CHAPTER 2

THE SYNTAX AND SEMANTICS OF TWO SIMPLE LANGUAGES

1. THE LANGUAGE L_0

We begin by considering a very simple language, which anyone familiar with symbolic logic will recognize as essentially the propositional calculus with propositions analyzed into predicates and arguments. Truth-conditional semantics was first developed in connection with logical languages, and it is instructive to look at such cases to understand the motivations for certain features that may appear peculiar in the context of natural languages.

In the logical tradition it is customary to specify the syntax of a language not by a phrase-structure grammar but by a recursive definition of the *well-formed expressions* of the language. This is done by giving first a list of *basic expressions* divided into various categories. (These will correspond to the terminal symbols and lexical categories of a phrase-structure grammar.) Then a set of *formation rules* states how expressions of various categories are combined into other more complex expressions. These rules apply recursively, and each is of the following form: given expressions $\alpha, \beta, \ldots, \eta$ of categories $C_\alpha, C_\beta, \ldots, C_\eta$, respectively, these expressions can be combined in a specific way (stated by the rule) to yield an expression of category C_ω. One notices certain similarities here to the phrase-structure rules of a phrase-structure grammar but certain differences as well. We will have more to say on this point later, but let us first consider as a specific instance the syntax of our language L_0.

1. *Syntax of L_0*

A. The *basic expressions* of L_0 are of three syntactic categories:

(2-1)

Category	*Basic Expressions*
Names	*d, n, j, m*
One-place predicates	*M, B*
Two-place predicates	*K, L*

B. The *formation rules* are of two kinds. First, there are rules for combining predicates with an appropriate number of names to produce *atomic sentences*

(sentences having no other sentences as parts). These are rules 1. and 2. below. Second, there are rules (3.–7. below) which form a sentence out of one or more other sentences.

(2-2)
1. If δ is a one-place predicate and α is a name, then $\delta(\alpha)$ is a sentence.
2. If γ is a two-place predicate and α and β are names, then $\gamma(\alpha, \beta)$ is a sentence.
3. If ϕ is a sentence, then $\neg\phi$ is a sentence.
4. If ϕ and ψ are sentences, then $[\phi \wedge \psi]$ is a sentence.
5. If ϕ and ψ are sentences, then $[\phi \vee \psi]$ is a sentence.
6. If ϕ and ψ are sentences, then $[\phi \rightarrow \psi]$ is a sentence.
7. If ϕ and ψ are sentences, then $[\phi \leftrightarrow \psi]$ is a sentence.

Note that in the statements of the foregoing rules we used the Greek letters "α," "β," etc. to refer to expressions of L_0. These symbols function in effect as variables taking expressions of L_0 as values, and it is important not to confuse such symbols with symbols of the language itself. Here L_0 is the *object language*, the language under study. The rules in (2-2) are statements *about* certain expressions of L_0 and are couched in English (albeit of a somewhat stilted, technical variety). Thus, English is the *meta-language*, the language used in talking about the object language and "α," "β," etc. are *meta-language variables*, or simply *meta-variables*. To help forestall possible confusion between meta-variables and object language symbols, we will adopt henceforth the following notational convention:

(2-3)

Notational Convention 1: Lower-case letters of the Greek alphabet are used only as meta-language variables, never as symbols of an object language.

The rules in (2-2) constitute a recursive definition of the infinite set of expressions of the category "sentence" in L_0. Rules 1. and 2., together with (2-1), comprise the base of the recursive definition, and rules 3.–7. together make up the recursion. It is assumed in all such statements of formation rules that nothing else is a member of any syntactic category except what qualifies by virtue of the rules; i.e., the exclusion clause is assumed whether explicitly stated or not.

By rule 1. we can form, for example, the atomic sentence $M(d)$ from the one-place predicate M and the name d. Similarly, rule 2. allows the formation of the atomic sentence $K(d, j)$ from the two-place predicate K and the names d and j. Now given that $M(d)$ and $K(d, j)$ are sentences, we can form by rule 3. the (non-atomic) sentences $\neg M(d)$ and $\neg K(d, j)$. The reader can verify on the basis of the formation rules given that each of the following is a well-formed sentence of L_0:

(2-4)
1. $[K(d, j) \wedge M(d)]$
2. $\neg[M(d) \vee B(m)]$
3. $[L(n, j) \rightarrow [B(d) \vee \neg K(m, m)]]$
4. $[\neg\neg\neg B(n) \leftrightarrow \neg M(n)]$

One should note carefully that the parentheses, comma, and square brackets are symbols of L_0 just as K, d, etc. are. They are not basic expressions, however, and are not assigned membership in any syntactic category. Such symbols are called *syncategorematic* because they are introduced into expressions by the formation rules along with the regular, or *categorematic*, symbols. Besides the symbols that might be regarded as punctuation, other syncategorematic symbols of L_0 are the *connectives* "$\wedge$," "$\vee$," "$\neg$," "$\rightarrow$," and "$\leftrightarrow$."

2. *Semantics of* L_0

Our strategy for determining the semantic values of the sentences and other constituents of L_0, for connecting each well-formed expression of L_0 with some "object in the world," will follow the Principle of Compositionality mentioned in Chapter 1. We first assume that a semantic value has been given to each basic expression (somehow or other – the way in which this is done does not concern us at the moment). We then state *semantic rules*, whose job it is to determine the semantic value of each larger constituent in terms of the semantic values of its components. In other words, semantic values are assigned to successively more inclusive constituents of the sentence until finally the semantic value of the entire sentence has been determined.

With respect to sentence (2-4) 2., for example, we would assume the semantic values for the basic expressions "M," "d," "B," and "m" to be antecedently given. Then by means of the semantic rules, we would determine the semantic values of the constituents "$M(d)$," "$B(m)$," "$M(d) \vee B(m)$," and finally of the whole sentence. Since the semantic rules will be designed in such a way that they retrace or "track" the syntactic structure, every well-formed sentence of L_0 as well as every well-formed constituent of such a sentence will be assigned a semantic value by this procedure.

What sorts of things are the semantic values to be? First, it is a basic assumption of the truth-conditional approach that semantic values come in various varieties or types, and that *in general* (but not always, as we shall see later), members of different syntactic categories take on semantic values of different types. Different theories of truth-conditional semantics will, however, make different assumptions about the exact range of possible semantic values and about the pairing of syntactic categories with types of values.

For the moment, we will assume a rather elementary version of the theory. In this system, names take as their semantic values just what our common sense would probably tell us they ought to take, namely, individuals. We are not obliged to say at this point just what counts as an individual; we would certainly want to include human beings, animals, and other countable physical objects, but we need not take a stand on such philosophically controversial problems as whether events, propositions, actions, etc. are to count. For illustrative purposes, let us assume that the universe consists of just four individuals, namely, Richard Nixon, Noam Chomsky, John Mitchell and Muhammad Ali. To each name in L_0 we might then assign one of these individuals as its semantic value, as in the following table:

(2-5)	*Name*	*Semantic Value*
	d	Richard Nixon
	n	Noam Chomsky
	j	John Mitchell
	m	Muhammad Ali

Other pairings would have served as well, but let us adopt this one now for purposes of illustration. We will in general insist that each basic expression be assigned a single semantic value, so that cases of lexical ambiguity will be treated as separate lexical items that happen to have the same pronunciation. We will not require, however, that every individual in the domain of discourse be assigned to a name as its semantic value. Thus, there could be individuals for whom our language had no names. It is also allowed for one and the same individual to have two or more names (just as "Samuel Clemens" and "Mark Twain" are names of the same person). In short, what we want is a *function*, in the mathematical sense, from the names of the language into the set of individuals in the domain of discourse; more specifically, it must be a *total function*, since no names are to be left unpaired. The table in (2-5) is intended to represent one such total function.

PROBLEM (2-1). Given a language with *n* distinct names and a universe of discourse consisting of m distinct individuals, how many different assignments of semantic values to names are possible?

It is important to recognize that there are two very different sorts of entities involved in (2-5). In the left-hand column there are *linguistic entities* – lexical items of a particular syntactic category in a particular language. In the right-hand column we find not linguistic entities but *"real-world" entities*, in this case, real people. There would be little chance of confusion on this matter were it not for the fact that we are communicating with the reader by means of the printed page, and so we could not put in (2-5) the people themselves but rather have let them be represented by their conventional names in English. The reader is encouraged to mentally transcend this limitation and to imagine that in (2-5) we have persuaded Messrs. Nixon, Mitchell, Chomsky and Ali to participate in a *tableau vivant* in which they wear their respective names from L_0 as, say, signs hanging around their necks. In this way, the reader will not be tempted to think that the semantic value of d is "Richard Nixon" (i.e., Mr. Nixon's name); rather, it is Richard Nixon himself, the ex-President of the United States of America, the man who said "I am not a crook," etc. The point is worth belaboring since it is central to the program of truth conditional semantics, as we said in the preceding chapter, that a connection is made between language and extra-linguistic reality, i.e. "the world." (The sanitizing quotes here are prompted by the fact that we will eventually want to consider not only the world in which we live as it actually is but also the world as it was, as it will be, as it might have been, etc. i.e., other "possible worlds").

What sort of semantic value should the one-place predicates B and M have? What "objects in the world" could we connect these predicates with? For purposes of L_0, we will let this semantic value be a set of individuals – intuitively, the set of individuals of which the predicate is true. Our semantics is said to be *extensional* because our semantic treatment of a predicate here involves *only* its "extension." Thus, we will let the semantic value of B be the set of all individuals that are bald and the semantic value of M be the set of all individuals who have moustaches. We might digress at this point to admit that from the point of view of a speaker's understanding of the meanings of predicates, it is not very natural to identify our understanding of, say, "is bald" with the set of bald persons; there are, after all, bald individuals we have never met. Instead, what seems more relevant is our grasp of a certain attribute or characteristic that all these individuals share that distinguishes them from others. (For this reason we have avoided use of the term "meaning" here and have employed the more neutral term "semantic value.") But one implicit effect of the use of "is bald" by all speakers of English is potentially to single out this set of persons, however this is done mentally by individual speakers, and this set will serve our present purposes quite well. Our treatment of the semantic values of predicates in later languages will be more complex and will more closely approximate our intuitive understanding of their "meaning."

We can now see how the truth value of a sentence formed from a name and a one-place predicate is to be determined. A sentence like $M(j)$ should be true if and only if the individual denoted by the name j (John Mitchell in this case) is a member of the set of individuals denoted by the predicate M (the set of individuals that have moustaches). Since we will need to use the phrase "the semantic value of α" often, we will introduce a special notation for it:

(2-6) Notational Convention 2: For any expression α, we use $[\![\alpha]\!]$ to indicate the semantic value of α.

Thus our semantic rule for sentences formed from a one-place predicate and a name will say that if δ is a one-place predicate and α is a name, then $\delta(\alpha)$ is true iff (if and only if) $[\![\alpha]\!] \in [\![\delta]\!]$.

For two-place predicates, we will adopt an approach that parallels that used for one-place predicates: we let the semantic value of a two-place predicate be a set of pairs of individuals; intuitively, these are the pairs of which the predicate is true when the first argument names the first individual and the second argument names the second individual. In particular, we will let $[\![K]\!]$ be the set of pairs in which the first knows the second, and $[\![L]\!]$ will be the set of pairs in which the first loves the second. For example, ⟨Richard Nixon, John Mitchell⟩ is a member of $[\![K]\!]$, but to the extent of our knowledge, ⟨Muhammad Ali, Noam Chomsky⟩ is not a member of $[\![K]\!]$. Our semantic rule for sentences formed from a two-place predicate and two names will then state that if γ is a two-place predicate and α and β are names, then $\gamma(\alpha, \beta)$ is true iff $\langle [\![\alpha]\!], [\![\beta]\!] \rangle \in [\![\gamma]\!]$ (that is, if the ordered pair consisting of the denotations of the two names, in that order, is a member of the set of pairs denoted by the predicate). By this rule, for example, $K(d, j)$ is true, because $\langle [\![d]\!], [\![j]\!] \rangle \in [\![K]\!]$, in other words, because ⟨Richard Nixon, John Mitchell⟩ $\in \{\langle x, y \rangle \mid x \text{ knows } y\}$. Note that we want the semantic values of two-place predicates to be sets of *ordered* pairs (and not merely sets of two-member sets) in order to allow for the possibility that $\gamma(\alpha, \beta)$ can be true while

$\gamma(\beta, \alpha)$ is false (or vice versa), for some two-place predicate γ and names α and β.

