Chapter 6: Montague Grammar

6.1 Introduction

Thus far, we have noted some fairly significant differences between a sentence's surface syntax and its underlying meaning. This poses a challenge for giving a systematic compositional account of natural language, since mismatches like these make it unclear how to give a rule-governed mapping of syntactic structures to meanings. Consider the fact that the meaning of all subject-predicate sentences does not simply consist in having an individual as the subject, a property as the predicate, and an attribution of that property to the subject. For example, the syntactic structure of the sentence 'No one is perfect' is that of a subject-predicate sentence. Yet, the meaning of that sentence does not consist in having the individual No One as its subject, to which we attribute the property of being perfect. But, if natural language is systematically compositional, we should expect the same rule to apply for composing the meaning of any sentence with the syntactic surface structure of a subject-predicate sentence. But, as we just saw, this is false. There are different ways to approach this mismatch problem.

One approach that we've seen, stemming from the theory of Transformational Grammar, attempts to eliminate the mismatch problem by positing various systematic rules that govern the mapping of a sentence's deeper syntactic structure to the complex meanings of complex expressions. This approach locates the mismatch as occurring between structures within natural language itself – between its surface syntax, and its deeper underlying syntactic structures. On this view, giving a compositional theory of natural language involves finding the underlying structure of a sentence and giving a systematic mapping of those structures to meanings. But this isn't the only way to approach the problem.

A second approach locates the source of the mismatch elsewhere – between the language being studied, and the language being used to represent the language under study. This approach emphasizes the fact that any theorist modeling natural language, uses formal invented languages to do so. Why use formal languages to model natural language? Well, because formal languages accomplish the goal of taking sets of expressions, and specifying a function that maps each expression to a specific meaning. The very same goal we have in analyzing natural language. Therefore, a formal language is exactly the kind of system needed in giving a theory of meaning. The only difference is that when using formal language to represent natural language, the functions that map expressions to a specific meaning must map natural language expressions to their intuitive meanings, or the meanings grasped by speakers of that language.

Historically speaking, however, formal languages have not been very good theories of, haves not produced a close mapping of, the structure of natural language. For this reason, some have thought that using formal languages to model natural language is simply a mistake. However, there is an enormous disadvantage to drawing that conclusion. We risk losing our only known means for giving a compositional theory of natural language. Formal languages are, by their nature, perfectly compositional. Therefore, if we can use a formal language to model the structure of natural language, we have no fear of failing to provide a compositional analysis of it. It would be guaranteed in virtue of the compositionality of the formal language that represents it. For this reason alone, it is worth considering all possibilities for using a formal language to represent natural language.

A more optimistic way of understanding the second approach to the mismatch problem is due to Richard Montague (1970). Instead of seeing the failure of a formal language to represent the structure of a natural language as a reason for rejecting altogether the use of formal languages in the study of natural language, Montague instead believed that any mismatches arising between the surface structures of natural language, and the meanings they are mapped to, via the formal language modeling it, simply signals the inadequacy of the particular formal language we happen to be using. There is, after all, more than one formal language. Once we see this, we might think that resolving the mismatch problem is simply a matter of getting the formal language correct. For Montague, once we find that correct formal language, its structure will correspond to the surface structures in natural language. Note that accepting Montague's view requires rejecting the project of Transformational Grammar as misguided, since if we had the correct formal language for modeling a natural language, there would be no need to posit underlying structures in order to explain the mismatch problem, since there would no longer be any mismatch to explain.

6.2 Montague's Project and His Theoretical Commitments

Before getting into the details of Montague's account, we need to examine the kind of approach Montague took to meaning in general. First, Montague was committed to characterizing meaning in purely truth-oriented terms, in which the meaning of a sentence is understood in terms of its truth value, and also in terms of the condition a sentence must satisfy, the way the world must be, in order for it to be true.

Second, Montague understood the fundamental mode of composition in natural language in terms of function application. That is, for any given complex expression, its meaning is the result of applying a function to the meanings of that expression's parts that allows us to combine those parts into a complex whole. For example, the meaning of a sentence like 'John loves Mary' is a function of the meanings of that sentence's parts, the meaning of ‘John’, of ‘Mary’, and of ‘loves’, as well as the order of the expressions in that sentence – whether ‘John’ comes first or second in that sentence clearly makes a significant difference to what that sentence means.

Compositionality and its preservation was central to Montague’s project. Note that in order to have a complete account of the compositional nature of natural language, it is not enough to simply provide rules for how to produce sentences compositionally. We also need an account of the structural relations between sentences in language; we need an account of the semantic entailment relations between sentences. A sentence **semantically entails** another sentence just in case the first sentence guarantees the truth of the second sentence. That is, one sentence entails another sentence just in case, in inferring the second sentence from the first sentence, there is no possibility of deriving something false from something true. Capturing these relations systematically is difficult, since many syntactically similar sentences seem to have different types of logical consequences. Furthermore, it is made even more difficult given that Montague had as his goal that these relations should be derived from the meanings assigned only to the surface structures of sentences alone. Take for example, the sentences 'Harry found a unicorn' and 'Harry seeks a unicorn'. Clearly, the first carries the implication that there is at least one unicorn, whereas the second does not. If compositionality is to be preserved, Montague must also account for these types of mismatches.

To capture the logical consequences of the previous sentences, Montague supplemented his theory with **meaning postulates** – rules about how interpret expressions in relation to certain other expressions. Specifically, in the previous case, the meaning postulates concern how to interpret sentences containing **propositional attitude** verbs, verbs whose meanings involve a speaker’s mental orientation towards a proposition, such as taking the attitude of belief to the proposition that it is raining, or the attitude of hoping that it is raining, or imagining that it is raining, and so on. For Montague, the reason that ‘Harry seeks a unicorn' and ‘Harry found a unicorn’ differ in their logical consequences is that the meanings corresponding to ‘seeks’ and ‘found’ express different relations to the meanings of the object phrase ‘unicorn’ when occurring in those sentences. The first sentence does not entail that there is at least one unicorn because the propositional attitude verb 'seeks' relates the meaning of a noun phrase like ‘unicorn’ only to a mental representation – to seek something like a unicorn, all that is needed is an idea of what is being sought. In contrast, the propositional attitude verb ‘found’ requires something more. It must relate the meaning of a noun phrase like ‘found’ to something existent. In this case, its truth would require the existence of a unicorn.

