1

An Overview of Knowledge Representation

John Mylopoulos
Hector J. Levesque
University of Toronto

1. Introduction

This is a brief overview of terminology and issues related to Knowledge Representation (hereafter KR) research, intended primarily for researchers from the Database or Programming Language area.

Knowledge Representation is a central problem in Artificial Intelligence (AI) today. Its importance stems from the fact that the current design paradigm for "intelligent" systems stresses the need for the availability of expert knowledge in the system along with associated knowledge handling facilities. This paradigm is in sharp contrast to earlier ones which might be termed "power-oriented" [GP77] since they placed an emphasis on general purpose heuristic search techniques [NILS71].

The basic problem of KR is the development of a sufficiently precise notation with which to represent knowledge. Following [HAYE74] we shall refer to any such notation as a (knowledge) *representation scheme.* Using such a scheme one can specify a *knowledge base* consisting of *facts.* For the purposes of this paper, a knowledge base will be treated as a model of a world/enterprise/slice of reality.[1]

Hector Levesque is currently affiliated with Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, California.

This is a revised and updated version of a paper that appeared in [BZ81].

[1] For other ways of viewing a knowledge base see [BS80] (p. 68).

A number of important papers on the subject already exist. [HAYE74] deals with central issues of KR theory, and [BC75] includes a fine collection of papers on KR theory and practice. More recently, [FIND79] [WH79] [GM78] have compiled important collections of papers on semantic network, production system, and logical representation schemes respectively. [BOBR77] contains an interesting collection of short presentations on a number of state-of-the-art schemes, and [GP77] relates KR to other important problems in AI. A recent SIGART Newsletter issue, [BS80], contains questionnaire results from more than 80 research groups working on or using a representation scheme. Finally, [WM77] examines KR issues and searches for counterparts in Data Modelling research.

2. A Taxonomy of Representation Schemes

When trying to classify representation schemes we consider the world as a collection of *individuals* and as a collection of *relationships* that exist between them. The collection of all individuals and relationships at any one time in any one world constitutes a *state,* and there can be *state transformations* that cause the creation/destruction of individuals or that can change the relationship among them. Depending on whether the starting point for a representation scheme is individuals/relationships, true assertions about states, or state transformations, we have a (semantic) *network, logical,* or *procedural* scheme respectively. A number of schemes proposed recently adapt more than one viewpoint and will be considered separately.

2.1 Logical Representation Schemes

Logical Representation Schemes employ the notions of constant, variable, function, predicate, logical connective, and quantifier in order to represent facts as logical formulas in some logic (first or higher order/multi-valued/modal/fuzzy, *etc.*). A knowledge base, according to this view, is a collection of logical formulas that provide a partial description of a state. Modifications to the knowledge base occur when the introduction or deletion of logical formulas occurs. In this sense, logical formulas serve as atomic units for knowledge base manipulation in such schemes.

An important advantage of logical schemes is the availability of inference rules in terms of which one can define proof procedures. Such procedures can be used for information retrieval [REIT78a], semantic constraint checking [NY78], and problem solving [GREE69].

[NILS71] presents a review of early results, applications, and promises of theorem-proving research, whereas [GM78] contains a representative sample of more recent work on logical schemes and theorem-proving and their applications to Databases.

Another strength of logical schemes is the availability of a clean, well understood and well accepted formal semantics [MEND64], at least for "pure" logical schemes that are quite close to first order logic. As one moves to representation schemes that try to deal with knowledge acquisition [MD78], beliefs [MOOR77], and defaults [REIT78b], the availability of a clean formal semantics becomes more problematic and is an area of active research. The chapter by Levesque dealing with the semantics of incomplete knowledge within a logical framework gives a good indication of what is and isn't provided by classical logic to the KR researcher.

A third strength of logical schemes is the simplicity of the notation employed, thus facilitating knowledge base descriptions that are understandable. Yet another strength is the conceptual economy encouraged by such schemes, allowing each fact to be represented once, independently of its different uses during the course of its presence in the knowledge base.

An important drawback of logical schemes is the lack of organizational principles for the facts that constitute a knowledge base. A large knowledge base, like a large program, needs organizational principles to be understandable as a unit. Without them, a knowledge base can be as unmanageable as a program written in a programming language that does not support abstraction facilities.

A second drawback is the difficulty in representing procedural and heuristic knowledge[2] such as:

> "If you are trying to do A while condition B holds, try strategies C_1, C_2, . . . , C_n."

An interesting departure from logical representation schemes has been proposed by Kowalski [KOWA74], who argues in favour of a dual semantics for logical formulas of the form:

> "B_1 *and* B_2 *and* . . . *and* B_m *implies* A"

The first is the traditional Tarskian semantics; the second is a procedural semantics that interprets the formula as:

> "If you want to establish A, try to establish B_1 and B_2 and . . . and B_m."

---

[2] [HAYE77] argues against this point.

The language PROLOG [KOWA74] exemplifies this idea and has gained many supporters because it combines the advantages of logical and procedural representation schemes.

Another attempt to integrate logical and procedural representations has resulted in the representation language FOL [WEYH80]. Here procedures can be used as referents of logical expressions. Reasoning in FOL can be carried out either in terms of inference rules or procedures, thus combining the strengths of both approaches to KR.

Logical schemes are strongly related to Codd's Relational Model [CODD70], and it is fair to argue that such schemes have their counterparts in Database Management. The chapter by Reiter explores this relationship and argues for a proof-theoretic view of the Relational Model.

2.2 Network Representation Schemes

Semantic networks come in such a wide variety of forms and are used in so many ways that it is difficult to pinpoint what is common to all of them. To a large extent, this diversity is explained by the history of these networks that is summarized in the chapter by Israel and Brachman. In its most basic form, however, a semantic network represents knowledge in terms of a collection of objects (nodes) and binary associations (directed labelled edges), the former standing for individuals (or concepts of some sort), and the latter standing for binary relations over these. According to this view, a knowledge base is a collection of objects and relations defined over them, and modifications to the knowledge base occur through the insertion/deletion of objects and the manipulation of relations. Ever since they were originally proposed [QUIL68], most network schemes have favoured the use of binary relations as a means of representing binary or components of n-ary relationships. A network knowledge base has an obvious graphical representation where each node denotes an object and each labelled edge $(n_1, R, n_2,)$ indicates that $(n_1, n_2,) \in R$, R being one of the relations used in the knowledge base.

Early versions of network schemes tended to encourage a proliferation of relations that had little or no semantics when new kinds of knowledge were represented. Indeed, to some, semantic networks were nothing more than a (cute) notation in search of a semantics. This practice and other deficiencies of earlier network schemes are criticized in influential papers by Woods [WOOD75] and Schubert [SCHU76]. Such criticism has triggered a trend towards using network schemes that have formal semantics and are *descriptively adequate* (*i.e.*, can be used to represent any fact expressible in a logical scheme). Some of these schemes simply view network knowledge bases as convenient notations and/or implementations of logical knowledge bases ([SHAP79]

[SCHU76], *etc.*). Others, notably KL-ONE [BRAC79] view network schemes as tackling a different set of representational issues, and they propose a set of primitive relations accordingly.

A crucial issue of network schemes is the organizational axes they offer for structuring a knowledge base. Some of the axes that have been used are discussed briefly below.

Classification

According to classification, an object (*e.g.*, John Smith) should be associated with its generic type(s) (*e.g.*, STUDENT, MALE, PERSON). Including this organizational axis in a network scheme forces a distinction between *tokens* (*e.g.*, John Smith) and *types* (*e.g.*, PERSON). Some network schemes use classification recursively to define (meta) types with instances types, *etc.* (*e.g.*, PSN [LM79]).

Aggregation

This axis relates an object (*e.g.*, John Smith) to its components or parts. For example, the parts of John Smith, viewed as a physical object, are his head, arms *etc.* When viewed as a social object, they are his address, social insurance number, *etc.* As with classification, aggregation can be applied recursively so that one can represent the components of the components of an object, *etc.* Thus, aggregation defines a second organizational dimension for network schemes.

Generalization

Generalization relates a type (*e.g.*, STUDENT) to more generic ones (*e.g.*, PERSON). The generalization relation between types, often called is-a, is a partial order and organizes types into a *generalization* or *is-a hierarchy*. A common use of this hierarchy in semantic networks has been to minimize storage requirements by allowing properties associated with general object types to be inherited by more specialized ones. In addition, generalization and the other primitive association types provide the means for the overall organization and management of a large knowledge base.

Partitions

Another method of organizing network knowledge bases is proposed in [HEND75], and it involves grouping objects and elements of relations into *partitions* that are organized hierarchically, so that if partition A is below partition B, everything visible or present in B is also visible in A unless otherwise specified. Partitions have been found useful in representing time, hypothetical worlds, and belief spaces (*e.g.*, [COHE78]).

Not all network schemes treat the organizational principles mentioned above in the same way. For example, NETL [FAHL79] and others identify classification with generalization.

Due to their nature, network schemes directly address issues of information retrieval, since the associations between objects define access paths for traversing a network knowledge base. Another important feature of network schemes is the availability of organizational principles. A third is the graphical notation that can be used for network knowledge bases and that enhances their understandability.

A major drawback of network schemes has been the lack of formal semantics and standard terminology. The chapter by Israel and Brachman provides a brief history of semantic networks and a thorough account of their semantic deficiencies.

2.3 Procedural Representation Schemes

Such schemes view a knowledge base as a collection of active agents or processes. Most procedural schemes have been influenced quite heavily by LISP, which has been used almost exclusively as the implementation language for AI systems. Indeed, in the past, LISP itself was a favorite representation scheme due to, among other things, its basically symbolic nature and the dynamic run-time environment it offers its users.

Procedural schemes beyond LISP can be classified on the basis of the stand they take with respect to two issues. The first is concerned with the activation mechanism offered for processes. The second involves the control structures that are available.

On the first issue, PLANNER [HEWI71] [HEWI72] introduced the notion of *pattern directed procedure invocation*. A knowledge base is viewed in PLANNER as a global database of assertions and a collection of *theorems* (or *demons*) that watch over it and are activated whenever the database is modified or searched. Each theorem has an associated *pattern* which, upon the theorem's activation, is matched against the data about to be inserted/removed or retrieved from the database. If the match succeeds, the theorem is executed. Thus with theorems the usual procedure calling mechanism is replaced with one in which procedures are called whenever a condition is satisfied.

Production systems [WH79] offer a procedural scheme that is in many ways similar to PLANNER. A knowledge base is a collection of *production rules* and a global database. Production rules, like theorems, consist of a pattern and a body involving one or more *actions*. The database begins in some initial state, and rules are tried out in some prespecified order until one is found whose pattern matches the database. The body of that rule is then executed, and matching of other rules continues. This account is an idealization of production systems and most of them vary in the form of rules they follow and the order in which they are tried [DK75].

There are major differences between the activation mechanism of a PLANNER theorem and a production system rule as well. The order in which theorem patterns are matched is undetermined in PLANNER (although the user can define one for any particular situation in which he tries to tamper with the database). "Standard" production systems, like Markov algorithms, have a fixed ordering of rules that determine when each rule will be matched against the database. Another important difference is that theorems can directly call other theorems whereas productions can do so only indirectly by placing appropriate information in the database. Thus, a production system database can be viewed as a workspace or a bulletin board that provides the only means of communication between rules.

Turning to control structures, several proposals exist which extend or otherwise modify the usual hierarchical control structure of LISP or ALGOL. As indicated earlier, production systems offer one where there is no direct communication or control between rules. Thus a production system knowledge base consists of a collection of *loosely coupled* rules, and this feature renders such knowledge bases easy to understand and modify.

PLANNER's control structure for theorems uses *backtracking,* and when a theorem's body is executed and fails to achieve a predetermined goal, the side-effects of the unsuccessful theorem are erased and other theorems are tried until one is found that succeeds. It has been argued quite convincingly that backtracking is an unwieldy control structure [SM72] and it should be avoided at all costs.

An extreme proposal with regard to control structures is Hewitt's ACTOR formalism [HBS73] [HG74] which views *all* objects that are part of a knowledge base as *actors* (*i.e.,* active agents that play a role on cue according to a script). Actors are capable of sending and receiving *messages* which, naturally, are also actors. Thus, writing a program in the ACTOR formalism involves deciding on the objects in the domain, the messages each object should receive, and what each object should do when it receives each kind of message. The ACTOR formalism basically does not impose a preconceived control structure on its user. Instead, it provides him with control primitives so that he can define his own. The ACTOR formalism was inspired by the Smalltalk programming language [BYTE81] which has been under development at Xerox PARC for more than a decade.

Procedural schemes have, in principle, one major advantage and one major drawback compared with other types of schemes. They allow the specification of direct interactions between facts, thus eliminating the need for wasteful searching [WINO75]. On the other hand, a procedural knowledge base, like a program, is difficult to understand and modify. Each of the proposed schemes discussed in the previous paragraphs goes some distance toward eliminating the drawbacks of pure procedural schemes while maintaining their advantages.

2.4 Frame-Based Representation Schemes

Since 1975, when Minsky originally proposed it [MINS75], the notion of *frame* has played a key role in KR research. A frame is a complex data structure for representing a stereotypical situation, such as being in a certain kind of living room or going to a child's birthday party. The frame has slots for the objects that play a role in the stereotypical situation as well as relations between these slots. Attached to each frame are different kinds of information, such as how to use it, what to do if something unexpected happens, default values for its slots, *etc.* A knowledge base is now a collection of frames organized in terms of some of the organizational axes discussed earlier, but also other "looser" principles such as the notion of *similarity* between two frames.

Minsky's original frame proposal essentially provided a framework for developing representation schemes that combined ideas from semantic networks, procedural schemes, linguistics, *etc.* Several representation schemes proposed since then have further developed the frame proposal. Below we present brief descriptions of four of them.

FRL [GR77]

An FRL knowledge base consists of frames whose slots carry information such as comments on the source of a value bound to the slot, a default value, constraints, and procedures that are activated when a value is bound, unbound, or needed for a slot. All frames are organized into a hierarchy which appears to be a combination of classification and generalization as described in Section 2.2. The procedures attached to a slot are expressed in LISP.

