Book Reviews

Knowledge Representation: Logical, Philosophical, and Computational Foundations

John F. Sowa

Pacific Grove, CA: Brooks/Cole, 2000,
xiv+594 pp; hardbound, ISBN
0-534-94965-7, $67.95

Reviewed by
Stuart C. Shapiro
University at Buffalo, The State University of New York

For Sowa, "Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domain" (p. xii). "This book is a general textbook of knowledge-base analysis and design, intended for anyone whose job is to analyze knowledge about the real world and map it to a computable form" (p. xi). From these statements, one may gather that Sowa takes knowledge representation to be a broader topic than a subarea of artificial intelligence, and, indeed, he says, "AI design techniques have converged with techniques from other fields, especially database and object-oriented systems," (p. xi) and he lists the "major knowledge representations" to be discussed as "rules, frames, semantic networks, object-oriented languages, Prolog, Java, SQL, Petri nets, and the Knowledge Interchange Format (KIF)" (p. xii), a broader list than most knowledge representation authors would employ. These are mostly discussed rather briefly. The major notations used throughout the book are predicate calculus and conceptual graphs. "Conceptual graphs are a two-dimensional form of logic that is based on the semantic networks of AI and the logical graphs of C. S. Peirce. Both notations are exactly equivalent in their semantics [more about this later in this review], and instructors may choose to use either or both in lectures and exercises" (p. xii). However, any instructor who does not like conceptual graphs and tries to ignore them will have a hard time fighting the proselytizing attitude of the book.

The field of knowledge representation is usually called "knowledge representation and reasoning," because knowledge representation formalisms are useless without the ability to reason with them. Sowa acknowledges this, but gives much less attention to reasoning: "Although the focus of this book is on representation rather than reasoning, the choice of representation can have a major effect on the way the reasoning is carried out and on its ultimate success or failure" (p. 245).

This book is "intended for anyone whose job is to analyze knowledge," and practitioners will find it useful. However, it is also designed for the student, and includes an extensive set of exercises at the end of each chapter and answers to selected exercises at the end of the book. Appendix C, "Extended Examples," contains several-page descriptions of several example applications that could be used as the specifications of longer projects.

The rest of this review will discuss the book chapter by chapter, with appendices interspersed.

The first chapter is an introduction to logic. Sowa shows himself as probably the most scholarly person writing about knowledge representation, in that he ties current issues and techniques to their precedents in the work of historical figures in philosophy and logic. In Section 1.1, he sketches the history of logic through Plato, Aristotle, Porphyry, the Scholastics, Lull, Leibniz, Boole, Peirce, Frege, Schröder, Peano, and Russell. Even readers familiar with artificial intelligence may be surprised to learn that:

- "Aristotle's method of defining new categories by *genus* and *differentiae* is fundamental to AI systems, to object-oriented systems, and to every dictionary from the earliest days to the present" (p. 4).
- Leibniz's "definition of modality in terms of possible worlds is still used today in the semantics for *modal logic*" (p. 7).
- "Leibniz saw that accounting machines could also be used for mechanical reasoning—an insight that should entitle him to be called the grandfather of artificial intelligence" (p. 8).
- "Modern systems of predicate calculus are based on the algebraic notation developed by C. S. Peirce in 1883" (p. 10).
- "Peano began the practice of turning letters upside-down or backward to represent the logical symbols ... Bertrand Russell adopted this notation from Peano for the *Principia Mathematica*" (p. 11).

Then he suggests that "readers who have not had a course in logic or who would like a quick review should read [Appendix A.1]" (p. 11). However, this suggestion is probably premature; reading Appendix A.1 is probably best left for after all of Chapter 1 has been read.

The title of Section 1.2 is "Representing knowledge in logic" (pp. 11–18). This phrase and others like it, such as "Logic itself is a simple language" (p. 15) and "Logic is an *ontologically neutral* notation" (p. 16, italics in the original), have always bothered me. To me, logic is not a particular knowledge representation language, but is the study of correct reasoning. There are many systems of logic, each of which may be called *a* logic, and knowledge representation research may be viewed as a search for an appropriate logic to underlie commonsense reasoning. I am sure that Sowa does not actually disagree with this. Section 1.3 is titled "Varieties of logic" (pp. 18–29), and at the beginning of Section 1.5, he says, "many notations for logic have been invented over the years ... To be a logic, a knowledge representation language must have four essential features: Vocabulary ... Syntax ... Semantics ... Rules of inference" (pp. 39–40). The logics and variants of logics discussed in Chapter 1 include typed (sometimes called "sorted") logics, lambda calculus, modal logic, higher-order logic, KIF, and conceptual graphs. There are also discussions of important issues in logical representation, including the choice of predicates, expressing definitions, object- vs. meta-language, names, types, measures, and the unique-naming convention. An unusual application that is first discussed in this chapter, and then reappears throughout the book, is the representation of a musical piece.

Appendix A supplements Chapter 1 by providing more complete introductions to propositional and predicate logic (Appendix A.1), conceptual graphs (Appendix A.2), and the Knowledge Interchange Format (Appendix A.3).

