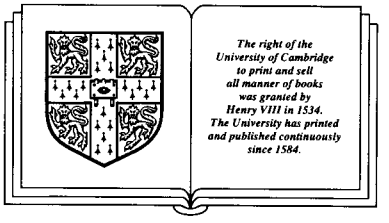
FORMAL SEMANTICS AND PRAGMATICS FOR NATURAL LANGUAGE QUERYING

JAMES CLIFFORD
New York University



CAMBRIDGE UNIVERSITY PRESS
Cambridge
New York Port Chester Melbourne Sydney

Published by the Press Syndicate of the University of Cambridge
The Pitt Building, Trumpington Street, Cambridge CB2 1RP
40 West 20th Street, New York, NY 10011, USA
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

First published 1990

Printed in Great Britain at the University Press, Cambridge

Library of Congress cataloguing in publication data available

British Library cataloguing in publication data available

ISBN 0 521 35433 1

Contents

List of Figures

CHAPTER ONE

INTRODUCTION

It is difficult to imagine a successful semantic theory that does not include time as an integral component. Yet all of today's major data models – models which purport to provide a general theory on how to represent information for convenient and rapid storage and retrieval on digital computers – completely ignore this essential aspect of semantics. In this work we examine the connection between two areas of semantics, namely the semantics of databases and the semantics of natural language, and link them together via a common view of the semantics of time. In the first part we argue that an essential ingredient for the success of efforts to incorporate more *real world* semantics into database models is a coherent theory of the semantics of time. We describe such a database theory, and then proceed to present a formally defined English database query language whose semantic theory makes explicit reference to the notion of denotation with respect to a moment of time.

The idea that time might be an important consideration in providing an enriched database semantics is not new to this work, but it is nonetheless a relatively recent concern of database research. [Bub77], [Ser80], [Klo81], [And81], [AM82], [CW83], [Sno84], [GV85], [CC87], [GY88], and [CC88a] are among the many works that have lately investigated ways in which time might be added to a database model. Two recent surveys of the literature on time and databases, [BADW82] and [Sno86], and a recent international conference, [TAI87], provide excellent references to the growing interest and literature in this field. Our own approach (first presented in [Cli82a] and [CW83] and further investigated in [CC87]) owes a debt to the philosophical tradition represented in such works as [Car47], [Pri67], and [RU71], and particularly in the works of the logician Richard Montague [Mon74]. This tradition makes a case for a far more pervasive theory of the importance of time in a theory of meaning than one which asserts that time is just one among many equally important aspects.

In Chapter 2 we present a brief introduction to the Montague Semantics (MS) framework, including a definition of the intensional logic IL_s. Then in Chapter 3 we present a formal definition of the Historical Relational Database Model (HRDM), our formal incorporation of a semantics for time within the context of the relational

database model ([Cod70]). Unlike much of the recent database work involving time, we do not attempt to incorporate any notions of how time is encoded (e.g., in the Julian calendric system.) Such theories are important, but it seems that before exploring some of the minutiae of time a theory that captures more of its essence is needed. Theories concerned with months of the year or days of the week are best discussed at the level of interfaces or data representation schemes, and not at the level of a basic semantic theory. We therefore present a very general theory, one which ascribes only the simplest and most intuitive properties to time (order and density) and defines a very simple relationship between time and the other elements of the relational database model.

Our emphasis, as discussed in Chapters 3 and 4, is on the centrality of time with respect to database modelling, and we argue that time is central to our understanding of database semantics. We present the point of view that it is not simply expressions such as "previous employee" or "salary increase" that require reference to the notion of time for their understanding, but rather that every aspect of database theory is understood better when its relationship to the phenomenon of time is considered. In particular, a new insight into the nature of the database distinction between *key* and *non-key* attributes is provided by an understanding of their relationship to time.

In order to make a concrete presentation of these ideas, we present them within the context of two well-known database models, the relational model [Cod70] and the entity-relationship model [Che76]. We adopt the relational model both because it is a well-studied and formalized database model, and because it is increasingly being used as a model for implemented systems. The entity-relationship model is used not simply because of its growing popularity as a tool for modelling database semantics, but also because its ontological theory – that the world consists of entities and relationships among them – is in close accord with the ontology of the philosophical logic tradition.