KRL [BW77]

This is a more ambitious representation language than FRL. Like FRL, the basic units of a KRL knowledge base are frames that have slots and that have several kinds of information attached to each slot. Unlike FRL, where this information provides details about how to instantiate a frame, KRL is much more concerned with a matching operation for frames. All on-going processes are controlled by a multiprocessor agenda that can be scheduled by the designer of the knowledge base. KRL also supports belief contexts that can serve to define an attention focusing mechanism. "Self knowledge" can be included in a knowledge base by providing descriptions of other descriptions.

OWL [SHM77]

Unlike other frame-oriented schemes, OWL bases its features on the syntactic and semantic structure of English, taking as its founding principle the Whorfian Hypothesis that a person's language plays a key role in determining his model of the world and thus in structuring his thought. An OWL knowledge base can be viewed as a semantic network whose nodes are expressions representing the meaning of natural language sentences. Each node, called a concept, is defined by a pair (genus, specializer) where "genus" specifies the type or superconcept and "specializer" serves to distinguish this concept from all other concepts that have the same genus.

KL-ONE [BRAC79]

A KL-ONE knowledge base is a collection of concepts, and each concept is a highly structured object, having slots to which one can attach a variety of information (defaults, modalities, *etc.*). To a concept one can also attach structural descriptions that express constraints on the values that can be bound to the different slots of the concept. Concepts provide purely descriptional structure and make no assertions about existence of a referent or coreference of descriptions. A separate construct called a *nexus* is used to make assertions about the world being modelled. Also, KL-ONE offers procedural attachment as a means of associating procedural information (expressed at this time in LISP), with a concept. Another important feature of KL-ONE is the strong organization of concepts it encourages through a version of the generalization axis discussed in Section 2.2.

Two other important representation schemes are introduced in later chapters. Omega is a description-based scheme and is sketched briefly in the chapter by Hewitt and de Jong. The Plan Calculus, described by Rich, is a frame-based scheme intended for the representation of programming knowledge. The chapter by Borgida, Mylopoulos, and Wong derives many of its key ideas from PSN, yet another frame-oriented scheme described in [LM79].

3. Distinguishing Features of Representation Schemes

The reader who has a background in Databases and/or Programming Languages must have already noticed the similarity in basic goals between KR research as we have described it in this paper and research on Semantic Data Models or Program Specifications. In all three cases the aim is to provide tools for the development of descriptions of a world/enterprise/slice of reality which correspond *directly* and *naturally* to our own conceptualizations of the object of these descriptions. The tools under consideration involve a representation scheme/semantic data model/specification language that serves as the linguistic vehicle for such descriptions. Below we list some of the more technical (and less vague) characteristics of representation schemes whose qualities distinguish them from their semantic data model/program specification language cousins.

3.1 Multiple Uses of Facts

Unlike a database, whose facts are used exclusively for retrieval purposes, or a program, whose facts are used only during the execution of some procedure, a knowledge base contains facts that may have multiple uses. A representation scheme must take this into account in terms of the tools it offers. Below we list some possible uses [BOBR75].

Reasoning

Given a collection of facts, new facts may be deduced from them according to given rules of inference without interaction with the outside world. Some inferences have the flavour of inference techniques in logic. For knowledge bases, however, it is also sometimes useful to derive facts by means of specialized procedures that exploit given facts only in fixed ways. For example, a procedure that determines whether a pair is in the transitive closure of some binary relation can perform reasoning of a very specialized nature and is only applicable to facts associated with a transitive relation. Also, a knowledge base may be represented in such a way that there are "preferred inferences." The use of defaults is a good example of such a mechanism.

Deductive reasoning, which has a formal, special purpose or heuristic flavour, is not the only kind of reasoning. There are also inductive [BROW73] and abductive reasoning [POPL73], which have played a role in some knowledge bases.

Given this variety of reasoning mechanisms, the question for a designer of a representation scheme is not how he can include all of them in his scheme, but which one, if any, he is going to include. Logical schemes clearly have an advantage over other types of schemes when considered from the point of view of (general purpose) reasoning facilities.

Access

Access (and storage) of information in a knowledge base for question-answering purposes constitutes an all-important use of the knowledge base. The associationist viewpoint of network schemes, particularly their organizational axes, make them strong candidates for access-related uses.

Matching

Matching as a knowledge base operation can be used for a variety of purposes, including:

1. *classification* (*i.e.*, determining the type of an unknown input)
2. *confirmation* where a possible candidate to fit a description is matched against it for confirmation purposes
3. *decomposition* where a pattern with a substructure is matched against a structured unknown and the unknown is decomposed into subparts corresponding to those of the pattern
4. *correction* where the nature of a pattern match failure leads to error correction of the unknown input

The matching operation itself can be:

1. *syntactic* where the form of the unknown input is matched against another form
2. *parametric* in the tradition of Pattern Recognition research [DH73]
3. *semantic* where the function of the components of the pattern is specified and the matcher attempts to find elements of the input to serve this function
4. *forced matching* as in MERLIN [MN74] where a structure is viewed as though it were another and matches of corresponding items may be forced

KRL has paid special attention to matching as a knowledge base operation.

3.2 Incompleteness

Except for situations in which a knowledge base models artificial "microworlds" (*e.g.*, [WINO72]), it cannot be assumed that the knowledge base is a complete description of the world it is intended to model. This observation has important consequences for the operations defined over a knowledge base (inference, access, matching) as well as the design methodologies for knowledge bases.

Consider first the operations on a knowledge base. Incompleteness of the knowledge base can lead to very different answers to questions such as:

"Is there a person who lives in Toronto?"

and the answer will depend on whether it is assumed that the persons currently represented in the knowledge base are the only persons in the world being modelled. If the knowledge base is taken to be complete, it may be sufficient to search through the objects related in a certain way to the object representing Toronto. If the knowledge base is possibly incomplete, however, the answer can be "yes" without there being any corresponding object in the knowledge base. A second example is the question:

> "How many children does Mary have?"

which might be answered, under the completeness assumption, by counting representational objects that satisfy some criteria. Without this assumption much more complex forms of reasoning (such as reasoning by cases, reductio ad absurdum and the like) might be required to determine the answer. Similarly, from the facts:

> "Someone is married to Mary"
> "John is not married to Mary"

one can draw different conclusions if George is the only other person represented in the knowledge base, depending on whether it is assumed that John, Mary and George are the only persons in the world being modelled. Similar remarks apply for matching.

Until recently much of the work on KR ignored the problem of incompleteness or dealt with it in an *ad hoc* way. The chapters in this volume by Levesque and Reiter can be seen as attempts to correct this situation. Reiter shows how different forms of incompleteness (and especially the null values of the Relational Model) can be explained in terms of the proof theory of first order logic. Levesque begins with the very general form of incompleteness allowed by first order logic and investigates a query language appropriate for knowledge bases that are radically incomplete.

Viewing a knowledge base as an incomplete and approximate model of a world that can always be improved but can never be quite complete, leads to design methodologies for knowledge bases that are drastically different from design methodologies that are designed for programs. Thus, in Programming Languages the leading design methodology encourages a "once and for all" process where the designer begins with a clear idea of the algorithm he wants to realize and proceeds to construct a complete design (*e.g.*, [WIRT81]). In AI, a knowledge base is developed over a period of time that can be as long as its lifetime by means of different *knowledge acquisition* processes that can range from interactive sessions with an expert (*e.g.*, [DAVI77]) to the automatic generation of new facts based on the system's "introspections" (*e.g.*, [LENA77]). Organizational principles underlying the structure of a knowledge base can play a crucial role in determining the direction of knowledge acquisition (*i.e.,* which facts should be acquired first and which facts should be acquired later).

3.3 Self Knowledge

There are many kinds of self knowledge, and some of them were described in the previous section. For instance, the statement:

"All students are known (to the knowledge base)"

says something about the state of the knowledge base, not the world. Facts that describe the form or allowable configurations of other facts (*e.g.,* type definitions) constitute an important class of self knowledge. Making such facts available for question answering and inference, by representing them the same way as other facts, is an important capability of *declarative* schemes (*i.e.,* logical and network schemes) generally not shared by procedural schemes. A good example of the use of such self-knowledge for knowledge acquisition is provided in TEIRESIAS [DAVI77].

A second kind of self-knowledge involves the ability of a system to answer elementary questions about its actions as in SHRDLU [WINO72], or about the strategies it uses to debug problem solving procedures as in HACKER [SUSS75].

A very general introspective architecture is proposed and investigated in [DOYL80]. It is shown that one's reasoning about what inferences to make can be used in making decisions and in taking action. All relevant aspects of the intentional state of the system (such as its goals, beliefs, *etc.*) are subject to scrutiny and are therefore explicitly represented.

[SMIT82] defines a new dialect of LISP in which programs can "reflect" on their own execution. At any stage of the computation, a program can jump to the level of its interpreter and examine what state it was in as encoded in the data structures of the interpreter. In particular, a program can look at what it has left to do on the stack and perhaps decide to do something completely different.

4. Current Issues

While there is perhaps no general agreement about the major unresolved issues in KR, there is a definite trend away from the more implementational issues and towards more formal and conceptual investigations of representation schemes. This has led, among other things, to a reappraisal of the role of formal logic in KR [NEWE81] [MOOR82]. The most apparent result of this is the recent trend towards hybrid schemes which incorporate both logical and nonlogical sublanguages [RICH82] [ISRA82] [BL82]. These hybrid schemes do not attempt to overlay features of a logical language on top of, for example, a semantic network, as was often done in the past, but instead partition the knowledge representation task so that the network and the logical languages are given separate responsibilities.

In [BL82], for example, a knowledge base is factored into a *terminological* component that maintains the technical vocabulary of a domain, and an *assertional* component that maintains a collection of facts about that domain. A terminological sublanguage in the style of KL-ONE is used to provide a set of term-forming facilities that allow new terms to be appropriately placed on a taxonomy relative to previously defined ones. A first order logical sublanguage is used to manage the assertional component and provides facilities for forming a theory and reasoning about the domain using a theorem prover. The point of contact between the two components is the predicate and function symbols of the logical language: these nonlogical symbols are, in fact, the technical terms of the terminological component. This means that deduction within the assertional component must treat the nonlogical symbols not as primitives (as in a standard logic), but as structured terms that have a complex meaning to be derived from the terminological component. The claimed advantage of the separation of the two areas, however, is that each component can be optimized independently and that neither has to suffer from the limitations of the other.

Another important issue in KR that recently has received considerable attention is the formalization of default reasoning. Standard logical deduction schemes are *monotonic* since new axioms never invalidate previous theorems. Common sense belief revision, on the other hand, is obviously *nonmonotonic* since the acquisition of knowledge can cause old beliefs (specifically, those that were held by default) to be discarded. The papers in [BOBR80] examine formal systems that have this nonmonotonicity property, and more recent developments are discussed in [ISRA80] [REIT81] [KONO82] [REIT82].

5. Acknowledgement

Alex Borgida read an earlier draft of this paper and offered many thoughtful suggestions for improvement.

6. References

[BOBR75] [BC75] [BOBR77] [BOBR80] [BW77] [BRAC79] [FIND79] [BS80] [BZ81] [BL82] [BROW73] [BYTE81] [CODD70] [COHE78] [DAVI77] [DK75] [DOYL80] [DH73] [FAHL79] [FEIG77] [FIND79] [GM78] [GP77] [GR77] [GREE69] [HAYE74] [HAYE77] [HEND75] [HEWI71] [HEWI72] [HBS73] [HG74] [ISRA80] [ISRA82] [KONO82] [KOWA74] [LENA77] [LM79] [MD78] [MEND64] [MINS75] [MOOR77] [MOOR82] [MN74] [NEWE81] [NY78] [NILS71] [POPL73] [QUIL68] [REIT78a] [REIT78b] [REIT81] [REIT82] [RICH82] [SCHU76] [SHAP79] [SMIT82] [SUSS75] [SM72] [SHM77] [WH79] [WEYH80] [WINO72] [WINO75] [WIRT71] [WM77] [WOOD75]

References

[ABRI74]
Abrial, J.R., "Data Semantics," in J.W. Klimbie and K.L. Koffeman (eds.), *Data Management Systems,* North-Holland, Amsterdam, The Netherlands, 1974.

[ACC82]
Atkinson, M., K. Chisholm, and P. Cockshott, "PS-ALGOL: An ALGOL with a Persistent Heap," *SIGPLAN Notices,* Vol. 17, No. 7, July 1982.

[AGBB77]
Ambler, A.L., D.I. Good, J.C. Browne, W.F. Burger, R.M. Cohen, C.G. Hoch and R.E. Wells, "Gypsy: A Language for Specification and Implementation of Verifiable Programs," *SIGPLAN Notices,* Vol. 12, No. 3, March 1977, pp. 1-10.

[AGP78]
Arvind, (no initial), K.P. Gostelow, and W. Plouffe, "The (Preliminary) Id Report," Technical Report 114, Dept. of Information and Computer Science, Univ. of California, Irvine, 1978.

[AI80]
Artificial Intelligence, Special Issue on Non-Monotonic Logic, D. Bobrow (ed.), Vol. 13, Nos. 1 and 2, April 1980.

[AM81]
Ariav, G., and H.L. Morgan, "MDM: Handling the Time Dimension in Generalized DBMSs," Decision Sciences Working Paper 81-05-06, Wharton School, Univ. of Pennsylvania, 1981.

[ANSI75]
ANSI/X3/SPARC (Standards Planning and Requirements Committee), "Interim Report from the Study Group on Database Management Systems," *FDT* (Bulletin of ACM SIGMOD), Vol. 7, No. 2, 1975.

[AS81]
Attardi, G. and M. Simi, "Semantics of Inheritance and Attributions in the Description System Omega," *Proc. International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, August 1981.

[AU79]
Aho, A. V., and J. D. Ullman, "Universality of Data Retrieval Languages," *Proc. 6th ACM Symposium on Principles of Programming Languages,* 1979.

[AV80]
Apt, K. R., and Van Emden, M. H., "Contributions to the Theory of Logic Programming," Research Report CS-80-12, Dept. of Computer Science, Univ. of Waterloo, Waterloo, Ont., Canada, 1980.

