such as noun phrases, verb phrases, sentences, etc. Since the syntax allows sentences (and other complex expressions) to be embedded under other sentences (or complex expressions of the same kind)—the property known as *recursion*—, it generates infinitely many sentences. Notice that the syntax itself is a finite mechanism, but it produces infinitely many expressions.

The idea we will pursue is that the semantics does essentially the same thing as the syntax, except that instead of combining expressions, it combines meanings. For instance, it takes the meaning of 'a' and the meaning of 'man' and combine them to produce the meaning of 'a man'.

Furthermore, we assume that the semantics combines meanings of various expressions, simple or complex, in such a way that the **Compositionality Principle** is respected:

(1) **The Compositionality Principle**: The meaning of a complex phrase is determined solely by the meanings of its parts and their syntax.

A semantic system obeying this principle can deal with infinitely many expressions. Before explaining why, however, let's consider some examples. Recall that it is easy to know the truth-condition of any grammatical declarative sentence. Let's take 'A man saw Mary'. According to the Compositionality Principle, the truth-condition of this sentence is determined only by the meanings of its parts, namely 'a', 'man', 'saw', and 'Mary', and how these expressions are arranged.

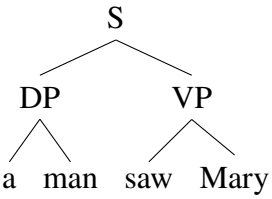
This sounds like a truism. However, it is not so hard to imagine an artificial language whose semantics doesn't obey the Compositionality Principle. In such a language, the meaning of 'a man', for example, might have nothing to do with the meaning of 'a' or the meaning of 'man'. Or take a more abstract 'language' in which sentences are sequences of mouse clicks. Suppose that one click means 'Select the file that the cursor is currently on', two clicks means 'Open the file the cursor is currently on', and three clicks means 'Delete the file the cursor is currently on.' The sentence with three clicks is syntactically decomposable into three individual clicks or one click plus two clicks, but clearly, the meaning of three clicks can be analysed as three instances of the meaning of one click, or the meaning of one click combined with the meaning of two clicks. In languages like these that do not obey the Compositionality Principle, there is no general way of computing the meaning of a complex expression based on the meanings of its parts. In other words, the meanings of complex expressions need to be remembered. The point here is that natural languages are not like this, and obey the Compositionality Principle.

If natural language semantics obeys the Compositionality Principle, there must be a way to combine any given two grammatical phrases. And if there are only a finite number of such ways to combine meanings that will collectively cover all grammatical sentences, we have a finite semantic system that computes the meanings of any grammatical sentence, by combining the meanings of its parts.

It should be stressed here that how the words are arranged, i.e. the syntactic structure, has semantic consequences. It is easy to see: clearly, 'A man saw Mary' and 'Mary saw a man' have different truth-conditions/meanings, despite the fact that these sentences are made up of the same morphemes. So the semantics is in a way dependent on the syntax. In order to incorporate this idea directly to our semantic theory, we understand the Compositionality Principle in a strict way. That is, we assume that the meaning of a complex phrase is determined by the meanings of *its immediate daughters*.

Here is a concrete example illustrating this idea. Let's assume that the sentence 'A man saw Mary' has the following syntactic structure.

(2)



The Compositionality Principle in the strict sense says that the meaning of the sentence 'A man saw Mary' with the above syntactic structure should be determined by the meaning of 'a man' and the meaning of 'saw Mary' and nothing else. It furthermore tells us that the meaning of 'a man', in turn, should be determined by the meaning of 'a' and the meaning of 'man', and nothing else. Likewise, the meaning of 'saw Mary' should be determined by the meaning of 'saw' and the meaning of 'Mary', and nothing else. Again, an important consequence of this is that all we need to remember is the meanings of these individual words/morphemes and how to combine them.

2 A Toy Syntax

As is clear from the above discussion, our semantic system will be made up of two finite components. They are the **lexicon** and the **compositional rules**. The lexicon is a list of the meanings of words, or more precisely, of morphemes. The compositional rules are instructions as to how to combine different meanings. We will take a step-by-step approach here and start with a very small grammar with a small lexicon and just one compositional rule. Then, we will enrich both the lexicon and the inventory of compositional rules throughout this course so that our theory will be able to account for more grammatical constructions.

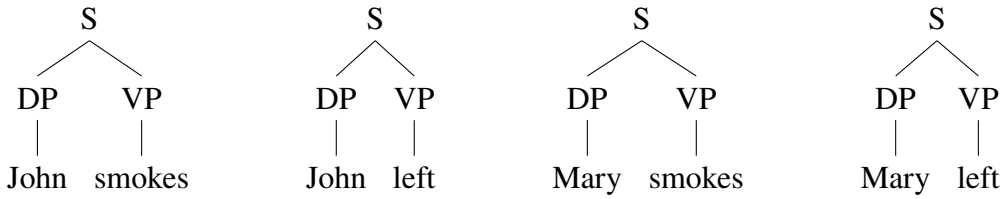
First, we need some sort of syntax. Let us assume a very simple syntax in (3).