Before going on to a formal statement of the semantic rule or rules involved, we must pause to consider our obligations to the Principle of Compositionality. We want our semantic rules to be such that the semantic value of a syntactically complex expression is always a function of the semantic values of its syntactic components and of their "mode of combination," i.e., the way the parts are combined to form the expression in question. As we said in Chapter 1, in order to ensure that the semantic component adheres to the Principle of Compositionality, it is common practice to construct the semantic rules so that they are in one-to-one correspondence with the syntactic rules. For example, $\neg[M(j) \wedge K(j, d)]$ is generated by the grammar of L_0 through the application of the four syntactic rules B1, B2, B4 and B3 of (2–2). Therefore, four of the semantic rules we formulate will correspond to these syntactic rules. The semantic rules will, in effect, "compute" the semantic values of successively larger parts of this sentence, starting with the semantic values of the basic expressions.

The semantic rules for L_0 that remain to be formulated are therefore those that will correspond to the syntactic rules producing $\neg\phi$, $[\phi \wedge \psi]$, $[\phi \vee \psi]$, $[\phi \rightarrow \psi]$, and $[\phi \leftrightarrow \psi]$ from sentences ϕ and ψ. These will be designed to have the effect of the familiar truth tables for these connectives and thus require little comment. Though here formulated in our English metalanguage in a way that requires our understanding of English "and," "or," "not," etc., we will see shortly that these semantic rules could, if desired, be given a mechanical formulation that avoids these metalanguage words. We now state the complete semantic system for L_0: the assignment of semantic values to basic expressions, and the rules that recursively determine the semantic value for any sentence of L_0 in terms of the basic expressions and syntactic rules from which it is formed.

(2-7) A. Basic Expressions:

$[\![d]\!]$ = Richard Nixon $[\![j]\!]$ = John Mitchell

$[\![n]\!]$ = Noam Chomsky $[\![m]\!]$ = Muhammad Ali

$[\![M]\!]$ = the set of all living people with moustaches

$[\![B]\!]$ = the set of all living people who are bald

$[\![K]\!]$ = the set of all pairs of living people such that the first knows the second.

$[\![L]\!]$ = the set of all pairs of living people such that the first loves the second.

(2-8) B. Semantic Rules:

1. If δ is a one-place predicate and α is a name, then $\delta(\alpha)$ is true iff $[\![\alpha]\!] \in [\![\delta]\!]$.
2. If γ is a two-place predicate and α and β are names, then $\gamma(\alpha, \beta)$ is true iff $\langle [\![\alpha]\!], [\![\beta]\!] \rangle \in [\![\gamma]\!]$.
3. If ϕ is a sentence, then $\neg\phi$ is true iff ϕ is not true.
4. If ϕ and ψ are sentences, then $[\phi \wedge \psi]$ is true iff both ϕ and ψ are true.
5. If ϕ and ψ are sentences, then $[\phi \vee \psi]$ is true iff either ϕ or ψ is true.
6. If ϕ and ψ are sentences, then $[\phi \rightarrow \psi]$ is true iff either ϕ is false or ψ is true.
7. If ϕ and ψ are sentences, then $[\phi \leftrightarrow \psi]$ is true iff either ϕ and ψ are both true or else ϕ and ψ are both false.

Given these semantic rules and the assignments of semantic values to basic expressions, one can, in principle, determine the truth values of all sentences of L_0. We say "in principle" because values are assigned to the predicates M, B, K, and L in terms of what is in fact the case in the world at present. Anyone's knowledge of this factual situation may be limited, but the functions which are the semantic values are well-defined nonetheless (if we ignore for the sake of the example any problems connected with the assumed preciseness of the predicates; i.e., we assume that we can determine for any living individuals whether they are bald or not, have moustaches or not, etc.). Thus, at the time of writing, $B(n)$ is false, since Noam Chomsky is not bald, but $B(j)$ is true, since John Mitchell is bald. Given this, it follows that $\neg B(j)$ is false (since $B(j)$ is true), and $\neg B(n)$ is true (since $B(n)$ is false). $K(d,j)$ is true, since Richard Nixon knows John Mitchell, and thus $[B(j) \wedge K(d,j)]$ is true, both conjuncts being true. $K(n, m)$, we suspect, is false, since we doubt that Noam Chomsky knows Muhammad Ali.

PROBLEM (2-2). Determine the truth values, insofar as your knowledge allows you to do so, of each of the sentences of L_0 given in (2-4).

This system of syntactic and semantic rules thus specifies an infinite set of sentences of L_0 and assigns to each a semantic value, either *true* or *false* (and, as it happens, the assignments are unique since L_0 contains no "syntactically ambiguous" sentences). But in the preceding chapter we promised a system that would supply for each sentence of a language its *truth conditions* – necessary and sufficient conditions for the truth of that sentence – and it appears that we have delivered only a *truth value*. Where is the rest? The answer is that the foregoing semantic rules do in fact give the truth conditions for sentences of L_0, and they do so, as it were, *while* determining truth values. Consider an atomic sentence of L_0, for example, $M(n)$. Let us ask under what circumstances it would be true. This information is contained in semantic rule B.1, where we learn that for any sentence of the form $\delta(\alpha)$, it will be true just in case $[\![\alpha]\!] \in [\![\delta]\!]$. That is, for the case of sentence $M(n)$, it will be true just in case $[\![n]\!] \in [\![M]\!]$, which, given the assumed semantic values for M and n, amounts to saying that Noam Chomsky is a member of the set of all living people with moustaches. In summary, $M(n)$ is true iff Noam Chomsky has a moustache.

If we examine any sentence of L_0 of the form $[\phi \wedge \psi]$ and ask for its truth conditions, we find that they are given by rule B.4. $[M(n) \wedge B(d)]$ is true, for example, just in case both $M(n)$ is true and $B(d)$ is true. But we know under what conditions each of these is true by virtue of rule B.1, so we therefore know the truth conditions of $M(n) \wedge B(d)$. Similar reasoning applies to any sentence of L_0, and so we have in fact delivered what we promised.

A further, related point may perhaps make this clearer yet. As we will explain in section IV, a quite different status is possessed by the assignment of semantic values to basic expressions in (2-7) and by the semantic rules in (2-8). Precisely what semantic values are assigned in (2-7) depends crucially on certain facts about the world, e.g., on just who is bald, who in fact knows whom, etc. If these facts were different than they actually are, different semantic values would have to be assigned to "*B*", "*K*", etc. But regardless of how these facts might change, the semantic rules in (2-8) would stay the same. These rules are an integral part of the language L_0, as opposed to the world this language might be used to talk about. They state relationships between the semantic value of one expression and the semantic values associated with other, syntactically related expressions – relationships which must hold just by virtue of what language L_0 is, irrespective of precisely what objects in the world turn out to be connected with L_0's expressions. The game plan of model theory is to describe the semantics of a language by characterizing these necessary relationships between associated objects, capitalizing on the fact that the relationships do not change depending on contingent facts about the condition of things in the world.

We have given the syntactic and semantic rules for L_0 in the general form usually adopted for formal languages by logicians. But in fact many of the features of our statements of these rules were essentially abritrary, as we could have defined a language with the same expressive capability as L_0 in somewhat different ways while still adhering to the Principle of Compositionality and the goal of a truth-conditional semantics. For example, while we used recursive definitions to specify the syntax of L_0, we could instead have used a phrase-structure grammar of the sort linguists are accustomed to, and instead of introducing the connectives $\wedge$, $\vee$, $\rightarrow$, etc., syncategorematically, we could have treated them as basic expressions in a category of sentence conjunctions. Similarly, several aspects of our semantics could have received alternative but equivalent formulations. In order to better illustrate which features of L_0 are crucial to our program and which are matters of convenience and, at the same time, to show how this program can be applied to a language that resembles English to a much greater degree than L_0, we now turn to an English-like but semantically similar language which we will call L_{0E}.

II. THE LANGUAGE L_{0E}

1. *Syntax of* L_{0E}

The syntax of L_{0E} is given by the following context-free phrase-structure grammar:[1]

(2-9)

$$S \rightarrow \begin{Bmatrix} S\ Conj\ S \\ Neg\ S \\ N\ VP \end{Bmatrix} \qquad VP \rightarrow \begin{Bmatrix} V_i \\ V_t\ N \end{Bmatrix}$$

Conj → *and*, *or*
N → *Sadie*, *Liz*, *Hank*
V_i → *snores*, *sleeps*, *is-boring*
V_t → *loves*, *hates*, *is-taller-than*
Neg → *it-is-not-the-case-that*

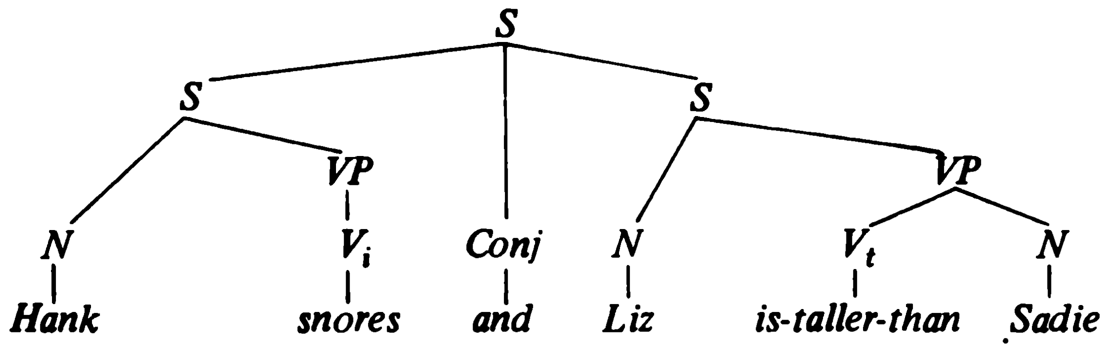
By trying a few derivations with this grammar the reader will see that it generates a small fragment of quasi-English. For example, the following are grammatical according to (2-9):

(2-10)
1. *Sadie snores.*
2. *Liz sleeps.*
3. *It-is-not-the-case-that Hank snores.*
4. *Sadie sleeps or Liz is-boring and Hank snores.*
5. *It-is-not-the-case-that it-is-not-the-case-that Sadie sleeps.*

One should note that the hyphenated items *is-boring*, *is-taller-than*, and *it-is-not-the-case-that* are understood as unanalyzable terminal symbols of L_{0E} just as *Liz* and *snores* are. This is merely a device to allow us to introduce a bit of variety into this rather limited language without having to deal with certain syntactic complexities prematurely. We do not wish to suggest that the corresponding expressions "is boring," etc. should be treated this way in a more complete and accurate grammar of English, and there is nothing essential to our semantic approach involved here.

L_{0E} contains five *lexical categories* (those which are immediately rewritten as terminal symbols), namely N, V_i, V_t, *Neg*, and *Conj*. For convenience, let us call these by their traditional linguistic designations: proper nouns (or names), intransitive verbs, transitive verbs, negation, and (co-ordinating) conjunctions, respectively. The two remaining non-lexical categories are *VP* (for verb phrase) and S (for sentence). We assume that the derivation of any sentence of L_{0E} can be represented by a tree structure in the usual way (described in Note 1), and given such a tree we will use the commonly accepted terminology, saying, for example, that any string of terminal symbols which is exhaustively dominated by a category stands in the "is a" relation to that category. The sentence *Hank snores and Liz is-taller-than Sadie* would thus be associated with the following tree structure according to the grammar of (2-9):

(2-11)



With respect to this tree, we can say that *Hank* stands in the "is a" relation to N, or briefly, that *Hank* is an N. Similarly, *is-taller-than Sadie* is a *VP*, the entire terminal string is an S, "snores and" is not a constituent at all, etc.

Although the grammar in (2-9) generates only a fragment of English (or near-English), it is worth noting that it generates an infinite language – by means of the recursive rules $S \to S$ *Conj* S and $S \to$ *Neg* S. Further, there are sentences, in fact an infinite number of them, which have more than one syntactic derivation. Sentence (2-10) 4. is one such.

