1 Truth-conditional Semantics and the Fregean Program

1.1 Truth-conditional semantics

To know the meaning of a sentence is to know its truth-conditions. If I say to you

(1) There is a bag of potatoes in my pantry

you may not know whether what I said is true. What you do know, however, is what the world would have to be like for it to be true. There has to be a bag of potatoes in my pantry. The truth of (1) can come about in ever so many ways. The bag may be paper or plastic, big or small. It may be sitting on the floor or hiding behind a basket of onions on the shelf. The potatoes may come from Idaho or northern Maine. There may even be more than a single bag. Change the situation as you please. As long as there is a bag of potatoes in my pantry, sentence (1) is true.

A theory of meaning, then, pairs sentences with their truth-conditions. The results are statements of the following form:

Truth-conditions
The sentence "There is a bag of potatoes in my pantry" is true if and only if there is a bag of potatoes in my pantry.

The apparent banality of such statements has puzzled generations of students since they first appeared in Alfred Tarski's 1935 paper "The Concept of Truth in Formalized Languages."[1] Pairing English sentences with their truth-conditions seems to be an easy task that can be accomplished with the help of a single schema:

Schema for truth-conditions
The sentence "______" is true if and only if ______.

A theory that produces such schemata would indeed be trivial if there wasn't another property of natural language that it has to capture: namely, that we understand sentences we have never heard before. We are able to compute the meaning of sentences from the meanings of their parts. Every meaningful part of a sentence contributes to its truth-conditions in a systematic way. As Donald Davidson put it:

> The theory reveals nothing new about the conditions under which an individual sentence is true; it does not make those conditions any clearer than the sentence itself does. The work of the theory is in relating the known truth conditions of each sentence to those aspects ("words") of the sentence that recur in other sentences, and can be assigned identical roles in other sentences. Empirical power in such a theory depends on success in recovering the structure of a very complicated ability – the ability to speak and understand a language.[2]

In the chapters that follow, we will develop a theory of meaning composition. We will look at sentences and break them down into their parts. And we will think about the contribution of each part to the truth-conditions of the whole.

1.2 Frege on compositionality

The semantic insights we rely on in this book are essentially those of Gottlob Frege, whose work in the late nineteenth century marked the beginning of both symbolic logic and the formal semantics of natural language. The first worked-out versions of a Fregean semantics for fragments of English were by Lewis, Montague, and Cresswell.[3]

> It is astonishing what language accomplishes. With a few syllables it expresses a countless number of thoughts, and even for a thought grasped for the first time by a human it provides a clothing in which it can be recognized by another to whom it is entirely new. This would not be possible if we could not distinguish parts in the thought that correspond to parts of the sentence, so that the construction of the sentence can be taken to mirror the construction of the thought. . . . If we thus view thoughts as composed of simple parts and take these, in turn, to correspond to simple sentence-parts, we can understand how a few sentence-parts can go to make up a great multitude of sentences to which, in turn, there correspond a great multitude of thoughts. The question now arises how the construction of the thought proceeds, and by what means the parts are put together

> so that the whole is something more than the isolated parts. In my essay "Negation," I considered the case of a thought that appears to be composed of one part which is in need of completion or, as one might say, unsaturated, and whose linguistic correlate is the negative particle, and another part which is a thought. We cannot negate without negating something, and this something is a thought. Because this thought saturates the unsaturated part or, as one might say, completes what is in need of completion, the whole hangs together. And it is a natural conjecture that logical combination of parts into a whole is always a matter of saturating something unsaturated.[4]

Frege, like Aristotle and his successors before him, was interested in the semantic composition of sentences. In the above passage, he conjectured that semantic composition may always consist in the saturation of an unsaturated meaning component. But what are saturated and unsaturated meanings, and what is saturation? Here is what Frege had to say in another one of his papers.

