Lecture I:

BACKGROUND AND BEGINNING

Today's lecture has two parts. One of them is a very general introduction where I will talk about the question: "What is meaning?" and say a little bit about the history of semantics—the study of meaning—in recent and not so recent linguistics. And then we will begin, right away today, with a more specific introduction to a more technical subject, the study of semantics from a certain point of view, that of model-theoretic semantics. This is the general topic of the eight lectures.

Although what we understand about the semantics of natural language is surely very much less than what we do not understand, I believe some progress has been made in the last 15 years or so. I also believe that we are currently in the midst of a number of quite exciting developments, and I would like to concentrate here on some of these developments. I intend these lectures to be largely self-contained. I do this for two reasons. The first is that I would like the lectures to be accessible to everyone that is here, and I'm sure that you have a wide variety of backgrounds. More important, I believe strongly that it should be possible for a specialist or technician to explain what he or she is doing to anyone who is interested and who is willing to go along and do a little work, and that trying to do this is important for me also, because it forces me to think hard about why I am doing what I

© 1989 State University of New York, Albany

do and whether it is important. Specialists can get caught up in the details of their work and forget why it is that they do what they do.

I apologize that I am so ignorant of the linguistic tradition of your country. There are not very many places in the world where an independent linguistic tradition has developed; China is one of them. I spent the first years of my life in Japan, and enough of my early experience has remained with me to help me recognize that China has been to the East what Greece and Rome have been to the West. And wherever we have records of the earliest intellectual wonderings of humans, we find records of people wondering about language. Some of the questions people have asked are these:

Why are there so many different languages?
How different and how much alike are different languages?
What is the relation between language and the world?
How can words be so obscure and yet admit of wrong and right?

The last question is from Zhūang Zǐ in a passage from *The Inner Chapters*. The passage raises some very central questions that we will consider here. It also presents very clearly the central assumption of semantics. I quote: "Words are not just blown air, they have a meaning." The main job of semantic theories is to explain how words and other linguistic expressions, such as sentences and phrases, can have meanings and to say what these meanings are. I think that this point, that linguistic expressions have meanings, is very obvious, and that the ordinary person would consider it so obvious that he or she would suppose that linguists, those people whose job it is to understand language, would naturally take semantics, the study of meaning, to be one of their central concerns. Yet, this has not always been the case in the history of linguistics. I will say a little bit about this history today.

Now what could a meaning be? Again, I think the ordinary person would say that a meaning must be something that is not language, except in the case of words about words. Words refer to things. Sentences are about happenings in the world. We use words and sentences to talk about the world, about our own feelings and concerns and needs. Once again this point seems obvious, but linguists and philosophers who have concerned themselves with language have not always agreed. We will take up this question at several points in these lectures. The point of view that I will follow there is one that makes two assumptions:

I. Language has meaning.
II. Meanings are things that are not language.

What are meanings? Meanings are the things that language is about. This is apparently what makes words different from the twitterings of birds, which

© 1989 State University of New York, Albany

are not *about* anything, as far as we can tell. Semantics is the study of the meanings that expression of language can have.

So far, I've used the word *meaning* for something that linguistic expressions have and semantics is about. But this word itself—the word *meaning*—has many meanings as does the corresponding verb *mean*. Some of these different meanings can be illustrated in a few examples.

1. Giving you these flowers means that I love you.
2. Those mountains ahead mean trouble.
3. He said that he would join us, but he didn't mean it.
4. When I say X, I mean Y.
5. *Àiren* means *spouse*.

In working out a scientific theory, we have to be careful about the terms we use. We often have to adopt special terminology that departs from ordinary language and gets its meaning from the way in which it is used in our theories. Let us agree to continue to use *meaning* as a term to cover lots of different kinds of relationships, but we will adopt some special terminology for the more special and technical aspects of *meaning*.

In order to focus on some of these more specialized ideas that we need to use, let us consider the examples just given and what they seem to mean. In the sentence, "Giving you these flowers means that I love you," I do a certain act, I give you some flowers; by this act, I want to convey to you something about my feelings for you, that I love you. So *mean,* the English word *mean* in this sentence, seems to designate a relation between an act that I do and some meaning, intention, feeling, or attitude that the action is supposed to convey. This is not the sense of *meaning* that I want to focus on here.