[BACH77]
Bachman, C. W., "The Role Concept in Data Models," *Proc. 3rd International Conference on Very Large Databases,* Tokyo, Japan, 1977.

[BACK78]
Backus, J., "Can Programming be Liberated from the von Neumann Style?," *Communications of the ACM,* Vol. 21, No. 8 August 1978, pp. 613-641.

[BALZ81]
Balzer, R., "Transformational Implementation: An Example," *IEEE Transactions on Software Engineering,* Vol. SE-7, No. 1, January 1981.

[BARB82]
Barber, G. R., "Office Semantics," Ph.D. thesis, Massachusetts Institute of Technology, 1982.

[BARR80]
Barron, J. L., Dialogue Organization and Structure for Interactive Information Systems, Master's thesis, (CSRG Technical Report) Dept. of Computer Science, Univ. of Toronto, January 1980.

[BARS77]
Barstow, D. R., "Automatic Construction of Algorithms and Data Structures Using A Knowledge Base of Programming Rules," Stanford AIM-308, November 1977.

[BARW81]
Barwise, J., "Some Computational Aspects of Situation Semantics (Abstract)," Unpublished manuscript, 1981.

[BBC80]
Bernstein, P. A., B. T. Blaustein, and E. M. Clarke, "Fast Maintenance of Integrity Assertions Using Redundant Aggregate Data," *Proc. 6th International Conference on Very Large Databases,* Montreal, Que., Canada, October 1980.

[BBD77]
Bell, M. L., D. C. Bixler, and M. E. Dyer, "An Extendible Approach to Computer-Aided Software Requirements Engineering," *IEEE Transactions on Software Engineering*, Vol. SE-3, No. 1, January 1977.

[BBDG81a]
Bauer, F. L., M. Broy, W. Dosch, R. Gnatz, F. Geiselbrechtinger, W. Hesse, B. Krieg-Brueckner, A. Laut, T. A. Matzner, B. Moeller, H. Partsch, P. Pepper, K. Samelson, M. Wirsing, H. Woessner, "Report on a Wide Spectrum Language for Program Specification and Development" (tentative version), Institut fuer Informatik der TU

Muenchen, TUM-I8104, 1981; also in: *Lecture Notes in Comptuer Science,* Springer-Verlag, New York, 1981.

[BBDG81b]
Bauer, F. L., M. Broy, W. Dosch, R. Gnatz, B. Krieg-Brueckner, A. Laut, M. Luckmann, T. A. Matzner, B. Moeller, H. Partsch, P. Pepper, K. Samelson, R. Steinbrueggen, M. Wirsing, H. Woessner, "Programming in a Wide Spectrum Language: a Collection of Examples," *Science of Computer Programming* Vol. 1, 1981, pp. 73-114.

[BBG78]
Beeri, C., P. A. Bernstein, and N. Goodman, "A Sophisticate's Introduction to Database Normalization Theory," *Proc. 4th International Conference on Very Large Databases,* West Berlin, September 1978.

[BC75]
Bobrow, D., and A. Collins, (eds.), *Representation and Understanding,* Academic Press, New York, 1975.

[BD77]
Burstall, R. M., and J. L. Darlington, "A Transformation System for Developing Recursive Programs," *Journal of the ACM,* Vol. 24, No. 1, January 1977.

[BD81]
Barr, A., and J. Davidson, "Representation of Knowledge," in A. Barr and E. Feigenbaum (eds.), *Handbook of Artificial Intelligence,* William Kaufmann Inc., 1981.

[BDMN73]
Birtwistle, G. M., O.-J. Dahl, B. Myhrhaug, K. Nygaard, *Simula Begin,* Van Nostrand Reinhold, New York, 1973.

[BF79]
Buneman, O. P., and R. E. Frankel, "FQL — A Functional Query Language," *Proc. 1979 ACM SIGMOD International Conference on the Management of Data,* Boston, Mass., May 1979.

[BFN81]
Buneman, O. P., R. E. Frankel, and R. Nikhil, "A Practical Functional Programming System for Databases," *Proc. ACM Conference on Functional Programming and Archictecture,* New Hampshire, 1981.

[BFN82]
Buneman, O. P., R. E. Frankel, and R. Nikhil, "An Implementation Technique for Database Query Languages," *ACM Transactions on Database Systems,* Vol. 7, No. 2, June 1982.

[BG80a]
Bobrow, D. and Goldstein, I., "Representing Design Alternatives," *Proc. Society for Study of Artificial Intelligence and Simulation of Behavior Conference,* Amsterdam, The Netherlands, July 1980.

[BG80b]
Burstall, R. M., and J. A. Gougen, "The Semantics of CLEAR: A Specification Language," *Proc. Copenhagen Winter School on Abstract Software Specification,* Copenhagen, Denmark, 1980.

[BHR80]
Bayer, R., H. Heller, and A. Reiser, "Parallelism and Recovery in Database Systems," *ACM Transactions on Database Systems,* Vol. 5, No. 2, June 1980.

[BI82]
Borning, A.H., and D.H. Ingalls, "Multiple Inheritance in Smalltalk-80," *Proc. AAAI National Conference,* Pittsburgh, Penn., August 1982.

[BISK81]
Biskup, J., "Null Values in Data Base Relations," in [GM78].

[BJ78]
Bjorner, D., and C.B. Jones, *The Vienna Development Method,* Springer-Verlag, New York, 1978.

[BL82]
Brachman, R.J., and H. Levesque, "Competence in Knowledge Representation," *Proc. AAAI National Conference*, Pittsburgh, Penn., August 1982, pp. 189-192.

[BM76]
Basu, S., and J. Misra, "Some Classes of Naturally Provable Programs," *2nd International Conference on Software Engineering,* San Francisco, October 1976.

[BM77]
Buneman, O.P., and H.L. Morgan, "Alerting Techniques for Database Systems," *IEEE COMPSAC Conference,* Chicago, Ill., November 1977.

[BMS80]
Burstall, R.M., D.B. MacQueen, and D.T. Sanella, "HOPE: an Experimental Applicative Language," *Proc. Lisp Conference,* Stanford, Calif., 1980.

[BMW82]
Borgida, A.T., J. Mylopoulos, and H.K.T. Wong, "Methodological and Computer Aids for Interactive Information System Design," in H.J. Schneider and A. Wasserman (eds.), *Automated Tools for Information System Design — Proc. of IFIP Conference,* North-Holland, Amsterdam, The Netherlands, 1982.

[BOBR75]
Bobrow, D.G., "Dimensions of Representations," in [BC75].

[BOBR77]
Bobrow, D.G., "A Panel on Knowledge Representation," *Proc. 5th International Joint Conference on Artificial Intelligence,* Cambridge, Mass., August 1977.

[BORG81]
Borgida, A.T., "On the Definition of Specialization Hierarchies for Procedures," *Proc. 7th International Joint Conference on Artificial Intelligence,* Vancouver, B.C., Canada, August 1981.

[BORG82a]
Borgida, A. T., "Conceptual Modeling for Information System Development," *Proc. 1st AUC Conference on Databases,* Medellin, Colombia, August 1982.

[BORG82b]
Borgida, A. T., "Prospectus for Research on Flexible Information Systems," Dept. of Computer Science, Rutgers Univ., 1982.

[BOWL77]
Bowles, K. L., *Microcomputer Problem Solving Using Pascal,* Springer-Verlag, New York, 1977.

[BP80]
Barwise, J., and J. Perry, "The Situation Underground," unpublished manuscript, 1980.

[BP81a]
Barwise, J., and J. Perry, "Situations and Attitudes," *Journal of Philosophy,* Vol. 78, No. 11, October 1981, pp. 668-691.

[BP81b]
Barwise, J., and J. Perry, "Semantic Innocence and Uncompromising Situations," in P. A. French, T. E. Uehling, Jr., and H. K. Wettstein, (eds.), *The Foundations of Analytic Philosophy,* Midwest Studies in Philosophy, Vol. VI, Univ. of Minnesota Press, Minneapolis, 1981, pp. 387-404.

[BP81c]
Broy, M., and P. Pepper, "Program Development as a Formal Activity," *IEEE Transactions on Software Engineering,* Vol. SE-7, No. 1, January 1980.

[BPP76]
Bracchi, G., P. Paolini, and G. Pelagatti, "Binary Logical Associations in Data Modelling," in J. M. Nijssen (ed.), *Modelling in Database Management Systems* (Proc. IFIP TC2 Conference, Freudenstadt), North-Holland, Amsterdam, The Netherlands, 1976.

[BR70]
Buxton, J. N., and B. Randell (eds.), *Software Engineering Techniques,* NATO, 1970 (report on a conference sponsored by the NATO Science Committee, Rome, Italy, October 27-31, 1969).

[BRAC76]
Brachman, R. J., "A Structural Paradigm for Representing Knowledge," BBN Report No. 3605, Bolt, Beranek and Newman Inc., Cambridge, Mass., 1976.

[BRAC79]
Brachman, R. J., "On the Epistemological Status of Semantic Networks" in [FIND79], pp. 3-50.

[BRAC80a]
Brachman, R. J., "I Lied about the Trees," unpublished manuscript, 1980.

[BRAC80b]
Brachman, R. J., "An Introduction to KL-ONE," in R. J. Brachman, *et al.* (eds.), *Research in Natural Language Understanding, Annual Report (1 Sept. 78-31 Aug. 79),* Bolt, Beranek and Newman Inc., Cambridge, Mass., 1980, pp. 13-46.

[BRES72]
Bressan, A., *A General Interpreted Modal Calculus,* Yale Univ. Press, New Haven, Conn., 1972.

[BRIN75]
Brinch Hansen, P., "The Programming Language Concurrent Pascal," *IEEE Transactions on Software Engineering,* Vol. SE-1, No. 4, June 1975, pp. 199-207.

[BROD78]
Brodie, M. L., "Specification and Verification of Database Semantic Integrity," Ph.D. thesis (Computer Systems Research Group Technical Report No. 91), Univ. of Toronto, April 1978.

[BROD80a]
Brodie, M. L., "The Application of Data Types to Database Semantic Integrity," *Information Systems*, Vol. 5, No. 4, 1980.

[BROD80b]
Brodie, M. L., "Data Abstraction, Databases and Conceptual Modeling," *Proc. 6th International Conference on Very Large Databases,* Montreal, Que., Canada, October 1980.

[BROD80c]
"Data Quality in Information Systems," *Information & Management,* Vol. 3, 1980.

[BROD81a]
Brodie, M. L., "Association: A Database Abstraction for Semantic Modelling," *Proc. 2nd International Entity-Relationship Conference*, Washington, D.C., October 1981.

[BROD81b]
Brodie, M. L., "On Modelling Behavioural Semantics of Data," *Proc. 7th International Conference on Very Large Databases,* Cannes, France, September 1981.

[BROD82]
Brodie, M. L., "Axiomatic Definitions for Data Model Semantics," *Information Systems*, Vol. 7, No. 2, 1982.

[BROO75]
Brooks, F. P., Jr., *The Mythical Man-Month: Essays on Software Engineering,* Addison-Wesley, Reading, Mass., 1975.

[BROW73]
Brown, J. S., "Steps Towards Automatic Theory Formation," *Proc. International Joint Conference on Artificial Intelligence,* Palo Alto, Calif., August 1973.

[BROW80]
Browne, J.C., The Interaction of Operating Systems and Software Engineering," *Proc. IEEE,* Vol. 68, No. 9, September 1980.

[BRUC75]
Bruce, B., "Case Systems for Natural Language," *Artificial Intelligence,* Vol. 6, 1975, pp. 327-360.

[BS78]
Brodie, M.L., and J.W. Schmidt, "What is the Use of Abstract Data Types?," *Proc. 4th International Conference on Very Large Databases,* West Berlin, September 1978.

[BS79]
Bentley, J.L., and M. Shaw, "An Alphard Specification of a Correct and Efficient Transformation on Data Structures," *Proc. IEEE Conference on Specifications of Reliable Software,* April 1979, pp. 222-237.

[BS80]
Brachman, R.J., and B. Smith, Special Issue on Knowledge Representation, *SIGART Newsletter,* No. 50, February 1980.

[BS82a]
Bobrow, D.G., and M.J. Stefik, "Loops: An Object Oriented Programming System for Interlisp," Draft Report, Xerox PARK, 1982.

[BS82b]
Brodie, M.L., and J.W. Schmidt (eds.), "Final Report of the ANSI/X3/SPARC DBS-SG Relational Database Task Group," *SIGMOD Record,* Vol. 12, No. 4, July 1982.

[BS82c]
Brodie, M.L., and E.O. Silva, "Active and Passive Component Modelling: ACM/PCM," in [OSV82], pp. 41-91.

[BSD82]
Byrd, R.J., S.E. Smith, S.P. de Jong, "An Actor-Based Programming System," *SIGOA Conference on Office Information Systems,* June 1982.

[BSR80]
Bernstein, P.A., D.W. Shipman, and J.B. Rothnie, "Concurrency Control in a System for Distributed Databases (SDD-1), *ACM Transactions on Database Systems,* Vol. 5, No. 1, March 1980.

[BUNE79]
Bunemann, O. and E. Clemons, "Efficiently Monitoring Relational Databases," *ACM Transactions on Database Systems,* Vol 4., No. 3, September 1979.

[BURG75]
Burge, W.H., *Recursive Programming Techniques*, Addison-Wesley, Reading, Mass., 1975.

[BW77]
Bobrow, D., and T. Winograd, "An Overview of KRL, a Knowledge Representation Language," *Cognitive Science*, Vol. 1, No. 1, January 1977.

[BW81]
Borgida, A. T., and H. K. T. Wong, "Data Models and Data Manipulation Languages: Complimentary Semantics and Proof Theory," *Proc. 7th International Conference on Very Large Databases*, Cannes, France, September 1981, pp. 260-271.

[BYTE81]
Special Issue on Smalltalk, *BYTE*, August 1981.