> Statements in general, just like equations or inequalities or expressions in Analysis, can be imagined to be split up into two parts; one complete in itself, and the other in need of supplementation, or "unsaturated." Thus, e.g., we split up the sentence
>
> "Caesar conquered Gaul"
>
> into "Caesar" and "conquered Gaul." The second part is "unsaturated" – it contains an empty place; only when this place is filled up with a proper name, or with an expression that replaces a proper name, does a complete sense appear. Here too I give the name "function" to what this "unsaturated" part stands for. In this case the argument is Caesar.[5]

Frege construed unsaturated meanings as functions. Unsaturated meanings, then, take arguments, and saturation consists in the application of a function to its arguments. Technically, functions are sets of a certain kind. We will therefore conclude this chapter with a very informal introduction to set theory. The same material can be found in the textbook by Partee et al.[6] and countless other sources. If you are already familiar with it, you can skip this section and go straight to the next chapter.

1.3 Tutorial on sets and functions

If Frege is right, functions play a crucial role in a theory of semantic composition. "Function" is a mathematical term, and formal semanticists nowadays use it in exactly the way in which it is understood in modern mathematics.[7] Since functions are sets, we will begin with the most important definitions and notational conventions of set theory.

1.3.1 *Sets*

A *set* is a collection of objects which are called the "*members*" or "*elements*" of that set. The symbol for the element relation is "$\in$". "$x \in A$" reads "x is an element of A". Sets may have any number of elements, finite or infinite. A special case is the *empty set* (symbol "$\emptyset$"), which is the (unique) set with zero elements.

Two sets are equal iff[8] they have exactly the same members. Sets that are not equal may have some overlap in their membership, or they may be *disjoint* (have no members in common). If all the members of one set are also members of another, the former is a *subset* of the latter. The subset relation is symbolized by "$\subseteq$". "$A \subseteq B$" reads "A is a subset of B".

There are a few standard operations by which new sets may be constructed from given ones. Let A and B be two arbitrary sets. Then the *intersection of* A and B (in symbols: $A \cap B$) is that set which has as elements exactly the members that A and B share with each other. The *union of* A and B (in symbols: $A \cup B$) is the set which contains all the members of A and all the members of B and nothing else. The *complement of* A *in* B (in symbols: $B - A$) is the set which contains precisely those members of B which are not in A.

Specific sets may be defined in various ways. A simple possibility is to define a set by *listing its members*, as in (1).

(1) Let A be that set whose elements are a, b, and c, and nothing else.

A more concise rendition of (1) is (1′).[9]

(1′) $A := \{a, b, c\}$.

Another option is to define a set by *abstraction*. This means that one specifies a *condition* which is to be satisfied by all and only the elements of the set to be defined.

(2) Let A be the set of all cats.

(2′) Let A be that set which contains exactly those x such that x is a cat.

(2′), of course, defines the same set as (2); it just uses a more convoluted formulation. There is also a symbolic rendition:

(2″) A := {x : x is a cat}.

Read "{x : x is a cat}" as "the set of all x such that x is a cat". The letter "x" here isn't meant to stand for some particular object. Rather, it functions as a kind of place-holder or *variable*. To determine the membership of the set A defined in (2″), one has to plug in the names of different objects for the "x" in the condition "x is a cat". For instance, if you want to know whether Kaline ∈ A, you must consider the statement "Kaline is a cat". If this statement is true, then Kaline ∈ A; if it is false, then Kaline ∉ A ("x ∉ A" means that x is not an element of A).

1.3.2 *Questions and answers about the abstraction notation for sets*

Q1: If the "x" in "{x : x is a positive integer less than 7}" is just a place-holder, why do we need it at all? Why don't we just put a blank as in "{_ : _ is a positive integer less than 7}"?

A1: That may work in simple cases like this one, but it would lead to a lot of confusion and ambiguity in more complicated cases. For example, which set would be meant by "{_ : {_ : _ likes _} = ∅}"? Would it be, for instance, the set of objects which don't like anything, or the set of objects which nothing likes? We certainly need to distinguish these two possibilities (and also to distinguish them from a number of additional ones). If we mean the first set, we write "{x : {y : x likes y} = ∅}". If we mean the second set, we write "{x : {y : y likes x} = ∅}".