Or consider the example: "Those mountains ahead mean trouble." This seems to mean something like this: "Continuing our journey (or whatever) is going to be hard for us because we will have to cross those mountains." So here, *mean* seems to designate a relation between something—the mountains—and some consequence for us with respect to some purpose or goal. This, again, is not the sense of meaning that I want to focus on here.

Consider the next examples: "He said that he would join us, but he didn't mean it." This seems to concern questions of sincerity. Someone says something, but does he really mean it? That is, is he really sincere in what he says? Here, *mean* seems to designate a relation between a person and something that that person says. Again, this is not the sense of *meaning* that I want to focus on here, although there will be some points where we will take up the kind of question raised by this example having to do with language and what people do with language.

The next example—"When I say X, I mean Y."—seems to mean something like this: I say something, there is a usual meaning associated with

© 1989 State University of New York, Albany

what I say, but I am telling you that what I really intend to convey to you is something different. So here *mean* has to do with the relation between a person, something that he says, and something else that he, the person, means by that. This example requires references to the kind of *meaning* I want to concentrate on here, but it still does not directly express that sense.

Let us now look at the last example: "*Aìren* means *spouse*." This example is closely related to the sense of *meaning* I want to start with here. What are we talking about? You will notice that I have underlined the Chinese word *aìren* and the English word *spouse*. A more precise way of stating what this example is intended to mean is: The word *aìren* means the same thing as what is meant by *spouse*. What I want to focus on here is, what is it that we mean by "the same thing?" *Aìren* means this thing and *spouse* means this thing. What is the thing that these two words mean or designate? Let us call this thing, whatever it is, the *denotation* of the linguistic expression in question. So, another way of saying the same thing is to say, the denotation of this word, *aìren,* is the same as the denotation of *spouse*. And again we are talking about something that is not a feeling but something in the world, whatever it is that is referred to by these two words. Now this is still not quite right. The trouble is that without reference to a language, we do not really know what these quoted expressions are supposed to be. There might be a different language in which something like this word *aìren* meant a certain kind of flower and another language in which *spouse* meant newspaper or something like that. So, to be very definite about it, we need to say something like this: The denotation of *aìren* in a certain language (Chinese) is the same as the denotation of *spouse* in English. Therefore, when we talk about denotations of expressions, we presuppose that we know that there is a certain language that the expression is being used in.

Now, I have just given a very quick look at what I take semantics to be, or at least some important part of semantics. Let us reflect on what will be required for such a view of semantics to work. Evidently, this program has two parts. First, we need to show how to assign denotations to all the basic or lexical elements in the language, Chinese, English, or whatever. And then, we need to show how to put together the denotations of the simple expressions, words like *spouse* or *airen,* and to show how the denotations of complex expressions can be made from the denotations of the simple ones. And so on, and so on, and so on.

Here is a simple example of what we need to do. Suppose we say that *John* denotes a certain individual—me, for instance, or any particular person whose name is John. And suppose that we say that *walk* denotes a certain set or collection of individuals—the set of walkers. To be able to say what "John walks" means, we want to say something like this: it is a true statement to say that the individual denoted by *John* is a member of the set of individuals that walk. That is a very simple example of how we might want to treat the meaning of complex expressions by putting together the meanings or

© 1989 State University of New York, Albany

denotations of the expressions that go into them, as we make more and more complex expressions in our syntax.

So, we apparently need two kinds of things, thus far, to talk about the denotations of expressions in a language. We need to be able to talk about individuals, to have a set of individuals that are denoted by words like names. John is one such individual. And we need to be able to talk about sets or collections of the individuals. Consequently, in giving this example of what a denotation might be as part of an answer to the question, "What is meaning?" I have already made a certain choice. The choice is that denotations are something like things in the world, not language but things in the world—people, tables, cups, books, and so on. That is the main kind of theory I will be talking about here. But it is important to know that other kinds of answers could be given and have been given. One such answer is that meanings are mental objects of some sort, things in my head, concepts or thoughts. So, the answer I have adopted is a controversial answer. Not everyone will agree with that choice.

I would like to put this question aside for now and for the next few lectures simply assume that an interesting way to talk about meanings is to talk in this way about the denotations as being things, sets of things, and so on. Later in the lectures I will return and look at some of the other sorts of answers that might be given to the question, "What is a meaning?" and compare different theories with different answers to the question.