[BZ81]
Brodie, M. L., and S. N. Zilles (eds.), *Proc. Workshop on Data Abstraction, Databases, and Conceptual Modelling, SIGART Newsletter*, No. 74, January 1981; *SIGMOD Record*, Vol. 11, No. 2, February 1981; *SIGPLAN Notices*, Vol. 16, No. 1, January 1981.

[CB74]
Chamberlin, D. D., and R. F. Boyce, "SEQUEL: a Structured English Query Language," *Proc. 1974 ACM SIGMOD International Conference on the Management of Data*, 1974.

[CBLL82]
Curry, G., L. Baer, D. Lipkie, B. Lee, "Traits: An Approach to Multiple-Inheritance Subclassing," *Proc. Conference on Office Information Systems, SIGOA Newsletter*, Vol. 3, Nos. 1 and 2, June 1982.

[CHAM75]
Chamberlin, D. D., *et. al.*, "Views, Authorization and Locking in a Relational Data Base System," *Proc. 1975 National Computer Conference*, Anaheim, Calif., May 1975.

[CHAM76]
Chamberlin, D. D., "Relational Database Management Systems," *Computing Surveys*, Vol. 8, No. 1, March 1976.

[CHEA81]
Cheatham, T. E., "Program Refinement by Transformation," *Proc. 5th International Conference on Software Engineering*, San Diego, Calif., March 1981.

[CHEN76]
Chen, P. P. S., "The Entity-Relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976.

[CHEN78]
Chen, P. P. S., *The Entity-Relationship Approach to Logical Database Design*, Monograph No. 6, QED Information Sciences, Wellesley, Mass., 1978.

[CHIL77]
Childs, D. L., "Extended Set Theory," *Proc. 3rd International Conference on Very Large Databases,* Tokyo, Japan, October 1977.

[CHUR40]
Church, A., "A Formulation of the Simple Theory of Types," *Journal of Symbolic Logic,* Vol. 5, 1940, pp. 56-68.

[CHUR41]
Church, A., "The Calculi of Lambda-Conversion," *Annals of Mathematics Studies No. 6,* Princeton Univ. Press, 1941.

[CL73]
Chang, C. L., and R. C. T. Lee, *Symbolic Logic and Mechanical Theorem Proving,* Academic Press, New York, 1973.

[CLAR78]
Clark, K. L., "Negation as Failure," in [GM78].

[CLIN81]
Clinger, W. D., "Foundations of Actor Semantics," Technical Report MIT/AI/TR-633, MIT Laboratory for Artificial Intelligence, May 1981.

[CM79]
Carlson, E. and W. Metz, "A Design for Table-Driven Display Generation and Management Systems," Technical Report, IBM Research Laboratory, San Jose, Calif., 1979.

[CODA71]
CODASYL Data Base Task Group, April 1971 Report.

[CODD70]
Codd, E. F., "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM,* Vol. 13, No. 6, June 1970, pp. 377-387.

[CODD71]
Codd, E. F., "A Database Sublanguage Founded on the Relational Calculus," *Proc. SIGFIDET Workshop,* San Diego, Calif., 1971.

[CODD72]
Codd, E. F., "Relational Completeness of Database Sublanguages," in R. Rustin (ed.), *Data Base Systems,* Prentice-Hall, Englewood Cliffs, N.J., 1972.

[CODD79]
Codd, E. F., "Extending the Database Relational Model to Capture More Meaning," *ACM Transactions on Database Systems*, Vol. 4, No. 4, December 1979, pp. 397-434; IBM Research Report RJ2599, San Jose, Calif., August 1979.

[CODD82]
Codd, E. F., "Relational Database: A Practical Foundation for Productivity," *Communications of the ACM,* Vol. 25, No. 2, February 1982.

[COHE78]
Cohen, P.R., *On Knowing What to Say: Planning Speech Acts*, Ph.D. thesis (TR-118), Dept. of Computer Science, Univ. of Toronto, 1978.

[CWAM75]
Collins, A., E. Warnock, N. Aiello, and M. Miller, "Reasoning from Incomplete Knowledge," in [BC75].

[DATE81]
Date, C.J., *An Introduction to Database Systems,* 3rd ed., Addison-Wesley, Reading, Mass., 1981.

[DATE83]
Date, C.J., *An Introduction to Database Systems Volume II,* Addison-Wesley, Reading, Mass., 1983.

[DAVI58]
Davis, M., *Computability and Unsolvability,* McGraw-Hill, New York, 1958.

[DAVI77]
Davis, R., "Interactive Transfer of Expertise: Acquisition of New Inference Rules," *Proc. 5th International Joint Conference on Artificial Intelligence,* Cambridge, Mass., August 1977.

[DB82]
Dayal, U., and P.A. Bernstein, "On the Updatability of Network Views—Extending Relational View Theory to the Network Model," *Information Systems,* Vol. 7, No. 1, 1982.

[DD80]
Demers, A.J., and J.E. Donahue, "Data Types, Parameters, and Type Checking," *Proc. ACM Symposium on Principles of Programming Languages,* SIGACT and SIGPLAN, January 1980, pp. 12-23.

[DEJO80]
de Jong, S.P., "The System for Business Automation (SBA): a Unified Application Development System," *Proc. 1980 IFIP Congress,* Tokyo, Japan, 1980.

[DENN74]
Dennis, J.B., "First Version of a Data Flow Procedure Language," *Proc. Symposium on Programming,* Institut de Programmation, Univ. of Paris, Paris, France, April 1974, 241-271.

[DEUT81]
Deutsch, L.P., "In Summary of Workshop Session on Types," *Proc. Workshop on Data Abstraction, Databases, and Conceptual Modelling, SIGPLAN Notices,* Vol. 16, No. 1, January 1981, p. 49.

[DH72]
Dahl, O.-J., and C.A.R. Hoare, "Hierarchical Program Structures," in O.-J. Dahl, E.W. Dijkstra, and C.A.R. Hoare (eds.), *Structured Programming,* Academic Press, New York, 1972, pp. 175-220.

[DH73]
Duda, R. O., and P. E. Hart, *Pattern Classification and Scene Analysis,* Wiley-Interscience, New York, 1973.

[DIJK68]
Dijkstra, E. W., "Goto Statement Considered Harmful," *Communications of the ACM,* Vol. 11, No. 3, March 1968, pp. 147-148.

[DIJK72]
Dijkstra, E. W. "Notes on Structured Programming," in O.-J. Dahl, E. W. Dijkstra, and C. A. R. Hoare (eds.), *Structured Programming,* Academic Press, New York, 1972.

[DIJK76]
Dijkstra, E. W., *A Discipline of Programming*, Prentice-Hall, Englewood Cliffs, N.J., 1976.

[DISE77]
diSessa, A., "On Learnable Representations of Knowledge: A Meaning for the Computational Metaphor," Memo MIT/AIM-441, MIT Laboratory for Artificial Intelligence, September 1977.

[DK75]
Davis, R., and J. King, "An Overview of Production Systems," Memo AIM-271, Stanford Artificial Intelligence Laboratory, 1975.

[DK76]
DeRemer, F., and H. H. Kron, "Programming-in-the-Large vs. Programming-in-the-Small," *IEEE Transactions on Software Engineering,* Vol. SE-2, No. 2, June 1976, pp. 80-86.

[DMN68]
Dahl, O.-J., B. Myrhaug, and K. Nygaard, *Simula 67 Common Base Language,* Pub. S-22, Norwegian Computing Center, Oslo, 1968.

[DN66]
Dahl, O.-J., and K. Nygaard, "SIMULA — An ALGOL-Based Simulation Language," *Communications of the ACM,* Vol. 9, No. 9, September 1966, pp. 671-678.

[DoD78]
Department of Defense, *Steelman Requirements for High Order Computer Programming Languages,* June 1978.

[DoD79]
Department of Defense, *Revised Steelman Requirements for High Order Computer Programming Languages,* 1979.

[DoD80]
Department of Defense, *Requirements for Ada Programming Support Environments: Stoneman,* February 1980.

[DOYL80]
Doyle, J., "A Model for Deliberation, Action and Introspection," Technical Report MIT/AI/TR-581, MIT Laboratory for Artificial Intelligence, 1980.

[DR79]
Davis, A.M., and T.G. Rauscher, "Formal Techniques and Automatic Processing to Ensure Correctness in Requirements Specifications," *Proc. IEEE Conference on Specifications of Reliable Software,* IEEE Catalog No. 79 CH1401-9C, 1979, pp. 15-35.

[DT80]
Deutsch, L.P., and E.A. Taft, "Requirements for and Experimental Programming Environment," Xerox PARC Report CSL-80-10, 1980.

[DV77]
Davis, C.G., and C.R. Vick, "The Software Development System," *IEEE Transactions on Software Engineering,* Vol. SE-3, No. 1, January 1977.

[EARL71]
Earley, J., "Toward an Understanding of Data Structures," *Communications of the ACM,* Vol. 14, No. 10, October 1971, pp. 617-627.

[EGL76]
Eswaren, K.P., J.N. Gray, R.A. Lorie, and I.L. Traiger, "The Notions of Consistency and Predicate Locks in a Database System," *Communications of the ACM*, Vol. 19, No. 11, November 1976.

[ESWA76]
Eswaren, K.P., "Specifications, Implementations and Interactions of a Trigger Subsystem in an Integrated Database System," IBM Research Report RJ1820, San Jose, Calif., August 1976.

[FAHL79]
Fahlman, S.E., *NETL: A System for Representing and Using Real-World Knowledge*, MIT Press, 1979.

[FALK80]
Falkenberg, E.D., "Conceptualization of Data," *Infotech State-of-the-Art Report on Data Design,* Pergamon Infotech Limited, London, 1980.

[FAUS81]
Faust, G., "Semiautomatic Translation of COBOL into HIBOL," M.S. thesis (Technical Report MIT/LCS/TR-256), MIT Laboratory for Computer Science, March 1981.

[FEIG77]
Feigenbaum, E.A., "The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering," *Proc. 5th International Joint Conference on Artificial Intelligence,* Cambridge, Mass., August 1977.

[FIND79]
Findler, N.V., *Associative Networks: Representation and Use of Knowledge by Computer*, Academic Press, New York, 1979.

[FLOY67]
Floyd, R.W., "Assigning Meanings to Programs," in J.T. Schwartz (ed.), *Proc. Symposium in Applied Mathematics,* Vol. 19, American Mathematical Society, 1967, pp. 19-32.

[FN79]
Feiertag, R., and P.G. Neumann, "The Foundations of a Provably Secure Operating System (PSOS)," *Proc. National Computer Conference,* 1979, pp. 329-334.

[FOGG82]
Fogg, D., "Parser Support for Abstract Data Types in INGRES," Masters Report, Univ. of California, Berkeley, September 1982.

[FW76a]
Friedman, D.P., and D.S. Wise, "CONS Should Not Evaluate its Arguments," in *Automata, Languages, and Programming,* Edinburgh Univ. Press, Edinburgh, Scotland, 1976.

[FW76b]
Friedman, D.P., and D.S. Wise, "The Impact of Applicative Programming on Multiprocessing," *Proc. ACM International Conference on Parallel Processing,* 1976, pp. 263-272.

[GANE80]
Ganes, C.P., "Data Design in Structured System Analysis," in P. Freeman and A.I. Wasserman (eds.), *Tutorial on Software Design Techniques,* 1980.

[GBM82]
Greenspan, S., A.T. Borgida, and J. Mylopoulos, "Capturing More World Knowledge in the Requirements Specification," *Proc. 6th International Conference on Software Engineering,* Tokyo, Japan, 1982.

[GERH75]
Gerhart, S.L., "Knowledge About Programs: a Model and Case Study," *Proc. of International Conference on Reliable Software,* June 1975, pp. 88-95.

[GH78]
Guttag, J.V., and J.J. Horning, "The Algebraic Specification of Abstract Data Types," *Acta Informatica,* Vol. 10, 1978, pp. 27-52.

[GH80]
Guttag, J.V., and J.J. Horning, "Formal Specification as a Design Tool," *Proc. ACM Symposium on Principles of Programming Languages,* SIGACT and SIGPLAN, January 1980, pp. 251-261.

[GHM78]
Guttag, J.V., E. Horowitz, and D.R. Musser, "Abstract Data Types and Software Validation," *Communications of the ACM,* Vol. 21, No. 12, December 1978, pp. 1048-1064.

[GKB82]
Gustafsson, M.R., T. Karlsson, and Bubenko, J.A., Jr., "A Declarative Approach to Conceptual Information Modelling," in [OSV82], pp. 93-142.

[GM78]
Gallaire, H., and J. Minker (eds.), *Logic and Data Bases,* Plenum Press, New York, 1978.

[GM80]
Goodenough, J. B., and C. L. McGowan, "Software Quality Assurance: Testing and Validation," *Proc. IEEE,* Vol. 68, No. 9, September 1980.

[GMS77]
Geschke, C. M., J. H. Morris, Jr., and E. H. Satterthwaite, "Early Experience with Mesa," *Communications of the ACM,* Vol. 20, No. 8, August 1977, pp. 540-553.

[GMW79]
Gordon, M. J., A. J. Milner, and C. P. Wadsworth, "Edinburgh LCF," *Lecture Notes in Computer Science,* No. 78, Springer-Verlag, New York, 1979.

[GOLD73]
Goldberg, J., "Proceedings of a Symposium on the High Cost of Software," Technical Report, Stanford Research Institute, Stanford, Calif., September 1973.

[GOOD77]
Good, D. I., "Constructing Verified and Reliable Communications Processing Systems," *Software Engineering Notes,* Vol. 2, No. 5, October 1977, pp. 8-13.

[GP77]
Goldstein, I., and S. Papert, "Artificial Intelligence, Language, and the Study of Knowledge," *Cognitive Science,* Vol. 1, No. 1, 1977.

[GR77]
Goldstein, I., and R. B. Roberts, "NUDGE: A Knowledge-Based Scheduling Program," *Proc. 5th International Joint Conference on Artificial Intelligence,* Cambridge, Mass., August 1977.

[GRAY78]
Gray, J. N., "Notes on Data Base Operating Systems," *Proc. Advanced Course on Operating Systems,* Munich, West Germany, in *Lecture Notes in Computer Science,* No. 60, Springer-Verlag, New York, 1978.