Q2: Why did you just write "{x : {y : y likes x} = ∅}" rather than "{y : {x : x likes y} = ∅}"?

A2: No reason. The second formulation would be just as good as the first, and they specify exactly the same set. It doesn't matter which letters you choose; it only matters in which places you use the same letter, and in which places you use different ones.

Q3: Why do I have to write something to the left of the colon? Isn't the condition on the right side all we need to specify the set? For example, instead of "{x : x is a positive integer less than 7}", wouldn't it be good enough to write simply "{x is a positive integer less than 7}"?

A3: You might be able to get away with it in the simplest cases, but not in more complicated ones. For example, what we said in A1 and A2 implies that the following two are different sets:

$$\{x : \{y : x \text{ likes } y\} = \varnothing\}$$
$$\{y : \{x : x \text{ likes } y\} = \varnothing\}$$

Therefore, if we just wrote "$\{\{x \text{ likes } y\} = \varnothing\}$", it would be ambiguous. A mere statement enclosed in set braces doesn't mean anything at all, and we will never use the notation in this way.

Q4: What does it mean if I write "{California : California is a western state}"?

A4: Nothing, it doesn't make any sense. If you want to give a list specification of the set whose only element is California, write "{California}". If you want to give a specification by abstraction of the set that contains all the western states and nothing else but those, the way to write it is "{x : x is a western state}". The problem with what you wrote is that you were using the name of a particular individual in a place where only place-holders make sense. The position to the left of the colon in a set-specification must always be occupied by a place-holder, never by a name.

Q5: How do I know whether something is a name or a place-holder? I am familiar with "California" as a name, and you have told me that "x" and "y" are place-holders. But how can I tell the difference in other cases? For example, if I see the letter "a" or "d" or "s", how do I know if it's a name or a place-holder?

A5: There is no general answer to this question. You have to determine from case to case how a letter or other expression is used. Sometimes you will be told in so many words that the letters "b", "c", "t", and "u" are made-up names for certain individuals. Other times, you have to guess from the context. One very reliable clue is whether the letter shows up to the left of the colon in a set-specification. If it does, it had better be meant as a place-holder rather than a name; otherwise it doesn't make any sense. Even though there is no general way of telling names apart from place-holders, we will try to minimize sources of confusion and stick to certain notational conventions (at least most of the time). We will normally use letters from the end of the alphabet as place-holders, and letters from the beginning of the alphabet as names. Also we will never employ words that are actually used as names in English (like "California" or "John") as place-holders. (Of course, we could so use them if we wanted to, and then we could also write things like "{California : California is a western state}", and it would be just another way of describing the set {x : x is a western state}. We could, but we won't.)

Q6: In all the examples we have had so far, the place-holder to the left of the colon had at least one occurrence in the condition on the right. Is this necessary for the notation to be used properly? Can I describe a set by means of a condition in which the letter to the left of the colon doesn't show up at all? What about "{x : California is a western state}"?

A6: This is a strange way to describe a set, but it does pick one out. Which one? Well, let's see whether, for instance, Massachusetts qualifies for membership in it. To determine this, we take the condition "California is a western state" and plug in "Massachusetts" for all the "x"s in it. But there are no "x"s, so the result of this "plug-in" operation is simply "California is a western state" again. Now this happens to be true, so Massachusetts has passed the test of membership. That was trivial, of course, and it is evident now that any other object will qualify as a member just as easily. So {x : California is a western state} is the set containing everything there is. (Of course, if that's the set we mean to refer to, there is no imaginable good reason why we'd choose this of all descriptions.) If you think about it, there are only two sets that can be described at all by means of conditions that don't contain the letter to the left of the colon. One, as we just saw, is the set of everything; the other is the empty set. The reason for this is that when a condition doesn't contain any "x" in it, then it will either be true regardless of what value we assign to "x", or it will be false regardless of what value we assign to "x".