Of course, nothing in the syntax rules out interpreting ‘seeks’ in the same way as ‘found’, only Montague’s meaning postulates rule this out, and for this very reason, Montague’s meaning postulates create barriers to reaching his ultimate goal. First, in the absence of an independent motivation for meaning postulates, Montague’s theory seems ad hoc – that the postulates were introduced specifically for the purpose of making the framework systematic without any further justification for them. Second, the introduction of these postulates undermines Montague’s entire project of giving a compositional theory of natural language that relies only on the resources of its surface structure and the meanings of its parts. It now relies on these plus some additional meaning postulates.

Most contemporary theorists of formal semantics explain the different logical entailments of sentences containing verbs like ‘seeks’ and ‘found’ by building the features that make for those differences into the meanings of the words themselves – into the basic lexicon. So, for example, it is part of the meaning of 'seeks' that it is not a relation between an individual and another individual, rather it is a relation between an individual and something more abstract, like a mental representation. And, this is false for the word ‘found’. If this is an acceptable move, we can have the compositionality Montague strived for, since the basic parts of the system — the meanings of the words themselves — encode the features of those expressions that led Montague to introduce his meaning postulates in the first place. And compositionality does not dictate the meanings of the basic parts of the language, only how those parts must fit together to produce complex wholes. So, by building the differences between ‘seeks’ and ‘found’ into the meanings of the expressions themselves, we can now offer a non-ad-hoc compositional theory, as well as one that respects Montague’s way of thinking about this – that in offering a compositional theory of natural language, we need not and should not appeal to any structures or rules in addition to that governing surface syntax in getting the interpretations of sentences and their logical entailments correct.

In addition to his general commitment to a compositional account of natural language, Montague had another theoretical commitment that specified what counts as a compositional system. In Montague’s framework, for every syntactic rule of combination that produces a certain kind of grammatical string, there corresponds a semantic rule that gives that grammatical string a specific kind of meaning. For example, consider the syntactic rule that says that combining a proper name PN with a verb phrase VP results in a sentence. The corresponding semantic rule would specify the kind of meaning produced by that specific kind of syntactic operation. For example, in Montague’s system, our semantic rule would tell us that combining the meaning of a proper name with the meaning of a verb phrase produces a truth bearer as its meaning, something we can evaluate for truth. This is known as the **rule-**

to-rule hypothesis (Bach, 1976). While, on some accounts, compositionality does not entail the rule-to-rule hypothesis, the hypothesis does entail compositionality. If the rule-to-rule hypothesis is correct, we could, then, give a fully compositional account of language.

We now have a grasp of the general commitments Montague held concerning what it is to give a compositional account of natural language. The next step is to examine the formal language Montague used to model the structure of natural language.

6.3 The Formal Language of Montague Grammar

Montague's formal system, known as Montague Grammar, is widely understood as potentially eliminating the mismatch problem, since it does, in fact, yield a systematic mapping of at least some of the surface syntax of natural language to a complex expression's meaning**.** Specifically, Montague gives a compositional account of the meaning of quantifier expressions, something that, up until that point, had been rather elusive.

As before, we discussed the idea that the mismatch problem is simply an artifact of choosing the wrong formal system to model natural language. Montague, recognizing this, sought to rely on a formal language that could capture the structure of natural language. For Montague, this involved using what is known as the **lambda calculus**, developed by Alonzo Church in the 1930's, which is concerned with the nature of the relation between arguments and functions. Montague rejected the standard practice of using **first order classical predicate logic** to model natural language, a logic primarily concerned with the properties of particular individuals.

In addition to the difference in focus between the lambda calculus and first order predicate logic, yet another difference between them is that the lambda calculus is a species of what we might think of as a **hyperintensional logic**, a logic that expands the domain of meaning beyond both extensions, and intensions, which standard classical logic does not do.

The **extension** of an expression is simply the individual(s) to which an expression applies. For example, the extension of the expression 'dog' is composed of the individuals in the actual world that are dogs, and the extension of the expression 'Fido' is the actual individual Fido. The **intension** of an expression is that aspect of an expression's meaning that gives its definition. For example, the definition of 'dog' is not simply individuals that are dogs, but some kind of animal with a certain genetic structure. Intensions are often thought of as functions that when, applied to an expression, yield that expression's extension, but are nevertheless not identical to its extension. Specifying the exact nature of an intension is notoriously difficult, since it is difficult to say exactly what the domain of a function serving as an intension could be. On the most widely accepted view, an intension is a function that maps an expression to all of its possible extensions. So while the extension of the expression 'dog' is composed of the individuals that are actual dogs, its intension includes much more than this; it includes not only the actual dogs, but also the possible dogs as well. This makes sense. When we want to know the definition of an expression, we want to know the necessary and sufficient conditions for its application. We don't just want to know what it happens to apply to, but what it must apply to. Our definitions of expressions ought to capture the essence of what is to be a dog. To see why we need intensions, imagine that all of the actual dogs have tails. It seems to follow, on a purely extensional picture of meaning, that part of the meaning of the word 'dog' is having a tail. But we know this is false. Dogs can lose their tails and still be dogs. Appealing to intensions can explain this fact. The meaning of the expression 'dog' is its intension and its intension includes dogs without tails as well as those with tails since there are dogs that possibly lack tails. While the lambda calculus is more than an intensional language, it is at least that.

To see another reason for why we need a logic that is intensional, consider the following puzzle: if all there was to the meaning of an expression was an extension, we could not distinguish between the meanings of 'renate' and 'cordate', since these designate the very same individuals in the actual world – they are extensionally equivalent. But these expressions do, in fact, have different meanings. As we've seen, a solution is to extend meaning beyond extensions to their intensions. However, as it turns out, not even intensions can fully distinguish the meanings of all expressions.

On the intensionalist picture, two expressions have the same meaning just in case those expressions have the same extension in every possible world, not just in the actual world. That is, they have the same meaning just in case they necessarily designate the same individuals. This way of thinking of meaning resolves the problem with thinking of meaning as purely extensional in nature, since the expressions 'renate' and 'cordate' will differ in meaning, given that it is possible for a renate to fail to be a cordate. That is, there is at least one possible world in which renates fail to be cordates. On an intensionalist picture, this would show that these expressions differ in meaning.