Before we begin to look more closely at a semantic theory, I would like to spend a few minutes on the recent history of linguistic theory in the United States and Europe. The year 1957 was an important one in linguistics. That was the year in which Noam Chomsky published his small book *Syntactic Structures*. This book had a profound impact on linguistic theory, not only in the United States but in many other parts of the world. To my mind, the most important idea and the one that was hardest to grasp for linguists who had worked in earlier traditions was the idea of a generative grammar.

What is a *generative grammar?* It is supposed to be an explicit statement of what the classes of linguistic expressions in a language are and what kind of structures they have. This notion is important for us here because the kind of semantics that I wish to concentrate on presupposes the existence of an explicit grammar of this sort. Therefore, I want to spend a little time reflecting on what such a grammar is and what it does. The most important thing to keep in mind, and the one thing that is hardest to become accustomed to at first, is the idea that the grammar actually says explicitly what is in the language. That is the basic idea of a generative grammar. An example will make this idea more concrete. Given a generative grammar of a language, it should be possible for you to construct various kinds of expressions in the language by completely mechanical means, without knowing anything about the language ahead of time. So, I want to present a

© 1989 State University of New York, Albany

small grammar for a small artificial language that will be important for us as we progress. And I would like to develop it to illustrate what we are thinking about.

I will call this language *PC* (an abbreviation for predicate calculus). PC has several kinds of expressions. It has what I call *terms,* and they are of two sorts:

1. *variables,* and these will look like this:
   x, y, z, . . . (late letters of the English alphabet);
2. *individual constants,* and they are chosen from these letters
   a, b, c, . . . (early letters of the English alphabet); and
3. it has further expressions, two sorts of *predicates,* that we will call *one-place predicates,* and these will be expressions like *Run, Walk, Happy, Calm,* . . ..

(You may think that this is English, but it is not English. For our purposes right now, think of them as simply meaningless symbols that belong to these different classes of expressions.) And then we will have *two-place predicates*. And these will be words again that look like English, but are not:

Love, Kiss, Like, See, ...

These are the only kinds of basic expressions or, if you like, *lexical expressions* in the language. I included ellipses here because we may imagine many more of them exist, but we only need a few examples of each kind. The other expressions in our language have to be constructed by rules I now give you to show you how to make some more complex expressions. So, we are going to make one further class of expressions, which I will call *formulas*. Here are the rules for making formulas:

R1. If *P* is a one-place predicate and *T* is a term, then *P(T)* is a formula.

This means if I pick something that is a term, an *x* or *y* or *a* or *b* and pick something that is a one-place predicate, I can construct a formula by putting together the first thing—say *Run*—and writing the second thing—a term like a—in parentheses after it: so *Run(a)* would be a formula according to this first rule. You can probably guess what the second rule is:

R2. If *R* is a two-place predicate and *X* and *T* are terms, then *R(X,T)* is a formula.

So, we can now write things like this: *See(a,b)*.

© 1989 State University of New York, Albany

This is only the syntax of the language. There is no semantics for the language at this point. But given this grammar, we can already say about certain expressions that they belong to certain classes in the grammar because we have said explicitly what these classes are. So we know for example that the expression *x* is a term and that *b* is a term. And we know that these are formulas in the language:

Run(x), Like(c,y), Calm(c), and *Love(x,y).*

And, while we do not know anything about what they mean, we have, in effect, a very simple generative grammar for this language PC. All we need to generate examples from this language is to be able to recognize the symbols or signs and to check whether they belong to the appropriate categories or classes that are listed in the grammar.

The grammar that I have just given—for a very simple language—follows the form that logicians like Carnap and Tarski use to define the syntax of the formal languages of symbolic logic and other artificial systems. It is an example of what some people call a "formal system." One way of characterizing what Chomsky did is to say that Chomsky put forward a certain thesis or hypothesis about natural languages, namely, that a natural language, a language like Chinese or English, can be described as a formal system. And I call that "Chomsky's Thesis" from 1957. Chomsky's way of constructing a grammar was rather different from the way in which I presented this language. But a simple system like the one I have illustrated is enough for us to begin. In a later lecture, I return to the question of what kind of grammars seem most appropriate for natural languages like English or Chinese.