Q7: When a set is given with a complicated specification, I am not always sure how to figure out which individuals are in it and which ones aren't. I know how to do it in simpler cases. For example, when the set is specified as "$\{x : x + 2 = x^2\}$", and I want to know whether, say, the number 29 is in it, I know what I have to do: I have to replace all occurrences of "x" in the condition that follows the colon by occurrences of "29", and then decide whether the resulting statement about 29 is true or false. In this case, I get the statement "$29 + 2 = 29^2$"; and since this is false, 29 is not in the set. But there are cases where it's not so easy. For example, suppose a set is specified as "$\{x : x \in \{x : x \neq 0\}\}$", and I want to figure out whether 29 is in this one. So I try replacing "x" with "29" on the right side of the colon. What I get is "$29 \in \{29 : 29 \neq 0\}$". But I don't understand this. We just learned that names can't occur to the left of the colon; only place-holders make sense there. This looks just like the example "{California : California is a western state}" that I brought up in Q5. So I am stuck. Where did I go wrong?

A7: You went wrong when you replaced all the "x" by "29" and thereby went from "$\{x : x \in \{x : x \neq 0\}\}$" to "$29 \in \{29 : 29 \neq 0\}$". The former makes sense, the latter doesn't (as you just noted yourself); so this cannot have been an equivalent reformulation.

Q8: Wait a minute, how was I actually supposed to know that "$\{x : x \in \{x : x \neq 0\}\}$" made sense? For all I knew, this could have been an incoherent definition in the first place, and my reformulation just made it more transparent what was wrong with it.

A8: Here is one way to see that the original description was coherent, and this will also show you how to answer your original question: namely, whether $29 \in \{x : x \in \{x : x \neq 0\}\}$. First, look only at the most embedded set description, namely "$\{x : x \neq 0\}$". This transparently describes the set of all objects distinct from 0. We can refer to this set in various other ways: for instance, in the way I just did (as "the set of all objects distinct from 0"), or by a new name that we especially define for it, say as "$S := \{x : x \neq 0\}$", or by "$\{y : y \neq 0\}$". Given that the set $\{x : x \neq 0\}$ can be referred to in all these different ways, we can also express the condition "$x \in \{x : x \neq 0\}$" in many different, but equivalent, forms – for example, these three:

"$x \in$ the set of all objects distinct from 0"
"$x \in S$ (where S is as defined above)"
"$x \in \{y : y \neq 0\}$"

Each of these is fulfilled by exactly the same values for "x" as the original condition "$x \in \{x : x \neq 0\}$". This, in turn, means that each can be substituted for "$x \in \{x : x \neq 0\}$" in "$\{x : x \in \{x : x \neq 0\}\}$", without changing the set that is thereby defined. So we have:

$$
\begin{aligned}
& \{x : x \in \{x : x \neq 0\}\} \\
= \ & \{x : x \in \text{the set of all objects distinct from } 0\} \\
= \ & \{x : x \in S\} \text{ (where S is as defined above)} \\
= \ & \{x : x \in \{y : y \neq 0\}\}.
\end{aligned}
$$

Now if we want to determine whether 29 is a member of $\{x : x \in \{x : x \neq 0\}\}$, we can do this by using any of the alternative descriptions of this set. Suppose we take the third one above. So we ask whether $29 \in \{x : x \in S\}$. We know that it is iff $29 \in S$. By the definition of S, the latter holds iff $29 \in \{x : x \neq 0\}$. And this in turn is the case iff $29 \neq 0$. Now we have arrived at an obviously true statement, and we can work our way back and conclude, first, that $29 \in S$, second, that $29 \in \{x : x \in S\}$, and third, that $29 \in \{x : x \in \{x : x \neq 0\}\}$.

Q9: I see for this particular case now that it was a mistake to replace *all* occurrences of "x" in the condition "$x \in \{x : x \neq 0\}$" by "29". But I am still not confident that I wouldn't make a similar mistake in another case. Is there a general rule or fool-proof strategy that I can follow so that I'll be sure to avoid such illegal substitutions?