However, there are some things that the standard intensionalist framework cannot accommodate: expressions that have different meanings but that necessarily pick out the same individuals – that pick out the same individuals in every possible world. For example, consider the expressions '2+2' and 'the square root of 16'. These two expressions will pick out the number 4 in every possible world. That is, these expressions have the same extension in every possible world, namely, the number 4. Because these expressions have exactly the same extension in every possible world, it follows, on an intensionalist picture, that they must have the same meaning.

It seems, then, that only a system that goes beyond intensions, a hyperintensional system, like the lambda calculus, can distinguish the meanings of these two expressions. While Montague did rely on the lambda calculus as a formal language for modeling natural language, he did not exploit its resources for dealing with the problems created by the standard intensional account. As a hyperintensional account, the lambda calculus is based on treating meaning as consisting in sets of rules for deriving them as opposed to merely their intensions and their determined extensions. While these rules might yield the same intensions for two expressions, this does not entail that those expressions have the same meaning. For instance, if we rely on the idea of rules for deriving meanings, we can distinguish the difference between the expressions ‘2+2’ from ‘the square root of 16’ since the rules used in deriving the meanings of these two different expressions will differ significantly, given the different mathematical operations we would have to apply to yield the number 4. Unfortunately, further detailed explanation of the lambda calculus, and its attendant notion of hyperintensionality, remain beyond the scope of this book. What is essential for our present purposes is to understand only that it differs from classical logic, and how it does so, not by being merely extensional, or evem intensional, for that matter,, but instead by being hyperintensional.

6.4 The Syntax and Semantics of Montague Grammar

The nature of the compositional theory offered by Montague involves subscribing to grammatical types – sentences, verbs, adjectives, and so on. Montague Grammar then is a sophisticated version of Phrase Structure Grammar also known as Categorial Grammar, since it begins by categorizing expressions as belonging to certain grammatical types. These grammatical types correspond to **semantic types** – the meaningful units apt for combination, whose rules for combination correspond with the syntactic rules for combination in a natural language.

Before we can grasp the nature of Montague's semantic types, we must first distinguish two separate components of meaning, important for understanding how to apply Montague's theory. While Montague himself did not explicitly distinguish these components in the way we will, we can safely deploy them in explaining his work given Frege's influence on his work. In addition, Montague himself must have recognized the problem of making a sentence's meaning consist only in its truth value. Namely, that it entails that all sentences with the same truth value have the same meaning. And, of course, this cannot be correct. A solution, Frege's solution, is to introduce two different component's to an expressions meaning.

The first component of an expression's meaning is its **semantic value** – how that expression determines the truth value of a sentence of which it is a part, or in the case of a sentence, its truth value. Returning to our previous example of the sentence 'John loves Mary', the semantic value of the expression 'loves Mary' is a function that takes the semantic value of an expression like 'John', an argument expression, and maps it to either true or false. A predicate like 'loves Mary' then contributes to a sentence's truth value in virtue of its being a function whose output is a truth value. The subject of the sentence 'John' contributes to the truth value of a sentence by providing the input for a predicate's semantic value – an argument – which then allows us to derive a truth value for that sentence.

The second component of an expression's meaning is its **semantic content** – what that expression contributes to the way the actual world, or the way possible worlds must be, in order for a sentence of which it is a part to be true, or in the case of a sentence, its truth condition. Semantic content, then, corresponds with either its extensional content, or both its extensional and intensional content, depending upon whether one accepts a Russellian or a Fregean theory of semantic content. In the case of the sentence 'John loves Mary', a simple understanding of the semantic content of 'loves Mary' is the property of loving Mary, and the semantic content of 'John' is the individual John. The whole sentence 'John loves Mary' is true just in case John, the individual, has the property of loving Mary.

Semantic values are related to semantic content in virtue of the fact that semantic value encodes the formal semantic structure for representing a sentence's semantic content. Both concepts are important for understanding an expression's meaning, since simply understanding a sentence's semantic value would tell us only about its functional role, which by itself, is not a complete account of an expression's meaning. For example, we might know that the semantic role of an expression like 'John' when it occurs in the sentence 'John loves Mary' is that of providing the argument for the predicate 'loves Mary', and that this is its role in producing a truth value for that sentence – that of providing the input for an instance of applying the function expressed by 'loves Mary' in order to produce an output, a truth value. This tells us a lot about the semantic composition of a sentence like 'John loves Mary', and the role each of its parts plays in that composition, but it does not tell us everything we would need to know to fully understand what is being expressed by the sentence 'John loves Mary'. To fully understand that sentence, we need to also understand what information it conveys, understood standardly as that sentence's truth condition.

Montague Grammar, since it is supposed to model the formal structure of natural language, is an account of the semantic values of types of expressions and the rules governing their combination. Like Frege, Montague is committed to the idea that a sentence is fundamentally made up of two parts: a subject and a predicate, whose semantic values, as before, respectively correspond to an argument and a function. Since functions and arguments naturally combine, the idea that the fundamental principle of natural language composition is a matter of function application is a promising strategy for explaining compositionality.

Given Montague's understanding of compositionality, and his commitment to the idea that the semantic value of a sentence is its truth value, each semantic type must belong either to the category of an argument, a function, or a truth value. In Montague's system, these basic semantic types are known respectively as either **e-type** expressions, **<e,t>-type** expressions, and **t-type** expressions.

Expressions of type e are expressions whose semantic value provides the basic or atomic inputs for functions – arguments – that cannot themselves be understood in functional terms, as having any complexity or structure. In classical logic, these would correspond to the logical constants of a language. Grammatically, these expressions correspond to simple subject expressions such as proper names or PN's. A natural understanding of the semantic content of an expression of type e would be specific individuals – the basic atoms of the world. For example, 'Fido' has the semantic type e, and its semantic value is the basic argument with which the expression 'Fido' is associated. The semantic content of 'Fido', however, is not an argument, but rather is the individual Fido.

In contrast, expressions of type <e,t> are those expressions that take basic arguments as inputs and map those basic arguments to truth values. Grammatically, these expressions correspond to the predicative element of a sentence. Their semantic values are functions, expressions whose semantic value has structure. The traditional corresponding semantic content of an expression like this is that of a property, which concerns a complex state of affairs involving certain individuals being certain ways. For example, a predicate like 'barks' is assigned the semantic type <e,t>, which is explained as a function that takes basic arguments and maps them to a truth value dependent, in this instance, upon whether the predicate 'barks' does or does take a particular basic argument to the value true. The corresponding semantic content for the predicate 'barks' would be the property of being a barker.