The PC language is not very interesting at this point because we can only make very simple sentences. We cannot make anything very complicated, and there are many things we cannot express in this language, so I want to add two further rules about formulas. The first new rule says:

R3. If *F* is a formula then so is this: −F.

the fourth rule says:

R4. If *F* and *G* are formulas so are these two things:
(F & G) and *(F* v *G).*

Theses rules tell us that we can make more complicated formulas out of simple ones. We can take one formula and another formula and put them together with a sign in between and parentheses around and we will have another new formula. We can take that formula and put the sign "−" in

© 1989 State University of New York, Albany

front of it and we will have another formula. So, now we can make sentences that are as long as we want from this grammar.

Grammars for natural languages also must have this capacity because there is no longest sentence in any language. If you give me a very long sentence in English I can always add something to make it more complex and make a longer sentence out of it. With these rules, then, we have not only the possibility of making very short sentences, now we can make sentences as long as we want by a few added rules.

We have been thinking about a very simple language, the language PC. And, thus far we have confined ourselves to the syntax of this language. *Syntax* is the study of language from a purely formal point of view with no attention to meaning. If we were to talk about a natural language, we could go on and say a great deal about the language from this purely formal point of view. Some linguists in our century seem to imply that this is all there is to say, that the only important thing about language is the network of formal relationships and contrasts that exist in the language. In the United States, some of the most influential linguists before Chomsky seemed to have this idea. One was Leonard Bloomfield, another was Zellig Harris, who was Chomsky's teacher. Of course, they recognized that words have meanings, but they seemed to think that the study of meaning could not be done in a precise and scientific way. In this respect, they agreed with many philosophers and logicians who said that natural languages are so vague and ambiguous that they cannot be described in the same way that artificial languages, such as PC, can be described.

One philosopher who did not agree with this view was Richard Montague. In Montague's papers on natural language, which were written in the late 1960s and early 1970s, Montague claimed that natural languages could be treated in just the same way as the formal artificial languages of the logician. We may state this as a second thesis. I said something about Chomsky's thesis before. This is what I like to call "Montague's Thesis": Natural languages can be described as interpreted formal systems. Remember, I said Chomsky's thesis was that natural languages can be described as formal systems. Montague added to this the idea that natural languages can be described as interpreted formal systems. Montague took over from the logical tradition, the philosophical tradition, the methods of so-called model-theoretic semantics. This is the view I mentioned. Semantics assigns to sentences and other expression interpretations that are something other than language, in particular, it assigns to sentences the interpretations that have to do with whether they are true or false. In general, to determine whether a sentence is true or false, two things are necessary: (1) you must know what the sentence means and (2) you must face the sentence with some situation in the real world and see whether it corresponds to the meaning of the sentence.

(By the way, we can now see that the term *formal semantics,* which is

© 1989 State University of New York, Albany

used in the general title of these lectures, is quite misleading. The most essential thing about model-theoretic semantics is that it is not just about relations among expressions in some language or languages, but about relations between language and nonlanguage. The way in which the word *formal* is used in the term *formal semantics,* it means instead something similar to *explicit* or *precise*. Another footnote on that title: *informal* is intended to mean something like this: *without undue use of strange formalism*. A point I hope you will come to appreciate is that you can be quite precise in ordinary language. A formalism should pay for itself in increased perspicuity and understanding; it is not an end in itself.)