[GREE69a]
Green, C., "The Application of Theorem Proving to Question-Answering Systems," Ph.D. thesis, Dept. of Electrical Engineering, Stanford Univ., 1969.

[GREE69b]
Green, C., "Theorem Proving by Resolution as a Basis for Question-Answering Systems," in D. Michie and B. Meltzer (eds.), *Machine Intelligence 4*, Edinburgh Univ. Press, Edinburgh, Scotland, 1969.

[GRIF76]
Griffiths, P., and B. Wade, "An Authorization Mechanism for a Relational Data Base System," *ACM Transactions on Database Systems,* Vol. 2, No. 3, September 1976.

[GTW78]
Goguen, J. A., J. W. Thatcher, and E. G. Wagner, "An Initial Algebra Approach to the Specification, Correctness, and Implementation of Abstract Data Types," in R. Yeh (ed.), *Current Trends in Programming Methodology,* Vol. IV, Prentice-Hall, Englewood Cliffs, N.J., 1978.

[GUAR78]
Guarino, L. R., "The Evolution of Abstraction in Programming Languages," Technical Report CMU-CS-78-120, Carnegie-Mellon Univ., May 1978.

[GUTT77]
Guttag, J. V., "Abstract Data Types and the Development of Data Structures," *Communications of the ACM,* Vol. 20, No. 6, June 1977, pp. 396-404.

[GUTT80]
Guttag, J. V., "Notes on Type Abstraction (Version 2)," *IEEE Transactions on Software Engineering,* Vol. SE-6, No. 1, January 1980, pp. 13-23.

[GW79]
Gerhart, S. L., and D. S. Wile, "Preliminary Report on the Delta Experiment: Specification and Verification of a Multiple-User File Updating Module," *Proc. IEEE Conference on Specifications of Reliable Software,* IEEE Catalog No. 79 CH1401-9C, 1979, pp. 198-211.

[GY76]
Gerhart, S. L., and L. Yelowitz, "Observations of Fallibility in Applications of Modern Programming Methodologies," *IEEE Transactions on Software Engineering,* Vol. SE-2, No. 5, September 1976, pp. 195-207.

[HABE73]
Habermann, A. N., "Critical Comments on the Programming Language Pascal," *Acta Informatica,* Vol. 3, 1973, pp. 47-57.

[HAL79]
Hewitt C., G. Attardi, and H. Lieberman, "Specifying and Proving Properties of Guardians for Distributed Systems," *Proc. Conference on Semantics of Concurrent Computation,* INRIA, Evian, France, July 1979.

[HARD80]
Hardgrave, W. T., "Positional Set Notation," internal report, National Bureau of Standards, February 1980.

[HAS80]
Hewitt, C., G. Attardi, and M. Simi, "Knowledge Embedding with a Description System," *Proc. 1st AAAI National Conference,* August 1980.

[HAYE74]
Hayes, P. J., "Some Problems and Non-Problems in Representation Theory," *Proc. AISB Summer Conference,* Essex Univ., Essex, Great Britian, 1974.

[HAYE77]
Hayes, P. J., "In Defense of Logic," *Proc. 5th International Joint Conference on Artificial Intelligence,* Cambridge, Mass., 1977, pp. 559-565.

[HAYE78]
Hayes, P. J., "The Ontology of Liquids," unpublished manuscript, 1978.

[HAYE79]
Hayes, P. J., "The Logic of Frames," in D. Metzing (ed.), *Frame Conceptions and Text Understanding,* Walter de Gruyter and Co., Berlin, 1979, pp. 46-61.

[HAZE76]
Hazen, A., "Expressive Completeness in Modal Language," *Journal of Philosophical Logic,* Vol. 5, 1976.

[HB77]
Hewitt, C., and H. Baker, "Laws for Communicating Parallel Processes," *Proc. 1977 IFIP Congress,* 1977.

[HBS73]
Hewitt, C., P. Bishop, and R. Steiger, "A Universal Modular ACTOR Formalism for Artificial Intelligence," *Proc. International Joint Conference on Artificial Intelligence,* Palo Alto, Calif., August 1973.

[HEND75]
Hendrix, G., "Expanding the Utility of Semantic Networks through Partitioning," *Proc. International Joint Conference on Artificial Intelligence,* Tbilisi, USSR, September 1975.

[HENI79]
Heninger, K. L., "Specifying Software Requirements for Complex Systems: New Techniques and Their Applications," *Proc. IEEE Conference on Specifications of Reliable Software,* IEEE Catalog No. 79 CH1401-9C, 1979, pp. 1-14.

[HENK50]
Henkin, L., "Completeness in the Theory of Types," *Journal of Symbolic Logic,* Vol. 15, 1950, pp. 81-91.

[HEWI69]
Hewitt, C. E., "PLANNER: A Language for Proving Theorems in Robots," *Proc. International Joint Conference on Artificial Intelligence,* Washington, D.C., May 1969.

[HEWI71]
Hewitt, C. E., "PLANNER: A Language for Proving Theorems in Robots," *Proc. International Joint Conference on Artificial Intelligence,* London, Great Britian, August 1971.

[HEWI72]
Hewitt, C. E., *Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot,* Ph.D. thesis, Dept. of Mathematics, MIT, 1972.

[HEWI75]
Hewitt, C. E., "How To Use What You Know," *Proc. International Joint Conference on Artificial Intelligence,* Tbilisi, USSR, August 1975.

[HEWI77]
Hewitt, C. E., "Viewing Control Structures as Patterns of Passing Messages," *Artificial Intelligence,* Vol. 8, 1977, pp. 323-364.

[HEWI80]
Hewitt, C. E., "The Apiary Network Architecture for Knowledgeable Systems," *Conference Record of the 1980 Lisp Conference,* Stanford Univ., Stanford, Calif., August 1980.

[HG74]
Hewitt, C. E., and I. Greiff, "Actor Semantics of PLANNER-73," Working Paper No. 81, MIT Laboratory for Artificial Intelligence, 1974.

[HINT62]
Hintikka, J., *Knowledge and Belief: An Introduction to the Logic of the Two Notions,* Cornell Univ. Press, 1962.

[HK81]
Hecht, M. S., and L. Kershberg, "Update Semantics for the Functional Data Model," Data Base Research Report No. 4, Bell Labs, Holmdell, N.J., 1981.

[HL82]
Haskin, R. L., and R. A. Lorie, "On Extending the Functions of a Relational Database System," *Proc. 1982 ACM SIGMOD International Conference on the Management of Data,* Orlando, Fla., 1982, pp. 207-212.

[HM75]
Hammer, M., and D. McLeod, "Semantic Integrity in a Relational Database System," *Proc. 1st International Conference on Very Large Databases,* Framingham, Mass., September 1975, pp. 25-47.

[HM76]
Hammer, M., and D. McLeod, "A Framework for Database Semantic Integrity," *Proc. 2nd International Conference on Software Engineering,* San Francisco, Calif., October 1976.

[HM78]
Hammer, M., and D. McLeod, "The Semantic Data Model: a Modelling Mechanism for Database Applications," *Proc. 1978 ACM SIGMOD International Conference on the Management of Data,* Austin, Texas, May-June 1978.

[HM81]
Hammer, M., and D. McLeod, "Database Description with SDM: A Semantic Database Model," *ACM Transactions on Database Systems,* Vol. 6, No. 3, September 1981.

[HOAR69]
Hoare, C. A. R., "An Axiomatic Basis for Computer Programming," *Communications of the ACM,* Vol. 12, October 1969, pp. 576-580, 583.

[HOAR72a]
Hoare, C. A. R., "Proof of Correctness of Data Representations," *Acta Informatica,* Vol. 1, No. 4, 1972, pp. 271-281.

[HOAR72b]
Hoare, C. A. R., "Notes on Data Structuring," in O.-J. Dahl, E. W. Dijkstra, and C. A. R. Hoare (eds.), *Structured Programming,* Academic Press, New York, 1972, pp. 83-174.

[HOAR78]
Hoare, C. A. R., "Communicating Sequential Processes," *Communications of the ACM,* Vol. 21, No. 8, August 1978.

[HOT76]
Hall, P., J. Owlett, and S. J. P. Todd, "Relations and Entities," in J. M. Nijssen (ed.), *Modelling in Database Management Systems,* Elsevier North-Holland, New York, 1976.

[HOUS77]
Housel, B. C., "A Unified Approach to Data Conversion," *Proc. 3rd International Conference on Very Large Databases,* Tokyo, Japan, 1977.

[HOWD79]
Howden, W. E., "An Analysis of Software Validation Techniques for Scientific Programs," Technical Report DM-171-IR, Dept. of Mathematics, Univ. of Victoria, Victoria, B.C., Canada, March 1979.

[HR79]
Hunt, H. B., and D. J. Rosenkrantz, "The Complexity of Testing Predicate Locks," *Proc. 1979 ACM SIGMOD International Conference on the Management of Data,* Boston, Mass., May 1979.

[HSW75]
Held, G., M. Stonebraker, and E. Wong, "INGRES: A Relational Database System," *Proc. AFIPS 1975 National Computer Conference,* Vol. 44, 1975.

[HW73]
Hoare, C. A. R., and N. Wirth, "An Axiomatic Definition of the Programming Language PASCAL," *Acta Informatica*, Vol. 2, No. 4, 1973, pp. 335-355.

[HWY79]
Housel, B. C., V. Waddle, and S. B. Yao, "The Functional Dependency Model for Logical Database Design," *Proc. 5th International Conference on Very Large Databases,* Rio de Janeiro, Brazil, October 1979.

[HY79]
Hevner, A. R., and S. B. Yao, "Query Processing in Distributed Database Systems," *IEEE Transactions on Software Engineering,* Vol. SE-5, No. 3, 1979.

[IBHK79]
Ichbiah, J. D., J. P. G. Barnes, J.-C. Heliard, B. Krieg-Brueckner, O. Roubine, and B. A. Wichmann, "Rationale for the Design of the Ada Programming Language," *SIGPLAN Notices,* Vol. 14, No. 6, Part B, 1979.

[IEEE79a]
IEEE Computer Society (eds.), *Workshop on Quantitative Software Models for Reliability, Complexity, and Cost: an Assessment of the State of the Art,* IEEE Catalog No. TH0067-9, 1979.

[IEEE79b]
IEEE Computer Society (eds.), *Proc. Conference on Specifications of Reliable Software,* IEEE Catalog No. 79 CH1401-9C, 1979.

[IKWL79]
Ichbiah, J. D., B. Krieg-Brueckner, B. A. Wichmann, H. F. Ledgard, J.-C. Heliard, J.-R. Abrial, J. P. G. Barnes, and O. Roubine, "Preliminary Ada Reference Manual," *SIGPLAN Notices,* Vol. 14, No. 6, Part A, 1979.

[IKWL80]
Ichbiah, J. D., B. Krieg-Brueckner, B. A. Wichmann, H. F. Ledgard, J.-C. Heliard, J.-R. Abrial, J. P. G. Barnes, M. Woodger, O. Roubine, P. N. Hilfinger, and R. Firth, *Reference Manual for the Ada Programming Language: Proposed Standard Document,* Department of Defense, US Government Printing Office 008-000-00354-8, July 1980; also in: *Lecture Notes in Computer Science,* No. 106, Springer-Verlag, New York, 1981.

[IKWL82]
Ichbiah, J. D., B. Krieg-Brueckner, B. A. Wichmann, H. F. Ledgard, J.-C. Heliard, J.-L. Gailly, J.-R. Abrial, J. P. G. Barnes, M. Woodger, O. Roubine, P. N. Hilfinger, and R. Firth, *Reference Manual for the Ada Programming Language,* Draft Revised MIL-STD 1815; Draft Proposed ANSI Standard Document for Editorial Review, US Department of Defense, Honeywell Inc., and Alsys, July 1982; also available from AdaTEC, ACM order no. 825820.

[IMB81]
Islam, N., T. J. Myers, and P. Broome, "A Simple Optimizer for FP-like Languages," *Proc. ACM Conference on Functional Programming and Architecture,* New Hampshire, 1981.

[INGA78]
Ingalls, D. H., "The Smalltalk-76 Programming System: Design and Implementation," *Conference Record of the Fifth Annual ACM Symposium on Programming Languages,* Tucson, Arizona, January 1978.

[IOS79]
International Organization for Standardization, *Draft Specification for the Computer Programming Language Pascal,* ISO/TC 97/SC 5 N, 1979.

[ISRA80]
Israel, D. J., "What's Wrong with Non-Monotonic Logic?," *Proc. 1st National Conference on Artifical Intelligence,* American Association for Artificial Intelligence, Stanford, Calif., 1980, pp. 99-101.

[ISRA82]
Israel, D. J., "On Interpreting Semantic Network Formalisms," *International Journal of Computer Mathematics,* (to appear in a special issue on Computational Linguistics edited by N. Cercone); also available as BBN Report No. 5117, Bolt, Beranek and Newman Inc., Cambridge, Mass., 1982.

[IVER79]
Iverson, K. E., "Operators," *ACM Transactions on Programming Languages and Systems,* Vol. 1, No. 2, 1979.

[JACO82]
Jacobs, B. E., "On Database Logic," *Journal of the ACM,* Vol. 29, No. 2, April 1982, pp. 310-332.

[JL76]
Jones, A. K., and B. H. Liskov, "An Access Control Facility for Programming Languages," MIT Computation Structures Group and Carnegie-Mellon Univ., MIT Memo 137, 1976.

[JS82]
Jarke, M., and J. W. Schmidt, "Query Processing Strategies in the Pascal/R Relational Database Management System," *Proc. 1982 ACM SIGMOD International Conference on the Management of Data,* Orlando, Fla., June 1982.

[JW74]
Jensen, K., and N. Wirth, *Pascal User Manual and Report,* Springer-Verlag, New York, 1974.

[KAHN79]
Kahn, K. M., "Creation of Computer Animation from Story Descriptions," Ph.D. thesis, MIT, 1979.

[KENT78]
Kent, W., *Data and Reality,* Elsevier North-Holland, New York, 1978.