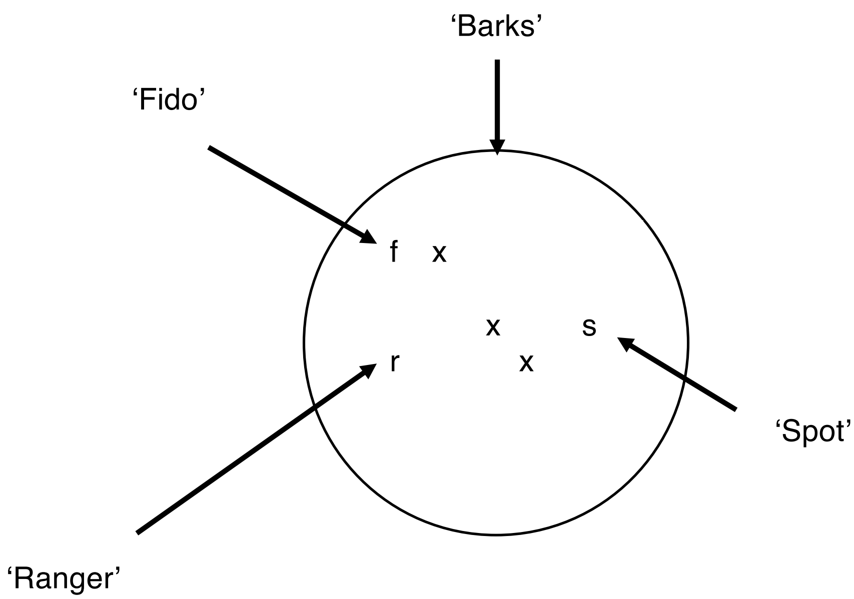
Combining the previous two semantic types produces a t-type expression – the basic outputs resulting from taking the semantic value of a predicate and applying it to the semantic value of a simple subject. Expressions of type t are expressions that have truth values as their semantic values, and that have as their semantic content, a truth condition. Grammatically, expressions belonging to this semantic type correspond to subject-predicate sentences. For instance, the sentence 'Fido barks' is assigned a truth value that is determined by whether the basic argument – the subject – supplied by the expression 'Fido' is one that the function associated with 'barks' – the predicate – maps to true. The semantic content of the sentence 'Fido barks', in contrast, would be that it is true just in case the individual Fido has the property of being a barker.

6.5 Applying Montague Grammar

To make the concepts here more accessible, we will represent semantic values in set theoretic terms. For instance, we can think of a predicate, whose semantic value is a function, a set of **ordered pairs**, sets for which the order of each member matters. In the case of a predicate, its semantic value is a set of ordered pairs of basic arguments and truth values. For any given argument, a predicate will pair that argument with either the value true or the value false. It is easy then, assuming there are only two truth values, to think of the semantic value of a predicate as the set containing all and only those arguments that pair with the value true. On this picture, the corresponding semantic content of a predicate is the set of all of the semantic contents of basic arguments – of individuals – that have the property corresponding to the predicate.

To make this more concrete, let's look at one example of how the previous conception applies to a simple sentence like 'Fido barks.' Syntactically, an item like 'Fido' can combine with other items like 'barks' to form a complete sentence. The generic semantic value of an e type expression is a variable x for which we can substitute logical constants – atomic elements of a system that ultimately have individuals as their semantic contents. For example, an expression like ‘Fido’ is the argument, or logical constant, to which an expression like ‘Fido’ applies, represented as ‘f’. The semantic value of a predicate like ‘barks’, now understood in set theoretic terms, is the set of arguments which ‘barks’ maps to the value true, or to which ‘barks’ applies. For example, ‘barks’ might apply to arguments provided by the expressions ‘Fido’, ‘Ranger’, and ‘Spot’, but also many others. Figure 6.1 represents the semantic value of both argument expressions like ‘Fido’ as well as predicate expressions like ‘barks’, where the arrows are intended to be pointing to the semantic value of the expressions under analysis.

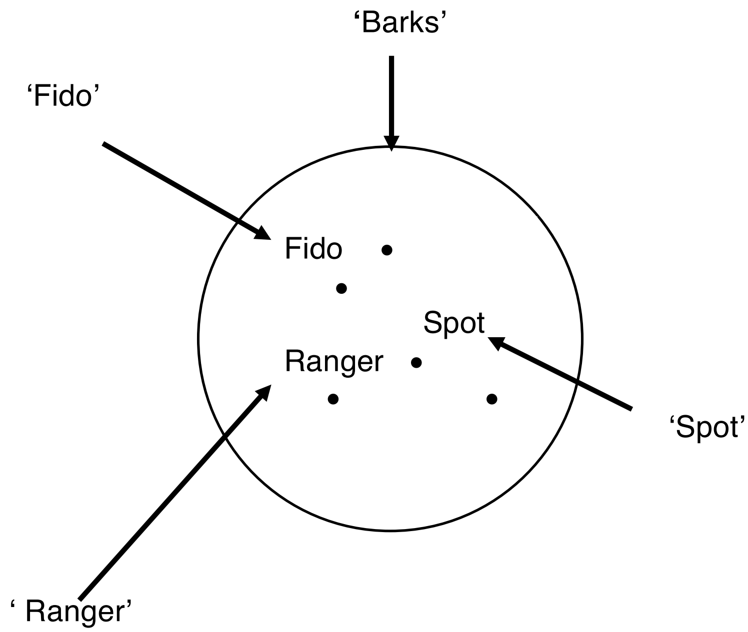
Fig. 6.1



We can now understand the semantic value of the sentence ‘Fido barks’ as true just in case the argument serving as the semantic value of ‘Fido’ is a member of the set of arguments to which ‘barks’ applies.

The semantic content of a predicate, of say a predicate like ‘barks’ would be the set of individuals, like Fido, Ranger, and Spot that bark, as we all many other individuals which for which we will use • as a variable that ranges over actual individuals as Fig 6.2 illustrates. The arrows in this case point out extensions associated with various expressions.

Fig. 6.2



The semantic content of the sentence 'Fido barks' then is that it is true just in case Fido is member of the set of individuals that bark.

