The Formalization of 'Meaning' [1]

1. Obtaining a System That Derives Truth Conditions

(1) **The Goal of Our Enterprise**
To develop a system that, for every sentence S of English, derives the truth-conditions of S from (i) the meanings of the component pieces of S, and (ii) the syntax of S.

(2) **Outline Sketch of a System Satisfying (1)**

SEMANTICS($[_{NP}$ Dave]) + SEMANTICS($[_{VP}$ smokes]) =

SEMANTICS($[_{S}$ Dave smokes]) = **TRUTH-CONDITIONS($[_{S}$ Dave smokes])**

Some Important Questions:

- How do we construct a theory that obtains this result for us?

- What are the 'meanings' of "Dave" and "smokes" such that combining them together can yield for us the truth conditions of "Dave smokes"?

... more in a moment!

2. More about the Meaning of 'Meaning': An Excursus of 'Extensions'

We've already seen that the everyday word 'meaning' is vague and ambiguous in a number of ways…*here's another:*

(3) **The Meaning of the Phrase 'The President'**

a. In one sense, the *meaning* of the NP "the president of the United States" changed in January 2009. It went from *meaning* George Bush (with all of the attendant connotations) to *meaning* Barack Obama.
('denotation', 'reference')

b. In another sense, the *meaning* of the NP is the same now as it was in 2008. It still *means* 'the chief executive officer of the United States government'.
('sense', 'concept')

Instead of using the word 'meaning' in this vague and ambiguous fashion, let's introduce two different terms to refer unambiguously to these two different 'senses' of the word "means".

[1] These notes are based on material in Heim & Kratzer (1998: 13-26), as well as Chierchia & McConnell-Ginet (2000: 53-73, 99-104).

(4) **Extension vs. Intension (of NPs)**

a. The *extension* of an NP is the thing in the world that the NP (currently) refers to.

- The *extension* of “the president” is *Barack Obama.*

b. The *intension* of an NP is the ‘general concept’ behind the NP, which determines (for a given time/situation) what the extension of the NP is.

- The *intension* of “the president” is *the chief executive officer of the US*

So, the ‘meaning’ of an NP can be broken up into its ‘extension’ and its ‘intension’…
…can the meaning of a sentence likewise be broken up in this way?

(5) **Intension of a Sentence = ‘Truth Conditions’**

We might take the ‘intension’ of a sentence to be (something like) its *truth conditions*…

As we’ve already seen, the truth conditions of a sentence are akin to what we might vaguely describe at the sentence’s (asserted) ‘informational content’.

… But if the ‘intension’ of a sentence is its truth conditions, what is its ‘extension’?...

(6) **Extension of a Sentence = Truth Value**

If we take the ‘intension’ of a sentence to be its truth conditions, then we should take the ‘extension’ of a sentence to be its *truth value*.

Why?
Recall that the ‘intension’ determines for a given time/situation what the *extension* is.
Truth conditions determine for a given time/situation what the *truth value is.*

Side-Note:

The (crucial) idea that the extension of a sentence is its *truth value* is one of those ‘weird ideas’ that you just have to get used to…

For some further motivation behind this crucial concept:
a. Chierchia & McConnell-Ginet (2000: 62-65)
b. Our later discussion of the ‘truth-functional sentence connectives’ (Section 5.2)

(7) **The General Picture**

a. Intension of "X":

A kind of 'concept'/ 'definition' which – for any given time/situation – determines what "X" 'picks out' at that time/situation.

b. Extension of "X"

The thing which, at a given time/situation "X" 'picks out'

c. Illustrative Paradigm:

i. Intension of "the president" = *the chief executive officer of the US*

ii. Extension of "the president" = Barack Obama

iii. Intension of "Barack smokes" = *Barack smokes*

iv. Extension of "Barack smokes" = TRUE

What's the point of all of this?

As we will presently see, you can actually build a decent theory of how meanings can 'compose' to yield truth-conditions by paying attention to extensions (defined as above)!