(3) a. $S \rightarrow DP\ VP$
b. $DP \rightarrow$ John | Mary
c. $VP \rightarrow$ smokes | left

This is a kind of grammar called a Phrase Structure Grammar (or Context Free Grammar). Perhaps it is not the kind of syntactic framework you are familiar with, but we will use it here for expository purposes. And as we will see, our semantics is compatible with a more modern type of syntax.

The way to read syntactic rules like (3) is intuitive. (3a), for example, says "A node labeled S branches off to DP and VP". The rule in (3b) says "A node labeled DP yields John or Mary" (the vertical bar | means 'or'). Consequently, this syntax generates the following four sentences.

(4)



Whether the syntax is completely adequate for English or not is not our main concern here. You might want to have different labels, e.g. TP instead of S, another node below VP, etc. We abstract away from these details, because the main purpose here is to demonstrate how to construct a semantics for a syntax like this one. And in Lecture 2, we will only talk about the meanings of sentences and the meanings of DPs and proper names. The meanings of VPs and verbs will be the topic of the next lecture.

For expository purposes, we also use the bracket notation for syntactic objects. For example, the four sentences above are sometimes represented as (5).

(5) a. [$_S$ [$_{DP}$ John] [$_{VP}$ smokes]]
b. [$_S$ [$_{DP}$ John] [$_{VP}$ left]]
c. [$_S$ [$_{DP}$ Mary] [$_{VP}$ smokes]]
d. [$_S$ [$_{DP}$ Mary] [$_{VP}$ left]]

3 Sentences Denote Truth-Values

Recall that a native speaker knows the truth-condition of any given grammatical declarative sentence. For instance, we already know the truth-condition of 'John smokes', namely, it is true iff John smokes. Notice that whether the sentence is true or false is relative to some state of affairs. This makes sense because (most) sentences are not inherently true or false. One can only say that a sentence is true or false in a given situation. In other words, in order for you to determine the truth of falsity of the sentence 'John smokes', you have to inspect the situation and know (i) who John is and (ii) whether that person smokes or not.

We represent information like this in a mathematical structure called a **model**. You don't need to know the details of this (If you are interested, please see the end of the lecture notes for next week). All you have to know is that a model is a mathematical representation of a situation or state of affairs that contains enough information for you to decide the truth or falsity of the sentence. A model is often denoted by M (or M_1, M_2, etc.). A semantic theory that uses models is called **model theoretic semantics**, and this is the framework we will adopt.

If a model M is given, you can say a given sentence S is true or false in the situation represented by M. Or equivalently, we say S **denotes** truth or falsity in M, or the **denotation** of S is truth or falsity in M. The truth and falsity, furthermore, are standardly denoted by 1 and 0, respectively, which are called **truth-values**. We follow this tradition here. So, with respect to a model M S denotes 1 or 0. Furthermore, the denotation of any expression X with respect to a model M is written $[\![X]\!]^M$. Therefore, for any declarative sentence S and for any model M, we have $[\![S]\!]^M = 0$ (which means that S is false in the situation modeled by M) or $[\![S]\!]^M = 1$ (which means that S is true in the situation modeled by M).

It is important to keep in mind the difference between truth-values and truth-conditions. Both are in some sense 'meanings' of sentences, but truth-conditions are more abstract than truth-values. Truth-values are denotations of sentences in particular models, which represent particular states of affairs, and they are either 0 or 1. Truth-conditions are conditions that tell you in what kind of situation the sentence denotes 0 and in what kind of situations the sentence denotes 1. So two completely independent sentences, for instance, 'It is raining in London'

and 'John likes sushi', can have the same denotation in some models, but their truth-conditions remain distinct.

Using the new notation we introduce above, we can state the truth-condition of a given sentence S as follows.

(6) For any model M, $[\![S]\!]^M = 1$ iff S in M.

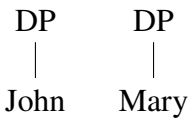
For example, the truth-condition of 'There is a circle in a square' can be stated as (7).

(7) For any model M, $[\![\text{There is a circle in a square}]\!]^M = 1$ iff there is a circle in a square in M.

We often omit 'for any model M', when it is obvious.

4 Proper Names Denote Individuals

Let us now turn to the meanings of DPs. We have so far only discussed the meanings and denotations of sentences, but we actually have intuitions about the meanings of DPs of the following kind.