It seems, however, that there must be more semantic types than only those we have discussed. Take, for instance, the sentence 'All dogs bark'. Before Montague's formalization of some of Frege's ideas, most treated the semantic value of a quantifier like 'all' as an operator that took the semantic value of two predicates as arguments, and produced a truth-value. To give a more explicit characterization, relying on set theoretic notions, the meaning of the quantifier 'all', for instance in the sentence 'All dogs bark', was believed to take the semantic value of the expression 'dog' and the semantic value of 'barks' and represent the degree of overlap between the members of these two sets. A sentence such as 'All dogs bark' would be true just in case the arguments contained in the set to which 'dog' applied are also all members of the set that map true. In our previous notation, the quantifier in the sentence

‘All dogs bark’ would have as its semantic type **<<e,t>,<<e,t>,t>>** – a function that takes two ordered pairs as an argument and maps this argument to a truth value.

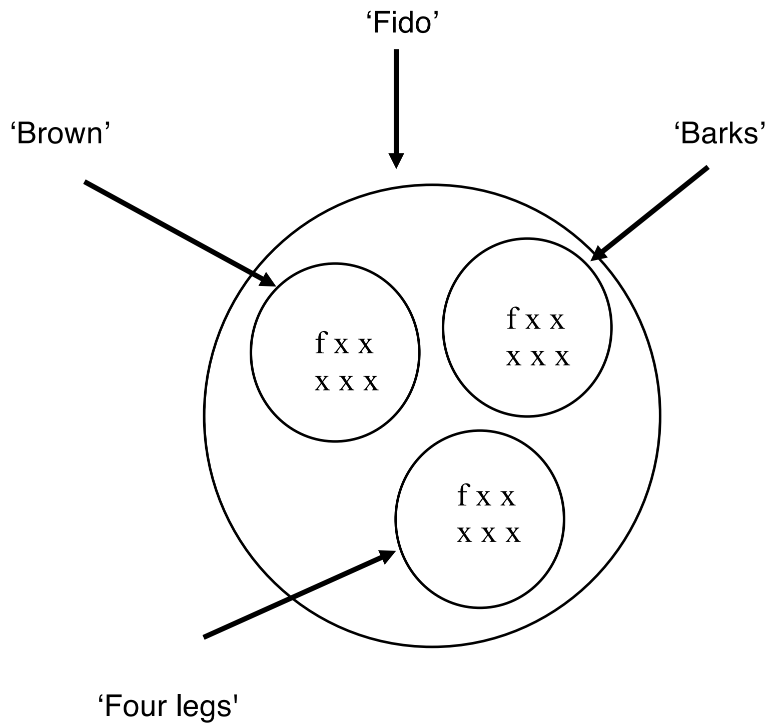
But where did the semantic type <<e,t>,<<e,t>,t>> come from? While of course, we can construct any type we like, doing so leaves us with a fairly vacuous account of compositionality. In addition, it is not in keeping with the basic form of composition proposed initially by Frege and developed by Montague — a kind of function application in which expressions of type e – basic arguments – are the arguments for expressions of type <e,t>. Additionally, and most importantly for Montague, it does not accurately represent the surface syntactic role that 'all dogs' plays in the sentence 'All dogs bark'.

Syntactically, 'All dogs' appears to be similar to expressions that belong to the semantic type e — they can both occur in the subject position in subject-predicate sentences. But expressions of type e supply arguments that are not functions or sets as their contribution to the truth value of a sentence – they are not complex. In contrast, the semantic value of ‘All dogs’ clearly does appear to be a set – the set of arguments to which the expression ‘All dogs’ applies, which is complex. The semantic contents of these types also appear to be distinct. As we know, expressions of type e like ‘Fido’ typically have as their semantic content a particular individual Fido, whereas the semantic content of an expression like ‘All dogs’ is not a specific individual, but is instead the group of individuals that have the property of being a dog.

Once again, we have a mismatch between surface structure and the kinds of meanings sentences have that threatens to undermine a commitment to compositionality. Montague’s innovation was to treat both expressions of type e and quantified expressions as of the same semantic type. In particular, rather than think of subjects in a subject-predicate sentence as having arguments as their semantic value, we should instead think of them as having functions as their semantic value. That is, quantifier phrases and expressions of type e, both of which occur as subjects in subject predicate sentences, are really expressions whose semantic role is to take the semantic values of predicates as arguments. On Montague's picture, then, expressions once thought of as of belonging to the semantic type e are really expressions that belong to the semantic type <<e,t>,t> -- an expression that takes expressions with the semantic type <e,t> and maps them to an expression of the semantic type t. That is, expressions of type e are now understood as expressions that take functions as arguments, and produce truth values as a result – itself a function. This is likewise true for quantified expressions.

But does this work for the case of an expression of type e, previously understood as associated with a basic argument that is not itself complex? Well, if we are going to give the same semantic treatment to both expressions of type e and quantifiers, then we must be able to think of them as corresponding to a set of some kind. The idea is that we can treat what once served as the semantic value of expressions of type e as a set of various sets of basic arguments. Which sets of arguments? Those sets for which the semantic value of an expression of type e, a basic argument, belongs. Consider again the semantic value of 'Fido' as a basic argument f. We just now described a very different way of thinking of the semantic value of 'Fido'. Not as providing a basic argument, but instead as a providing a set of the semantic values of predicates, or sets of basic arguments. What kinds of sets of arguments would characterize the semantic value of 'Fido'? Well, all of those sets for which the original semantic value of Fido – the basic argument with which 'Fido' is associated – is a member. In other words, the semantic value of an expression like 'Fido' is equivalent to the set of arguments that get mapped to true by, say, the barks function, that get mapped to true by the brown function, and that get mapped to true by the four-legged function. Fig 6.3 illustrates what this means more clearly.

