# Chapter 1
## Overview

1. Theses ........................................................................................................................................ 2
2. Semantics is King (or at least Autonomous)............................................................................... 2
3. Semantic-Composition Is Hugely Expanded .............................................................................. 3
4. Expanded-Composition Produces Over-Generation (which we cure)......................................... 3
5. Case-Marking Requires Expanded Lambda-Abstraction ............................................................ 4
6. Open Expressions do not Have the Same Types as Closed Expressions................................. 4
7. A New Theory of Quantifiers and Quantifier Phrases ................................................................. 5
8. Pronoun-Binding........................................................................................................................ 5
9. There is just one Type of Noun.................................................................................................. 7
1. **Theses**

1. **Semantics is King**
   Although the other branches of grammar are important, and are indeed crucial to linguistic competence, they all serve semantics.

2. **Semantics involves Translation**
   Fundamentally, we follow Montague in doing Indirect Semantics. We insist that the meaning of a phrase in the object language is conveyed by a corresponding phrase in the target language, which makes sense of the original phrase.

3. We reject open formulas as translations of anything.

4. Composition is critical to semantics; readings are data, not theory.

5. Composition is not unique. Although the meaning of a phrase is systematically constructed out of its parts, it is not a unique function of these parts and their arrangement. The very same tree can produce different semantic output.

6. We by and large reject LF.

7. We thoroughly reject Frege's Thesis that all composition is function-application.

8. We reject that syntactic category determines semantic category.

9. We insist that case-marking is ubiquitous and critical to understanding meaning.

10. We reject Montague's analysis of quantifiers in favor of an infinitary-type-theory account. This allows us to posit many quite distinct quantifiers.

11. We reject the fundamental distinction between definite-nouns and indefinite-nouns; all nouns are fundamentally indefinite.

12. We reject magic binding principles [lambda-generalization].

2. **Semantics is King (or at least Autonomous)**

Language-use consists in the processing [production and recognition] of signs/symbols, which are variously realized – by gestures, sounds, and graphs. It involves many skills, and linguistics seeks to understand how we acquire and master (and muster!) all of them.

Even though many skills are involved, central among our linguistic capacities is our ability to **communicate** – which happens by and large even with bad spelling, bad penmanship, bad pronunciation, and bad "grammar". Our central linguistic skill is **understanding** phrases and phrase-composition, which is largely independent of sensibilities about well-formedness, broadly understood. In particular, semantic-composition allows meaning to find its way even when the expressions in question set off our syntax alarms, and even when these alarms "howl".

For example, speaking of howling, why is it that ‘*every dog of mine*’ is fine, but not ‘*every my dog*’, which has exactly the same meaning components, and which can be understood equally well with just a little effort. It's fine that our syntax rejects the latter, but this *alone* does not mean that our semantics should reject it too.
3. Semantic-Composition Is Hugely Expanded

According to orthodox semantics, following Frege, all composition is function-application. Following category theory in math, \(A \rightarrow B\) is the set of all functions from \(A\) to \(B\). Then Frege's idea is that an item of type \(A \rightarrow B\) (i.e., a function from \(A\) to \(B\)) combines with an item of type \(A\) (i.e., an element of \(A\)) to produce an item of type \(B\) (i.e., an element of \(B\)).

Given the way we propose to notate types, function-application looks like modus ponens. Our account of composition pursues this analogy, but includes compositions based on all valid argument forms, not just modus ponens. The obvious question then is which formal logic best models semantic-composition, and the answer appears to be Multi-Linear Logic.

Since composition corresponds to logical-derivation, the same input can give rise to many different outputs. Although the output is systematically constrained, it need not be unique – in contrast to how compositionality is usually understood.

4. Expanded-Composition Produces Over-Generation (which we cure)

Having such a powerful compositional machine has its drawbacks – over-generation. For example, an unwelcome result is that one meaning of ‘Jay respects Kay’ is that Kay respects Jay!