A9: A very good policy is to write (or rewrite) your conditions in such a way that there is no temptation for illegal substitutions in the first place. This means that you should never reuse the same letter unless this is strictly necessary in order to express what you want to say. Otherwise, use new letters wherever possible. If you follow this strategy, you won't ever write something like "$\{x : x \in \{x : x \neq 0\}\}$" to begin with, and if you happen to read it, you will quickly rewrite it before doing anything else with it. What you would write instead would be something like "$\{x : x \in \{y : y \neq 0\}\}$". This (as we already noted) describes exactly the same set, but uses distinct letters "x" and "y" instead of only "x"s. It still uses each letter twice, but this, of course, is crucial to what it is meant to express. If we insisted on replacing the second "x" by a "z", for instance, we would wind up with one of those strange descriptions in which the "x" doesn't occur to the right of the colon at all, that is, "$\{x : z \in \{y : y \neq 0\}\}$". As we saw earlier, sets described in this way contain either everything or nothing. Besides, what is "z" supposed to stand for? It doesn't seem to be a place-holder, because it's not introduced anywhere to the left of a colon. So it ought to be a name. But whatever it is a name of, that thing was not referred to anywhere in the condition that we had before changing "x" to "z", so we have clearly altered its meaning.

Exercise

The same set can be described in many different ways, often quite different superficially. Here you are supposed to figure out which of the following equalities hold and which ones don't. Sometimes the right answer is not just plain "yes" or "no", but something like "yes, but only if . . .". For example, the two sets in (i) are equal only in the special case where $a = b$. In case of doubt, the best way to check whether two sets are equal is to consider an arbitrary individual, say John, and to ask if John could be in one of the sets without being in the other as well.

(i) $\{a\} = \{b\}$
(ii) $\{x : x = a\} = \{a\}$

(iii) $\{x : x \text{ is green}\} = \{y : y \text{ is green}\}$
(iv) $\{x : x \text{ likes a}\} = \{y : y \text{ likes b}\}$
(v) $\{x : x \in A\} = A$
(vi) $\{x : x \in \{y : y \in B\}\} = B$
(vii) $\{x : \{y : y \text{ likes } x\} = \emptyset\} = \{x : \{x : x \text{ likes } x\} = \emptyset\}$

1.3.3 *Functions*

If we have two objects x and y (not necessarily distinct), we can construct from them the *ordered pair* $\langle x, y\rangle$. $\langle x, y\rangle$ must not be confused with $\{x, y\}$. Since sets with the same members are identical, we always have $\{x, y\} = \{y, x\}$. But in an ordered pair, the order matters: except in the special case of $x = y$, $\langle x, y\rangle \neq \langle y, x\rangle$.[10]

A *(2-place) relation* is a set of ordered pairs. Functions are a special kind of relation. Roughly speaking, in a function (as opposed to a non-functional relation), the second member of each pair is uniquely determined by the first. Here is the definition:

(3) A relation f is a *function* iff it satisfies the following condition:
For any x: if there are y and z such that $\langle x, y\rangle \in f$ and $\langle x, z\rangle \in f$, then $y = z$.

Each function has a *domain* and a *range*, which are the sets defined as follows:

(4) Let f be a function.
Then the *domain* of f is $\{x$: there is a y such that $\langle x, y\rangle \in f\}$, and the *range* of f is $\{x$: there is a y such that $\langle y, x\rangle \in f\}$.

When A is the domain and B the range of f, we also say that f is *from* A and *onto* B. If C is a superset[11] of f's range, we say that f is *into* (or *to*) C. For "f is from A (in)to B", we write "$f : A \rightarrow B$".

The uniqueness condition built into the definition of functionhood ensures that whenever f is a function and x an element of its domain, the following definition makes sense:

(5) $f(x) :=$ the unique y such that $\langle x, y\rangle \in f$.

For "f(x)", read "f applied to x" or "f of x". f(x) is also called the "*value* of f *for the argument x*", and we say that *f maps x to y*. "$f(x) = y$" (provided that it is well-defined at all) means the same thing as "$\langle x, y\rangle \in f$" and is normally the preferred notation.