PROBLEM (2-3). Construct all phrase-structure trees associated with the sentences in (2-10). The fourth sentence should have two trees. Set your results aside for use in Problem (2-8).

PROBLEM (2-4). Are there any syntactically ambiguous sentences (those having distinct tree structures) in L_{0E} which do not involve an application of the phrase-structure rule $S \rightarrow S\ Conj\ S$?

2. *Semantics of* L_{0E}

Even though we have used a phrase-structure grammar to formulate the syntax of L_{0E} we will still adopt the Principle of Compositionality in our interpretation of L_{0E} in much the same way as with L_0. We will assign a semantic value to each lexical item as we did to each basic expression in L_0, and for each syntactic constituent there will be a rule for determining its semantic value from the semantic value(s) of its sub-constituents. Though we may think of the phrase-structure rules as defining a tree "from top to bottom," our semantic rules will be formulated as proceeding from the bottom of the tree (its terminal nodes) to the top. Nevertheless, there will be a semantic rule corresponding to each phrase structure rule in the semantics of L_{0E}, just as there was a one-to-one correspondence between syntactic and semantic rules in L_0. More formally, for each phrase structure rule $\alpha \rightarrow \beta_1\ \beta_2 \ldots \beta_n$, there will be a semantic rule for determining the semantic value of the constituent labelled α in a tree in terms of the semantic values of the constituents $\beta_1, \beta_2, \ldots \beta_n$ which the node α dominates.

Purely for convenience, we will make a slight change in the way the semantic values for sentences are stated. Instead of simply classifying sentences into the (meta-language) categories "true" and "false" as we did for L_0, we will select two objects to represent truth and falsity, respectively, and assign one of these to each sentence as its semantic value. Following common practice among mathematical logicians (including Montague), we will select the number 1 to indicate truth and 0 to indicate falsity. The intuitive significance of these semantic values is the same as before: sentences assigned 1 are to be thought of as those that correspond to some (real or hypothetical) state-of-affairs, while sentences assigned 0 are those that don't. Thus our choice of these two objects has no particular ontological significance; we could just as well have selected the Empire State Building for the value assigned to true sentences and the planet Venus for false ones.

Turning now to the lexical categories of L_{0E}, we will first assign values to the names. As with L_0, we will want names to denote individuals, so we may assign the names *Sadie*, *Liz* and *Hank* values as in (2-12):

(2-12) $[\![\textit{Sadie}]\!]$ = Anwar Sadat
$[\![\textit{Liz}]\!]$ = Queen Elizabeth II
$[\![\textit{Hank}]\!]$ = Henry Kissinger

We wish to give the semantic values of intransitive verbs, V_i, the effect of singling out a set of individuals, just as we did with the one-place predicates of L_0. We will achieve this effect in a different way here, however.

Sets of individuals are in a one-to-one correspondence with functions that map individuals to 0 or 1, as will emerge momentarily. It is very convenient to follow Montague in the common mathematical practise of not distinguishing between the two isomorphic sorts of object in cases like this, and to consider an object sometimes as a set, sometimes as a certain kind of function. For this reason we digress briefly to define the association which justifies this identification.

If A is the set of individuals and S is any subset of A, we define a function f_S on the set A by letting

$$f_S(a) = \begin{cases} 1 \text{ if } a \in S \\ 0 \text{ if } a \notin S \end{cases}$$

for each a in A. This function is called the *characteristic function* of S (with respect to A) and belongs to $\{0, 1\}^A$, where X^Y is in general the set of all functions from Y into X. The characteristic function divides the domain A into two parts, the subset mapped into 1 (namely S) and the complementary subset, which is mapped into 0.

Two fundamental properties of sets guarantee that sets of individuals and their characteristic functions are in a one-to-one correspondence. First, membership in a set S is a strictly yes-or-no matter, i.e., each particular individual either does or else does not belong to S. Thus every set included in A is characterized by some way of saying "true" or "false" to each individual. Secondly, two sets are distinct if they differ in membership. Therefore, different ways of saying "true" or "false" to individuals correspond to different sets.

The semantic values of V_i's in L_{0E} will all be characteristic functions of sets of individuals. Assuming for the sake of simplicity that the three individuals mentioned in (2-12) are the only individuals in the world, we might for example stipulate that the V_i *snores* has as its semantic value the following function:

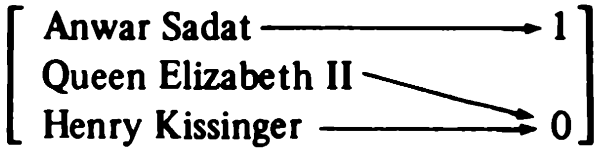
(2-13)

$$[\![\textit{snores}]\!] = \begin{bmatrix} \text{Anwar Sadat} \longrightarrow 1 \\ \text{Queen Elizabeth II} \longrightarrow 1 \\ \text{Henry Kissinger} \longrightarrow 0 \end{bmatrix}$$

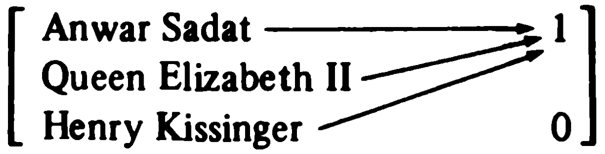
(Recall that a function is technically a set of ordered pairs: thus (2-13) is simply a convenient graphic representation of the set {⟨Anwar Sadat, 1⟩, ⟨Queen Elizabeth II, 1⟩, ⟨Henry Kissinger, 0⟩}). Note that this semantic value is a set-theoretic construct made from individuals (NB: here, real people!) and truth values. (In the extensional semantic theory we will be constructing in this and the following two chapters, *every* kind of semantic value will in fact be made out of these same basic ingredients – individuals and truth values – by means of the combinatory apparatus of set theory.)

For the sake of completeness, let us assume that semantic values of the remaining V_i's are as follows:

(2-14) ⟦*sleeps*⟧ =



(2-15) ⟦*is-boring*⟧ =



PROBLEM (2-5). Assuming that the world contains n individuals, how many different semantic values for V_i's are possible?

Given now the semantic value for both the names and the intransitive verbs of L_{0E}, what kind of semantic rule will be required to "compute" the semantic value of a sentence of the form $N + V_i$? It is clear that the simplest rule will be: apply the function which is the semantic value of the V_i to the argument which is the semantic value of the N. The result will be the semantic value (i.e. 1 or 0) of the sentence. For example, given that ⟦*Sadie*⟧ = Anwar Sadat and that ⟦*snores*⟧ is the function in (2-13), the truth value of the sentence *Sadie snores* will be the value of the function at the argument Anwar Sadat, i.e., 1 (true). In the usual notation for functions, in which the argument is written to the right of the name of the function and enclosed in parentheses, the foregoing could be expressed as:

(2-16) ⟦*snores*⟧ (⟦*Sadie*⟧) = 1

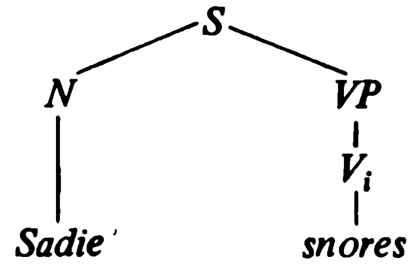
In the same way we could determine that given our assumed semantic values, ⟦*Hank sleeps*⟧ = 0 and ⟦*Liz is-boring*⟧ = 1.

In L_0 we let one-place predicates denote sets and specified that a sentence formed from such a predicate plus a name was to count as true just in case the individual denoted by the name belonged to the set denoted by the predicate. The function given in (2-13) as the semantic value of *snores* is the characteristic function of the set {Anwar Sadat, Queen Elizabeth II}, and applying such a function to an individual, as we did in (2-16), results in the value 1 (truth) just in case that individual belongs to the set characterized by this function, false just in case that individual does not belong to that set. Thus, our semantic treatment of one-place predicates in L_0 turns out to be equivalent to that of sentences with V_i's in L_{0E}. In many approaches to truth conditional semantics, sets rather than characteristic functions are assigned as semantic values of certain syntactic categories. As we see, nothing crucial is involved in this choice, since sets and characteristic functions are essentially two ways of looking at what amounts to the same thing. It may be more elegant to formalize semantic values as characteristic functions rather than sets in that the semantic rules which produce a truth value as output are assimilated to other rules which work by applying a function to an argument. Montague preferred the elegance and uniformity of stating semantic rules as *rules of functional application* wherever possible, and thus we will adhere to his practice of using characteristic functions rather than sets in the formal definitions. But, again following Montague's practice, we will often talk in terms of sets rather than functions when it is intuitively more congenial to do so. The reader should be prepared to make the necessary conversion without being explicitly directed to do so in each case.

As a further preliminary to stating the semantic rules for L_{0E}, we note again that sentences like *Sadie sleeps or Liz is-boring and Hank snores* are derivable in nonequivalent ways. It is intuitively clear that the semantic value such a sentence has may depend on how it is derived, in particular on the phrase-structure tree associated with its derivation, so that it will be semantically as well as syntactically ambiguous. For this reason, we shall not assign semantic values directly to sentences in an ambiguous language like L_{0E}, but in the first instance to phrase-structure trees. Otherwise we could not continue assigning a unique semantic value to each part of the language we interpret. Sentences and other phrases naturally inherit the semantic values assigned to their one or more tree structures.

Turning now to the semantic rules of L_{0E}, we will provide a semantic rule for each syntactic rule used in producing sentences. In order to interpret the structure (2-17) of the sentence *Sadie snores*, which will ultimately involve applying the function $[\![snores]\!]$ to $[\![Sadie]\!]$, we need semantic rules for the phrase-structure rules that introduce the intervening nodes N, V_i and VP.

(2-17)



Clearly the semantic value of the nodes labelled with the lexical categories N and V_i should just be the semantic values of the respective lexical items which they immediately dominate. Thus, the semantic rule corresponding to the syntactic rule $N \rightarrow$ *Sadie* should be something like the following:

(2-18) If α is N and β is *Sadie*, then $[\![\alpha]\!] = [\![\beta]\!]$.
$\qquad\qquad\quad |$
$\qquad\qquad\quad \beta$

The semantic rule corresponding to $V_i \rightarrow$ *snores* would be similar, and in fact we could abbreviate all such semantic rules by means of the following rule schema:

(2-19) If α is γ, where γ is any lexical category and β is any lexical item,
$\qquad\qquad\quad |$
$\qquad\qquad\quad \beta$
and $\gamma \rightarrow \beta$ is a syntactic rule, then $[\![\alpha]\!] = [\![\beta]\!]$.

For the grammar of L_{0E}, this schema is instantiated by twelve semantic rules, each one corresponding to a lexical rule of the grammar.

Corresponding to the nonlexical syntactic rule $VP \rightarrow V_i$, we will want a semantic rule which attaches the semantic value of the V_i node to the VP node. Here and below, we use triangles (in a way familiar to linguists) as meta-variables over trees; e.g., $\underset{\triangle}{V_i}$ stands for any tree rooted in the node V_i.