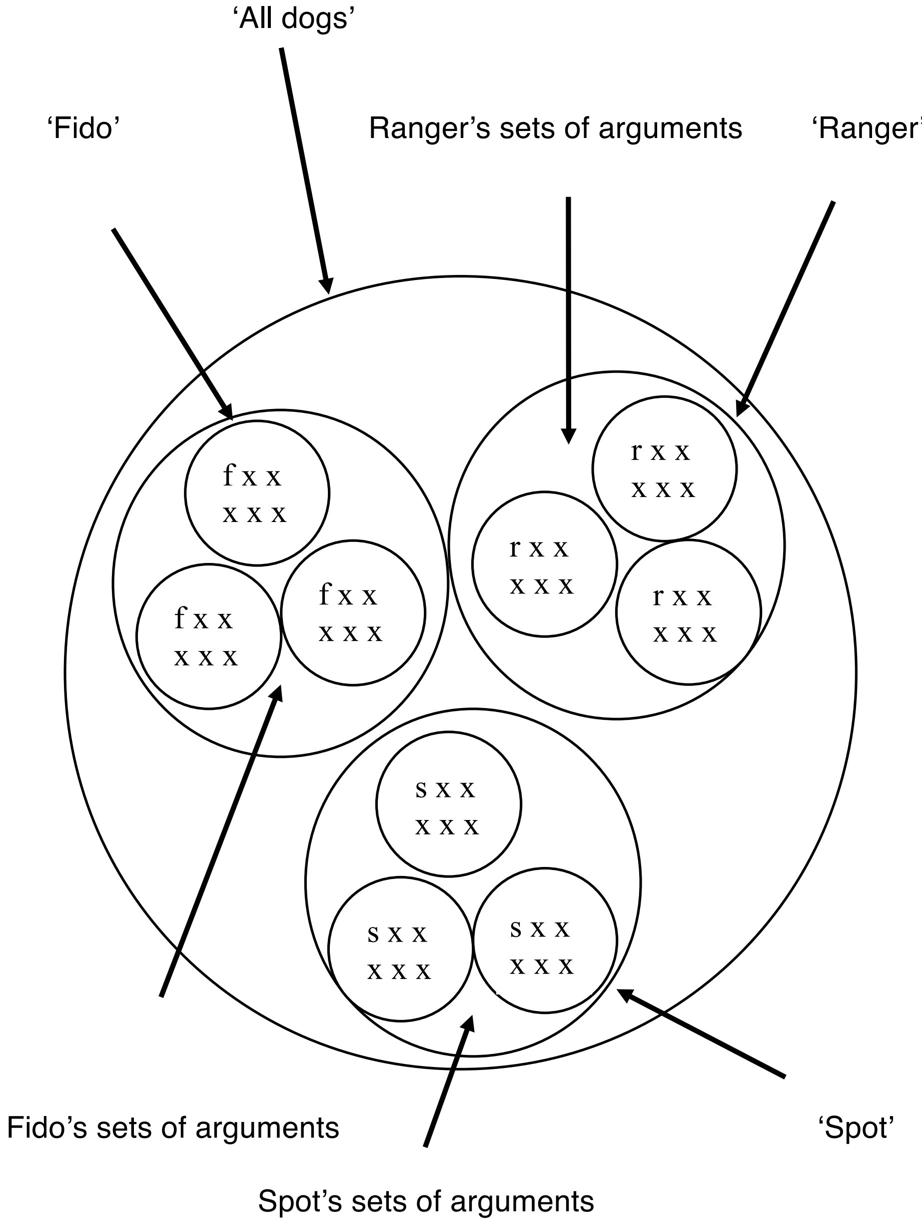
Fig. 6.3



The semantic content of ‘Fido’, on this picture, would be the sets of individuals that those individuals are a member of. That is, the semantic content of ‘Fido’ would be equivalent to the set of properties that characterize the individual Fido.

The semantic value of a quantified expression, on Montague’s picture, is also a set. However, unlike proper names, quantified expressions are not simply associated with a set of sets of basic arguments, but are instead a set of sets of sets of basic arguments. For example, the expression ‘All dogs’ would be associated with the semantic value of each dog, which is now understood as a set of sets – the set of sets of basic arguments that characterize what was formerly an e type expression. A quantified expression like ‘All dogs’ then would be a set of each set of sets that characterize each and every dog – a set of sets of sets. This way of thinking of the semantic value of 'all dogs' can be illustrated with the following diagram:

Fig. 6.4



As you can see, the semantic value of 'all dogs' is just the set of the sets of sets of arguments that characterize the semantic value of each dog. The semantic content of an expression like 'All dogs' is the set of sets of sets of individuals, or properties, that characterize each dog.

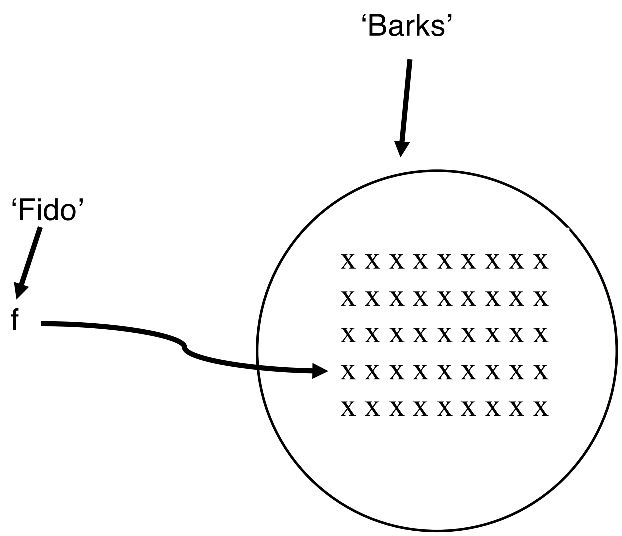
In the case of both ‘Fido’ and ‘all dogs’, we can now see that each has as its semantic value a set. In the first case, a set of sets of basic arguments, and in the second, a set of sets of sets of basic arguments, but individual sets nonetheless. This makes 'Fido' and 'all dogs' both of the same semantic type.

To summarize, Montague's solution to the threat posed to compositionality by the mismatch problem was to develop a uniform semantics for all noun phrases, which corresponded to its surface syntax, so that the logical form of every sentence is encoded in its surface syntax. Both the noun phrases 'Fido' and 'all dogs' play the same syntactic and semantic roles in the sentences in which they occur, thereby respecting the standard definition of compositionality originally formulated in Frege’s work.

6.6 Evaluating Sentences: Truth Values and Truth Conditions

Two questions now arise that concern Montague’s proposed semantic value for a specific kind of e-type expression: a proper name. That is, how do we assign sentences containing proper names truth values? Montague cannot rely on the idea that a sentence like ‘Fido barks’ gets the value true just in case an argument associated with ‘Fido’ is a member of the set of arguments that map the Fido-argument to true. But, at the level of semantic content, this makes sense; our intuitive grasp of what makes it the case that Fido barks is just that Fido has the property of barking, previously understood as the set of individuals that bark. But, recall, Montague rejects the claim that the semantic value of a proper name like ‘Fido’ is a basic argument. So, he must give a different account of what makes a sentence containing a proper name true. Figure 6.5 illustrates the standard account of predication for evaluating the semantic value of a sentence like ‘Fido barks’, where the wavy arrows indicate what would have to hold for the sentence to be true. In figure 6.5, for instance, it would have to be the case that the Fido argument belongs to the set of barks arguments.

