Introduction

We know a lot about the semantic structure of natural language. If you think that semantics is about the connection between linguistic entities, words, sentences and such, and non-linguistic entities, things 'in the world', then the way to do semantics will be to specify the linguistic entities, then specify the 'things in the world' which are to be their meanings, and then connect up the two. Chapters 1 and 2 illustrate this by setting out in detail a very simple formal language together with its semantics. This semantics allows us to address the philosophical issues with a particular example in mind. (These chapters can be omitted by those who already have a background in formal semantics.) The kind of semantic theory set out is what is called possible-worlds semantics, and is based on the idea that the meaning of a sentence is the conditions under which it is true, and that these conditions are simply the class of possible worlds in which the sentence is true.

Many years ago, in Cresswell 1978, I defended this approach to semantics by arguing that speakers of a language know the truth conditions of the sentences they utter, and that it is this knowledge which constitutes their semantic competence. It was David Lewis who, when I gave this talk at Princeton in 1975, convinced me that there was a problem. I had argued that to know meaning is to know truth conditions. Lewis asked what it is to know truth conditions. If an interpretation to a language is a pairing of expressions with their meanings then the fact that certain expressions are paired with certain meanings will be a *mathematical* (or logical) fact. But it is obvious that someone who knows no English is not suffering a deficiency of mathematical knowledge. The empirical fact is that a certain formal structure correctly models the behaviour of a linguistic population in ways in which alternative structures do not. Thus an interpretation in which the word ***horse*** refers to horses is correct in a sense in which an interpretation in which ***horse*** refers to lobsters is wrong.

So the question becomes: What is it to model correctly the linguistic behaviour of a population? Lewis 1975 has produced one answer, and his answer is discussed in chapter 7 of this book. That answer links the semantics of languages with the content of propositional attitudes, specifically with belief and desire. In Cresswell 1978 my reply to the comment that I did not say what it was for certain sentences to have the truth conditions they do, was that such facts exist even if we cannot say what they are. In Cresswell 1982, I took this a little further. I argued that although semantic facts may supervene on psychological (and presumably social) facts, yet there may be no way of *reducing* semantic facts to other kinds of facts. If this is so then semantics must be taken as autonomous in the sense that any semantic theory simply has to take as primitive and not further analysable that thus and such a sentence has thus and such truth conditions in the language of a linguistic population. (This seems to me the conclusion that Schiffer 1987 *ought* to have drawn.)

Although you may not be able to give any reductive analysis of semantic facts in non-semantic terms – and this is the 'conclusion' I come to in chapter 10 – yet you should be able to say *something* about what it is that justifies semantic structures, and about why possible-worlds semantics has the plausibility it does. In chapters 3–5 I try to set the stage by adding to the possible-worlds framework already introduced via the interpretations to the language set out in chapters 1 and 2, language users and relations between them and their languages. Along the way I say something about the possible-worlds metaphysics that I presuppose. In chapter 6 I discuss Putnam's twin-earth example to see whether it really does cast doubt on possible-worlds semantics as Putnam seems to have thought. Chapter 7 discusses Lewis's attempt to analyse semantic facts in terms of propositional attitudes, and also looks at the discussion in Field 1972 of the view that Tarski shewed that semantic facts have an even simpler reductive analysis.

Whatever the details may be about exactly what kind of non-semantic facts constitute semantic facts or constitute the content of propositional attitudes, it seems pretty clear that in a broad sense they are facts about the causal interactions between the linguistic behaviour of speakers and the facts in the world that they are speaking about. In chapter 8 I argue that the kind of causation involved is best analysed using Lewis's account, in Lewis 1973b, of causation in terms of counterfactuals. If in turn you analyse counterfactuals as Lewis does in

Lewis 1973a, so that 'If α had been the case then so would β' is true in a possible world *w* iff β is true in the world most like *w* in which α is true, then by putting language users into the possible worlds used to model their language you have a very intuitive explanation of why possible-worlds semantics is so plausible.