(8)


What do we do with expressions like these? We **refer** to particular people. Of course, who exactly we refer to varies across conversational contexts. By '[$_{\text{DP}}$ John]', we could refer to John Lennon, or to John Harris in our department, but in each of such cases, we have a particular **referent** of the DP.

Importantly, who the referent is matters for the truth-condition of the sentence containing the proper name. If we refer to John Harris, the truth-condition of 'John smokes' is about John Harris, and if we refer to somebody else, John Harris will be irrelevant for the truth or falsity of the sentence. So it makes sense to equate the denotation of expressions like (8) to their referents.

How do we know who the referent is in a give model? We simply assume that a model M fixes the denotation of [$_{\text{DP}}$ John] to a particular person, just as M fixes the denotation of a sentence to 1 or 0. For our limited purposes here, we do not need to know exactly how this is done (and there is a lot of debate on this in the literature). We just assume that the denotation of the DP is some person (probably named John). Or using our notation above, for any model M, $[\![[_{\text{DP}}\ \text{John}]]\!]^M$ is an **individual**. In formal semantics, it is common to use the word **individual** to talk about people as well as any objects and other entities, including cups, desks, chairs, etc.

Now we know the denotation of the DP [$_{\text{DP}}$ John]. What would be the denotation of the proper name itself, i.e. 'John'? Notice that in projecting the DP, no new word is added to the structure. Given the Compositionality Principle in the strict sense, it is natural to assume that in such a case, the meaning does not change. That is, we have:

$$\left[\!\!\left[\begin{array}{c}\text{DP}\\ |\\ \text{John}\end{array}\right]\!\!\right]^M = [\![\text{John}]\!]^M$$

5 Summary

Let us summarize the discussion so far. To achieve the goal of constructing a finite semantics that is capable of assigning meanings to infinitely many sentences, we assume the **Compositionality Principle**.

(1) **The Compositionality Principle**: The meaning of a complex phrase is determined solely by the meanings of its parts and their syntax.

Our semantics will have two finite components, the **lexicon** and the **compositional rules**. We have not yet discussed what exactly they should look like, but from the above discussion, we already know that proper names, which are morphemes, denote individuals.

(9) For any model M, $[\![\text{John}]\!]^M$ is some particular individual in M.

Here $[\![X]\!]^M$ is the denotation of the expression X in the model M. We also discussed the denotations of sentences, which are truth-values. For any declarative sentence S, we have:

(10) Fro any model M, $[\![S]\!]^M = 0$ or $[\![S]\!]^M = 1$, where 0 represents falsity and 1 represents truth.

In the next lecture, we will discuss the denotations of verb phrases like [$_{\text{VP}}$ smokes].

6 Functions

At this point, let us touch on the concept of **functions**. As we will see, functions play a particularly important role in formal semantics.

Functions are simply a kind of 'mappings'. You can think of a function as a kind of blackbox that takes an input and returns an output. A function is special in that for the same input, it always returns the same output. Thus, functions are mappings where each input has a unique output, or equivalently, they are deterministic mappings.

As a matter of convention, we write the output of a function F **applied to** an input (or **argument**) A as $F(A)$. You must be familiar with this notation from high school mathematics, e.g. if your function is $g(x) = x^2 + 2 \times x$, then $g(2) = 2^2 + 2 \times 2 = 8$. That is, g returns 8 when 2 is the input.

In order for a mapping F to count as a function, the following must be true: for any argument A, there is only one $F(A)$. Not all mappings have this property. Here are some examples of functions and non-functions.

- Suppose that the mapping S maps any positive or negative integer n to n^2. This mapping is a function, because for each input value, there is only one output value. Notice that in this case there are some outputs that can be produced by different inputs, e.g. $S(2) = S(-2) = 4$. But this is fine, because what matters is whether there is a unique output for each input.
- Now consider the mapping G that maps any person to the languages they speak. This is not a function, because for some inputs, there are multiple outputs. For instance, Andrew Nevins speaks several languages.

In order to define a function, you need to specify the following three things:

- **Domain**: the set of objects the function takes as inputs.
- **Range**: the set of objects the function returns as outputs.
- What the function does to each input.

In the case of the squaring function S above:

- The domain of S ($\text{dom}(S)$) is the set of all integers, which is often denoted by $\mathbb{Z}$.
- The range of S ($\text{ran}(S)$) is the set of all natural numbers (including 0), $\mathbb{N}$. Here you can also say the range is $\mathbb{Z}$, because $\mathbb{N}$ is a subset of $\mathbb{Z}$.
- For any integer n, $S(n) = n^2$.

The domain and range are sometimes written together as '$S : \mathbb{Z} \rightarrow \mathbb{N}$'. This means the mapping S is a function from integers to natural numbers.

If you want to know more about functions, I recommend Chapters 2 and 3 of Partee, ter Meulen, & Wall (1990) *Mathematical Methods in Linguistics*.