[KH81]
Kornfeld, W. A., and Hewitt, C., "The Scientific Community Metaphor," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-11, No. 1, January 1981.

[KING82]
King, R., "A Semantics-Based Methodology for Database Design and Evolution," Ph.D. thesis (Technical Report), Computer Science Dept., Univ. of Southern California, October 1982.

[KL80a]
Keller, R. M., and G. Lindstrom, "Parallelism in Functional Programming through Applicative Loops," Technical Report, Univ. of Utah, 1980.

[KL80b]
Krieg-Brueckner, B., and D.C. Luckham, "Anna: Towards a Language for Annotating Ada Programs," *SIGPLAN Notices,* Vol. 15, No. 11, 1980, pp. 128-138.

[KLEE71]
Kleene, S.C., *Introduction to Metamathematics,* Elsevier North-Holland, New York, 1971.

[KLVO82]
Krieg-Brueckner, B., D.C. Luckham, F.W. von Henke, and O. Owe, *Anna: a Language for Annotating Ada Programs,* Springer-Verlag, New York (to appear).

[KM82a]
King, R., and D. McLeod, "Semantic Database Models," in S.B. Yao (ed.), *Principles of Database Design,* Prentice-Hall, Englewood Cliffs, N.J. (to appear).

[KM82b]
King, R., and D. McLeod, "The Event Database Specification Model," *Proc. Second International Conference on Databases: Improving Usability and Responsiveness,* Jerusalem, Israel, June 1982.

[KM82c]
King, R., and D. McLeod, "A Methodology and Tool for Database Life-Cycle Management," Technical Report, Computer Science Dept., Univ. of Southern California, November 1982 (submitted for publication).

[KNUT73]
Knuth, D.E., *The Art of Computer Programming, Vol. 1: Fundamental Algorithms,* Second edition, Addison-Wesley, Reading, Mass., 1973.

[KOLA82]
Kolata, G., "How Can Computers Get Common Sense?," *Science,* Vol. 217, No. 4566, September 24, 1982.

[KONO82]
Konolige, K., "Circumscriptive Ignorance," *Proc. AAAI National Conference,* Pittsburgh, Penn., August 1982.

[KORN82]
Kornfeld, W., "Concepts in Parallel Problem Solving," Ph.D. thesis, MIT, 1982.

[KOWA74]
Kowalski, R., "Predicate Logic as a Programming Language," *Proc. IFIP Congress,* 1974, pp. 569-574.

[KOWA78]
Kowalski, R., "Logic for Data Description," in [GM78].

[KOWA79]
Kowalski, R., *Logic for Problem Solving,* Elsevier North-Holland, New York, 1979.

[KP76]
Kernighan, B. W., and P. J. Plauger, *Software Tools,* Addison-Wesley, Reading, Mass., 1976.

[KR81]
Kung, H. T., and J. T. Robinson, "On Optimistic Methods for Concurrency Control," *ACM Transactions on Database Systems,* Vol. 6, No. 2, June 1981.

[KUHN67]
Kuhns, J. L., "Answering Questions by Computer—a Logical Study," Memorandum RM 2428 PR, Rand Corporation, Santa Monica, Calif., December 1967.

[LAND65]
Landin, P. J., "A Correspondence Between ALGOL 60 and Church's Lambda Notation," *Communications of the ACM,* Vol. 8, Nos. 2 and 3, 1965.

[LAND66]
Landin, P. J., "The Next 700 Programming Languages," *Communications of the ACM,* Vol. 9, No. 3, 1966.

[LENA77]
Lenat, D. B., "The Ubiquity of Discovery," *Proc. International Joint Conference on Artificial Intelligence,* Cambridge, Mass., August 1977.

[LEVE81a]
Levesque, H., "A Formal Treatment of Incomplete Knowledge Bases," Ph.D. thesis, Dept. of Computer Science, Univ. of Toronto, 1981; also available as Technical Report No. 3, Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, Calif.

[LEVE81b]
Levesque, H., "The Interaction with Incomplete Knowledge Bases: a Formal Treatment," *Proc. International Joint Conference on Artificial Intelligence*, Univ. of British Columbia, Vancouver, B.C., Canada, 1981.

[LEVI77]
Levin, R., "Program Structures for Exceptional Condition Handling," Ph.D. thesis, Carnegie-Mellon Univ., June 1977.

[LEWI68]
Lewis, D., "Counterpart Theory and Quantified Modal Logic," *Journal of Philosophy,* Vol. 65, 1968, pp. 113-126.

[LGHL78]
London, R. L., J. V. Guttag, J. J. Horning, B. W. Lampson, J. G. Mitchell, and G. J. Popek, "Proof Rules for the Programming Language Euclid," *Acta Informatica,* Vol. 10, No. 1, 1978 pp. 1-26.

[LHLM77]
Lampson, B. W., J. J. Horning, R. L. London, J. G. Mitchell, and G. J. Popek, "Report on the Programming Language Euclid," *SIGPLAN Notices,* Vol. 12, No. 2, February 1977, pp. 1-79.

[LIEB81a]
Lieberman, H., "A Preview of Act-1," AI Memo No. 625, MIT Artificial Intelligence Laboratory, 1981.

[LIEB81b]
Lieberman, H., "Thinking About Lots of Things At Once Without Getting Confused: Parallelism in Act-1," AI Memo No. 626, MIT Artificial Intelligence Laboratory, 1981.

[LIPS78]
Lipski, W., Jr., "On Semantic Issues Connected with Incomplete Information Data Bases," PAS Report 325, Institute of Computer Science, Warsaw, Poland, 1978.

[LIPS79]
Lipski, W., Jr., "On Semantic Issues Connected with Incomplete Information Databases," *ACM Transactions on Database Systems,* Vol. 4, No. 3, September 1979, pp. 262-296.

[LISK77]
Liskov, B., *et al.,* "Abstraction Mechanisms in CLU," *Communications of the ACM,* Vol. 20, No. 8, August 1977, pp. 564-576.

[LM79]
Levesque, H., and J. Mylopoulos, "A Procedural Semantics for Semantic Networks," in [FIND79].

[LMW79]
Linger, R. C., H. D. Mills, and B. I. Witt, *Structured Programming Theory and Practice*, Addison-Wesley, Reading, Mass., 1979.

[LOCK79]
Lockmann, P. *et al.,* "Data Abstractions for Data Base Systems," *ACM Transactions on Database Systems,* Vol. 4, No. 1, March 1979.

[LOND75]
London, R. L., "A View of Program Verification," *Proc. International Conference on Reliable Software,* IEEE Computer Society, April 1975, pp. 534-545.

[LORI77]
Lorie, R. A., "Physical Integrity in a Large Segmented Database," *ACM Transactions on Database Systems,* Vol. 2, No. 1, March 1977.

[LS80]
Lamersdorf, W., and J. W. Schmidt, "Specification of Pascal/R," Report No. 73/74, Fachbereich Informatik, Univ. of Hamburg, July 1980.

[LSAS77]
Liskov B., A. Snyder, R. Atkinson, and C. Schaffert, "Abstraction Mechanism in CLU," *Communications of the ACM,* Vol. 20, No. 8, August 1977, pp. 564-576.

[LW76]
Lorie, R., and B. Wade, "The Compilation of a Very High Level Data Language," IBM Research Report RJ2008, San Jose, Calif., May 1977.

[LZ74]
Liskov, B., and S. Zilles, "Programming With Abstract Data Types," *SIGPLAN Notices,* Vol. 9, No. 4, April 1974, pp. 50-59.

[LZ75]
Liskov, B., and S. Zilles, "Specification Techniques for Data Abstractions," *IEEE Transactions on Software Engineering,* Vol. SE-1, No. 1, March 1975, pp. 7-19.

[LZ77]
Liskov, B., and S. Zilles, "An Introduction to Formal Specifications of Data Abstractions," in R. Yeh (ed.), *Current Trends in Programming Methodology, Vol. I,* Prentice-Hall, Englewood Cliffs, N.J., 1977.

[MANN74]
Manna, Z., *Mathematical Theory of Computation,* McGraw-Hill, New York, 1974.

[MBW78]
Mylopoulos, J., P. A. Bernstein, and H. K. T. Wong "A Preliminary Specification for TAXIS," Technical Report CCA-78-02, Computer Corporation of America, Cambridge, Mass., January 1978.

[MBW80]
Mylopoulos, J., P. A. Bernstein, and H. K. T. Wong, "A Language Facility for Designing Interactive Database-Intensive Applications," *ACM Transactions on Database Systems,* Vol. 5, No. 2, June 1980, pp. 185-207.

[MCAL80]
McAllester, D. A., "An Outlook on Truth Maintenance," Memo MIT/AIM-551, MIT Laboratory for Artificial Intelligence, August 1980.

[MCCA62]
McCarthy, J., *et al., LISP 1.5 Programmer's Manual,* MIT Press, Cambridge, Mass., 1962.

[MCCA80]
McCarthy, J., "Circumscription — A Form of Non-Monotonic Reasoning," *Artificial Intelligence,* Vol. 13, Nos. 1 and 2, April 1980, pp. 27-39.

[MCDE80]
McDermott, D., "Non-Monotonic Logic II: Non-Monotonic Modal Theories," Research Reort No. 174, Dept. of Computer Science, Yale Univ., February 1980.

[MCGE76]
McGee, W. C., "On User Criteria for Data Model Evaluation," *ACM Transactions on Database Systems,* Vol. 1, No. 4, December 1976.

[MD78]
McDermott, D., and J. Doyle, "Non-Monotonic Logic I," Memo MIT/AIM-486, MIT Laboratory for Artificial Intelligence, 1978.

[MEND64]
Mendelson, E., *Introduction to Mathematical Logic,* Van Nostrand, Princeton, N.J., 1964.

[MG77]
Miller, M. L., and I. Goldstein, "Problem Solving Grammars as Formal Tools for Intelligent CAI," *Proc. ACM,* 1977.

[MH69]
McCarthy, J., and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in D. Michie and B. Meltzer (eds.), *Machine Intelligence 4,* Edinburgh Univ. Press, Edinburgh, Scotland, 1969.

[MILL76]
Millen, J. K., "Security Kernel Validation in Practice," *Communications of the ACM,* Vol. 19, No. 5, May 1976, pp. 243-250.

[MILL79]
Miller, E. (ed.), *Tutorial: Automated Tools for Software Engineering,* IEEE Computer Society, IEEE Catalog No. EHO 150-3, 1979.

[MILN78]
Milner, R., "A Theory of Type Polymorphism in Programming," *Journal of Computer and System Sciences,* Vol. 17, 1978, pp. 348-375.

[MINS75]
Minsky, M., "A Framework for Representing Knowledge," in P. Winston (ed.), *The Psychology of Computer Vision,* McGraw-Hill, New York, 1975, pp. 211-277.

[MISR78]
Misra, J., "Some Aspects of the Verification of Loop Computations," *IEEE Transactions on Software Engineering,* Vol. SE-4, No. 6, November 1978, pp. 478-485.

[MN74]
Moore, J. and A. Newell, "How can MERLIN Understand?," in L. Gregg (ed.), *Knowledge and Cognition,* Lawrence Erlbaum Associates, Hillsdale, N.J., 1974.

[MONT74]
Montague, R., "The Proper Treatment of Quantification in Ordinary English," in R. Thomason (ed.), *Formal Philosophy,* Yale Univ. Press, New Haven, 1974, pp. 247-270.

[MOOR77]
Moore, R., "Reasoning About Knowledge and Action," *Proc. 5th International Joint Conference on Artificial Intelligence,* Cambridge, Mass., August 1977.

[MOOR81]
Moore, R., "Reasoning about Knowledge and Action," Technical Note 191, Artificial Intelligence Center, SRI International, Menlo Park, 1980.

[MOOR82]
Moore, R. C., "The Role of Logic in Knowledge Representation and Commonsense Reasoning," *Proc. AAAI National Conference,* Pittsburgh, Penn., August 1982.

[MORR73a]
Morris, J. H., "Types Are Not Sets," *Proc. ACM Symposium on Principles of Programming Languages,* 1973, pp. 120-124.

[MORR73b]
Morris, J. H., "Protection in Programming Languages," *Communications of the ACM,* Vol. 16, January 1973, pp. 15-21.

[MP82]
Manola, F., and A. Pirotte, "CQLF—A Query Language for CODASYL-Type Databases," *Proc. 1982 ACM SIGMOD International Conference on the Management of Data,* Orlando, Fla., 1982.

[MPBR82]
Manola, F., A. Pirotte, B. Blaustein, and D. R. Ries, "Family of Data Model Specifications for DBMS (Database Management System) Standards," Computer Corporation of America, Cambridge, Mass., December 1982; NBS-GCR-82-419, available as NTIS Report PB83-163394.

[MS73]
McDermott, D., and G. J. Sussman, "The Conniver Reference Manual," Memo MIT/AIM-259A, MIT Laboratory for Artificial Intelligence, 1973.

[MS81]
McLeod, D., and J. M. Smith, "Abstraction in Databases," in [BZ81].

[MSHI78]
McCarthy, J., M. Sato, T. Hayashi, and S. Igarashi, "On the Model Theory of Knowledge," Memo AIM-312, Dept. of Computer Science, Stanford Univ., 1978.

[MW80]
Mylopoulos, J., and H. Wong, "Some Features of the TAXIS Data Model," *Proc. 6th International Conference on Very Large Databases,* Montreal, Que., Canada, October 1980.

[MYER81]
Myers, T. J., Ph.D. Dissertation, Univ. of Pennsylvania, 1981.

[MYLO81]
Mylopoulos, J., "An Overview of Knowledge Representation," in [BZ81].

[NEWE62]
Newell, A., "Some Problems of Basic Organization in Problem-Solving Programs," Memorandum RM-3283-PR, Rand Corporation, Santa Monica, Calif., December 1962.

[NEWE80]
Newell, A., "Physical Symbol Systems," *Cognitive Science,* Vol. 4, No. 2, April-June 1980, pp. 135-183.