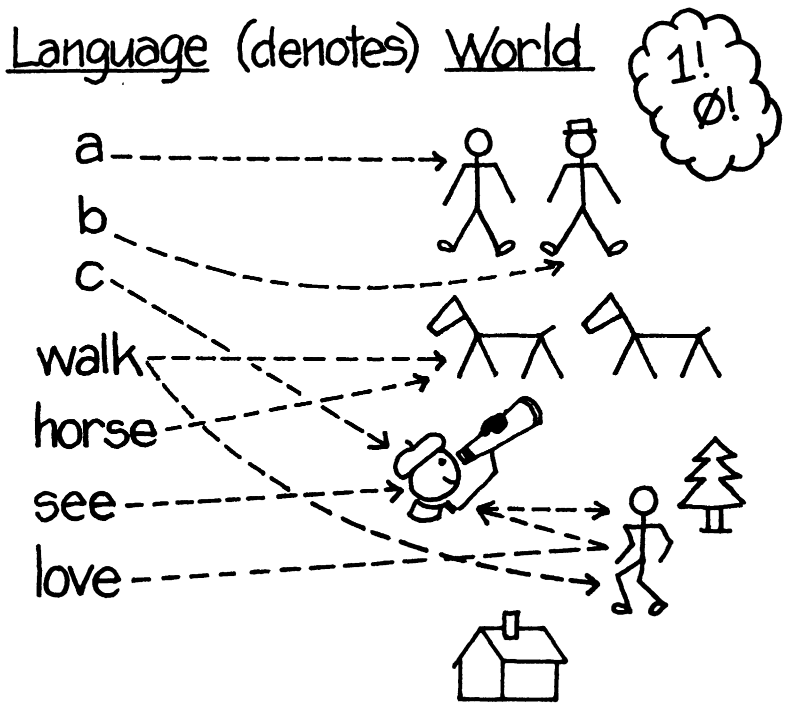
Now I want to take our simple language, PC, and show how we might go about giving an interpretation to this language by telling what the different kinds of expression in the language denotes. What we had before was the *syntax*. Now I am going to say something about the *semantics* of this language, something about the denotations of the different kinds of expression. And what I say is this: the terms denote individuals, the one-place predicates denote sets of individuals, the two-place predicates denote sets of pairs of individuals, and the formulas denote what I call truth-values. We will write *1* for True and *0* for False.

(You may think that the idea of letting the denotation of a predicate be a set is not very intuitive. Maybe *Walk* should rather mean something like the property of walking. If you think this, you are not alone. In later lectures I return to that idea. What we are looking at here is a standard theory about the denotations of a language developed by logicians and mathematicians who like to use *set theory* as a basic tool. It is a nice and well-understood theory and that is part of its appeal. But is has some drawbacks, as we will see. Similar remarks could be made about the idea of truth-values as the denotations of formulas.)

It is very important to understand what I am saying. What I am talking about here are these things in the world or in a model for the language which the different kinds of expressions in the language are supposed to refer to. So, think of a particular individual constant as having a person or a tea cup or a table as its semantic value or denotation. Here is a language, and here I am talking about a world of individuals and sets of individuals and pairs of individuals:

Now, before going on to say exactly how this all works, we need to think a little bit about the two kinds of terms: *individual constants* and *variables*. I said that they both denote individuals but they do it in a quite different way, and I have to spend a little time explaining that. The easiest thing is to think of individual constants as being like proper names in a language. So, these individual constants in this language, PC, work like *Emmon* or *Tom* or *John* or *Harry,* but we imagine that we can give names to many, many different kinds of things. So they always stand for some particular individual or thing. (For the time being, assume that constants—

© 1989 State University of New York, Albany



unlike real names in natural languages—always pick out a unique thing to which they refer. Consequently, you could think of each name as including a sort of *Universal Identification Number*.)

What about variables? Here we have to say that what variables denote depends on something that is called an *assignment of values to variables*. A variable is like a pronoun. So, variables work in this language very much like words, like *he, she,* or *it,* in a natural language like English. If we have a natural language sentence such as, "she is wise," how can we tell whether it is true or false? Well, we cannot determine its truth value unless we know who we are intending to mean by *she*. Therefore, part of the interpretation is an assignment of values to variables. And these assignments will always provide some particular individual for whatever variable we use, so that constants work like names and variables very much like variables in mathematics. A formula such as *Run(x)* cannot be judged true or false unless we know what individual is referred to by the variable. Therefore, we must have, in addition to what we have talked about already, an assignment of values or meanings to variables. Individual constants denote individuals very much like proper names do. Variables denote individuals under an assignment of values to the variables.

© 1989 State University of New York, Albany

Now, you will notice that this language is extremely limited thus far because we do not yet have any way of making general statements. We can only say things such as :"John runs," or "Mary runs," or "It is happy.†' We have no way of saying things such as: "Someone is happy," or "Everyone is sad," and so on. Furthermore, stating that these words denote particular individuals, as our constant terms do, would not make sense. Who is *everyone* or *someone?* There are no general expressions in the language, and we need to add two more things to be able to say things of this more general nature. Again, I first give the syntax of these new expressions and then describe their meanings. So, to the definitions of formulas, I add a fifth rule:

R5. If x is a variable and F is a formula, then
$\forall x\ F$ is a formula; and
$\exists x\ F$ is a formula.

These are going to correspond to general statements that say:

Every x is an F; and

Some x is an F.