Fig. 6.5

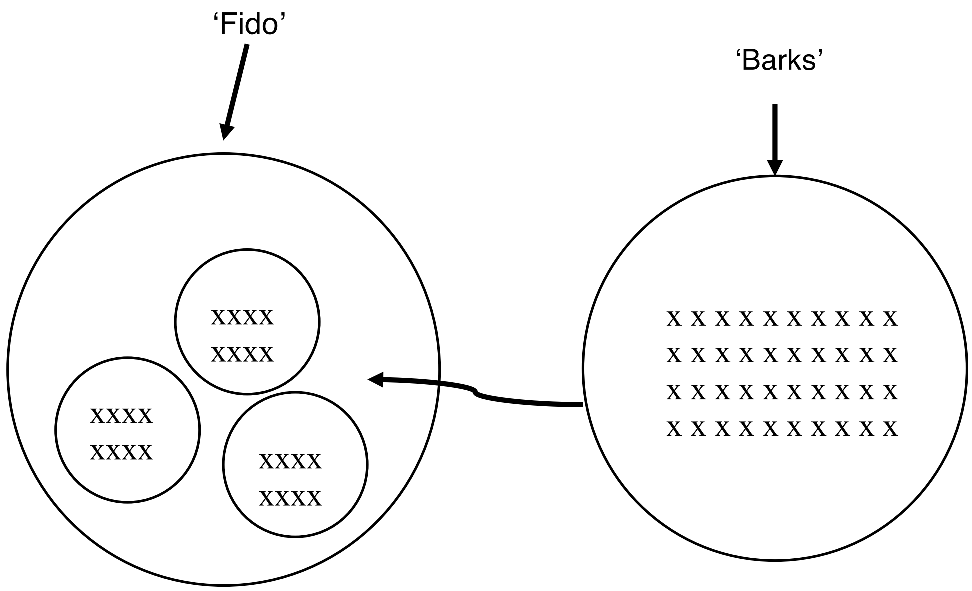


So, what is Montague's answer to the preceding issue? First, Montague is not giving a theory of what our utterances, or our uses of sentences, are intuitively about, but a theory of the semantic value of expression types. You can maintain that an utterance of 'Fido' on a given occasion refers to Fido, but nevertheless reject that idea when talking about the semantic value for the expression type 'Fido'. In other words, Montague is not giving a theory of the contents of our utterances in conversational contexts, but rather he is explaining the role that expression types play in a compositional semantic theory. The second question, however, still looms large. Montague simply cannot rely on our simple notion of predication, on the idea that a sentence like 'Fido barks' gets the value true just in case the argument provided by 'Fido' is a member of the set of arguments that maps those basic arguments to the value true.

Montague's answer is this: the semantic value of a sentence like 'Fido barks' is true just in case the semantic value of 'Fido', now the basic set of arguments that map the basic

Fido argument f to true, maps the semantic value of 'barks' to the value true. That is, the sentence 'Fido barks' gets the semantic value true, just in case the set of arguments to which 'barks' applies belongs to the set of arguments that characterize the semantic value of 'Fido'. Figure 6.6 illustrates this nicely.