Functions, like sets, can be defined in various ways, and the most straightforward one is again to simply *list* the function's elements. Since functions are sets of ordered pairs, this can be done with the notational devices we have already introduced, as in (6), or else in the form of a *table* like the one in (7), or in *words* such as (8).

(6) $F := \{\langle a, b\rangle, \langle c, b\rangle, \langle d, e\rangle\}$

(7) $F := \begin{bmatrix} a \rightarrow b \\ c \rightarrow b \\ d \rightarrow e \end{bmatrix}$

(8) Let F be that function f with domain $\{a, c, d\}$ such that $f(a) = f(c) = b$ and $f(d) = e$.

Each of these definitions determines the same function F. The convention for reading tables like the one in (7) is transparent: the left column lists the domain and the right column the range, and an arrow points from each argument to the value it is mapped to.

Functions with large or infinite domains are often defined by specifying a *condition* that is to be met by each argument–value pair. Here is an example.

(9) Let F_{+1} be that function f such that
$f : \mathbb{N} \rightarrow \mathbb{N}$, and for every $x \in \mathbb{N}$, $f(x) = x + 1$.
($\mathbb{N}$ is the set of all natural numbers.)

The following is a slightly more concise format for this sort of definition:

(10) $F_{+1} := f : \mathbb{N} \rightarrow \mathbb{N}$
For every $x \in \mathbb{N}$, $f(x) = x + 1$.

Read (10) as: "F_{+1} is to be that function f from $\mathbb{N}$ into $\mathbb{N}$ such that, for every $x \in \mathbb{N}$, $f(x) = x + 1$." An even more concise notation (using the λ-operator) will be introduced at the end of the next chapter.

Notes

1 A. Tarski, "Der Wahrheitsbegriff in den formalisierten Sprachen" (1935), English translation in A. Tarski, *Logic, Semantics, Metamathematics* (Oxford, Oxford University Press, 1956), pp. 152–278.

2 D. Davidson, *Inquiries into Truth and Interpretation* (Oxford, Clarendon Press, 1984), p. 24.
3 D. Lewis, "General Semantics," in D. Davidson and G. Harman (eds), *Semantics of Natural Languages* (Dordrecht, Reidel, 1972), 169–218; R. Montague, *Formal Philosophy* (New Haven, Yale University Press, 1974); M. J. Cresswell, *Logics and Languages* (London, Methuen, 1973).
4 G. Frege, "Logische Untersuchungen. Dritter Teil: Gedankengefüge," *Beiträge zur Philosophie des deutschen Idealismus*, 3 (1923–6), pp. 36–51.
5 Frege, "Function and Concept" (1891), trans. in M. Black and P. Geach, *Translations from the Philosophical Writings of Gottlob Frege* (Oxford, Basil Blackwell, 1960), pp. 21–41, at p. 31.
6 B. H. Partee, A. ter Meulen, and R. E. Wall, *Mathematical Methods in Linguistics* (Dordrecht, Kluwer, 1990).
7 This was not true of Frege. He distinguished between the function itself and its extension (German: *Wertverlauf*). The latter, however, is precisely what mathematicians today call a "function", and they have no use for another concept that would correspond to Frege's notion of a function. Some of Frege's commentators have actually questioned whether that notion was coherent. To him, though, the distinction was very important, and he maintained that while a function is unsaturated, its extension is something saturated. So we are clearly going against his stated intentions here.
8 "Iff" is the customary abbreviation for "if and only if".
9 We use the colon in front of the equality sign to indicate that an equality holds by definition. More specifically, we use it when we are defining the term to the left of ":=" in terms of the one to the right. In such cases, we should always have a previously unused symbol on the left, and only familiar and previously defined material on the right. In practice, of course, we will reuse the same letters over and over, but whenever a letter appears to the left of ":=", we thereby cancel any meaning that we may have assigned it before.
10 It is possible to define ordered pairs in terms of sets, for instance as follows: $\langle x, y \rangle := \{\{x\}, \{x, y\}\}$. For most applications of the concept (the ones in this book included), however, you don't need to know this definition.
11 The superset relation is the inverse of the subset relation: A is a superset of B iff $B \subseteq A$.