In the first formula, if we think about where the variable x appears in the formula, we make a general statement about everything, every individual in the interpretation. For the second, we make statements that correspond to English sentences such as "Someone runs." So, the first corresponds to a universal statement and the second to an existential sentence in logic. Now I have told you what kinds of denotations all of our expressions in this language have. Terms denote individuals. If they are constants, they are similar to names and denote particular individuals. If they are variables, like pronouns they denote individuals relative to a certain assignment of values to those variables. Predicates denote sets if they are one-place predicates. They denote sets of pairs of individuals, such as Mary and John, if they are two-place predicates. You must think of these pairs as ordered: Mary and (then) John is a different ordered pair from John and Mary. And all formulas denote truth-values: *1* for true and *0* for false.

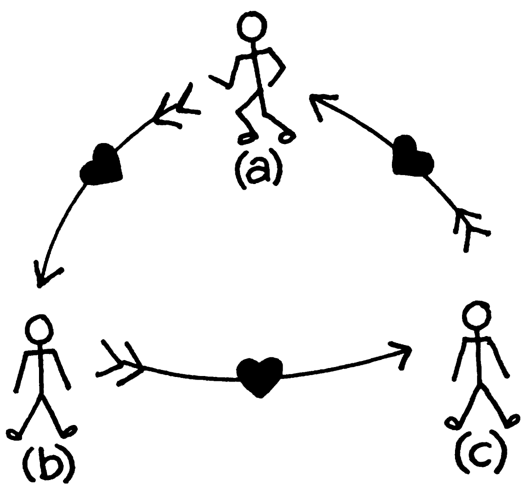
To give the semantics for this language, I have to say something more about what particular denotations we assign to formulas on the basis of the denotations of their simpler parts. Consequently, what I have to do is give you a definition of what it is to be a true formula in this system. I think it would be best, in presenting this truth definition, not to give a precise definition I would have to write on the board and would look very complicated, but rather to look at some examples and see how we would go

© 1989 State University of New York, Albany

about defining the denotations of particular expressions in this language. Let me take some simple examples of expressions such as:

Run(a).

We want to determine when a formula like *Run(a)* is true and when it is false. On the one hand, we have the language, PC, and on the other hand, we have the world or model, and we want to ask about the formula: When are we going to say that this formula is true? Well, the semantic value, the denotation of the formula will equal *1* (True) just in case the individual denoted by *a* is in the set denoted by *Run*. So, when would the formula *Run(a)* be true? Well, we have to look at the world:



We find in this world the individual that *a* is supposed to denote. We find in the world a lot of other individuals who are running. There are many other things in the world—trees and pots and so on—but if the individual denoted by *a* is a member of the set of runners, then we say the formula is true. How would we define the truth conditions for *Love(a,b)? Well, look at the world again. We have to find some individual denoted by b* and let us say again we find the same individual for *a*. We now have to look at pairs of things in the model not just single things. Let us assume we have three things in our interpretation now: *A, B,* and *C*. and somehow we find the set of pairs that are in the denotation of *Love*. How we do that is not part of semantics, but somehow we know that A loves B. We will make a very sad story: A loves B. B loves C. And, C loves A. So this is the way the world is—we

© 1989 State University of New York, Albany

have a set of pairs—A + B; B + C; C + A—and then we ask about this formula: is it true? Well, it is true according to this world because in this world we said A does love B, so the formula is true. What about the formula *Love(b,a)?* According to what I have said here, this formula is false. Because B unfortunately does not love A. B loves C. And if we write that out, we would say: this formula is true just in case—exactly under the circumstance, or if and only if—the pair of individuals A and B are in the set of pairs that are the denotation of *Love* in this model. So this gives us a beginning point towards defining True and False with respect to a certain world or model. Notice one thing about what I have just said: I assume that the specification of the interpretation gives us complete information about all the individuals, sets, and so on, that we need to give denotations to the formulas of the language being interpreted. It is only on this understanding that we can conclude, for example, that B does not love A. In later lectures, we will take up the problem of dealing with a less idealized setup, where we might have only incomplete information.

(In this example, I have slipped over a general convention that I will try to follow: I cannot hand you real things in a model, as I sometimes draw pictures. But pictures are not very convenient either, so I've used uppercased letters, such as A, B, C, to correspond to the things denoted by the constants in our language, *A* for the thing that *a* denotes, and so on.)