Although chapter 8 argues that possible-worlds semantics is plausible because it models so well the causal relations between language users and the world, it does not attempt to give any specific account of just what more 'basic' facts constitute the fact that our words mean what they do or that our beliefs and desires have the content they do. Both David Lewis and Robert Stalnaker have argued that the way to naturalize content is via the role of beliefs and desires in an explanation of behaviour. Chapter 9 explores this proposal. While I produce no arguments to shew that you cannot achieve a reductive analysis of this content an examination of the explicit proposals on pp. 27–40 of Lewis 1986a and in chapter 1 of Stalnaker 1984 shews how problematic it is to get a viable account, and in my view makes it likely that there is no account to be had. Some of the issues I shall be discussing have been raised by authors concerned with the advocacy in Davidson 1967 of the use of the theory of truth in Tarski 1936. One of the claims made there is that a semantical theory for a language can be given without any need to refer to what the speakers of that language know or do. I myself find semantical discussions in this tradition somewhat obscure and so I will be raising the issues in the context of possible-worlds semantics. Since most semantics which is actually *done*, as opposed to being talked about, is done in the possible-worlds framework, a discussion within this framework will give us access to a large body of literature.

Chapter 10 tries to say why it might be that there is no account to be had of the relation between semantic content and the non-semantic facts on which the semantic facts supervene. The nature of the dependence might just be too complicated. But if it is too complicated how can we *know* semantic facts? And clearly we *do* know semantic facts. I suggest that we might have a capacity for recognizing complicated patterns of psychological, social and other facts which are too complicated for us to have a theory of. This certainly seems to happen in knowledge of our own mental states, and that knowledge feels very like our instant recognition of the meanings of utterances that we and others produce. If this is right then the fact that we cannot naturalize semantics is only to be expected, and semantics must proceed as an autonomous discipline with its own source of semantic facts, known to speakers, as its data.

1

A simple formal language

This book is concerned with what makes one interpretation of a language the correct one and another the wrong one. I will be discussing this problem in the context of possible-worlds semantics and so, in order to set the scene, and by way of introduction for those who have not come across possible-worlds semantics for natural languages, I shall, in the first two chapters, set out a simple fragment so that we can see what is going on.

I hope that most of you will have a familiarity with at least the language of the first-order predicate calculus. (sometimes called the lower predicate calculus or LPC). I shall set out here a simple version of this, but without quantifiers, which I can generalize to a language rich enough for the points I want to make. So here is a language $\mathscr{L}$.

Sentences or *well-formed formulae* of $\mathscr{L}$ are finite sequences of what are called *symbols*. Although, in logical languages, symbols are often represented by letters, it is best to think of them as corresponding to *words* in natural language rather than letters. The sentences of $\mathscr{L}$ are those allowed by the formation rules. The formation rules are sensitive to the *syntactic category* of each symbol. It is time to be specific.

(1) $\mathscr{L}$ contains a category of *names*. Let these be ***Adriane***, ***Bruce***, ***Julie*** and ***David***.
(2) $\mathscr{L}$ contains a category of *predicates*. Let these be ***runs***, ***whistles*** and ***sees***.
(3) $\mathscr{L}$ contains a category of *sentential functors* (sometimes called connectives). Let these be ***not***, ***and*** and ***if***.
(4) $\mathscr{L}$ contains a left parenthesis, (, and a right parenthesis,).

Categories (2) and (3) may be further subdivided. For reasons which I will make clear ***runs*** and ***whistles*** are *one-place* predicates and ***sees*** is a two-place predicate. ***not*** is a one-place functor, while ***and*** and ***if*** are two-place functors.

The grammatically well-formed sequences, i.e. the *sentences*, of $\mathscr{L}$ are those and only those finite sequences of the symbols in (1)–(4) which satisfy the following *formation rules*:

FR1 If *a* is a name and *F* is a one-place predicate then *aF* is a sentence.
FR2 If *a* and *b* are names, not necessarily distinct, and *F* is a two-place predicate, then *aFb* is a sentence.
FR3 If α is a sentence then so are α***not***, (α ***and*** β) and (***if*** αβ).

The sentences formed by FR1 and FR2 can be called atomic sentences. Those which involve FR3 can be called non-atomic or complex sentences. Those familiar with predicate logic will know that *aF* is more usually written as *Fa* and *aFb* as *Fab*. I have chosen to keep close to the order of the English sentences which the sentences of $\mathscr{L}$ are mimicking. For similar reasons I have put ***not*** after a sentence, rather than in front of it. FR1 and FR2 have been stated using what are often called *metalinguistic* (or sometimes *metalogical*) variables. FR1 is a statement in the (English) metalanguage in which we are talking about $\mathscr{L}$, and it is convenient to enrich it by the variables '*a*' '*b*' and '*F*' to stand for the names and predicates of $\mathscr{L}$. Since (1) and (2) are small we could replace FR1 and FR2 by a finite list of all the atomic formulae:

Adriane runs	***Bruce runs***
Adriane whistles	***Bruce whistles***
Julie runs	***David runs***
Julie whistles	***David whistles***
Adriane sees Adriane	
Adriane sees Bruce	
Adriane sees Julie	
Adriane sees David	

and so on

Even with (1) and (2) so small I hope that you can see that replacing FR1 and FR2 by a list is extremely cumbersome. Further it fails to capture the reason *why* these are all sentences of $\mathscr{L}$. Take

(5) ***Adriane runs***

Adriane is a name and ***runs*** is a one-place predicate, so FR1 tells us that that is why (5) is a sentence. Replacing ***runs*** by the two-place predicate ***sees*** would give us

(6) ***Adriane sees***

which is not a sentence of $\mathscr{L}$. (A digression is in order here. The distinction between one-place and two-place predicates of $\mathscr{L}$ is like the distinction between intransitive and transitive verbs in English. Unfortunately it is almost impossible to find an English transitive verb which cannot also be used intransitively. There are many good uses of (6), despite the fact that ***sees*** also occurs as a transitive verb. But in $\mathscr{L}$ (6) is not well formed.)

FR3 could have been stated in a more general form using metalogical variables. We would make a distinction between one-place functors and two-place functors. ***not*** would be a one-place functor, while ***and*** and ***if*** would be two-place functors. FR3 could then be expressed as

> FR3′ If δ is an n-place functor (n = 1,2) and $\alpha_1,\ldots,\alpha_n$ are sentences, not necessarily distinct, then $(\delta\alpha_1\ldots\alpha_n)$ is a sentence.

FR3′ is stated for generalization to arbitrary n-place functors. It does however ignore the difference, reflected in FR3, that ***not*** is placed after the sentence it applies to, ***and*** goes between the two sentences it links while ***if*** goes in front of them. You can either regard these features as syntactically unimportant, or you can further subcategorize the two-place functors. As an example of how the formation rules operate look at how to prove that the following is a sentence of $\mathscr{L}$.

(7) (***if Adriane runs not*** (***Julie whistles and Bruce sees David***))

To shew that (7) is a sentence we first find its atomic parts. They are

(8) ***Adriane runs***
(9) ***Julie whistles***

and

(10) ***Bruce sees David***

Examples (8) and (9) are sentences because they each consist of a name followed by a one-place predicate, while (10) is a sentence because it consists of a two-place predicate between two names. In (8) the *a* of FR1 is ***Adriane*** while the *F* is ***runs***. In (9) *a* is ***Julie*** and *F* is ***whistles***. In (10) the *a* and *b* of FR2 are, respectively ***Bruce*** and ***David*** and the *F* is ***sees***.

Given (8) as a sentence, if (8) is the α of FR3 then FR3 tells us that

(11) ***Adriane runs not***

is also a sentence. And given (9) and (10) as sentences then if (9) is the α of FR3 and (10) is β, FR3 tells that

(12) ***(Julie whistles and Bruce sees David)***

is a sentence. Here the fact that α and β are metalinguistic *variables* is important. In getting (12) α referred to (9), while in getting (11) α referred to (8). We now use the fact that (11) and (12) are sentences to apply FR3 again. This time (11) is α and (12) is β; and the part of FR3 which applies is the last clause which says that two sentences (here (11) and (12)) preceded by ***if*** and enclosed in parentheses are a sentence.

I would like to mention an alternative description of the syntax of the predicates and functors of $\mathscr{L}$, a description which I hope will help later when I come to semantics. At a fairly intuitive level there seems to be a distinction between a name like ***Adriane*** and a predicate like ***runs*** along the following lines. The purpose of ***Adriane*** in $\mathscr{L}$ is to name someone, to name Adriane. Adriane is one of the items in the domain of discourse that $\mathscr{L}$ is talking about. (We are of course anticipating a little now - since so far $\mathscr{L}$ has been introduced as an uninterpreted purely syntactic system. But if we believe, with Montague (1974, p.223 n. 2), that the interest of syntax is as a preliminary to semantics, we shall want it to reflect at least some semantic structure.) The purpose of ***runs*** on the other hand is to enable a sentence to say something about whatever it is that is named - viz. that that item runs. While the purpose of ***Adriane*** is just to be a name the purpose of ***runs*** is, so to speak, to form the sentence (8) out of that name. The purpose of a sentence, on the other hand, is just to say something - that Adriane runs. Looked at in this light it is names and sentences which are the basic syntactical categories, with predicates being symbols which make sentences out of names. In fact what are called *categorial languages* are based on precisely this assumption, and I shall now describe $\mathscr{L}$ as a categorial language. The reason I am going to do this may at the moment appear technical, but in fact it will turn out to be philosophically important. This is because there are two competing dimensions of simplicity and complexity, even for a language like $\mathscr{L}$. In one dimension the symbols - names, predicates and functors - are all simple, while the sentences are all complex. In another dimension, in terms of their syntactic category, names and sentences are simple, while predicates and functors are complex. (And while in $\mathscr{L}$ names are simple in both dimensions there are considerations which support assigning them a complex syntactic category.) When we come to look at theories of what it is to understand a language the tension between these two dimensions turns out to be an important issue.