This is because we have in effect produced a "scrambling" language, which in turn underscores the fact that when we combine ‘Jay’, ‘Kay’, and ‘respects’, we have to know who is the subject and who is the object. This is accomplished in natural language by case-marking, which is done in various ways in various languages. So our theory needs case-marking.

We propose to use integers to encode cases, including ordinary cases such as
- nominative (1), accusative (2), dative (3), ablative (4),
- genitive (6), prolative (7), cumlative (8),
plus an extra "nullative" case, encoded by 0. We accordingly expand our types so that type \(D\) (domain, DP) may be case-marked to produce the derivative type \(D\theta\) for each case \(\theta\).

Many functions sub-categorize for case-marked input. For example, an ordinary VP sub-categorizes for a nominative-marked DP, and a genitive-noun (‘mother’, ‘father’, etc.) sub-categorizes for a genitive-marked DP. The following are examples.

- ‘is happy’ has type \(D_1 \rightarrow S\) function from nominative-marked DPs to sentences
- ‘respects’ has type \(D_2 \rightarrow (D_1 \rightarrow S)\)
- ‘mother’ has type \(D_6 \rightarrow (D_0 \rightarrow S)\)
- ‘woman’ has type \(D_0 \rightarrow S\)

Note carefully that, syntactically speaking, common-nouns are not functions; they do not take arguments. Nevertheless, semantically they are similar to predicates, so we treat them as predicates, although we distinguish them from genuine predicates (VPs and such) by employing the oddball case-marker 0.

---

1 Thus, I do not distinguish types from the sets of items that have those types. Further bear in mind that I switch between phrases and what they denote without comment, which is not good to teach intro students, but is convenient for me. Thus ‘D’ may mean determiner-phrase, or domain-entity, and ‘S’ may mean sentence or proposition (or perhaps truth-value). This is especially useful if we aren’t committed to whether we are doing direct or indirect semantics. Is a semantic-value an extra-linguistic object, or is it just another piece of language, albeit highly abstract?

2 This is our term for Linear Logic without Identity but with Conditional-Multiplication.

3 In a "logically perfect" language, case-marking is done entirely by word-order. Given how aggressive our theory of composition is, in semantically analyzing even such languages, we still must take "case" into account.

4 The Latin neologism ‘perlative’ is meant to suggest the word ‘per’, which is Latin for ‘by’ [among other things]. Similarly, ‘ablative’ suggests ‘ab’ [\(\approx\) ‘from’].

5 Formally, we have \(D_0 =_\approx D \times \{0\}\), an element of which is a pair \(\langle d,0\rangle\), where \(d\in D\). But we write the pair as \(d_0\).
5. Case-Marking Requires Expanded Lambda-Abstraction

As remarked earlier, our expanded types include $D_\theta$ for each $\theta$, and an item of type $D_\theta$ is an item $d_\theta$ where $d$ is an item in $D$, and $\theta$ is a case. In order to signify functions from $D_\theta$, it is convenient to expand lambda-abstraction. Here, although we don't have to, we choose to go all out – we allow lambda-abstact to take as input arbitrary open expressions (even lambda-abstracts!), not just variables. Note carefully, however, there are important restrictions on the application of lambda-conversion, since some such abstracts do not denote functions, but rather one-many relations.

As a simple example of expanded-abstraction, in the following expression

$$\lambda y_2\lambda x_1 [x \text{ respects } y]$$

the variables ‘$x$’ and ‘$y$’ have type $D$, but the compound expression ‘$x_1$’ has type $D_1$, and the compound expression ‘$y_2$’ has type $D_2$. Thus $[\text{respects}]$ takes a 2-marked entity and delivers a function that takes a 1-marked entity and delivers a proposition (or truth-value if you prefer). Notice that the case-markers behave a bit like post-its, and fall off in the proposition. Case-markers are ultimately merely part of the computational organization; post-its remind us where to put things, but having accomplished that they are discarded.