(2-20) If α is VP and β is $\underset{\triangle}{V_i}$, then $[\![\alpha]\!] = [\![\beta]\!]$.
$\qquad\qquad\quad |$
$\qquad\qquad\quad \beta$

Finally, we come to the more interesting semantic rule which corresponds to the branching syntactic rule $S \rightarrow N\ VP$:

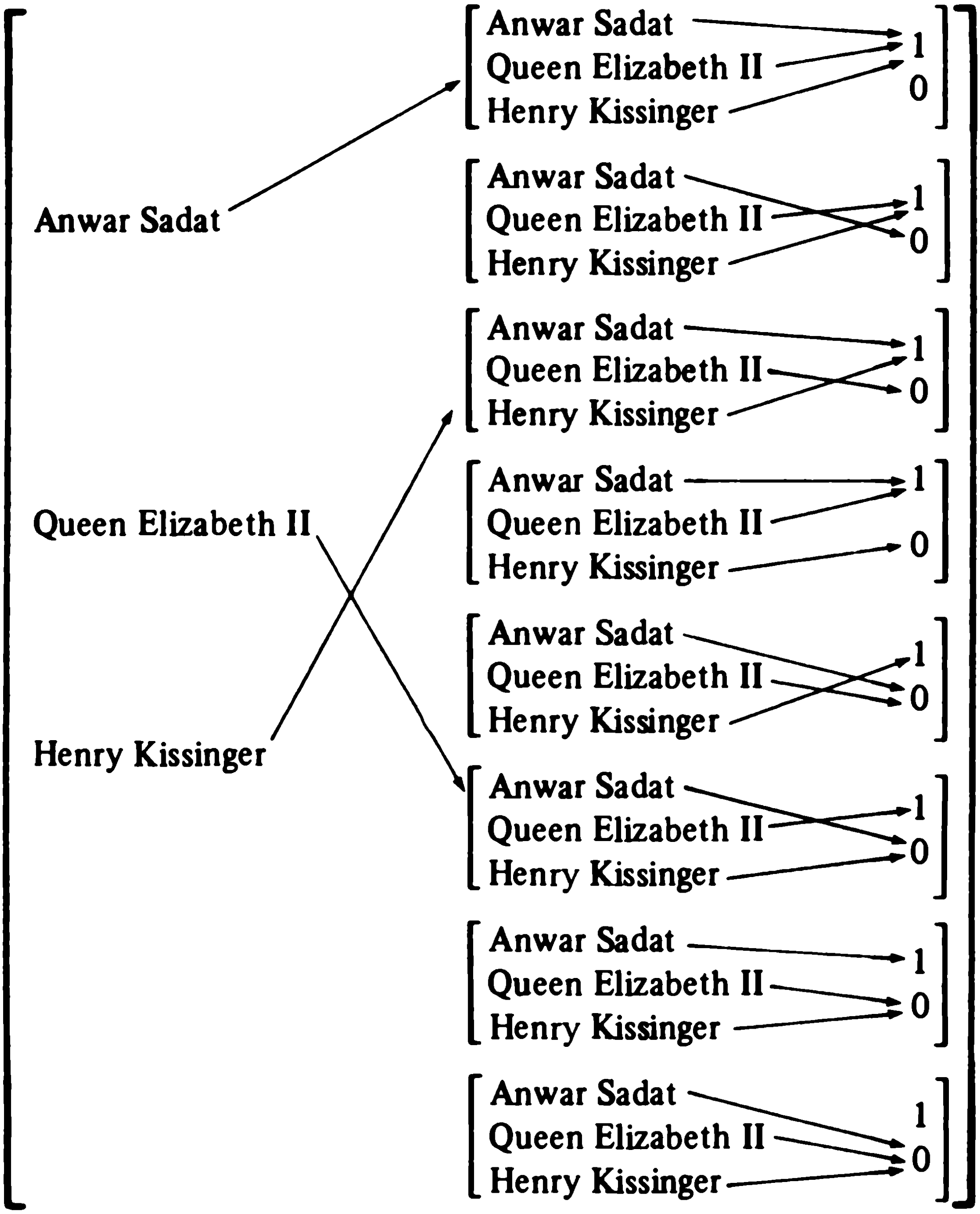
(2-21) If α is $\underset{\triangle}{N}$ and β is $\underset{\triangle}{VP}$, and if γ is $\underset{\alpha\ \ \beta}{S}$, then $[\![\gamma]\!] = [\![\beta]\!]([\![\alpha]\!])$.

PROBLEM (2-6). Determine by means of the semantic rules just given the semantic values of the phrase-structure trees of *Hank sleeps* and *Liz is-boring*.

As we emphasized in our discussion of the semantics of L_0, it is absolutely essential to keep in mind the distinctions among three different kinds of things: (1) expressions in the object language (e.g. *Sadie, is-boring*, and the tree in (2-17)), (2) entities serving as semantic values of expressions in the object language (e.g. Anwar Sadat and the functions in (2-13), and (3) expressions in the meta-language which are used in talking about the entities in (1) and (2) (e.g. α and the other expressions in (2-21)). In light of this discussion, the reader should now consider the following question: according to the semantic values given for L_{0E} so far and the semantic rules given for L_{0E}, what is the truth value of *Henry Kissinger sleeps*? Anyone who answers "0" or "false" has fallen into a trap. *Henry Kissinger sleeps* is not a sentence of L_{0E} since *Henry Kissinger* is not in its terminal vocabulary. Thus, not being a sentence of the object language, it is not assigned any semantic value by what has been given so far. Of course "Henry Kissinger sleeps" is a sentence of *English*, and in a semantics of that language we would want to be sure that some truth value is assigned to it.

Let us now ask what sort of values the V_t's should have. Semantically they seem to express relations between individuals. According to the grammar, a V_t followed by an N forms a VP. Is this consistent with the need for a VP to have as its semantic value a function from individuals to truth values (so that semantic rule (2-21) will function correctly)? The answer is "Yes" because we can take advantage of the isomorphism between relations and functions of a certain type. We need the semantic value of a V_t to be something that maps the semantic value of an N (i.e., an individual) into the semantic value of a VP (i.e., a function from individuals to truth values). Thus, we take the value of a V_t to be a function which yields other functions as its "outputs". Its domain will be the set of individuals, and its co-domain (i.e., set within which its "outputs" must lie) will be the set of all functions from individuals to truth values. For example, the V_t "loves" might have the following semantic value:

(2-22)



(In this diagram we have listed every characteristic function in the co-domain, including those that do not lie in the range of function (2-22).)

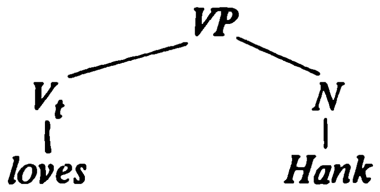
We now state the semantic rule corresponding to the syntactic rule $VP \rightarrow V_t\, N$:

(2-23) If α is V_t and β is N, and if γ is VP, then $[\![\gamma]\!]$ is $[\![\alpha]\!]([\![\beta]\!])$.

$\triangle$ $\triangle$ / \ α β

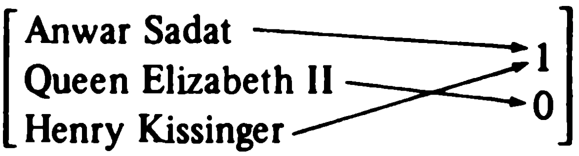
To illustrate, the grammar associates with the *VP* "loves Hank" the following tree structure:

(2-24)



Given that ⟦*Hank*⟧ = Henry Kissinger and that ⟦*loves*⟧ is the function given in (2-22), the semantic rule schema (2-19) will assign the corresponding semantic values to the trees rooted by V_t and N. Then, by semantic rule (2-23) we determine that the semantic value of the *VP* tree, i.e. of the verb phrase *loves Hank*, is:

(2-25)



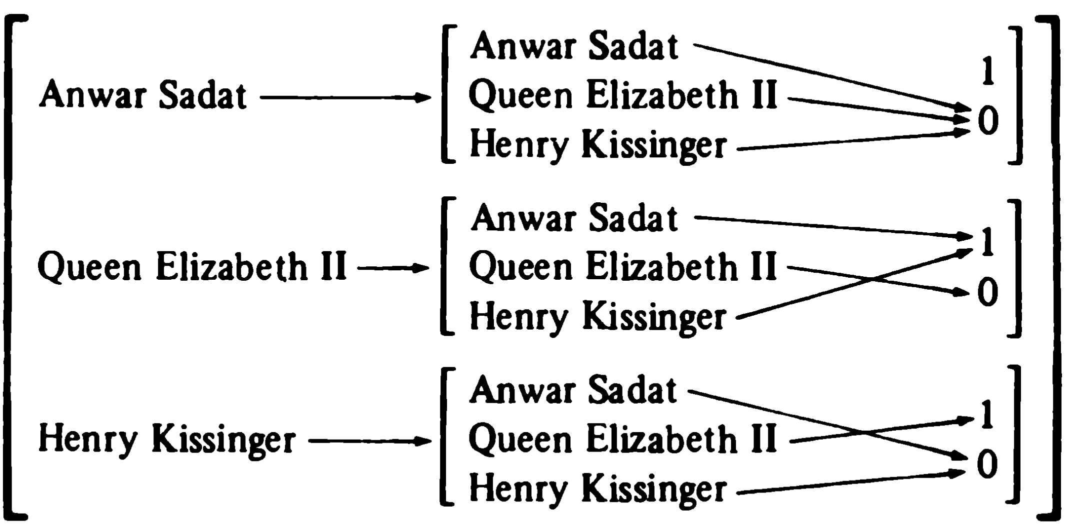
since this is the value of function (2-22) at the argument Henry Kissinger. Note that the semantic value of the *VP loves Hank* is, as it was designed to be, a function from individuals to truth values.

The reader should now be in a position to determine that the truth value that the sentence *Liz loves Hank* inherits from its tree structure is 0, according to our assumptions about ⟦*Liz*⟧, ⟦*Hank*⟧, ⟦*loves*⟧, and the semantic rules. (Recall that in L_0, two-place predicates denoted sets of ordered pairs of individuals. In fact, we will be able to show that by assigning V_t's a denotation like that in (2-22) we are giving a semantic treatment that is essentially equivalent to that in L_0; we will explain precisely why this is so on pp. 38–39, when we compare L_0 and L_{0E} in detail.)

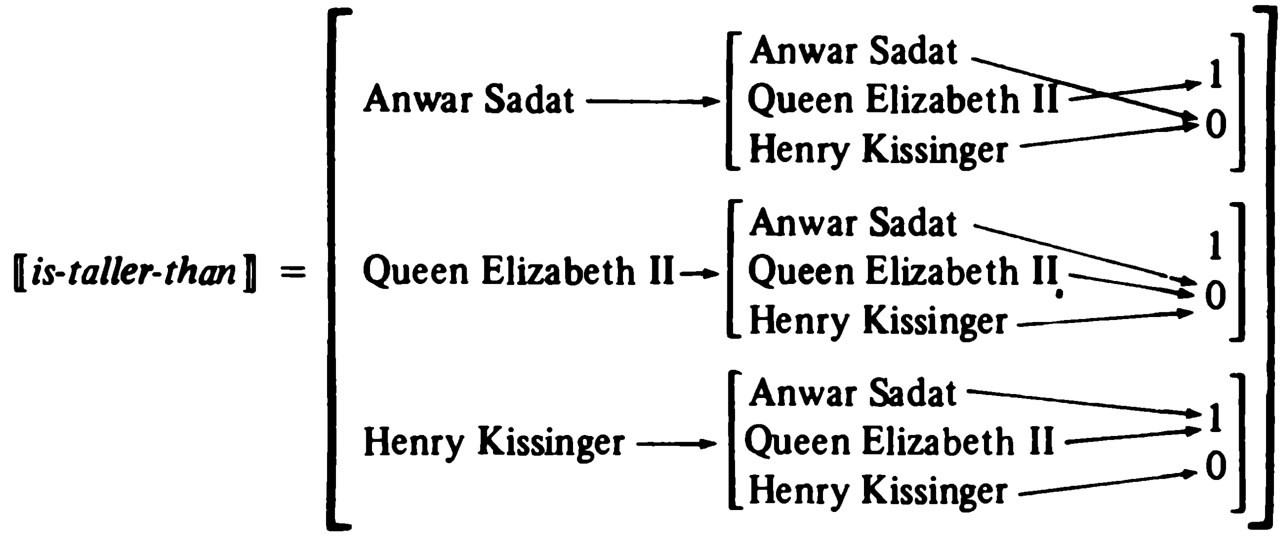
Returning to the specification of semantic values for lexical items of L_{0E}, let us assume that *hates* and *is-taller-than* have the following values. (In order to save space, we will not list the possible but unused values as we did in (2-22).)

(2-26)

⟦*hates*⟧ =



(2-27)



PROBLEM (2-7). Given our assumed universe consisting of three individuals, how many distinct semantic values are possible for a V_t? How many are possible given a universe of n individuals?