3. Towards a Compositional Semantics Based on Extensions

(8) **Some New Notation** [[X]] = the extension of "X"
(function from X to its extension)

(9) **The Compositionality of Extensions**

a. We've broken down the 'meaning' of a S/NP into its *extension* and its *intension*.

b. Recall, however, that our semantic system must be such that the 'meaning' of a complex expression should be *derived from* the 'meaning' of its component parts.

c. *Thus, the extension of a complex expression should be derived from the extensions of its component parts!*

d. In A Picture: [[Barack]] + [[smokes]] = [[Barack smokes]] = True

(10) **The Extension of a Predicate = Function**

a. In order to make the picture in (9d) work, the extension of the predicate "smokes" must be such that 'combining' it with the extension of "Barack" yields the extension of "Barack smokes", which is the value 'TRUE'.

b. *How can we model this?* Well, we know the following:

(i)	[[Barack]]	=	Barack
(ii)	[[Barack smokes]]	=	T(rue)

c. *Thus:* [[smokes]] + Barack = T

The extension of "smokes" 'combines' with Barack to yield T

d. *So, the extension of "smokes" is like a 'device' that takes Barack as input and yields T as output…*

e. ***So, the extension of "smokes" is like a FUNCTION!***

f. <u>The Core Idea:</u>

[[smokes]] = A function from entities to truth values, which yields T iff that entity smokes.

$$f : \{ x : x \text{ is an entity} \} \rightarrow \{ T, F \}$$
$$\text{for every } y \in \{ x : x \text{ is an entity} \},\ f(y) = T \textit{ iff } y \text{ smokes}$$

(11) **Interim Summary: Our Lexical Entries**

a. [[Barack]] = Barack

b. [[smokes]] = $f : \{ x : x \text{ is an entity} \} \rightarrow \{ T, F \}$
for every $y \in \{ x : x \text{ is an entity} \}$, $f(y) = T$ *iff* y smokes

We've almost got the picture in (9d) worked out… all we need is a rule for 'combining' the extensions of "Barack" and "smokes" to yield the extension of "Barack smokes"

(12) **The Rule of Function Application (Heim & Kratzer 1998: 44)**

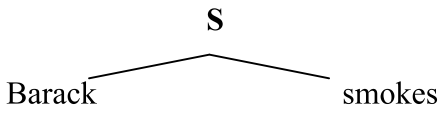
If X is a branching node that has two daughters – Y and Z – and if [[Y]] is a function whose domain contains [[Z]], then [[X]] = [[Y]]([[Z]])

With the rule in (12), we now have a system that derives the extension of the sentence "Barack smokes" from (i) the extension of its component pieces, and (ii) the syntax of the sentence.

(13) **Computing the Extension of 'Barack smokes'**

a. Syntactic Assumption (to be revised shortly):

The structure of the sentence *Barack smokes* is as follows:



b. Semantic Derivation:

(i)	[[S]]	=	(by F(unction) A(pplication))
(ii)	[[smokes]] ([[Barack]])	=	(by (11a))
(iii)	[[smokes]] (Barack)	=	(by (11b))
(iv)	f (Barack)	=	(by (11b) and the facts of the world)
(v)	True		

So, the system in (11) and (12) can derive the extension of a sentence (its truth value) from the extension of its component parts (given the facts of the world)…

So, we've obtained the system sketched in (9d)…

…Ok, but so what?...

4. From Extensions to Truth Conditions

Question: How does the system in (11) and (12) – which derives *extensions* – advance our goal of developing a formal system that derives *truth-conditions?*

(14) **Our Desired Semantic System**

A precise, formal system that for every sentence S of English, will derive a *correct* statement of the following form: *"S" is True iff X*

Believe it or not, our system for deriving extensions achieves what we want in (14)!!!