Now, what about the other ways of forming formulas? You may remember one of the rules stated that if we have a formula, we can make another formula by putting the sign "-" in front of it:

-Love(b,a)

This sign is going to correspond to *not* or *negation*. When do we want this formula to be true? We want this formula to be true just in case the formula without the negation is false. So, we can say:

the denotation of *-Love(b,a)* = *1* (is true)

iff

the denotation of *Love(b,a)* = *0* (is false)

(*iff* is often used as an abbreviation for "if and only if".)

What about the formula:

(Run(a) & Love(a,b))?

(The ampersand—&—corresponds to the word *and*.) And we want to say

© 1989 State University of New York, Albany

that this whole formula is true if and only if the first part is true *and* the second part is true. So that the formula

(Run(a) & Love(a,b))

is true just in case the two individual parts are true. Is it true or is it not? It is true. I said before: *a* is one of the runners, *(a,b)* stands for one of the pairs that are in the denotation of *Love,* so because the first formula is true and the second formula is true, the whole thing is true. And this is what the denotations of such formulas are supposed to be.

Finally, for these simple examples: What about this formula?

((Run(a) ∨ *Love(a,b))*

Well, this formula is going to be true just in case either one of the component formulas is true. Thus, the sign & is like *and* and the sign ∨ is like *or*. And this formula will be true just in case (if and only if) either the first part is true or the second part is true or both of them are true. So again, given this model—Is the formula true?—does it denote *1* in this model? Yes, because both of the parts of it are true. It would also be true if one of them was false. Just as long as at least one of them is true, then the whole thing is true. It would be false only if both of the sides of the disjunction were false.

The only thing left to complete the semantics for language PC is to describe the denotations of formulas such as:

∀*x Run(x)* and ∃*x Love(x,a).*

Here we need to think a little bit about assignments of values of variables. And we want to say what the truth of the whole expression is on the basis of just the part without the quantifier in each case, and we need to say it in terms of assignments of values to variables. The whole formula (1) is going to be true if and only if (2)—what we get when we take away ∀*x*—is true on every assignment of values to variables, that is, no matter what we take *x* to denote:

the denotation of (1) ∀*x Run(x)* = *1*

iff

the denotation of (2) *Run(x)* = *1* on *every* assignment of values to variables.

For this example, this statement is correct. Some complications arise from

© 1989 State University of New York, Albany

the fact that we do not know whether there are other variables in the formula or not, if we do not know about the internal structure of the inner formula, but, for this particular formula which has only the one relevant variable x in it, this statement works (I return to this point in the next lecture). You might guess now what is going to happen with the other formula, the existential sentence. Again, we need to say what the truth condition is for the whole formula is on the basis of the truth of the inner formula, what we get when we strip off $\exists x$. You need to say what the whole thing will denote on the basis of just what the second part will denote and here we will say this:

the denotation of $\exists x$ *Run(x)* = *1*

iff

the denotation of *Run(x)* = *1* on *some* assignment of values to variables.

Thus, in the first case no matter how we assign values, we have to get a true formula. Say we have the variable x. One assignment would say that x denotes this person. Another assignment would say x denotes that person. Another assignment would say x denotes that thing. For the universal quantifier: If for *every* assignment of values to x the formula is true, then the whole thing is true. For the existential quantifier: If on *some* assignment—at least one assignment—of values to variables the formula is true, then the whole thing is true.

So that, in a very rapid form, is the theory of quantification. If this is the first time you have heard such an explanation and you understood it, you should be very proud because it took logicians a very long time to develop this theory of quantification in all its complexities. If this is the first time you have heard it and you feel that you don't quite understand it—if you feel as though you need to think about it a little more and play with it to understand it—then you're perfectly justified. It is a complex thing, and I am presenting it in far too rapid a manner. I do not want to say too much about formal technical details. I want to give you the basic idea so that we can then talk about the general issues that these lectures are aimed at.

What I have just been going through is really a restricted version of the so-called predicate calculus, which is a formal logical system, and I have given you an interpretation. The way that we have done this is an example of the general approach of a model-theoretic interpretation. I have been talking about a world or model and a language and a relationship between that language and the world in terms of denotations or meanings of expressions of this predicate calculus or PC.