The remaining lexical items are the negation operator *it-is-not-the-case-that* and the two co-ordinating conjunctions *and* and *or*. We will assume that these have semantic values corresponding to the logical connectives "¬," "∧," and "∨" as defined by the customary truth tables:

(2-28)

p	$\neg p$
1	0
0	1

(2-29)

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

(2-30)

p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

Syntactically, *it-is-not-the-case-that* combines with a sentence to form another sentence. Therefore, given our "functional" approach to the semantics of L_{0E} we may treat it as a function mapping a truth value into a truth value, and this is in fact just what the truth table in (2-28) represents. Written in our diagrammatic notation it would appear as in (2-31):

(2-31) $\begin{bmatrix} 1 \rightarrow 0 \\ 0 \rightarrow 1 \end{bmatrix}$

This, then, is the semantic value we assign to *it-is-not-the-case-that*, and the semantic rule in which it figures is as follows:

(2-32) If α is $\underset{\triangle}{Neg}$ and ϕ is $\underset{\triangle}{S}$, and if ψ is $[_S\ \alpha\ \phi\ ]$, then $[\![\psi]\!]$ is $[\![\alpha]\!]([\![\phi]\!])$.

The logical connectives "$\wedge$" and "$\vee$" are two-place connectives forming a proposition from a pair of propositions. The conjunctions *and* and *or* similarly form one sentence from two, and the corresponding semantic operation ought therefore to take a pair of truth values and give a single truth value as a result. Accordingly, the semantic values assigned to *and* and *or* are the following functions:

(2-33)
$$[\![and]\!] = \begin{bmatrix} \langle 1,1\rangle \rightarrow 1 \\ \langle 1,0\rangle \rightarrow 0 \\ \langle 0,1\rangle \rightarrow 0 \\ \langle 0,0\rangle \rightarrow 0 \end{bmatrix}$$

(2-34)
$$[\![or]\!] = \begin{bmatrix} \langle 1,1\rangle \rightarrow 1 \\ \langle 1,0\rangle \rightarrow 1 \\ \langle 0,1\rangle \rightarrow 1 \\ \langle 0,0\rangle \rightarrow 0 \end{bmatrix}$$

Only one semantic rule is required, of course, corresponding to the syntactic rule $S \rightarrow S$ Conj S.

(2-35) If α is $\underset{\triangle}{Conj}$, ϕ is $\underset{\triangle}{S}$, and ψ is $\underset{\triangle}{S}$, and if ω is $[_S\ \phi\ \alpha\ \psi\ ]$, then $[\![\omega]\!]$ is $[\![\alpha]\!](\langle [\![\phi]\!], [\![\psi]\!] \rangle)$.

This completes the inventory of assumed semantic values for the terminal symbols of L_{0E} and of the semantic rules. Given these, the reader should now be able to determine the semantic value of any well-formed syntactic constituent of L_{0E}, and, in particular, of any sentence of L_{0E}.

PROBLEM (2-8). Determine the truth value assigned to each of the phrase-structure trees constructed in Problem (2-3), under the assumed assignments of semantic values to terminal symbols of L_{0E}. The fourth sentence in (2-10) would have posed a problem if we had attempted to assign semantic values to terminal strings rather than trees or labelled bracketings. Why? Why could we assign semantic values directly to terminal strings if we were dealing with a syntactically unambiguous language?

PROBLEM (2-9). Suppose new *Conj's* *if-and-only-if* and *only-if* were added to L_{0E} with semantic values appropriate to the logical connectives "$\leftrightarrow$" (the biconditional) and "$\rightarrow$" (the material conditional), respectively. Express their semantic values as functions of the sort given in (2-33) and (2-34). What semantic rules need to be added to accommodate these new *Conj's*?

3. *Alternative Formulations of L_{0E} and L_0*

The syntax and semantics of the two languages we have just described are similar in the overall effect of their semantic interpretations but differ in a number of details. In this section we digress to consider several "intermediate" languages borrowing various syntactic and semantic features of one or the other language. Our purposes here are to help us see what is essential to the truth conditional method and what is a matter of convenience or choice. We show, for example, that it is unimportant to our semantic method that L_{0E} is "English-like" and is defined syntactically by a phrase-structure grammar, while L_0 has neither of these properties, and we likewise show that it does not matter that two-place predicates were assigned relations as semantic values in L_0 while transitive verbs were assigned "function-valued" functions in L_{0E}. (Readers who already perceive these differences as inconsequential and do not desire additional practice in formulating languages of this sort may wish to skip directly to section III, p. 41.) The differences between the two languages may be summarized as follows:

1. While the basic expressions of L_0 look like those of the formal languages found in logic textbooks, as does the ordering of these in formulas, the lexical items of L_{0E} are deliberately designed to resemble English, and the word order in L_{0E} is much like that of English sentences.
2. We used recursive definitions to specify the syntax of L_0, but we used a context-free phrase structure grammar to specify that of L_{0E}.

3. Our semantic rules for L_0 classified sentences as "true" or "false" in the meta-language, while those of L_{0E} simply assigned 1 or 0 as the semantic value of a sentence.

4. The semantic values of one-place predicates were sets in L_0, but the semantic values of V_i's were characteristic functions of sets in L_{0E}.

5. The semantic values of two-place predicates were sets of pairs in L_0, but the semantic values of V_t's in L_{0E} were functions from individuals to the kind of semantic values assigned to V_i's.

6. The logical symbols $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ were treated syncategorematically in L_0 but the corresponding lexical items *it-is-not-the-case-that*, *and*, etc., in L_{0E} were introduced as members of the category *Neg* or *Conj*, and their semantic values were therefore defined as functions on truth values (or pairs of them), independently of the semantic rules corresponding to $S \rightarrow Neg\ S$ and $S \rightarrow S\ Conj\ S$.

However, these six characteristics are not in any essential way tied to the differences between natural languages and the formal languages of logicians, but rather are independent and somewhat arbitrary choices which we made for convenience (and for expository purposes). We could in fact alter the syntax or semantics of L_0 or L_{0E} with respect to almost any one of these characteristics while keeping the other five the same, as the reader can now assist us in demonstrating.

PROBLEM (2-10). Reformulate the syntax of L_{0E} as a set of recursive definitions. In this formulation the rules will produce only *sentences* of English (i.e. strings of words), not labelled bracketings or trees.

PROBLEM (2-11). Can the syntax of L_{0E} also be formulated as recursive definitions in such a way as to produce labelled bracketings for sentences (e.g. "$[_S [_N Sadie_N] [_{VP} [_{V_i} snores_{V_i}]_{VP}]_S]$" instead of *Sadie snores*)? If so, write a couple of rules of L_{0E} to illustrate how this is done.

The choice of sets versus characteristic functions for the semantic values of V_i's is likewise rather arbitrary, but does relate to the semantic rule (2-21). If we had assigned as the values of *snores*, *sleeps*, and *is-boring* the respective sets of which (2-13), (2-14) and (2-15) are the characteristic functions, then semantic rule (2-21) would have to be stated as follows:

(2-36) If α is $\underset{\triangle}{N}$ and β is $\underset{\triangle}{VP}$, and if γ is $[_S\ \alpha\ \beta]$ then $[\![\gamma]\!] = 1$ iff $[\![\alpha]\!] \in [\![\beta]\!]$, and is 0 otherwise.

Note that the functions given in (2-22), (2-26), and (2-27) as the semantic values of the V_t's are *not* characteristic functions inasmuch as their co-domains are not the set $\{1, 0\}$ (or any other set of two elements, one of which is specified as a "marked" or "distinguished" element). Rather, the *values* of these functions are themselves characteristic functions. If these values were reformulated as sets, in keeping with the modification (2-36) above, (2-22), for example, should be rewritten as:

(2-37)

Anwar Sadat ⟶ {Anwar Sadat, Queen Elizabeth II, Henry Kissinger}
Queen Elizabeth II ⟶ {Queen Elizabeth II}
Henry Kissinger ⟶ {Anwar Sadat, Henry Kissinger}

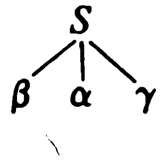
In this representation it is perhaps somewhat easier to grasp the import of the semantic value assigned to *loves*. Recall that the argument of (2-37) is the individual named by the N which is the direct object of *loves*. That is, if the N is *Hank*, then since $[\![$*Hank*$]\!]$ = Henry Kissinger, we take the value of (2-37) at the argument "Henry Kissinger" and find that it is the set {Anwar Sadat, Henry Kissinger}. This, then, is the set of individuals who "love Hank." (Compare this set with the characteristic function in (2-25)). Similarly, it is easy to see from (2-37) that the set of those who "love Sadie" is the entire domain of discourse and that the set of those who "love Liz" is just {Queen Elizabeth II}. Given this modification, only one more step is needed in order to think of a sentence $N + V_t + N$ in L_{0E} as expressing the proposition that two individuals named by the subject and object nouns stand in the binary relation named by the verb, just as we explicitly did with sentences like $L(j, m)$ in L_0. Given the universe we assumed for the semantics of L_{0E}, the semantic value given to the V_t *loves* would correspond to this set of ordered pairs:

(2-38)

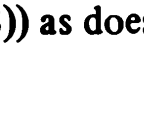
{⟨Anwar Sadat, Anwar Sadat⟩, ⟨Queen Elizabeth II, Anwar Sadat⟩,
⟨Henry Kissinger, Anwar Sadat⟩, ⟨Queen Elizabeth II, Queen Elizabeth II⟩,
⟨Anwar Sadat, Henry Kissinger⟩, ⟨Henry Kissinger, Henry Kissinger⟩}

While (2-38) represents exactly the same information as (2-22), notice that the function is 'backward' with respect to the ordered pairs. That is, an argument in (2-22) appears as a *second* coordinate of an ordered pair while individuals appearing in the values of the function turn up as *first* coordinates. Note, for example, that in (2-22) Anwar Sadat is mapped into a function which in turn maps Queen Elizabeth II into 1; thus, the ordered pair ⟨Queen Elizabeth II, Anwar Sadat⟩ appears in (2-38).

Representing the semantic value of a V_t like *loves* as a set of ordered pairs suggests that we might formulate the associated semantic rule in the following way:

(2-39) If α is V_t, β is N, and γ is N, and if δ is  , then $[\![\delta]\!]$ is 1 iff $\langle [\![\beta]\!], [\![\gamma]\!] \rangle \in [\![\alpha]\!]$, and is 0 otherwise.

While this would give the correct semantic values for sentences of the form $N + V_t + N$, it runs afoul of the condition that the semantic rules should be in one-to-one correspondence with the syntactic rules. Since the syntax does not contain a rule of the form $S \rightarrow N\, V_t\, N$, there will be no semantic rule of the form given in (2-39). It is nonetheless possible to adhere to our stipulation of one semantic rule for one syntactic rule and still represent the semantic value of a V_t as a set of ordered pairs. We simply restate the semantic rule corresponding to the syntactic rule $VP \rightarrow V_t + N$ as follows:

(2-40) If α is V_t and β is N, and if γ is  then $[\![\gamma]\!]$ is the set of all x such that $\langle x, [\![\beta]\!] \rangle \in [\![\alpha]\!]$.

PROBLEM (2-12). Verify that (2-40) yields the same semantic value for *loves Hank* (where $[\![$*loves*$]\!]$ is as in (2-38)) as does (2-23) (where $[\![$*loves*$]\!]$ is as in (2-22)).

It may be useful to many readers to consider in a bit more mathematical detail the identification we have made between relations and function-valued functions. Any binary relation R between members of sets A and B can be regarded in a standard way as a subset of $A \times B$, the set of ordered pairs $\langle a, b \rangle$ such that a is in A and b is in B. Being a set, R can thus be identified with a function in $\{0, 1\}^{A \times B}$ – recall our earlier discussion. From this characteristic function, f_R, of R we can define functions $g_{R,b}$, for every b in B, and the function h_R which we will identify with R. Let

$$g_{R,b}(a) = f_R(\langle a, b\rangle) \text{ for every } a \text{ in } A \text{ and each } b \text{ in } B,$$

$$h_R(b) = g_{R,b} \text{ for every } b \text{ in } B.$$

Each $g_{R,b}$ is a characteristic function of a subset of A, and thus the range of h_R is included in $\{0, 1\}^A$. Hence h_R, which maps each b in B to the set of members of A which stand in relation R to b, is a member of $(\{0, 1\}^A)^B$.

