Chapter I. Methodological Preliminaries.

1. Introduction.

This book explores some general issues related to the design of question answering systems; the logical representation of the contents of questions and answers, the logical analysis of data bases, and other issues concerning knowledge representation. It explores all these matters from one coherent perspective: the perspective of logical model theory.

Employing Tarski's (1936) idea of representing a state of the world by means of an interpretation of a logical language, precise definitions may be given of the meanings of the various data structures which play a role in question answering programs: representations of the contents of questions and answers, as well as data bases and other forms of "knowledge". This is demonstrated in some detail in the chapters of this book.

Chapter II criticizes existing proposals concerning the model-theoretic semantics of questions and answers, and puts forward an alternative.

The representation of knowledge in a way which dovetails with the logical representation of the contents of questions and answers is addressed later in the book. The goal of providing an interface with an "ordinary data base" will be focussed on particularly. The knowledge representation problem is therefore broken up into two parts. In Chapter IV data bases are construed as well defined objects within the framework of logical model theory, while in Chapter V a proposal is made for how to bridge the semantic gap between the natural language formulation of a query and the notions contained in the data base.

A question answering system which embodies the main ideas developed in this book is described in Chapter III. This system, PHILQA1, distinguishes itself favourably from other efforts in the field of computational linguistics because of its unusually refined modular structure, featuring explicitly defined interface languages between the modules and precise definitions of their tasks.

In the present introductory chapter, the model-theoretic perspective which constitutes the framework of this book is briefly contrasted with some alternative methodologies which have been brought to bear on similar problems: the Artificial Intelligence (A.I.) paradigm (whose adherents might take issue with our very goals of reliability and consistency), the idea of Procedural Semantics (which proclaims to be a superior alternative to model-theoretic semantics), and the use of Predicate Calculus (which is promoted by a research tradition which is focussed on a particular body of deductive techniques rather than on the use of well-defined meaning representations in general).
2. **Artificial Intelligence.**

The design of natural language question answering programs is often viewed as belonging to the general realm of Artificial Intelligence. The discussion in this book, however, will touch only tangentially on A.I. work on natural language and knowledge representation. This is because much of this work proposes knowledge representation data structures without specifying their semantics with any precision. Much A.I. work shows no interest in developing modular program structures with well-defined interface languages and thus sheds no light on the problem area concentrated upon here.

The difference between the A.I. approach and the one I wish to advocate is not a mere difference of opinion on how to attack a technical problem, but a much more fundamental disagreement about the nature of the enterprise one is engaged in.

A.I. research attempts to design working computer programs for tasks which seem intrinsically to involve the higher intellectual faculties. Natural language processing is often taken to be a paradigm example of this kind of task. In A.I., the goal of designing computer programs capable of performing tasks of this sort efficiently and correctly is taken to be somehow equivalent to the goal of modelling central aspects of human cognition. In my view, however, these enterprises are entirely different and there can be no virtue in confusing them. Let me expand on this somewhat.

A fundamental difference between the two activities is that they have different goals which overlap only slightly. Humans have many characteristics, for example, which one would just as soon do without in a computer program. We are often sloppy and unreliable, frequently forgetful and uncooperative. Therefore, if one could make a question answering system equalling human question answering performance, one should try to make a system which exceeds it. If the goals of question answering, automatic translation, expert systems and the like are construed as "cognitive modelling", then one is, in fact, undercutting the potential which electronic computers may have for such tasks.

Whether implemented computer programs can be expected to be structurally similar in any way to human mental processes is also open to serious doubt. The basic underlying hardware processes are so different in both cases that it is by no means self-evident that corresponding structures can be used to implement corresponding tasks.

Those working within hardcore A.I. tend to avoid facing the choice between doing A.I. as applied computer science or as theoretical psychology. By failing to deal clearly with the goals and purposes of the A.I. activity, practitioners need demonstrate neither correctness and efficiency in their
programs nor psychological validity. Comparisons with the human mind can be freely invoked to justify a lack of interest in the correctness, consistency and modularity of a program while the extent to which the program constitutes a model for human mental processing structures is left unspecified.¹ In light of this, it is best to view A.I. not as a scientific or technological discipline in the way in which these notions are normally considered but to see it rather as the experimental branch of computer science where new programming concepts are tried out without much concern for their theoretical underpinnings and unfettered by the constraints imposed by well-defined requirements of any sort.

This explains why one would not expect to deal in depth with A.I. work in natural language understanding in a discussion which is involved in evaluating different, well-described means for achieving the same (or similar) well-defined goals. This is why, in the rest of this book, references to A.I. work are made only where they are specifically relevant to the matter under discussion.