6. Open Expressions do not Have the Same Types as Closed Expressions.

For example, anaphoric pronouns do not have type $D$, and sentences containing unbound pronouns do not have type $S$. Remember, syntax does not rule!

Anaphoric pronouns do not denote entities, but rather create roles, which NPs fill, in precisely the same manner they fill ordinary functional roles (subject, object, etc.). Since there is no limit on how many such pronouns a phrase can contain, all with different antecedents, we posit infinitely-many such roles, which we propose to encode by negative integers. So in addition to nominative, accusative, etc., we now also have infinitely-many "negative" cases.

Consider the following example.

... respects his mother

This is syntactically a one-place predicate (a VP), and semantically it is also a VP provided the pronoun stem ‘he’ is demonstrative, but if ‘he’ is anaphoric, then semantically it is a two-place predicate – something like the following

$$\ldots \ldots \text{[1]} \text{ respects } \ldots \ldots \text{[-1]} \text{’s mother}$$

whose type is

$$\langle D_1 \times D_{-1} \rangle \rightarrow S$$

Often, the same object fills both roles, which means that it is marked both 1 and $-1$. So when we mark Jay, we have something like the following:

$$\text{Jay } + [+1] + [-1] = \text{Jay}_1 \times \text{Jay}_{-1}$$

where the latter is the ordered-pair consisting of Jay-marked-one and Jay-marked-minus-one.

Quantifier-phrases bind pronouns just like DPs do, although the type-theory is more complicated. For example, when we mark Jay both nominative and negative, we simply mark Jay, but when we mark a QP, the type-logic delivers something surprising, but welcome.

---

6 Here, we must resist seeing the subscripts as merely decorative, or as sortal modifiers. They are rather exactly like the super-scripts of arithmetic; ‘$x^2$’ is not a new kind of ‘$x$’; it is a compound expression in which ‘$2$’ actually refers to $2$.

7 An object cannot be marked with two theta-roles, but it can be marked with a theta-role and any number of alpha-roles.

8 As I use it, the cross-operator $\times$ applies both to semantic-objects and to types, so that (contrary to usual set-notation)

$$A \times B = \{ a \times b \mid a \in A \& b \in B \}$$

which means we have potential for fatal ambiguity, but context usually saves us.
In other words, the marked QP takes a two-place predicate and delivers a sentence – just what we want. We come back later to an example in which a QP binds a pronoun.

7. A New Theory of Quantifiers and Quantifier Phrases

Computations involving anaphora get gnarly for some QP examples, even grotesque. For this reason, but more for reasons involving empirical adequacy, we adopt a totally different theory of quantifiers and quantifier phrases.

According to the orthodox theory, a QP is a second-order predicate, and a Q is a function that takes a common-noun-phrase and delivers a QP. According to the revised account above, we add case-markers to the CNP [nullative] and the resulting QP [e.g., nominative].

According to the even-more-revised theory, a QP is an infinitary-operator. For example, ‘every’ denotes infinitary-conjunction, and the phrase ‘every man’ denotes the conjunction of all men

\[ \text{man}^1 \text{ and } \text{man}^2 \text{ and } \ldots \]

Here, I use super-scripts to distinguish enumeration from case-marking. Also, to avoid clutter, for the moment, I simply omit case-markers, leaving them understood.

Next, the rules of composition must be expanded, but the basic idea is that to combine a phrase with a conjunction is simply to combine it with all the conjuncts. So, for example,

\[ \text{man}^1 \text{ and } \text{man}^2 \text{ and } \ldots \]

combines with

is happy

to produce

\[ \text{man}^1 \text{ is happy and } \text{man}^2 \text{ is happy and } \ldots \]

On any given occasion in which ‘every man is happy’ is uttered felicitously, this is its content.

8. Pronoun-Binding

According to orthodoxy, as codified by Heim and Kratzer (1998), one can evaluate

Eros loves every woman

only by transforming this sentence into something approximating

every woman … Eros loves her.