The set of objects—or whatever it is we have in the model—I call a *model structure*. We have seen an example of a model structure for a

© 1989 State University of New York, Albany

particular language. One of the recurring questions and perhaps the main question of these lectures is: What kinds of model structures are most appropriate and revealing for studying the semantics of natural languages? not languages like PC, the predicate calculus, but English, Chinese, Russian, Thai, or whatever. What sorts of model structures do we want to set up if we want to try to pursue the semantics of natural languages in a model-theoretic manner? We will be studying various kinds of model structures for natural and artificial languages.

I would like to take a few minutes before we stop to say something about what we find in natural language that PC is not really able to cope with. The predicate calculus PC is too simple a system, and the model structure for it is too simple to be adequate for natural languages. Let me give you just a few examples of the way PC differs from a language like Chinese, or a language like English, by showing you what kinds of things we do not have.

In the parts of speech of PC, we have only three kinds of expressions—well, four maybe. We have terms, individual constants and variables, predicates of two kinds, and then we have things like *and* and *or* and parentheses and the universal quantifier and the existential quantifier, but we have only a very small number of kinds of expressions or parts of speech in PC. For natural language this won't do. We need more different kinds of expressions. For example, take a sentence such as, "John runs slowly." What is *slowly?* Nothing in PC corresponds to the adverb *slowly.* You have only predicates and individual terms. We can say something like, "John runs," but we can't say anything like, "John runs slowly." Or take an English sentence like, "Mary ran," as opposed to, "Mary runs." In English, of course, we must make an explicit choice of tense, and "Mary ran," does not mean the same thing as, "Mary runs." PC has nothing that corresponds to tense in natural language.

Or take a sentence such as, "Mary can run." We can only deal in PC with, "Mary runs," or "Mary does not run." What does it mean to say something like "Mary can run," or "It is possible that Mary will run."? Nothing corresponds to auxiliaries or moods, like subjunctives, words such as *can, should, may, must,* and so on. And PC does not have expressions of this sort. Natural languages do. We need to know something about how to have that sort of system in a model theory for natural language. We have no way of expressing conditionals. Sentences such as, "When it rains, it pours," or "If it is a nice day tomorrow, we will go to the beach," and so on. We have no way of connecting sentences with *if* or *when* yet. That is an easy thing to add. I will show you how to do that next time.

Furthermore, we do not have anything in this language that corresponds to what we need to interpret a sentence such as: "I live here." We can say, maybe, "John lives in Tianjin." We could say *live in* is a predicate and then *John* and then *T* or something for *Tianjin,* but we cannot say anything in this language like, "I live here." Now what does, "I live here," mean? *I* means

© 1989 State University of New York, Albany

me, but only if I am the one who is speaking. If you say, "I live here," then it means someone else. This language has no way of dealing with expressions that are determined by context. And, likewise, *here;* what does *here* mean? If I'm here, in this room, *here* might refer to this room. But if someone else says this sentence in another city, in another room then *here* in that sentence means something different. So, we need to think about context dependent words such as *I, here, now,* and so on.

In this language, we have no way of making complex sentences such as, "We will try to please you." That is where one sentence is somehow a component part of another sentence in a way that is not expressible in terms of simple *and, or, not,* and so on. Everything is a statement: John runs. Mary lives in Tianjin. Bill loves Sally. And so on. In this language, we have no way of asking questions. "Who lives in that house?" We have no way of making requests in PC. A language like PC, therefore, is very much limited to making statements about things and not asking questions or making requests, such as, "Please give me the time," and so.

The most important new thing we will be thinking about as we go along is the fact that in our interpretation of PC we thought only in terms of a single world or model. And, as we go along we will want to think about ways in which we can have a whole class of different models and think about the truth of sentences given different models. So that we will be able to think about possible different ways in which the world might be, rather than always in terms of a single actual model. And that brings in the notion of another possible world, which is simply a way in which things might be, not necessarily the way things are. We need this addition to our theory to be able to deal with sentences such as, "Mary can walk in the park." When is it true to say that Mary can walk in the park? Well, one way to answer that question is to say, "Mary can walk in the park," is true in this world if there is some other possible world, some other possible way in which thing might be, in which Mary does walk in the park. So we can explain modalities in terms of different ways in which the world might be. That will be the main topic of my next lecture.

© 1989 State University of New York, Albany