<u>Side-Note:</u> Some logical truths to keep in mind

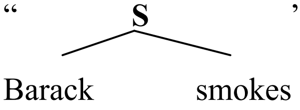
a. *Transitivity of 'iff'* A *iff* B and B *iff* C **entails** A *iff* C

b. *Substituting 'equals' for 'equals'* If x = y, then x = z *iff* y = z

(15) **Deriving the Truth Conditions of a Sentence via Our Theory of Extensions**

To Prove: "Barack smokes" is T iff Barack smokes

a. "Barack smokes" is T *iff* (by Syntactic Assumptions)

b. "  " is T *iff* (by definition of our notation)

c. [[**S**]] = T *iff* (by FA)

c. [[smokes]]([[Barack]]) = T *iff* (by (11))

d. f (Barack) = T *iff* (by definition of 'f' in (11b))

e. Barack smokes.

What just happened:

- In the preceding section, we first showed how a compositional extensional semantics can, *given the facts in the world*, compute the truth value of a sentence.
- *Conversely*, if we take as *hypothesis* that the truth value of a sentence is 'TRUE', our compositional extension semantics can 'work backwards', and compute *how the world must be constituted in order for the sentence to be true!*
- **Thus, we can use our extensional semantic function "[[ ]]" to compute the truth conditions of sentences!!!**

(16) **The Big Upshot**

An 'extensional semantics' – a formal system that maps complex structures to their extensions in the world – can provide us with a theory of how our brains recursively map sentences to their truth conditions.

(17) **Our Over-Arching Project, Redefined Again**

We wish to develop the *right* theory of the function "[[.]]", by examining the truth-conditions of particular sentences of English. Such a theory will consist of:

a. Primitive statements for the lexical items of the language

b. Rules for deriving the value of "[[ ]]" for larger structures from (i) the value of "[[ ]]" for their component parts and (ii) the syntax of the larger structures **(Principle of Compositionality)**

So our task is to adjust (a) and (b) until our theory predicts *exactly* the correct truth conditions for every sentence of English (or whatever language we're studying)!

(18) **A Review of Where We Are and How We Got Here**

a. We want to know how the 'meaning' of a sentence is computed from the 'meaning' of its parts. (Principle of Compositionality)

b. We found that (for declarative sentences) the term 'meaning' can (largely) be recast as *the truth-conditions of a sentence.*

c. Thus, we wish to know how the *truth conditions* of a sentence can be derived from the 'meanings' of its component parts.

(i) We took a detour and introduced the notion of the *extension* of a given linguistic expression.

(ii) We noted (stipulated) that the extension of a sentence is its *truth-value*.

(iii) We then developed a system that derives the extensions of sentences (their truth values) from the extensions of their component expressions.

(iv) The following are two key features of this system:

- The extensions of some expressions are *functions*

- A rule for 'combining' the extensions of two sister expressions to yield the extension of their mother is FUNCTION APPLICATION

d. **We found that this 'extensional semantics' can be used to derive the truth-conditions of sentences from the extensions of their component parts…**

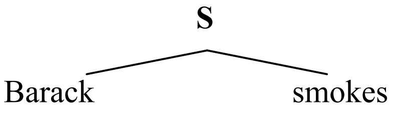
5. Some Follow-Up Technical Points

5.1 Deriving the Meaning of Non-Branching Nodes

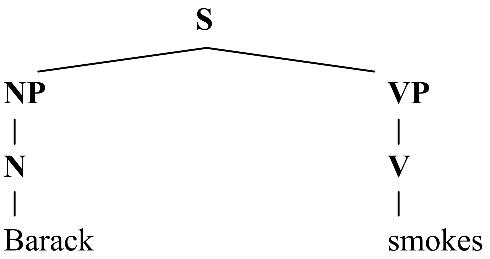
Thus far, we've been assuming that the structure of "Barack smokes" is that in (19a). However, a more realistic picture of its syntax might be that in (19b).

(19) **The Structure of "Barack smokes"**

a. Simplified Picture (No Non-Branching Nodes)



b. More Realistic Picture (With Non-Branching Nodes)



In order to formally compute the truth-conditions of structures like (19b), we will need to introduce some explicit rules for dealing with non-branching nodes like NP, N, VP and V.