[NEWE81]
Newell, A., "The Knowledge Level," *Proc. AAAI National Conference,* (presidential address) Stanford, Calif.; reprinted in *AI Magazine,* Vol. 2, No. 2, 1981.

[NG78]
Nicolas, J.M., and H. Gallaire, "Data Base: Theory vs. Interpretation," in [GM78].

[NILS71]
Nilsson, N., *Problem Solving Methods in Artificial Intelligence,* McGraw-Hill, Englewood Cliffs, N.J., 1971.

[NR69]
Naur, P., and B. Randell (eds.), *Software Engineering,* NATO, 1969 (report on a conference sponsored by the NATO Science Committee, Garmisch, West Germany, October 7-11, 1968).

[NS78]
Navathe, S.B., and M. Schkolnick, "View Representation in Logical Database Design," *Proc. 1978 ACM SIGMOD International Conference on the Management of Data,* Austin, Texas, May-June 1978.

[NY78]
Nicolas, J.M., and K. Yazdanian, "Integrity Checking in Deductive Databases," in [GM78].

[ONG82]
Ong, J., "Specification of an ADT Facility for a Relational Data Base System," Masters Report, Univ. of California, Berkeley, September 1982.

[ORGA76]
Organick, E.I. (chrm.), *Proc. of Conference on Data: Abstraction, Definition, and Structure, SIGPLAN Notices,* Vol. 11, No. 2, 1976.

[OSV82]
Olle, T.W., H.G. Sol, and A.A. Verjn-Stuart, *Information Systems Design Methodologies: A Comparative Review* (Proc. IFIP TC 8 Working Conference on Comparative Review of Information Systems Design Methodologies, Noordwijkerhout, Netherlands, May 1982), Elsevier North-Holland, Amsterdam, The Netherlands, 1982.

[OVER81]
Overmeyer, R., "A Time Expert for INGRES," M.S. thesis, Univ. of California, Berkeley, August 1981.

[PARN71]
Parnas, D.L., "Information Distribution Aspects of Design Methodology," *Proc. of IFIP Congress,* Booklet TA-3, 1971, pp. 26-30.

[PARN72a]
Parnas, D. L., "A Technique for Software Module Specification with Examples," *Communications of the ACM,* Vol. 15, May 1972, pp. 330-336.

[PARN72b]
Parnas, D. L., "On the Criteria to be Used in Decomposing Systems into Modules," *Communications of the ACM,* Vol. 15, No. 12, December 1972.

[PETE80]
Peters, L., "Software Design Engineering," *Proc. IEEE,* Vol. 68, No. 9, September 1980.

[POPL73]
Pople, H. E., "On the Mechanization of Abductive Logic," *Proc. International Joint Conference on Artificial Intelligence,* Palo Alto, Calif., August 1973.

[POWE82]
Powell, M., private communication.

[QUIL68]
Quillian, M. R. "Semantic Memory," in M. Minsky (ed.), *Semantic Information Processing,* MIT Press, 1968.

[RAMS79]
Ramshaw, L. H., "Formalizing the Analysis of Algorithms," Ph.D. thesis, Stanford Univ., 1979.

[RAPH71]
Raphael, B., "The Frame Problem in Problem-Solving systems," in N. V. Findler and B. Meltzer (eds.), *Artificial Intelligence and Heuristic Programming,* Edinburgh Univ. Press, Edinburgh, Scotland, 1971.

[RB82a]
Ridjanovic, D., and M. L. Brodie, "Semantic Data Model-Driven Design, Specification and Verification of Interactive Database Transactions," Computer Corporation of America, Cambridge, Mass., April 1982.

[RB82b]
Ridjanovic, D., and M. L. Brodie, "Defining Database Dynamics with Attribute Grammars," *Information Processing Letters,* Vol. 14, No. 3, May 1982.

[RB82c]
Ridjanovic, D., and M. L. Brodie, "Definition of Fundamental Concepts and Tools for Semantic Modelling of Data and Associated Operations," submitted for publication.

[RB82d]
Ridjanovic, D., and M. L. Brodie, "Disciplined Methodology for Database Transaction Design," submitted for publication.

[RB82e]
Ridjanovic, D., and M.L. Brodie, "Functional Specification and Implementation Verification of Database Transactions," submitted for publication.

[RB82f]
Ridjanovic, D., and M.L. Brodie, "Conceptual Modelling of Office Procedures," submitted for publication.

[RB83]
Ridjanovic, D., and M.L. Brodie, "Action and Transaction Skeletons: High Level Language Constructs for Database Transactions," *Proc. 1983 SIGPLAN Conference,* San Francisco, Calif., June 1983.

[REIT77]
Reiter, R., *An Approach to Deductive Question-Answering,* BBN Technical Report 3649, Bolt, Beranek and Newman, Inc., Cambridge, Mass., September 1977.

[REIT78a]
Reiter, R., "Deductive Question-Answering on Relational Databases," in [GM78], pp. 149-177.

[REIT78b]
Reiter, R., "On Closed World Data Bases," in [GM78], pp. 55-76.

[REIT78c]
Reiter, R., "On Reasoning by Default," *Proc. Second TINLAP Conference,* Urbana, Ill., July 1978, pp. 210-218.

[REIT80a]
Reiter, R., "Equality and Domain Closure in First Order Databases," *Journal of the ACM,* Vol. 27, No. 2, 1980, pp. 235-249.

[REIT80b]
Reiter, R., "Databases: A Logical Perspective," in [BZ80], pp. 174-176.

[REIT80c]
Reiter, R., "A Logic for Default Reasoning," *Artificial Intelligence,* Vol. 13, 1980, pp. 81-132.

[REIT81]
Reiter, R., "On Interacting Defaults," *Proc. International Joint Conference on Artificial Intelligence,* Vancouver, B.C., Canada, 1981.

[REIT82]
Reiter, R., "Circumscription Implies Predicate Completion (Sometimes)," *Proc. AAAI National Conference,* Pittsburgh, Penn., August 1982.

[REYN79]
Reynolds, J.C., "Reasoning About Arrays," *Communications of the ACM,* Vol. 22, No. 5, May 1979, pp. 290-298.

[RICH80]
Rich, C., "Inspection Methods in Programming," Ph.D. thesis (Technical Report MIT/AI/TR-604), MIT Laboratory for Artificial Intelligence, December 1980.

[RICH81]
Rich, C., "Multiple Points of View in Modeling Programs," *Proc. Workshop on Data Abstraction, Data Bases and Conceptual Modeling, SIGPLAN Notices,* Vol. 16, No. 1, January 1981, pp. 177-179.

[RICH82]
Rich, C., "Knowledge Representation Languages and Predicate Calculus: How to Have Your Cake and Eat it Too," *Proc. AAAI National Conference,* Pittsburgh, Penn., August 1982.

[ROSC75]
Rosch, E., "Cognitive Representations of Semantic Categories," *Journal of Experimental Psychology: General,* Vol. 104, 1975, pp. 192-233.

[ROSS77]
Ross, D.T., "Structured Analysis (SA): A Language for Communicating Ideas," *IEEE Transactions on Software Engineering,* Vol. SE-3, No. 1, January 1977.

[ROSS82]
Rosser, B.J., "Highlights of the History of the Lambda-Calculus," *Conference Record ACM Symposium on Lisp and Functional Programming,* Plymouth, Mass., August 1982.

[ROUS77]
Roussopoulos, N., "ADD: Algebraic Data Definition," *Proc. 6th Texas Conference on Computing Systems,* Austin, Texas, November 1977.

[ROUT79]
Routley, R., unpublished manuscript, 1980.

[ROWE82]
Rowe, L.A., *et al.,* "A Form Application Development System," *Proc. 1982 ACM SIGMOD International Conference on the Management of Data,* Orlando, Fla., June 1982.

[RS76]
Rich, C., and H.E. Shrobe, "Initial Report On A LISP Programmer's Apprentice," M.S. thesis (Technical Report MIT/AI/TR-354), MIT Laboratory for Artificial Intelligence, December 1976.

[RS79]
Rowe, L.A., and K.A. Schoens, "Data Abstraction, Views and Updates in RIGEL," *Proc. 1979 ACM SIGMOD International Conference on the Management of Data,* Boston, Mass., May 1979, pp. 71-81.

[RS81]
Reimer, M., and J.W. Schmidt, "Transaction Procedures with Relational Parameters," Report No. 45, Institut fuer Informatik, ETH Zurich, October 1981.

[RSW79]
Rich, C., H.E. Shrobe, and R.C. Waters, "An Overview of the Programmer's Apprentice," *Proc. 6th International Joint Conference on Artificial Intelligence,* Tokyo, Japan, August 1979.

[RSWS78]
Rich, C., H.E. Shrobe, R.C. Waters, G.J. Sussman, and C.E. Hewitt, "Programming Viewed as an Engineering Activity," NSF Proposal, (Memo MIT/AIM-459), MIT Laboratory for Artificial Intelligence, January 1978.

[RUTH73]
Ruth, G., "Analysis of Algorithm Implementations," Ph.D. thesis (MIT Project MAC Technical Report 130), 1973.

[SACE74]
Sacerdoti, E.D., "Planning in a Hierarchy of Abstraction Spaces," *Artificial Intelligence,* Vol. 5, No. 2, 1974, pp. 115-135.

[SC75]
Smith, J.M., and P.Y. Chang, "Optimizing the Performance of a Relational Algebra Database Interface," *Communications of the ACM,* Vol. 18, No. 10, 1975.

[SCHM77]
Schmidt, J.W., "Some High Level Language Constructs for Data of Type Relation," *ACM Transactions on Database Systems,* Vol. 2, No. 3, September 1977.

[SCHM78]
Schmidt, J.W., "Type Concepts for Database Definition," *Proc. International Conference on Data Bases,* Haifa, Israel, August 1978.

[SCHM82]
Schmidt, J.W., "Generalized Data Definition and Selection Mechanisms," Fachbereich Informatik, Univ. of Hamburg, 1982 (to appear).

[SCHU71]
Schuman, S.A. (ed.), *Proc. International Symposium on Extensible Languages, SIGPLAN Notices,* Vol. 6, No. 12, December 1971.

[SCHU76a]
Schubert, L.K., "Extending the Expressive Power of Semantic Networks," *Artificial Intelligence,* Vol. 7, No. 2, Summer 1976, pp. 163-198.

[SCHU76b]
Schuman, S.A., "On Generic Functions," in S.A. Schuman (ed.), *New Directions in Algorithmic Languages — 1975,* IRIA, Le Chesnay, France, 1976, pp. 169-192.

[SCHW75]
Schwartz, J.T., "On Programming," Interim Report on the SETL Project, Courant Institute of Mathematical Sciences, New York Univ., June 1975.

[SCOT72]
Scott, D. S., "Lattice Theoretic Models for Various Type-free Calculi," *Proc. 4th International Congress on Logic, Methodology and the Philosophy of Science,* Bucharest, Hungary, 1972.

[SE74]
Stonebraker, M., and E. Wong, "Access Control in a Relational Data Base System by Query Modification," *Proc. ACM Annual Conference,* San Diego, Calif., November 1974.

[SFFH78]
Shaw, M., G. Feldman, R. Fitzgerald, P. Hilfinger, I. Kimura, R. London, J. Rosenberg, and W. A. Wulf, "Validating the Utility of Abstraction Techniques," *Proc. ACM National Conference,* December 1978, pp. 106-110.

[SFL81]
Smith, J.M., S. Fox, and T. Landers, "Reference Manual for ADAPLEX," Technical Report CCA-81-02, Computer Corporation of America, Cambridge, Mass., 1981.

[SGC79]
Schubert, L. K., R. G. Goebel, and N. J. Cercone, "The Structure and Organization of a Semantic Net for Comprehension and Inference," in [FIND79], pp. 121-175.

[SHAP79]
Shapiro, S., "The SNePs Semantic Network Processing System," in [FIND79].

[SHAW79]
Shaw, M., "A Formal System for Specifying and Verifying Program Performance," Technical Report CMU-CS-79-129, Carnegie-Mellon Univ., June 1979.

[SHIP81]
Shipman, D. W., "The Functional Data Model and the Data Language DAPLEX," *ACM Transactions on Database Systems,* Vol. 6, No. 1, March 1981.

[SHM77]
Szolovits, P., L. Hawkinson, and W. A. Martin, "An Overview of OWL, A Language for Knowledge Representation," Technical Memo MIT/LCS/TM-86, MIT Laboratory for Computer Science, 1977.

[SHRO79]
Shrobe, H. E., "Dependency Directed Reasoning for Complex Program Understanding," Ph.D. thesis (Technical Report MIT/AI/TR-503), MIT Laboratory for Artificial Intelligence, April 1979.

[SK77]
Sibley, E. H., and L. Kershberg, "Data Architecture and Data Model Considerations," *Proc. AFIPS National Computer Conference,* Dallas, Texas, 1977.

[SK80a]
Silberschatz, A., and Z. Kedem, "Consistency in Hierarchical Database Systems," *Journal of the ACM,* Vol. 27, No. 1, January 1980.

[SK80b]
Stonebraker, M., and K. Keller, "Embedding Hypothetical Data Bases and Expert Knowledge in a Data Manager," *Proc. 1980 ACM SIGMOD International Conference on the Management of Data,* Santa Monica, Calif., May 1980.

[SL79]
Su, S. Y. W., and D. H. Lo, "A Semantic Association Model for Conceptual Database Design," *Proc. International Conference on the Entity-Relationship Approach to Systems Analysis and Design,* Los Angeles, Calif., December 1979.

[SM72]
Sussman, G. J., and D. McDermott, "Why Conniving is Better Than Planning," Memo MIT/AIM-255A, MIT Laboratory for Artificial Intelligence, 1972.

[SM80]
Schmidt, J. W., and M. Mall, "Pascal/R Report," Report No. 66, Fachbereich Informatik, Univ. of Hamburg, January 1980.

[SMIT82]
Smith, B. C., "Reflection and Semantics in a Procedural Language," Technical Report MIT/LCS/TR-272, MIT Laboratory for Computer Science, May 1982.

[SOWA76]
Sowa, J. F., "Conceptual Structures for a Database Interface," *IBM Journal of Research and Development,* Vol. 20, No. 4, July 1976, pp. 336-357.