Note that $h_R(b)(a)$ means a stands in R to b – i.e., $h_R(b)(a) = f_R(\langle a, b\rangle)$, where the notation reverses the order of a and b. We have chosen to identify R with a function on B because English syntax treats a transitive verb + direct object as a constituent of a sentence, rather than subject + transitive verb. One must take care not to confuse $h_R(b)(a)$ with $f_R(\langle b, a\rangle)$, which is defined if $b \in A$ and $a \in B$ – as, for example, with the relation expressed by *loves*, where $A = B$.

PROBLEM (2-13). Let A consist of the colors red, white, and blue, and B consist of the countries of the world; and let R be $\{\langle a, b\rangle | a \text{ appears in the flag of } b\}$.

What is h_R(Switzerland)? h_R(Great-Britain)?

PROBLEM (2-14). Let $A = B$ = {Anwar Sadat, Queen Elizabeth II, Henry Kissinger} and let R be the relation {⟨Anwar Sadat, Queen Elizabeth II⟩, ⟨Queen Elizabeth II, Henry Kissinger⟩, ⟨Henry Kissinger, Anwar Sadat⟩}. Diagram the corresponding function h_R in the form exemplified in (2-22).

PROBLEM (2-15). Of what set is (2-31) the characteristic function? If this set were assigned as the value of *it-is-not-the-case-that*, how would (2-32) be stated?

PROBLEM (2-16). Suppose *and* and *or* were assigned as semantic values the sets corresponding to the respective characteristic functions given in (2-33) and (2-34). How would semantic rule (2-35) then be stated?

PROBLEM (2-17). Find the two-place relation R on individuals such that $\langle x, y\rangle \in R$ iff α *hates* β is true, where $[\![\alpha]\!] = x$, $[\![\beta]\!] = y$, and $[\![\textit{hates}]\!]$ is as given in (2-26).

PROBLEM (2-18). Express (2-33) and (2-34) as functions with domain $\{1, 0\}$ and having as values functions from $\{1, 0\}$ to $\{1, 0\}$. Write the semantic rule corresponding to (2-35) under this new formulation of $[\![\textit{and}]\!]$ and $[\![\textit{or}]\!]$.

As we remarked, we could reformulate the syntax and semantics of L_0 in such a way that the logical symbols are treated as basic expressions and assigned semantic values directly. One way to do this would be as follows:

Add to the basic expressions of L_0:

Category	*Basic Expressions*
Sentence operators	$\neg$
Sentence connectives	$\wedge, \vee, \rightarrow, \leftrightarrow$

Replace formation rule 3. in (2-2) by:

3. If ϕ is a sentence and α is a sentence operator, then $\alpha\phi$ is a sentence.

and replace formation rules 4. through 7. by the single rule:

4. If ϕ and ψ are sentences, and α is a sentence connective, then $[\phi\alpha\psi]$ is a sentence.

The reader can easily determine that this modified syntax specifies exactly the same set of well-formed sentences as did (2-2) (although of course not the same set of well-formed expressions of all categories, since we have added new categories).

To the semantics we add the specifications of the semantic values of the sentence operator and the sentence connectives corresponding to their truth tables. "$\neg$," for example, will receive the semantic value given in (2-31), and "$\wedge$" will be assigned the same semantic value given to *and* in (2-33).

The semantic rules will then be amended in the following way. Replace semantic rule 3. in (2-8) by:

3. If ϕ is a sentence and α is a sentence operator, then $[\![\alpha\phi]\!] = [\![\alpha]\!]([\![\phi]\!])$.

and replace semantic rules 4. through 7. by the single rule:

4. If ϕ and ψ are sentences, and α is a sentence connective, then $[\![[\phi\alpha\psi]]\!] = [\![\alpha]\!](\langle [\![\phi]\!], [\![\psi]\!] \rangle)$.

The reader should have little difficulty in ascertaining that the same semantic values will be assigned to the sentences of L_0 as before.

PROBLEM (2-19). Revise the syntax and semantics of L_{0E} so that *and*, *or*, and *it-is-not-the-case-that* are introduced syncategorematically.

PROBLEM (2-20). Revise the syntax and semantics of L_0 so that the parentheses and square brackets as well as the logical connectives are treated as basic expressions.

PROBLEM (2-21). Give a syntax and semantics for L_0 in which each sentence connective (i.e., "$\wedge$," "$\vee$," "$\rightarrow$," and "$\leftrightarrow$") combines syntactically with a single sentence to yield an expression of the category "sentence operator." For example, $\wedge$ and $M(b)$ will form the sentence operator $M(b) \wedge$, which will, in turn, combine with another sentence, say $B(j)$, to give $[M(b) \wedge B(j)]$. Write the semantic rules in such a way that each non-basic sentence operator will be assigned an appropriate semantic value.

III. A SYNOPSIS OF TRUTH-CONDITIONAL SEMANTICS

We now summarize the essential points that are common to the applications of truth-conditional semantics we have presented so far.

First, we must emphasize that any truth-conditional semantics is always tightly interconnected with the syntax of the language in question. This is why the first step in the consideration of our example languages L_{0E} and L_0 was to specify the syntax. One should not infer from this, however, that in order to do truth-conditional semantics on, e.g., English, all syntactic questions must first be settled. Quite the contrary. The close interconnections between syntax and semantics mean that certain decisions made with respect to the syntax will have consequences for the semantics, and the converse is also true, but nothing prevents research in both areas of a language from proceeding in parallel. L_{0E} and L_0, of course, were deliberately chosen to be very simple languages with relatively few problems in the syntax so that we could illustrate the truth-conditional semantic method.

What is the minimum that a truth-conditional semantics requires of the syntax in order to operate? There must be at least a set of *syntactic categories*, one of which is the category "sentence" or something of the sort – the category associated with truth or falsity. There must in addition be some initial assignment of expressions of the language to these categories, and then, since in every interesting case we will be dealing with an infinite language, there must be rules which effect the assignment of the remaining well-formed expressions to their respective categories. We have seen how all this can be accomplished by a system of context-free phrase-structure rules in the case of L_{0E} or by a system of initial assignments of basic expressions to categories and recursive formation rules in the case of L_0. We will return in a moment to consider some important ways in which these systems of syntactic specification differ, but for now it suffices to note that either one accomplishes syntactically what is necessary for the semantics.

What, then, are the essential ingredients of a truth-conditional semantics, given the aforementioned syntactic information? They are as follows:

1. A set of things which can be assigned as semantic values. In the system assumed thus far these are (1) a set of individuals, (2) a set of truth values, and (3) various functions constructed out of these by means of set theory.

2. A specification for each syntactic category of the type of semantic value that is to be assigned to expressions of that category (e.g., names are to have individuals assigned, etc.) Sentences are to be assigned truth values.

3. A set of semantic rules specifying how the semantic value of any complex expression is determined in terms of the semantic values of its components.

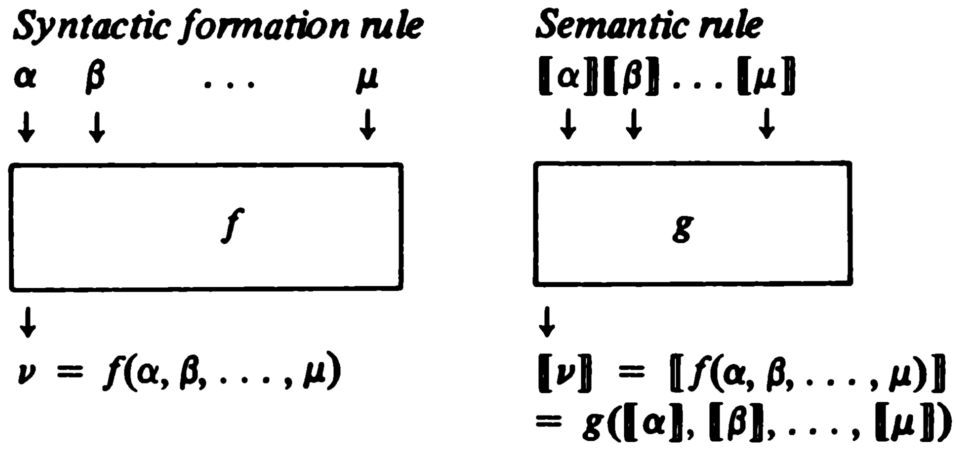
4. A specific assignment of a semantic value of the appropriate type to each of the basic expressions.

It might be helpful for the reader at this point to return to the relevant sections of this chapter and check that it was in fact just this information that was specified in giving the semantics of both L_{0E} and L_0.

Item 3 above contains an implicit assumption that our semantics is to adhere to the *Principle of Compositionality*, which we referred to earlier. As we saw, adherence to this principle leads us to construct our syntax and semantics so that they work in tandem.

Consider, for example, a syntactic formation rule of the form "if α is an A, and β is a B, . . . , and μ is an M, then $f(\alpha, \beta, \ldots, \mu)$ is an N" (where A, $B, \ldots, M, N$ are syntactic categories). The function f specifies how the inputs are to be mapped into the output, i.e., it specifies the mode of combination of the arguments. Corresponding to this syntactic rule there will be a semantic rule of the form "if α is an A, and β is a B, . . . , and μ is an M, then $[\![f(\alpha, \beta, \ldots, \mu)]\!]$ is $g([\![\alpha]\!], [\![\beta]\!], \ldots, [\![\mu]\!])$". Here, g is a function which, so to speak, specifies the "semantic mode of combination" of the semantic values which are its arguments. The situation can be represented by a diagram such as the following:

(2-41)



To use a precise mathematical term for this situation, we may say that there is a *homomorphism* from syntax to semantics. Since the mapping is not an isomorphism but only a homomorphism, different syntactic structures can receive the same semantic value.

As an example, consider the first syntactic formation rule for L_0 in (2-2). This rule takes as inputs a one-place predicate and a name and yields an expression of the category "sentence." The function f which expresses the syntactic mode of combination can be stated as follows: write the one-place predicate followed by a left parenthesis followed by the name followed by a right parenthesis. The corresponding semantic rule, rule 1. of (2-8), says that if δ is a one-place predicate and α is a name, then the semantic value of $\delta(\alpha)$, i.e., of $f(\delta, \alpha)$, is a function of the semantic values of δ and α; specifically, this function is one which assigns to $\delta(\alpha)$ the semantic value *true* just in case $[\![\alpha]\!]$ is a member of $[\![\delta]\!]$, and assigns it *false* otherwise. Examination of the remaining syntactic and semantic rules of L_0 in (2-2) and (2-8) reveals that the homomorphism dictated by the Principle of Compositionality is indeed present. It was in fact the ease of constructing the desired homomorphism between syntax and semantics that led logicians to give their syntactic rules the form that they did. To the linguist, who is accustomed to phrase-structure rules as devices for expressing immediate constituent structure, the logician's syntactic formation rules may at first appear rather strange, but then, after some reflection, the two systems may appear to be, in effect, notational variants. It is easy to see that a phrase-structure rule of the form $A \rightarrow B\ C \ldots M$ (where $A, B, C, \ldots, M$ are all non-terminal symbols) expresses roughly the same sort of syntactic information as does a formation rule of the form "If β is a member of category B, ..., and μ is a member of category M, then $\beta \ldots \mu$ is a member of category A." Note that the function which specifies the mode of syntactic combination in this rule merely concatenates the arguments in a particular order, and in fact any context-free phrase-structure grammar can be converted into an equivalent set of syntactic formation rules in which the mode of combination involves only concatenation, just as was demonstrated for L_{0E} in Problem (2-12). However, syntactic formation rules as used by Montague will allow considerably more complex modes of combination than this – they can, for example, be operations that are carried out by transformational rules of a transformational grammar – and for this reason it is not true that for every set of syntactic formation rules there is an equivalent context-free phrase-structure grammar. Formation rules which exploit this possibility of carrying out "transformation-like" operations figure prominently in Montague's system for English in PTQ.