(20) **The Rule for Non-Branching Nodes (Heim & Kratzer 1998: 44)**

If X is a non-branching node, and Y is its sole daughter, then [[X]] = [[Y]]

As we will see, it will also be useful to adopt a general rule for terminal nodes such as "Barack" and "smokes", rather than continually referring to their specific lexical entries (e.g. (11a,b))

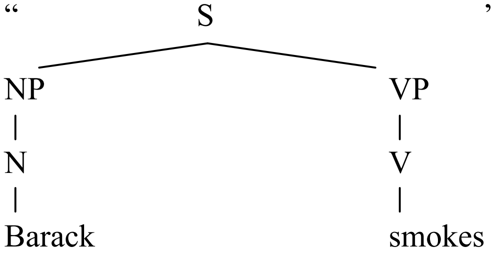
(21) **The Rule for Terminal Nodes (Heim & Kratzer 1998: 43)**

If X is a terminal node, then [[X]] is specified in the lexicon

In our proofs, we can abbreviate the rule in (20) as "NN" and (21) as "TN"

So, let's get some practice by using these rules in a proof!

(22) **Starting Off the Proof**

a. “ 

” is T *iff* (by notation)

b. [[S]] = T *iff* … uh oh!...

(23) **PROBLEM**

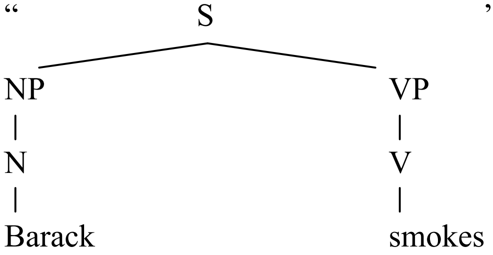
- We want to use our rule of ‘Function Application’ to compute [[S]]…
- But, we don’t know whether the rule can apply unless we know the values of [[NP]] and [[VP]].
- But, the values of [[NP]] and [[VP]] must now themselves be computed (rather than just easily looking them up in our lexicon)…

(24) **The Answer: ‘Subproofs’**

- A *subproof* is a proof inside a larger proof.
- The subproof proves/derives something that is then used in the larger proof.

On the following page, we can see how the proof above can be completed by using a ***subproof…***

(25) **Illustrative Derivation with Subproofs**

a. “ 

” is T *iff* (by notation)

b. [[S]] = T

c. **<u>Subproof</u>**

(i) [[NP]] = [[N]] (by NN)

(ii) [[N]] = [[Barack]] (by NN)

(iii) [[Barack]] = Barack (by TN)

(iv) [[NP]] = Barack (by c(i)-(iii))

d. **<u>Subproof</u>**

(i) [[VP]] = [[V]] (by NN)

(ii) [[V]] = [[smokes]] (by NN)

(iii) [[smokes]] = f (by TN)

(iv) [[VP]] = f (by d(i)-(iii))

e. [[S]] = T *iff* (by FA, **c**, **d**)

f. [[VP]]([[NP]]) = T *iff* (by **c**)

g. [[VP]](Barack) = T *iff* (by **d**)

h. f (Barack) = T *iff* (by def. of *f* in (11b))

i. Barack smokes.

Later on, when calculations start getting complicated, we will let you leave out ‘subproofs’… For now, however, make sure to include any ‘subproofs’ your proofs rely upon, using the conventions above

5.2 On Identifying 'Truth Values' as the Extensions of Sentences

Our semantic system has one core property that often first strikes people as 'odd' or 'confusing':

(26) **The Central (But Initially Confusing) Assumption of Our Semantic Theory**
Sentences of natural language have *extensions* ('denotations', 'reference'), and the *extension* of such sentences is a *truth value*.

We've given some general conceptual motivation to this claim, and have shown that it *does* contribute to our goal of having a formal system that derives truth-conditions….

…However, there is also something of an empirical argument for these two assumptions.