The former and the latter are declared to be equivalent, appealing to the idea of logical form. Alas, simply declaring two bits of syntax to be equivalent begs the semantic question. After all, logical form is just how a sentence looks when it is translated into logic. The point of formal semantics is to prove
the translation actually works! In that case, we must prove they are equivalent, not assume they are equivalent, in which case the *ad hoc* transformation into logical form is not needed in the first place.\(^\text{14}\)

We have no need for Q-movement, since expanded-composition allows \([\text{loves}]\) to combine *directly* with \([\text{every woman}]\) to produce a VP. But it's even easier with the new "big" theory of quantifiers, according to which ‘every woman’ denotes the conjunction of all women

\[
\text{woman}^1 \text{ and woman}^2 \text{ and } \ldots
\]

Then ‘loves’ combines with this by combining with each conjunct, to produce

\[
\text{loves woman}^1 \text{ and loves woman}^2 \text{ and } \ldots
\]

Finally, ‘Eros’ combines with each of these conjuncts to produce

\[
\text{Eros loves woman}^1 \text{ and Eros loves woman}^2 \text{ and } \ldots
\]

Once again, I have left out the case-marking, but it is easy to reinstate.

Even after appealing to Q-movement to obtain

\[
\text{every woman... Eros loves her}
\]

Heim and Kratzer still need to invoke a monster, in the form of lambda-abstraction insertion. What is its type? What are the rules for composing it with other things? Have we abandoned type-semantics and reverted to semantics using syncategorematic devices?

We have no need for this further *ad hoc* device, given our account of pronouns. Supposing ‘her’ is anaphoric, and not demonstrative, ‘Eros loves her’ is not *semantically* a sentence; it is rather a one-place predicate sub-categorizing for a negative-marked DP. The composition is tabulated as follows.\(^\text{15}\)

\[
\begin{array}{ccccccc}
\text{every} & \text{woman} & [-1] & \text{Eros} & [+1] & \text{loves} & (-1) & \text{her} & [+2] \\
(D \rightarrow S) \rightarrow ([D \rightarrow S] \rightarrow S) & D \rightarrow S & D & \rightarrow D_1 & D_2 \rightarrow (D_1 \rightarrow S) & D_1 \rightarrow D & \emptyset & D \rightarrow D_2 \\
(D \rightarrow S) \rightarrow S & (D \rightarrow S) \rightarrow S & D \rightarrow D_1 & D_1 \rightarrow D & \emptyset & D \rightarrow D_2 \\
(D \rightarrow S) \rightarrow S & (D_1 \rightarrow S) \rightarrow S & D_1 \rightarrow (D_1 \rightarrow S) & D_1 \rightarrow S \\
\end{array}
\]

Notice that the anaphoric-markers have two forms – \((-k)\) role-creating, and \([-k]\) role-filling, which are inverses of each other. Notice also that many of the compositions are simply *modus ponens* [function-application], but others are more obscure, although they are all valid in Linear Logic.

This employs the conventional account of QPs as second-order predicates. The example can also be done using infinitary-conjunction, in which case ‘every woman \([-1]\)’ denotes:

\[
\text{woman}^1, \ldots \text{ and woman}^2, \ldots \text{ and } \ldots
\]

which combines with: \(\lambda x.[\text{Eros loves } x]\) to produce:

\[
\text{Eros loves woman}^1 \text{ and Eros loves woman}^2 \text{ and } \ldots
\]

\(^{14}\) You may surmise that I take logical form to be a false idol.

\(^{15}\) My tabulae are, of course, just trees reformatted upside-down and with the branches flattened.
9. There is just one Type of Noun

Also part of my overall theory is the proposal that nouns have one semantic-type,¹⁶ not two as generally thought [definite-nouns versus common-nouns]. This accords with ordinary English, in which a proper-noun can be used as a common-noun, and a common-noun can be used as a proper noun, and it accords with some really huge languages, including Latin, Russian, and Mandarin, which do not have articles, and in which a sentence like ‘dog is barking’ is perfectly ok.