In the latter part of this work we argue that a fully formalized account can successfully be given for both the syntax and the interpretation of an English database query language. Moreover this account is independent of any performance model of how a processor (for example, a computer) might go about understanding such a language. The language theory that we present, with syntax paired with semantics, is once again a direct outgrowth of the work of Richard Montague, who argued quite seriously in a number of papers for the contention that there is "no important theoretical difference ... between formal and natural languages" ([Mon70a, p.188 in [Mon74]].) Working within the framework of Montague's syntactic and semantic theory we present a formalized fragment of English questions designed for the purposes of database querying.

Numerous systems for providing natural language access to databases have been described in the literature, including [WKN81], [Wal78], [Har78], and [HSSS78]. While these systems are dissimilar in a number of different respects, they all share what to us is the same defect, namely the lack of any fundamental formal theory of the semantics of the database or the semantics of the English query language.

We view the development of these and other such systems as belonging to the

first phase in the development of a formal theory of the semantics of database and of database querying, much as the early years in the design of computer languages such as FORTRAN were part of the first phase in the development of a theory of programming-language semantics. It awaited the impact of formal language theory, coupled with a theory of syntax-directed translation, for the area of programming language theory to be born. To complete an analogous development in the area of natural language querying would require the impact of formal language theory and a theory that coupled the syntax and the semantics of English. Many linguists today believe that Montague's theory of universal grammar [Mon70c] is the first successful attempt at formalizing such a uniform syntactic and semantic theory of natural language. We believe that some such formal theory of a query language is an important first step towards the development of provably correct and reliable natural language processing systems. For inherent in the notion of program *correctness* is the concept of a standard against which a program is to be judged.

In Chapter 5 we present an informal overview of a fragment of English for database querying which we call QE-III. We discuss the kinds of properties and abilities that a database query language in English should possess. Principal among these are an account of question semantics that possesses close analogs in database theory, an account of the semantics of multiple-WH questions, an account of the semantics of time, and a grammar that is conducive to a computer implementation. After examining a number of partial solutions to these problems, we introduce the notion of pragmatics as an additional formal component of our language's theory. We argue that assigning to the pragmatic component the task of providing a representation for the answer(s) to a question is both appropriate and elegant. Finally we discuss several other recent attempts at developing a formal theory of questions.

After this informal presentation we provide in Chapter 6 a formal definition of the query fragment QE-III as a Montague Grammar. This fragment represents a simplification of the semantic theory of Montague's fragment, presented in [Mon73] and known in the literature as PTQ, that offers a natural correspondence to the semantics of queries in a database context. An excellent introduction to the formal semantic approach to linguistic analysis which has come to be known as Montague Semantics can be found in [DWP81]. The fragment is provided with a formal syntax, semantics, and pragmatics, each component designed with the database application in mind. These components of the QE-III language are based upon the language IL_s introduced in Chapter 2. The inclusion of a formal pragmatic component is an interesting extension to the traditional conception of a Montague Grammar. Among the major extensions to the PTQ fragment embodied in QE-III are the inclusion of time-denoting expressions and temporal operators, an analysis of verb meanings into primitive meaning units derived from the database schema, and of course the inclusion of certain forms of direct questions. These extensions, and the semantics with which they are provided, are motivated by the ultimate goal of database access, but they are equally interesting in their own right. The syntactic theory presented is in some cases admittedly naive, for we have been primarily interested in getting the

interpretation right.

A more complete discussion of the details of the features of the QE-III fragment is presented in Chapter 7, where numerous example derivations and translations of typical database queries are examined and discussed. We also discuss briefly how QE-III can be adapted to different database domains, and how the logical translations of the English queries expressed in QE-III can be translated into a data manipulation language like the historical relational algebra of [CC87]. The chapter concludes with a discussion of some of the limitations of the fragment and of some possibilities for further extensions. Finally, we conclude in Chapter 8 with a review of the major ideas of the book and with a discussion of future work that this research suggests.