(27) **Empirical Argument for the Assumption in (26)**

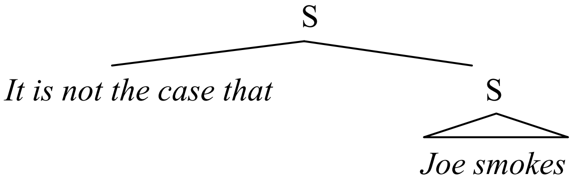
a. Fact:
There are a number of natural language expressions – called the 'logical connectives' – that have the following key properties:

(i) *Syntactically, they take sentences as complements / specifiers / sisters*
(ii) *Semantically, they seem to take truth values as arguments.*

b. Argument:
If the 'meaning' of one lexical item must take as argument the 'meanings' of its complements/specifiers, then such 'logical connectives' *independently* show that **at some level, the 'meaning' of a sentence is its truth value.**

(28) **Example: '*It is not the case that*'**

a. Syntax: Takes sentences as complement.



b. Semantics:
[[it's not the case that]] = h: { T, F } → { T, F }
for all y ∈ { T, F }, h(y) = T *iff* y = F

The extension of 'it is not the case that' is a function which takes as argument a truth-value y yields the value T iff y is F.

(29) **Question**
But, why should we assume the semantics in (28b)?

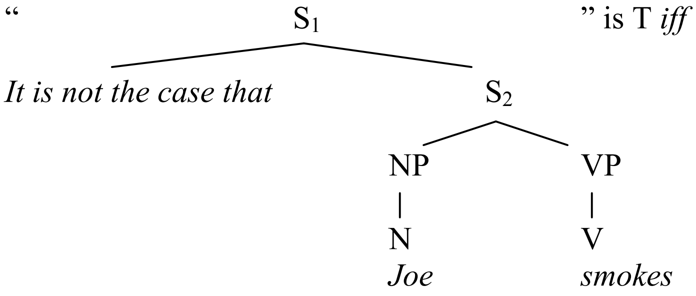
Answer: With the lexical entry in (28b), our system can derive the following truth-conditional statement:

“Its not the case that Joe smokes” is T *iff* Joe doesn’t smoke.

(30) **Illustrative Derivation**

a. “It is not the case that Joe smokes” is T *iff* (by Syntactic Assumptions)

b. “ ” is T *iff* (by notation)



c. [[S_1]] = T

d. **Subproof**

(i) [[NP]] = (by NN)

(ii) [[N]] = (by NN)

(iii) [[Joe]] = (by TN)

(iv) Joe

e. **Subproof**

(i) [[VP]] = (by NN)

(ii) [[V]] = (by NN)

(iii) [[smokes]] = (by TN)

(iv) f

f. **<u>Subproof</u>**

(i) [[S_2]] = (by FA, d, e)

(ii) [[VP]]([[NP]]) = (by d)

(iii) [[VP]](Joe) = (by e)

(iv) f(Joe)

g. [[S_1]] = T *iff* (by FA, NN, f)

h. [[it is not the case that]]([[S_2]]) = T *iff* (by TN)

i. h([[S_2]]) = T *iff* (by def. of *h* in (56b))

j. [[S_2]] = F *iff* (by f)

k. f(Joe) = F *iff* (by def. of *f* in (38b))

l. Joes doesn't smoke.

(31) **The Main Point**

a. The proof in (30) suggests that the lexical entry in (28b) is on the right track.

b. If the lexical entry in (28b) is correct, then the expression '*it is not the case that*' has as its extension a function that that takes **truth-values** as argument.

c. If the syntax in (28a) is correct, then the extension of '*it is not the case that*' must take as argument the extension of its sentential complement.

d. **Consequently, we must conclude that the extension of the sentential complement of '*it is not the case that*' is a <u>truth-value.</u>**

e. Thus, the extension of a sentence is a truth-value…

6. Where Do We Go From Here?

- Right now, we have a system that can derive truth-conditions for a *very limited* set of English sentences: *those headed by intransitive Vs and whose subjects are proper names.*

- We are going to quickly expand this system, so that it can interpret ever more complex (and therefore 'realistic') structures of English.

(32) **Some (Very Basic) Structures We Will Cover in the Next Few Months**

a. Transitive Verbs

b. Nouns

c. Adjectives

d. Relative Clauses (and Other Movement Structures)

e. Definite Descriptions

f. Pronouns

g. Quantifiers (and 'quantificational DPs')