IV. THE NOTION OF TRUTH RELATIVE TO A MODEL

Let us return now to the list of four essential components of a truth conditional semantics we gave earlier. If we look at them more carefully, we see that the items in the list fall into two broad classes corresponding to the kinds of factors which go into determining the semantic values of a sentence. We could say, roughly, that a particular sentence gets the semantic value that it does – is either true or false – because of certain formal structural properties that it has on the one hand, and on the other hand because of certain facts about the world. To take an example from L_{0E}, the sentence *Liz snores* gets the semantic value 1 (true) because all the following considerations interact to yield this result:

1. *Liz* has as its semantic value Queen Elizabeth II.
2. *snores* has as its semantic value a function from individuals to truth values which maps Queen Elizabeth II into 1.
3. *Liz* is a name, and *snores* is an intransitive verb, and the truth value of any sentence composed of a name plus an intransitive verb is determined by applying the function which is the semantic value of the intransitive verb to the argument which is the semantic value of the name.

Of these, the last is a theory-internal condition which is specified as a part of our semantic theory for L_{0E} in particular. The first two, however, are conditions which have to do with assumed facts about the connections between the language and the world and facts about the way the world is. That is, *Liz snores* might receive a different semantic value if *Liz* referred to (had as its value) someone other than Queen Elizabeth II or if *snores* referred to some other set of individuals (i.e., if different people snored than those who are assumed to snore). The truth value of *Liz snores* might also be different if we computed the semantic value of $N + V_i$ sentences in some way other than that given, but if so, that would be an essential modification of the semantic system of L_{0E}, and that is a different sort of variation than is involved in imagining, say, a different set of snorers.

All this discussion is by way of introducing the notion of a *model*. Formally, a model is an ordered pair $\langle A, F \rangle$, where A is a set, the set of individuals, and F is a function which assigns semantic values of the appropriate sort to the basic expressions. All the rest (for which there seems to be no standard name in the literature) is taken as the fixed part of the semantics for a particular language, and we may then examine the effect on the semantic values of expressions in the language as we allow the model to vary. The various choices of a model, then, are intended to represent the various ways we might effect the fundamental mapping from basic expressions to things in the world, while the fixed remainder represents the contribution to semantic values (and in particular, to truth values of sentences) made by the semantic theory itself. Having made this distinction, it is no longer sufficient to say that S is true *simpliciter*; rather, we must say that S is *true with respect to (a particular) model M*. As a notation for this, we use the notation for semantic values together with a superscript for the model.

(2-42) Notational Convention 3: For any expression α, we use $[\![\alpha]\!]^M$ to denote the semantic value of α with respect to model M.

The model we chose to illustrate the semantics of L_{0E} is $\langle A_{0E}, F_{0E} \rangle$, where A_{0E} is the set {Anwar Sadat, Queen Elizabeth II, Henry Kissinger}, and F_{0E} is the function which is the union of the sets given in (2-12) (represented as a function), (2-13), (2-14), (2-15), (2-22), (2-26), and (2-27). Why do we not include in this list the functions assigning values to *and*, *or*, and *it-is-not-the-case-that*, i.e., (2-31), (2-33), and (2-34)? The reason is that, although these are indeed a part of the assignment of semantic values to basic expressions, these particular basic expressions are distinguished from all the others in being a part of the "logical" vocabulary of the language; hence, their values are taken to be fixed once and for all and are not considered as part of the variable model. That is, after all, what we would expect of the constants of a particular logic: we can easily imagine different situations in which the set of snorers varies, but it is difficult to imagine a situation in which *and* means something other than what it does in our world. Thus, we amend our characterization of a model given above to say that the function F assigns semantic values to all the basic, *non-logical* expressions of the language.

Since the notion of *truth with respect to a model* plays a central role in most versions of truth-conditional semantics (and all versions discussed in this book), the term *model-theoretic semantics* is often used as a broad term for the kinds of approaches to semantics we are concerned with: we will henceforth adopt this term ourselves.

In the model we chose for L_0 the set A was the set of all living persons and the function F was given in (2-7). It may be helpful to consider some other possible models for L_0 in order for the reader to grasp firmly the point that a sentence may be true with respect to one model and false with respect to another.

The model M_1 $(= \langle A_1, F_1 \rangle)$

A_1 is the set of states of the United States. F_1 is defined as follows: $F_1(m) =$ Michigan, $F_1(j) =$ California, $F_1(d) =$ Alaska, $F_1(n) =$ Rhode Island, $F_1(M) = \{$Maine, New Hampshire, Vermont, Massachusetts, Connecticut, Rhode Island$\}$, $F_1(B) =$ the set of states that have Pacific coasts, $F_1(K) =$ the set of pairs of states such that some part of the first lies west of some part of the second (e.g., both $\langle$Washington, Oregon$\rangle$ and $\langle$Oregon, Washington$\rangle$ are in this set), $F_1(L) =$ the set of pairs of states such that the first is larger than the second. (Note that we have expressed the semantic values of K and L as ordered pairs. The reader should make the mental translation to functions from states to characteristic functions of sets of states.)

The model M_2 $(= \langle A_2, F_2 \rangle)$

A_2 is the set of all integers (positive and negative whole numbers and 0). $F_2(j) = 0$, $F_2(m) = 2$, $F_2(d) = 9$, $F_2(n) = -1$, $F_2(M) =$ the set of all odd integers, $F_2(B) =$ the set of all perfect squares, $F_2(K) =$ the set of all pairs of integers such that the first is greater than the second, i.e., $F_2(K) = \{\langle x, y \rangle | x > y\}$, and $F_2(L) =$ the set of all pairs of integers such that the first is the square of the second, i.e., $F_2(L) = \{\langle x, y \rangle | x = y^2\}$.

The model M_3 $(= \langle A_3, F_3 \rangle)$

A_3 is the set of all chemical elements. F_3 is defined as follows: $F_3(m) =$ magnesium, $F_3(j) =$ iodine, $F_3(d) =$ krypton, $F_3(n) =$ sodium, $F_3(M) =$ the set of "rare-earth" elements, $F_3(B) =$ the set of halogen elements, $F_3(K) =$ the set of all pairs of elements such that the first has a greater atomic number than the second, $F_3(L) =$ the set of all pairs of elements such that the two form a chemical compound containing no other elements.

The sentence $M(d)$ is false with respect to M_1 (because Alaska is not in $F_1(M)$, i.e., not a New England state), true with respect to M_2 (because 9 is odd), and false in M_3 (because krypton is not a rare-earth element). The sentence $B(j)$, on the other hand, happens to be true in all three models (because California has a Pacific coast, 0 is a perfect square, and iodine is a halogen). The sentence $K(j, n)$ is true in M_1 (California is west of Rhode Island), true in M_2 (0 is greater than -1), and true in M_3 (iodine has a greater atomic number than sodium). Finally, $L(n, m)$ is false in all three models (Rhode Island is not larger than Michigan, nor is -1 the square of 2, nor does sodium form a compound with magnesium).

PROBLEM (2-22). Construct a model for L_0 in which $M(d)$, $B(j)$, $K(j, n)$, and $L(n, m)$ are all true.

V. VALIDITY AND ENTAILMENT DEFINED IN TERMS OF POSSIBLE MODELS

There are various advantages that the notion of truth relative to a model has over the notion of truth *simpliciter*. The logician (or linguist, for that matter) may not actually be very interested in, say, the set of all bald persons and the question of just which persons belong to the set and which do not, or in similar questions about the denotations of other basic expressions, the answers to which would involve a great deal of empirical knowledge but would not be particularly enlightening for the overall theory of semantics. But it is nevertheless of interest to formulate one's semantics explicitly enough that *if* these basic denotations were ever specified in some way or other, then the precise definitions of truth for the sentences of the whole language would follow automatically. Or the logician might be interested in describing the syntax and basic semantic procedure for an "all-purpose" formal language that might profitably be put to use in talking about various domains of discourse.

But there is an even more important reason for being interested in the notion of truth relative to a model: Consider the difference between the sentences $[B(j) \wedge K(d,j)]$ and $[B(j) \rightarrow [K(d,j) \rightarrow B(j)]]$ of L_0 in the model given earlier in (2-7). Both of these sentences are true in this interpretation, but there is more to be said about the truth of the second sentence than just this. Its truth does not really depend in any way on the semantic values assigned to the basic expressions in it, but rather can be traced to general properties of its syntactic form and to the way we have given truth conditions for the conditional connective (viz., the semantic rule 6. in (2-8)). In fact, any sentence of the form $[\phi \rightarrow [\psi \rightarrow \phi]]$ will turn out to be true in L_0, given any possible model whatsoever. (A sentence with this form will no doubt be recognizable to the reader as one of the *valid sentences*, or *tautologies*, of the propositional calculus, of which our language is a rather simple extension.) The former sentence, on the other hand, will be true in some models and false in others, depending on the denotations assigned to B, K, d and j by the model. It turns out that by distinguishing those sentences of L_0 that are true with respect to *all* models from those that are true only with respect to *some* of the possible models, we can give a definition of *valid sentence* of L_0 (or *logically true sentence* of L_0) that satisfies the usual expectations as to which sentences of this language ought to count as logically valid:

(1) A sentence of L_0 is *valid* iff it is true with respect to every possible model for L_0.

Other familiar logical properties of sentences and relations between sentences can also be defined by using the notion of truth with respect to a model, quantifying over the class of possible models:

(2) A sentence of L_0 is *contradictory* iff it is false with respect to every possible model for L_0.

(3) Two sentences of L_0 are *logically equivalent* iff the first is true in exactly the same models in which the second is true and in no others.

(4) A sentence ϕ of L_0 is a *logical consequence* of a set of sentences Γ (or equivalently, Γ *logically entails* ϕ) iff every model in which all the sentences of Γ are true is a model in which ϕ is true also.

Now the possibility of giving definitions of these notions will be of as much interest to the linguist as to the logician since it is widely held among linguists that an account of these properties of English sentences and relations among English sentences is an important goal (some would say *the* goal) of semantics.

For the relation *logically equivalent* we would like to be able to substitute *synonymous* but cannot because most philosophers of language and linguists hold synonymy to be a much narrower relation among sentences, their synonymy taking into account various subtleties such as focus, conventional and conversational implicature, perhaps stylistic connotations of particular words, all of which we are ill-equipped to deal with formally at present. But fortunately the relation of logical equivalence seems to be a more workable and useful relation in the initial stages of developing a semantic theory for natural language.

Although we have defined logical entailment as a relation between a *set* of sentences Γ and a sentence ϕ, we can obviously consider entailment between a pair of sentences as the special case of this definition in which the set Γ contains only one sentence. Logical equivalence is of course simply mutual logical entailment between a pair of sentences.

PROBLEM (2-23). Find and example of each of the following in L_{0E}: (1) a valid sentence, (2) a sentence which is true in the model of L_{0E} given in the text but not valid, (3) a contradictory sentence, (4) a sentence which is false but not contradictory, (5) two sentences which are logically equivalent, (6) a non-empty set of sentences Γ and a sentence ϕ (not in Γ) such that ϕ is a logical consequence of Γ but not of any proper subset of Γ.

There are, to be sure, other properties of sentences or parts of sentences and relations among them that linguists have traditionally treated under the rubrics of synonymy (or logical equivalence) and entailment. For example, the validity of the sentence *If John is a bachelor, then he is an unmarried man* is attributed to the 'synonymy' of the phrases *bachelor* and *unmarried man*. Yet it is not at all obvious how the notion of truth with respect to a model can be extended to account for this example, since the unvarying truth of this last example cannot be traced to the syntactic form of the sentence but rather depends *also* on the particular basic expressions *bachelor*, *unmarried* and *man*. (If we were to follow Quine's terminology, we would refer to this latter kind of example as an *analytic sentence* and the former example, in which the syntactic form is responsible for validity, as a *logically true sentence*.) More will be said about such cases later.

VI. MODEL THEORY AND DEDUCTIVE SYSTEMS

The reader is probably aware that there is an older, more traditional way of characterizing validity and entailment for formal languages in terms of the notions of a *deduction* and *rules of inference*. This may be accomplished either through an *axiom system* or, in most contemporary logic texts, a system of *natural deduction* (cf., e.g. Blumberg, 1967). By the axiomatic method, a list of *axioms* (or *axiom schemata*) and a *rule* (or *rules*) *of inference* are given for a formal language, and then a *proof* is defined as any sequence of sentences of the language such that each sentence is either an axiom (or instance of an axiom schema) or follows from one or more of the preceding sentences of the sequence by some rule of inference. A *theorem* (which is to correspond to our intuitive notion of a logically true sentence) of the language is any sentence ϕ of the language for which there is a proof ending in ϕ. To illustrate, the following is a possible axiomatization for a propositional language which resembles L_0, except that it treats $\wedge$, $\vee$, and $\leftrightarrow$ as defined in terms of $\neg$ and $\rightarrow$. We have here axiom schemata rather than axioms proper because they are stated in terms of meta-language variables rather than actual sentences of L_0; hence each line is a schema for an infinite number of sentences of L_0 with similar syntactic forms, each of which is an axiom.