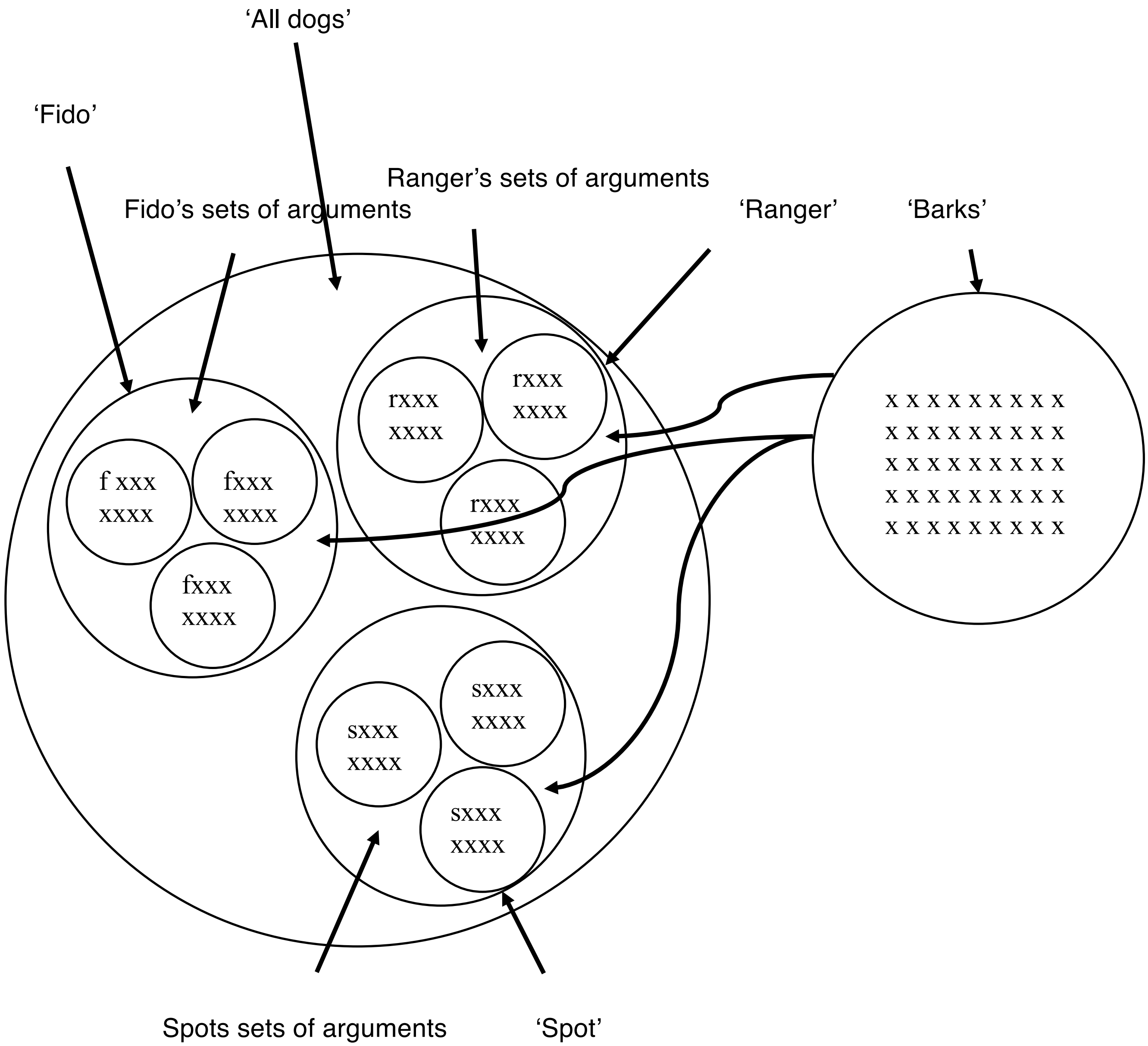
Fig. 6.6



The above diagram shows that the sentence 'Fido barks' gets the semantic value true just in case the set of arguments to which the bark-function applies is one of the sets of arguments that constitute the semantic value of 'Fido'. The truth condition for the sentence 'Fido barks' is that it is true just in case the semantic content of the predicate 'barks' – a property or set of individuals – is a member of set of properties, or sets of individuals, that constitute the semantic content of 'Fido'.

And, of course, Montague's system treats predication uniformly, so that a sentence like 'All dogs bark' gets the semantic value true just in case the semantic value of 'barks' is a member of each of the proper subsets that constitute the semantic value of 'all dogs' as illustrated figure 6.7 illustrates.

Fig. 6.7



In this diagram, we can see that the sentence 'All dogs bark' is true just in case the semantic value of each proper name PN above contains the set of arguments that the barks-function maps to true. In other words, for every set of arguments that characterize the semantic value of 'Fido', 'Ranger', and 'Spot', the set of arguments, which characterize the semantic value of 'barks', must be a member of each of their semantic values, if the sentence 'All dogs bark' is to count as true. The semantic content of the sentence 'All dogs bark' is that it is true just in case the set of individuals that bark – the property of being a barker – is a member of each set of properties that characterize each of the dogs.

While Montague's theory of predication represents a significant departure from standard accounts of representing how predication works, it is not at all clear that it does any damage. After all, it's not as if the standard account is itself intuitive. It's a set theoretic account of the nature of predication, something that requires quite a bit of background to understand. Montague's proposal concerning how to evaluate sentences like 'Fido barks' for truth is also not unintuitive for the same reason. Neither of these accounts encodes an intuitive notion of predication. Our intuitive notion of what it is to predicate involves property attribution. A sentence like 'Fido barks' is true just in case the attribution of the property of barking to Fido is correct. This is not particularly enlightening, and it does not entail a commitment to the standard account of predication or to Montague's notion either. As far as our intuitions are concerned, either picture could be said to be a way of individuals can have properties, and they both involve the application of set theory, relying on the notion of set membership to represent predication.

6.7 Concluding Remarks

Montague, then, is in many ways largely responsible for the current understanding in linguistics and philosophy of how semantic theory works formally. In fact, it is Montague's formal apparatus that is currently used for deriving the meaning of sentences.

We have now spent a considerable amount of time discussing the rules and structure of a language, and have examined a few proposals about how this might work. Our next chapter will explore the question of the nature of a rule and whether language should be thought of as rule-governed at all.

Main Points to Remember

- Mismatches between the surface form of sentences and their meanings could be a product of using the wrong logic to represent their logical form.
- Montague offered a system of logic that allows us to eliminate at least some mismatches between surface forms and meanings.
- Montague developed a specific type theory of linguistic expressions modeled on Frege's initial understanding of the rules of combination as instances of function application.
- In application, Montague offered an analysis of noun phrases that made them of the same semantic type (e.g., the phrases 'Fido' and 'All dogs').
- On Montague's view, all noun phrases are understood as sets of sets, or as sets of sets of sets.
- Instead of understanding sentences as being formed by expressions in which predicates semantically take individuals as arguments, Montague instead understood them as formed by expressions in which predicates semantically serve as arguments to noun phrases.
- This required a modification of our standard notion of predication.

Suggested Readings

Bach, Emmon. 1976. "An extension of classical transformational grammar", in Problems in Linguistic Metatheory (Proceedings of the 1976 conference). pp. 183–224. East Lansing, Michigan: Michigan State University.

Barwise, J. and Cooper, R. 1981. "Generalized quantifiers and natural language." Linguistics and Philosophy, 4: 159–219. Reprinted in Portner and Partee (eds.) 2002, pp. 75–126.

Carpenter, B. 1998. Type-Logical Semantics. Cambridge: MIT Press.

Gottlob, Frege. 1980. Translated as ‘Function and Concept’ by P. Geach in Translations from the Philosophical Writings of Gottlob Frege, P. Geach and M. Black (eds. and trans.), Oxford: Blackwell, third edition.
————.1974. Translated as The Foundations of Arithmetic: A logico-mathematical enquiry into the concept of number, by J.L. Austin, Oxford: Blackwell, second revised edition.
Lohndal, Terje. 2014. Phrase Structure and Argument Structure: A Case Study of the Syntax-semantics Interface. Oxford Studies in Theoretical Linguistics (49).

Jackendoff, R. S. 1972. Semantic Interpretation in Generative Grammar. Cambridge, MA: MIT Press.

Jacobson, P. 2014. Compositional Semantics. An introduction to the syntax/semantics interface. Oxford: Oxford University Press.

Montague, R. 1973. The proper treatment of quantification in ordinary English. in K. J. J. Hintikka, J. M. E. Moravcsik, and P. Suppes (eds.). Approaches to Natural Language (Synthese Library 49), Dordrecht: Reidel, 221–242.

Partee, B. H. and Hendriks, H. 1997. Montague grammar. In van Benthem and ter Meulen (eds.): 5–91.

Peters, Stanley. and Dag Westerståhl. 2008. Quantifiers in Language and Logic. Oxford University Press.

Zimmermann, E.. 1999. “Meaning postulates and the model-theoretic approach to natural language semantics.” Linguistics and Philosophy, 22: 529–561.