The system of syntactic categories needed for $\mathscr{L}$ is in fact very small. Take n and s as the basic categories of respectively name and sentence. Then a one-place predicate can be given category (s/n), for what it does is form a sentence (expression of category s) out of a name (expression of category n). A two-place predicate would be in category (s/nn) because it makes a sentence out of two names. The sentential connectives also belong to syntactic categories. ***not*** makes a sentence out of another sentence, and so is in category (s/s), while ***and*** and ***if*** each make a sentence out of two sentences and so are in category (s/ss). These are the only categories needed for $\mathscr{L}$, but it is not difficult to see that the representation of natural language suggests symbols in other categories. Here are one or two examples. Consider the word ***nobody***.

(13) ***nobody runs***

is a sentence whose structure might appear to be just like that of (8). But if so ***nobody*** would be a name, and we remember the trouble that Alice and the white king had in treating it as such. In logic ***nobody*** would be represented by a negated existential quantifier and would involve bound individual variables. For present purposes the addition of individual variables to $\mathscr{L}$ is an unnecessary complication. From a categorial point of view it is best to treat a quantifier like ***nobody*** as an operator which makes a sentence out of a one-place predicate. It would thus be of category (s/(s/n)). See how it goes: a one-place predicate is of category (s/n) because it makes an s (a sentence) out of an n (a name), and so an (s/(s/n)) makes a sentence out of something which makes a sentence out of a name. (This by itself doesn't solve all problems. The adventurous should look at why

(14) ***nobody sees Bruce***

or

(15) ***Julie sees nobody***

cannot be derived by these rules.)

Another kind of symbol is a one-place predicate modifier. Take the adverb ***quickly*** and look at

(16) ***Adriane runs quickly***

The right way to treat this sentence is to suppose that ***runs quickly*** is a complex one-place predicate, i.e. a complex expression in category (s/n). Thus it would be that complex expression, ***runs quickly***, which would make the sentence (16) out of the name ***Adriane***. This shews incidentally how a categorial language could have complex expressions in other categories besides s. In that respect this extended language is unlike $\mathscr{L}$. But if ***runs quickly*** is to be in category (s/n) how does it get there? Well, ***runs*** is also in category (s/n), and surely the function of ***quickly*** is to form the complex predicate ***runs quickly*** out of the simpler predicate ***runs***. ***quickly*** itself is therefore in category ((s/n)/(s/n)) since it makes an (s/n), a one-place predicate, out of an (s/n).

An even more elaborate category is that of preposition. Take the sentence

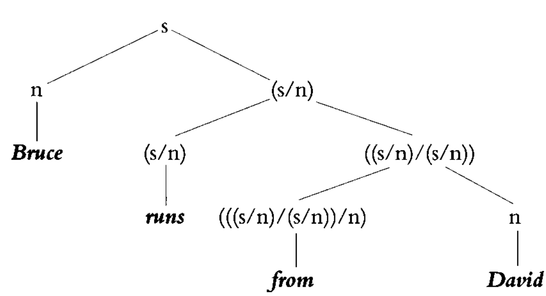
(17) ***Bruce runs from David***

In (17) it seems reasonable to take

(18) ***runs from David***

as a complex predicate – in category (s/n). So ***from David*** would be a predicate modifier in category ((s/n)/(s/n)). What does ***from*** do on its own then? Well it makes this modifier out of the name ***David***, and so is in category (((s/n)/(s/n))/n). Sometimes it is more perspicuous to represent facts like this in a tree:

(19)



Example (19) could be elaborated by annotating the higher nodes with the complex expressions derived at each stage.