(A1) $[\phi \rightarrow [\psi \rightarrow \phi]]$

(A2) $[[\phi \rightarrow [\psi \rightarrow \chi]] \rightarrow [[\phi \rightarrow \psi] \rightarrow [\phi \rightarrow \chi]]]$

(A3) $[[\neg\psi \rightarrow \neg\phi] \rightarrow [[\neg\psi \rightarrow \phi] \rightarrow \psi]]$

With these axiom schemata, one rule of inference would suffice to complete the axiomization of L_0, the rule of *modus ponens* (also known as the *rule of detachment*.) This is the rule that permits one, when given ϕ and $[\phi \rightarrow \psi]$, to infer ψ.

With such a deductive apparatus, definitions of properties of sentences and relations among sentences can be given which can be proved to correspond exactly to the semantic definitions given earlier in terms of possible models. This correspondence is illustrated by the table below:

Definitions in terms of possible models:	*Corresponding deductive definition*:
1. A sentence of L_0 is *valid* iff it is true with respect to every possible model for L_0.	1. A sentence of L_0 is a *theorem* of L_0 iff there is a proof of it from the above axiom schemata alone.
2. A sentence of L_0 is a *contradiction* iff it is false with respect to every possible model for L_0.	2. A sentence of L_0 is a *contradiction* iff its negation is a theorem of L_0.
3. A sentence ϕ of L_0 is *logically entailed* by a set of sentences Γ iff every model in which all the sentences of Γ are true is a model in which ϕ is true.	3. A sentence ϕ of L_0 is *deducible* (or *provable*) from a set of sentences Γ iff there is a sequence of sentences of L_0 such that each is either an axiom or belongs to Γ or else follows from some of the preceding sentences in the sequence by the rule(s) of inference, and ϕ is the last sentence of the sequence.
4. Two sentences of L_0 are *logically equivalent* iff they are true in exactly the same models (or equivalently, if each logically entails the other.)	4. Two sentences of L_0 are *logically equivalent* iff each is deducible from the other.

The possibility of giving corresponding deductive and semantic definitions for logical systems is of fundamental significance in modern logic, and indeed much research in logic is devoted to producing the semantics and corresponding axiomatizations for various logics and proving mathematically that the semantic definitions of validity and logical consequence are in fact exactly equivalent to the definitions of theoremhood and deducibility that result from the axiomatization. (See Blumberg 1967 and Henkin 1967 for additional discussion.)

The method of axiomatization is relevant to our present discussion because it has suggested to some linguists that an axiomatization (or perhaps natural deductive system) might be given for an appropriately formalized language of semantic representations (or as Lakoff (1972) has called it, a *natural logic*), thus enabling us to account for all the relations of entailment, logical equivalence, etc., that exist among English sentences, without appeal to model-theoretic semantics. This would be particularly appealing to the linguist in view of the ideal of the "autonomous semantic representation" that has no defined relation to non-linguistic (or non-mental) objects. 'Linguistics and Natural Logic' (Lakoff 1972) seems to suggest that a deductive system of some sort is to be the means of achieving Lakoff's goal of "a logic which is capable of accounting for all correct inferences made in natural language and which rules out incorrect ones." (p. 589).

But two points need to be made in response to this suggestion. First, to take the construction of an axiomatic system (but not a formally interpreted language) as the goal of the semantic analysis of natural language is, in an important sense, to miss the point of what semantics is all about. It is true that the advantage of symbolic logic, in its early form, was that it allowed one to completely ignore the meaning of the propositions involved in an argument and concentrate on the form of the argument entirely. Deductive systems are purposefully formulated in just such a way as to make interpretation of the primitive symbols irrelevant for carrying out proofs. Nevertheless, the ultimate interest in formal deductive systems for philosophers of language has always lain in the way they mimic certain properties of natural languages and how people use them, and languages in turn have the essential feature of referring to objects and situations beyond themselves. Without their reference to things in the world, human languages would be impossible to imagine. Notions such as "synonymy" and "entailment" thus always have lurking behind them the connection of languages with the world, and it is these connections which ultimately give the logical properties of sentences their interest for us, whether we temporarily ignore the connections with the world or not. The definition of truth with respect to a model has the advantage that it allows us to capture the definitions of logical truth, logical entailment, and related notions *and* at the same time to capture our intuitions of the essential "aboutness" of natural language; deductive systems satisfy only the first of these two objectives.

A second reason for preferring the semantic method to the deductive is that certain logics *cannot* be given axiomatic definitions of validity and entailment, though model-theoretic definitions of these notions are perfectly feasible for them. It can be proved mathematically that the set of valid sentences of second-order logic (logics involving quantification over predicates) cannot be finitely axiomatized (cf. Henkin 1950), whereas a semantic definition of validity and entailment can be given.

One might wonder whether there is an axiomatization of the fragment of English which Montague described in PTQ. We conjecture that such an axiomatization exists. A partial positive answer is given in Gallin (1975, p. 40), but the general question seems still to be unsettled. Natural language certainly contains devices (grammatical constructions and lexical items), however, whose semantic analysis precludes any complete axiomatization (Barwise, personal communication). For example, the semantic analysis of a sentence like (2-43)

(2-43) There is no way for all boys to take different girls to the party.

will necessarily involve quantifying over arbitrary functions from individuals to individuals (representing ways of pairing a boy with a girl whom he takes to the party).

Thus from this point on, we will concentrate exclusively on model-theoretic definitions of semantic entailment, validity and related notions, rather than deductive systems. This is not to say that the study of deductive systems has *no* interest for semantics and pragmatics of natural language. It might, for example, have particular applications in the psycholinguistic study of how people draw inferences from sets of sentences, or in artificial intelligence studies. Rather, this means that we can safely ignore formal deductive systems in what follows, since our model-theoretic method renders them superfluous for our purposes.

EXERCISES

1. Suppose the phrase-structure grammar in (2-9) were expanded to allow an adverb to modify the *VP* by adding the rules

$$VP \rightarrow VP\ Adv$$

and

$$Adv \rightarrow \textit{restlessly}, \textit{harmlessly}$$

Decide on an appropriate type of semantic value to assign to items belonging to the lexical category "*Adv*" and add the required semantic rule or rules. Though the same set of sentences would be generated if $VP \rightarrow VP\ Adv$ were replaced by $S \rightarrow S\ Adv$, this syntactic analysis would have untenable semantic consequences. What are they? (Hint: think about the semantic rule you would associate with $S \rightarrow S\ Adv$, and the difference between the type of semantic value associated with sentences versus verb phrases.)

2. The sentence *John sleeps restlessly* intuitively implies *John sleeps*. Amplify the semantic analysis you gave in Exercise 1 so as to guarantee formally that this implication holds. (Hint: Place a set-theoretic restriction on a function used in Exercise 1.) Note that the treatment we have given to "logical vocabulary" (e.g., *and*, *or*) is just the limiting case of what you must do for *restlessly*, *harmlessly*, etc. The restrictions on the logical vocabulary are so strong as to uniquely determine what semantic values may be assigned. The model-theoretic method of studying relationships between the semantic values of expressions makes it possible to capture necessary relationships of lexical meaning, like

those between *bachelor* and *man* or *unmarried* noted on p. 49, by mandating that certain relationships hold between the semantic values assigned to the items in question.

3. Write a set of formation rules for a language like L_0 but written in "Polish" notation. In this notation the connectives precede the formulas they connect, and parentheses are unnecessary. The letters N, K, A, C, E are generally used as symbols for the Polish connectives corresponding to "$\neg$," "$\wedge$," "$\vee$," "$\rightarrow$," and "$\leftrightarrow$" respectively. Thus, $[p \wedge [q \vee r]]$ in standard notation becomes $KpAqr$ in Polish notation, and $ECpqAqNr$ corresponds to $[[p \rightarrow q] \leftrightarrow [q \vee \neg r]]$ in standard notation. What changes must be made in the semantic rules for L_0 to accommodate the "Polish" syntax? What changes are necessary if the connectives are not treated syncategorematically but are assigned semantic values directly?

4. Give a syntax and a semantics for a language which is like L_0 except that it contains only one logical connective, "|," defined by the following truth table:

p	q	$[p \mid q]$
1	1	0
1	0	1
0	1	1
0	0	1

5. Give a syntax and a semantics for the propositional calculus (the language like L_0 except that propositions are not analyzed into predicates and names).

6. Suppose the following rules were added to the grammar in (2-9):

$$VP \rightarrow V_s\ S$$

$$V_s \rightarrow \textit{believes-that}, \textit{hopes-that}$$

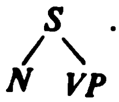
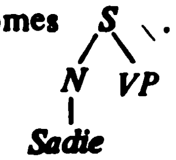
What type of semantic value would be appropriate for verbs belonging to the lexical category V_s? What difficulty arises in attempting to formulate the semantic rule for $V_s + S$ constructions?

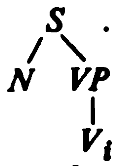
7. Consider a syntactic system in which each formation rule is of the form (2-41) where $\nu = \alpha\beta \ldots \mu$ (the concatenation of $\alpha, \beta, \ldots, \mu$). Prove that there is an equivalent context-free phrase structure grammar, i.e., one with the same categories and generating the same expressions of each category.

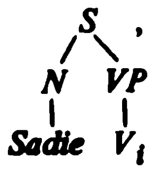
NOTE

[1] The notation is to be understood as follows. An expression of the form $A \rightarrow \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \cdot \\ \cdot \\ \cdot \\ \omega_n \end{Bmatrix}$ or $A \rightarrow \omega_1, \omega_2, \ldots, \omega_n$ abbreviates the n rules: $A \rightarrow \omega_1$, $A \rightarrow \omega_2, \ldots, A \rightarrow \omega_n$. The grammar may be used to derive sentences by first writing the symbol S (or, to derive phrases of any category A, by writing A) and then carrying out a series of steps using the rules to rewrite strings until no further rewriting is permitted. If at a given stage the last string produced is $\alpha_1 \alpha_2 \ldots \alpha_m$, and for some i there is a rule $\alpha_i \rightarrow \omega$, then it is permitted

to rewrite the string further and produce $\alpha_1 \ldots \alpha_{i-1} \omega \alpha_{i+1} \ldots \alpha_m$. For example, *S* may be rewritten as *S Conj S* or as *Neg S* or as *N VP*. If one rewrites it as *N VP*, the derivation may continue either by rewriting the latter string as *Sadie VP* or as *N* V_i (or as any of three other possibilities). It is irrelevant which of the symbols *N* or *VP* one replaces first; since the grammar is context-free any string of words that can be derived one way can also be derived the other way.

One can construct a structural description as the derivation proceeds which omits the irrelevant information about order of rewriting but records the very relevant facts about what string replaced each nonterminal symbol. The information can be represented as either a tree structure or a labelled bracketing. To construct, say, a tree, while carrying out a derivation, begin by simply writing the same symbol that initiated the derivation. As each step of the derivation replaces a nonterminal symbol by a string, augment the tree by adding the replacing string of symbols beneath the nonterminal symbol corresponding to the one replaced and connecting these new symbols to that one. If one starts from *S*, for instance, the tree begins simply as *S* too. If *S* is replaced by *N VP* the tree grows to  . If the string is then rewritten as *Sadie VP*, the tree becomes  .

On the other hand, of one had rewritten *N VP* as *N* V_i, the second three above would have grown instead to  .

If after replacing one of the symbols *N* and *VP*, one then makes the other replacement in the resulting string, he gets the string *Sadie* V_i in either case. The two different derivations of this string both correspond to the tree  , and are thus seen to be equivalent.