An important goal of the discussion in this book is to give definitions of notions such as the content of a question, the content of an answer, and knowledge about a subject domain in such a way that precise requirements for the correct operation of a question-answering program may be formulated. Elsewhere in the discussion, it will be shown that the notions of logical model-theory can be used very well for this purpose. In the present section, the virtues of model theory will be compared briefly with the properties of procedural semantics, a rather different formulation of semantics which developed out of Computational Linguistics and Artificial Intelligence.²

Model-theoretic semantics can be used as a tool to define the meaning of data structures independently of the programs which access them and to define the task of program modules independent of their implementation. A model-theoretic description of the content of a question specifies what answers would be correct in terms of a precisely defined notion of state of the

¹ A rather extreme position on this matter, which I would not want to ascribe to the A.I. Community as a whole, dismisses altogether the notion of a model as having an explicitly specified structural correspondence to the phenomenon it is supposed to model. Instead, it is simply stipulated that object A is a model for phenomenon B if people can be led to mistake A for B. This decision to declare mimesis to be the highest goal of the cognitive sciences is often credited to a philosophical essay by Turing (1950) which proposes a kind of "mimetic reductionism" of mentalistic notions. The work done by Colby constitutes the most explicit exemplification of this point of view. See, for instance, Colby et al. (1972) and Colby (1975).

² See Woods (1968, 1981); Woods et al. (1972); Winograd (1972).
world. How the answers will be computed is left open by this specification. Similarly, a model-theoretic description of the meaning of a piece of data shows how it constrains the states of the world which are considered possible, without specifying how this information is to be brought to bear on the queries. Thus, model-theoretic semantics may be used to introduce a considerable measure of precision into the discourse surrounding the structure and operation of a question-answering system without complicating matters with purely algorithmic considerations.

In procedural semantics, on the other hand, the meaning of a question is said to be a \textit{procedure} for computing the answer.

One problem arises because this procedure is assumed to be formulated in terms of certain primitive procedures which need not be further analyzed. But for the "primitive procedures" which yield factual knowledge concerning contingent states of affairs in the world, this view is not justified. Their values may not be known at the moment of the design of the system; it must be possible to change them if the state of affairs in the world changes. Therefore, we do not want to view these procedures as unanalyzed entities; they must be viewed as either containing or accessing data structures of some sort, and the meanings of these data structures must be described precisely.

Another problem with procedural semantics as it has been defined until very recently,\(^3\) is that it is too specific. Because the meaning of an expression encompasses its \textit{evaluation method} there is no room for logical equivalence transformations such as those which turn an expression into an equivalent but more efficiently evaluable one. The general criticism which should thus be made about this framework, from the point of view of model-theoretic semantics, is that it is conceptually too poor.

4. Limitations of Predicate Calculus.

When methods from formal logic are used in knowledge representation and natural language analysis, the formalism used is often the first-order predicate calculus. In fact, the notions of "logic" and "first-order predicate calculus" are all but interchangeable in the work of certain authors. (See, for example, the papers collected in Gallaire and Minker, 1978.)

Most syntactically and semantically well-defined languages used in computational systems for semantic representation derive more or less directly from first-order predicate calculus. (Woods, 1968; Green, 1969; Petrick, 1973). Proposals about semantic nets often pertain to pictorial representations of first-order predicate calculus expressions as well (Schubert, 1976).

\(^3\) Woods, the foremost exponent of procedural semantics, has kept redefining it, to the extent that it is now indistinguishable from model-theoretic semantics. Sentences are now analyzed as "abstract partial procedures". (Woods, 1981)
It must be emphasized, therefore, that the approach taken in this book does not embody any preconceptions about the syntactic form of the logical representation languages used. What has been of overriding concern, however, is that the expressions in the languages used be precisely defined and that their semantics (in the model-theoretic sense) be precisely defined as well. The first-order predicate calculus was rejected as a possible logical formalism for the general task of natural language modelling, because the semantic phenomena involved can not be accommodated within the limitations of this framework. For example, "collective quantification" requires functions or predicates on sets of individuals while "cumulative quantification" requires variables ranging over Cartesian products (i.e. over sets of n-tuples) and selection-operations on n-tuples. (Scha, 1981)

Comparatives result as well in quantificational structures not expressible in the minimal extensions of first order logic which are normally made. 4) "Bags" are necessary for dealing with the "exclusive-or" operator (Borowski, 1976) and with certain noun phrase denotations. 5) In addition, lambda-abstraction is necessary for using the technique of "translation specification", a most important possibility exploited here for knowledge representation. 6)

Therefore, a richer formal language than first-order predicate calculus was necessary to do the tasks we had in mind adequately: the languages introduced below as EFL (English-oriented Formal Language), DBL (Data Base Language) and WML (World Model Language) have the power needed to accomplish them. Appendix A specifies the syntax and semantics of these languages.

5. Conclusion.

To be able to design a reliably operating question-answering system with a modular structure, it is necessary to define notions of questions, answers and knowledge which are suited for a computationally effective treatment. These notions must be based on an explicit semantics, moreover, which makes it possible to formulate explicit correctness requirements for the algorithms employed in the system. In sections 2 and 3, above, the A.I. and procedural semantics approaches to question-answering system development were rejected because they are unequipped to deal with constraints of this sort. In section 4 the first-order predicate calculus was also rejected as a suitable framework because it lacks the richness and power needed to deal with linguistic complexities adequately. Let us now go on to tackle the main

4) Akzo bought more computers than Philips sold.

5) What is the sum of the prices of Akzo's computers?

6) See Chapter V.
subject areas we shall be dealing with: the semantic and computational aspects of the representation of questions, answers and knowledge. Because it is basic to the entire issue we address here, we will begin with the semantic analysis of questions and answers.