[SS75]
Schmid, H. A., and J. R. Swenson, "On the Semantics of the Relational Data Model," *Proc. 1975 ACM SIGMOD International Conference on the Management of Data,* San Jose, Calif., June 1975.

[SS77a]
Smith, J. M., and D. C. P. Smith, "Database Abstractions: Aggregation," *Communications of the ACM,* Vol. 20, No. 6, June 1977.

[SS77b]
Smith, J. M., and D. C. P. Smith, "Database Abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems,* Vol. 2, No. 2, June 1977, pp. 105-133

[SS78a]
Smith, J. M., and D. C. P. Smith, "Principles of Conceptual Database Design," *Proc. NYU Symposium on Database Design,* New York, May 1978.

[SS78b]
Steele, G. L., Jr., and G. J. Sussman, "The Art of the Interpreter, or, The Modularity Complex (Parts Zero, One, and Two)," Memo MIT/AIM-453, MIT Laboratory for Artificial Intelligence, May 1978.

[SS78c]
Steele, G. L., Jr., G. J. Sussman, "The Revised Report on SCHEME: A Dialect of LISP," Memo MIT/AIM-452, MIT Laboratory for Artificial Intelligence, January 1978.

[SS79]
Smith, J. M., and D. C. P. Smith, "A Database Approach to Software Specification," Technical Report CCA-79-17, Computer Corporation of America, Cambridge, Mass., April 1979.

[STAN67]
Standish, T. A., "A Data Definition Facility for Programming Languages," Ph.D. thesis, Dept. of Computer Science, Carnegie-Mellon Univ., 1967.

[STEE78]
Steele, G. L., "Rabbit: A Compiler for Scheme (A Study in Compiler Optimization)," Technical Report MIT/AI/TR-474, MIT Laboratory for Artificial Intelligence, May 1978.

[STON75]
Stonebraker, M., "Implementation of Integrity Constraints and Views by Query Modification," *Proc. 1975 ACM SIGMOD International Conference on the Management of Data,* San Jose, Calif., June 1975.

[STON76]
Stonebraker, M., *et al.,* "The Design and Implementation of INGRES," *ACM Transactions on Database Systems,* Vol. 2, No. 3, September 1976.

[STON80]
Stonebraker, M., "Retrospection on a Database System," *ACM Transactions on Database Systems,* Vol. 5, No. 2, June 1980, pp. 225-240.

[STON82a]
Stonebraker, M., *et al.,* "Document Processing in a Relational Database System," Report M82/20, Electronic Research Laboratory, Univ. of California, Berkeley, June 1982.

[STON82b]
Stonebraker, M., *et al.,* "A Rules System for a Relational Database System," *Proc. 2nd International Conference on Databases,* Jerusalem, Israel, June 1982.

[SUNA78]
Sunagren, B., "Database Design in Theory and Practice," *Proc. 4th International Conference on Very Large Databases*, West Berlin, September 1978.

[SUSS75]
Sussman, G. J., *A Computer Model of Skill Acquisition,* MIT Press, 1975.

[SUSS78]
Sussman, G. J., "Slices at the Boundary Between Analysis and Synthesis," in J.-C. Latombe (ed.), *Artificial Intelligence and Pattern Recognition in Computer Aided Design,* Elsevier North-Holland, New York, 1978.

[SW80]
Shaw, M., and W. A. Wulf, "Toward Relaxing Assumptions in Languages and Their Implementations," *SIGPLAN Notices,* Vol. 15, No. 3, March 1980, pp. 45-61.

[SW82]
Schneider, H.-S., and A. I. Wasserman, *Automated Tools for Information Systems Design* (Proc. IFIP WG 8.1 Working Conference on Automated Tools for Information Systems Design and Development, New Orleans, La., January 1982), Elsevier North-Holland, New York, 1982.

[SWC70]
Sussman, G. J., T. Winograd, and E. Charniak, "MICRO-PLANNER Reference Manual," Memo MIT/AIM-203, MIT Laboratory for Artificial Intelligence, 1970.

[SWL77]
Shaw, M., W. A. Wulf, and R. L. London, "Abstraction and Verification in Alphard: Defining and Specifying Iteration and Generators," *Communications of the ACM,* Vol. 20, No. 8, August 1977.

[TARK56]
Tarski, A., *Logic, Semantics, Metamathematics,* Oxford Univ. Press, 1956.

[TF76]
Taylor, D. C., and R. L. Frank, "CODASYL Database Management Systems," *Computing Surveys,* Vol. 8, No. 1, March 1976.

[TF80]
Teorey, T. J., and J. P. Fry, "The Logical Record Access Approach to Database Design," *Computing Surveys,* Vol. 12, No. 2, June 1980.

[TH77]
Teichroew, D., and E. A. Hershey, "PSA/PSL: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems," *IEEE Transactions on Software Engineering,* Vol. SE-3, No. 1, January 1977.

[THER82]
Theriault, D., "A Primer for the Act-1 Language," Memo MIT/AIM-672, MIT Laboratory for Artificial Intelligence, April 1982.

[TK78]
Tsichritzis, D., and A. Klug, "The ANSI/X3/SPARC DBMS Framework," *Information Systems,* Vol. 3, No. 4, 1978.

[TL76]
Tsichritzis, D., and F. Lochovsky, "Hierarchical Database Management: A Survey," *Computing Surveys,* Vol 8, No. 1, March 1976.

[TL82]
Tsichritzis, D., and F. Lochovsky, *Data Models,* Prentice-Hall, Englewood Cliffs, N.J., 1982.

[TSIC82]
Tsichritzis, D., "Form Management," *Communications of the ACM,* Vol. 25, No. 7, July 1982.

[TURI37]
Turing, A. M., "Computability and Lambda-Definability," *Journal of Symbolic Logic,* Vol. 2, 1937, pp. 153-163.

[TURN79]
Turner, D. A., "A New Implementation Technique for Applicative Languages," *Software—Practice & Experience,* Vol. 9, No.1, 1979.

[TURN81a]
Turner, D. A., "The Semantic Elegance of Applicative Languages," *Proc. ACM Conference on Functional Programming and Archictecture,* New Hampshire, 1981.

[TURN81b]
Turner, R., "Montague Semantics, Nominalization, and Scott's Domains," unpublished manuscript, 1981.

[TWW78]
Thatcher, J. W., E. G. Wagner, and J. B. Wright, "Data Type Specifications: Parameterization and Power of Specification Techniques," *Proc. SIGACT 10th Symposium on Theory of Computing,* May 1978, pp. 119-132.

[ULLM80]
Ullman, J. D., *Principles of Database Systems,* Computer Science Press, Potomac, Maryland, 1980.

[VASS79]
Vassiliou, Y., "Null Values in Database Management: A Denotational Semantics Approach," *Proc. 1979 ACM SIGMOD International Conference on Management of Data,* Boston, Mass., May 1979, pp. 162-169.

[VASS80]
Vassiliou, Y., "A Formal Treatment of Imperfect Information in Database Management," Ph.D. thesis, Dept. of Computer Science, Univ. of Toronto, 1980.

[VMPK75]
Van Wijngaarden, A., B. J. Maiuoux, J. E. L. Peck, C. H. A. Koster, C. M. Sintzoff, C. H. Lindsay, L. G. L. T. Meertens, and R. G. Fisker, "Revised Report on the Algorithmic Language Algol 68," *Acta Informatica,* Vol. 5, 1975, pp 1-236.

[WALD81]
Wadler, P., "Applicative Style Programming, Program Transformation and List Operators," *Proc. ACM Conference on Functional Programming and Architecture,* New Hampshire, 1981.

[WALK80]
Walker, A., "Time and Space in a Lattice of Universal Relations with Blank Entries," *XP1 Workshop on Relational Database Theory,* Stony Brook, N.Y., June-July 1980.

[WASS77]
Wasserman, A. I., "Procedure-Oriented Exception-Handling," Technical Report 27, Medical Information Science, Univ. of California, San Francisco, February 1977.

[WASS79]
Wasserman, A. I., "The Data Management Facilities of PLAIN," *Proc. 1979 ACM SIGMOD International Conference on the Management of Data,* Boston, Mass., May 1979.

[WATE78]
Waters, R. C., "Automatic Analysis of the Logical Structure of Programs," Ph.D. thesis (Technical Report MIT/AI/TR-492), MIT Laboratory for Artificial Intelligence, December 1978.

[WATE79]
Waters, R. C., "A Method for Analyzing Loop Programs," *IEEE Transactions on Software Engineering,* Vol. SE-5, No. 3, May 1979, pp. 237-247.

[WATE81]
Waters, R. C., "The Programmer's Apprentice: Knowledge Based Program Editing," *IEEE Transactions on Software Engineering,* Vol. SE-8, No. 1, January 1982.

[WB81]
Wirsing, M., and M. Broy, "An Analysis of Semantic Models for Algebraic Specifications," *International Summer School on the Theoretical Foundations of Programming Methodology,* Marktoberdorf, 1981.

[WE79]
Wiederhold, G., and R. El-Masri, "The Structural Model for Database Design," *Proc. International Conference on the Entity-Relationship Approach to Systems Analysis and Design,* Los Angeles, Calif., December 1979.

[WEYH80]
Weyhrauch, R. W., "Prolegomena to a Theory of Mechanized Formal Reasoning," *Artificial Intelligence,* Vol. 13, Nos. 1 and 2, April 1980, pp. 133-170.

[WH78]
Waterman, D. A., and F. Hayes-Roth (eds.), *Pattern-Directed Inference Systems,* Academic Press, New York, 1978.

[WILS75]
Wilson, M. L., "The Information Automata Approach to Design and Implementation of Computer-Based Systems," Technical Report FSD76-0093, IBM Federal Systems Division, Gaithersburg, Md., 1975.

[WINO72]
Winograd, T., *Understanding Natural Language,* Academic Press, New York, 1972.

[WINO73]
Winograd, T., "Breaking the Complexity Barrier (Again)," *Proc. SIGIR-SIGPLAN Interface Meeting,* November 1973.

[WINO75]
Winograd, T., "Frame Representation and the Declarative-Procedural Controversy," in [BC75].

[WIRS82]
Wirsing, M., "Structured Algebraic Specifications," *Proc. AFCET Symposium for Computer Science*, Paris, France, March 1982.

[WIRT71]
Wirth, N., "Program Development by Stepwise Refinement," *Communications of the ACM,* Vol. 14, No. 4, April 1971, pp. 221-227.

[WIRT73]
Wirth, N., *Systematic Programming, An Introduction,* Prentice-Hall, Englewood Cliffs, N.J., 1973.

[WIRT77]
Wirth, N., "Modula: A Language for Modular Programming," *Software—Practice & Experience,* Vol. 7, No. 1, January 1977, pp. 3-35.

[WKP80]
Walker, B.J., R.A. Kemmerer, and G.J. Popek, "Specification and Verificaton of the UCLA Security Kernel," *Communications of the ACM,* Vol. 23, No. 2, February 1980, pp. 118-131.

[WLGL78]
Wensley, J.H., L. Lamport, M.W. Green, K.N. Levitt, P.M. Melliar-Smith, R.E. Shostak, and C.B. Weinstock, "SIFT: Design and Analysis of a Fault-Tolerant Computer for Aircraft Control," *Proc. IEEE,* Vol. 66, No. 10, October 1978, pp. 1240-1255.

[WLS76]
Wulf, W.A., R.L. London, and M. Shaw, "An Introduction to the Construction and Verification of Alphard Programs," *IEEE Transactions on Software Engineering,* Vol. SE-2, No. 4, December 1976, pp. 253-265.

[WM77]
Wong, H.K.T., and J. Mylopoulos, "Two Views of Data Semantics: Data Models in Artificial Intelligence and Database Management," *INFOR,* Vol. 15, No. 3, 1977.

[WM81]
Weinreb, D., and D. Moon, "LISP Machine Manual," MIT Laboratory for Artificial Intelligence, March 1981.

[WONG81]
Wong, H. K. T., "Design and Verification of Interactive Information Systems Using TAXIS," Technical Report CSRG-129, CSRG, Univ. of Toronto, April 1981.

[WOOD75]
Woods, W. A., "What's in a Link: Foundations for Semantic Networks," in [BC75], pp. 35-82

[WPPD83]
Wirsing, M., P. Pepper, H. Partsch, W. Dosch, and M. Broy, "On Hierarchies of Abstract Data Types," *Acta Informatica,* 1983 (to appear).

[WS73]
Wulf, W. A., and M. Shaw, "Global Variables Considered Harmful," *SIGPLAN Notices,* Vol. 8, No. 2, February 1973, pp. 28-34.

[WSH77]
Welsh, J., M. J. Sneeringer, and C. A. R. Hoare, "Ambiguities and Insecurities in PASCAL," *Software — Practice & Experience,* Vol. 7, No. 6, 1977, pp. 685-696.

[WSHF81]
Wulf, W. A., M. Shaw, P. N. Hilfinger, and L. Flon, *Fundamental Structures of Computer Science,* Addison-Wesley, Reading, Mass., 1981.

[YANN82]
Yannakakis, M., "A Theory of Safe Locking Policies in Database Systems," *Journal of the ACM,* Vol. 29, No. 3, July 1982.

[YZ80]
Yeh, R. T., and P. Zave, "Specifying Software Requirements," *Proc. IEEE,* Vol. 68, No. 9, September 1980.

[ZANI77]
Zaniolo, C., "Relational Views in a Database System; Support for Queries," *Proc. IEEE Computer Applications and Software Conference,* Chicago, Ill., November 1977, pp. 267-275.

[ZILL80]
Zilles, S. N., "An Introduction to Data Algebras," in D. Bjoerner (ed.), *Abstract Software Specifications, Lecture Notes in Computer Science,* No. 86, Springer-Verlag, New York, 1980, pp. 248-272.

[ZISM78]
Zisman, M., "Use of Production Systems for Modelling Asynchronous Parallel Processes," in [WH78].

[ZLT82]
Zilles, S. N., P. Lucas, and J. W. Thatcher, "A Look at Algebraic Specification," submitted for publication.