The trick, of course, is to construct a general theory of how common-noun-phrases and article-headed-phrases combine with VPs to make a sentence. The basic ideas are two.

1. CNPs denote entity-sums, which often combine distributively with other phrases, and in a manner that sum often becomes disjunction/existential [sum becomes some!]

2. Articles ‘the’ and ‘a’ are semantically not determiners,¹⁷ but are rather adjectives, so ‘the dog’ and ‘a dog’ have the same type as ‘dog’. In particular, ‘the’ means ‘uniquely’¹⁸ and ‘a’ means ‘individual’.¹⁹

Then

a dog is barking

means

some individual-dog is barking,

and

the dog is barking

means

some uniquely-dog is barking.

The latter is just Russell's theory, which accordingly sits very nicely inside our theory of nouns.²⁰

Of course, ‘a dog’ can also appear after copular-be ["is of predication"]. But since [a dog] is a sum of entities, which sometimes behaves as a disjunction/existential, we can treat copular-be as transitive-be ["is of identity"]. One kind of noun, one kind of ‘is’.²¹

Sometimes nouns don't combine via existential quantification, but rather by universal quantification. This is accomplished because, in some contexts, logical-sum gets "promoted" to logical-product, which is in effect a super-wide-scope universal-quantifier. For example, in

if a body meet a body coming through the rye, if a body kiss a body

the first conditional promotes the first two sums to products. And in the example,

a dog can hear better than a human

the modality promotes the sums to products.

Sometimes logical-sums are not expanded at all, but remain "whole". For example,

humans like dogs

¹⁶ The exception are pronouns, which continue to be definite-nouns.

¹⁷ They are syntactically determiners, which explains why ‘the a dog’ sounds so bad – we eschew double-determiners.

¹⁸ Uniqueness seems more adverbial than adjectival. To be uniquely-P is to be the only P.

¹⁹ Note carefully that this account does not work for plural-nouns (’the dogs’) or mass-nouns (’the water’). For this we need to change ‘uniquely’ to ‘greatest’, following Godehard Link. This requires further explanation!

²⁰ Recall footnote 19. Russell's account must be adjusted to take account of plural-nouns and mass-nouns.

²¹ Plus the "is of existence" maybe!
is probably neither an existential nor a universal,\textsuperscript{22} but is rather more plausibly understood as a statement about humans-as-a-whole and dogs-as-a-whole.\textsuperscript{23}

Yet other times, both the distributive and the non-distributive readings are both plausible, as in:

\begin{quote}
I am looking for a small brown dog
\end{quote}

We can understand that this relates me to a particular small-brown-dog (my dog!), or alternatively to the sum of small-brown-dogs.\textsuperscript{24}

\textsuperscript{22} I say probably because the semantics permits a distributive reading as well as a non-distributive reading. Which reading we latch onto is not a purely semantic matter.

\textsuperscript{23} Key to this interpretation is the idea that the logical-sum of entities is yet another entity, which enables us to read ‘dogs’ as ‘dogs-as-a-whole’. Some verbs latch onto the whole-istic reading; others more naturally distribute over the class.

\textsuperscript{24} In either case, we are talking about actual dogs; we are speaking \textit{de re}. However, a more plausible interpretation is that this relates me to, not to an entity, whether a dog or dog-sum, but rather to a proposition or state of affairs – me finding a small brown dog. To look for a small brown dog is to seek [a world such] that I find a small brown dog. In this case, there are three readings – \textit{de re singular}, \textit{de re plural}, \textit{de dicto}. Also, notice that we must still say whether the seek-relation is distributive or not. It seems plausible that to want/seek/dread/… [a world such] that φ is not to want/seek/dread/… a particular world such that φ; rather one's relation is to the worlds as a whole.