Appendix A.1, "Predicate Calculus" (pp. 467–476), is a review of propositional and predicate logic that starts at the very basics, such as the truth tables for conjunction, disjunction, negation, material implication, and equivalence, but uses the terms "theorem" and "proof" without defining them, and even though Sowa had said that "to be a

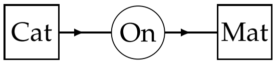

Figure 1
Conceptual graph for *A cat is on a mat* (p. 477).

logic, a knowledge representation language must have ... Semantics" (p. 39), he does not give the semantics of predicate logic, beyond the truth tables of the propositional connectives, either in Chapter 1 or in Appendix A.1. Sowa's choice of notations prefigures the techniques of conceptual graphs. For example, he presents a typed predicate logic, and both the "exactly-one quantifier" (∃!) and the "unique existential quantifier" (∃!!) (I did not know about the latter—it looks useful). Sowa mentions modal logic in Appendix A.1 without discussing it. He does introduce it in Section 1.3, which is one reason Appendix A.1 shouldn't be read right after reading Section 1.1, when Sowa suggests it.

Appendix A.2 (pp. 476–489) is an introduction to conceptual graphs. Again, I was disappointed to find this organized by a set of ten definitions, instead of by a specification of vocabulary, syntax, semantics, and rules of inference. The first definition is, "A conceptual graph *g* is a bipartite graph that has two kinds of nodes called concepts and conceptual relations" (p. 477). A simple example of a conceptual graph is shown in Figure 1. In the official linear notation, this conceptual graph is written as

`[Cat] → (On) → [Mat]`

In either notation, `[Cat]` and `[Mat]` are concepts, and `(On)` is a "conceptual relation." The use of the term *conceptual relation* is not fully justified. When I first introduced the distinction between conceptual and structural relations (Shapiro 1971), the idea was that structural relations were represented by arcs, and conceptual relations were conceptual entities in their own right, were represented by nodes, and could participate in relationships with other conceptual entities. It is true that the "conceptual relations" of conceptual graphs are nodes rather than arcs, but since conceptual graphs are bipartite, "there are no arcs that link ... relations to relations" (p. 478), and so conceptual graphs cannot represent information *about* so-called conceptual relations.

The concepts `[Cat]` and `[Mat]` are the simplest variety of concept: "Every concept has a *type t* and a *referent r* ... In the concept `[Bus]`, 'Bus' is the type, and the referent is a blank, ... In the concept `[Person: John]`, 'Person' is the type, and the referent 'John' is the name of some person" (p. 478, italics in the original).

Not only are conceptual graphs not defined by a specification of vocabulary, syntax, semantics, and rules of inference, they do not have their own independent semantics at all![1] For example, I would expect the concept `[Cat]` to denote some cat, but instead we read, "The concept `[Cat]` by itself simply means *There is a cat*" (p. 484, italics in the original). The real significance of conceptual graphs is that they are a notational variant of standard predicate logic: "There is a *formula operator* ϕ, which translates conceptual graphs to formulas in predicate calculus ... For Figure [1], ϕ generates the following formula: $(\exists x\colon \mathrm{Cat})(\exists y\colon \mathrm{Mat})\mathrm{on}(x,y)$" (p. 476–477, italics in the original). However, there are still problems. Figure 2 (in which the referents ∀ and @1 are quantifiers) is said to be the conceptual graph for the sentence *Every employee*

---

[1] I said this to Sowa in person. He agreed and said that I could say in this review that he agreed.

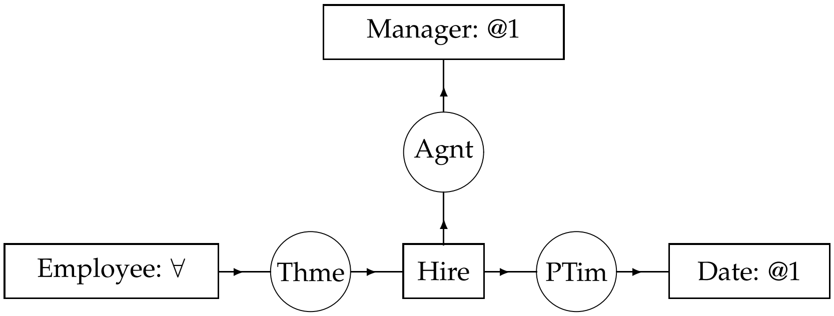

Figure 2
A conceptual graph for *Every employee is hired by some manager on some date* (p. 456).

is hired by some manager on some date, but there is nothing in the conceptual graph to indicate the scopes of the quantifiers. The quantifier structure could be any of these:

$$(\forall x : \mathit{Employee})(\exists! y : \mathit{Manager})(\exists! z : \mathit{Date}),$$

$$(\exists! y : \mathit{Manager})(\forall x : \mathit{Employee})(\exists! z : \mathit{Date}),$$

$$(\exists! z : \mathit{Date})(\forall x : \mathit{Employee})(\exists! y : \mathit{Manager}),$$

$$(\exists! y : \mathit{Manager})(\exists! z : \mathit{Date})(\forall x : \mathit{Employee}).$$

The basic text on conceptual graphs is Sowa (1984). The semantics of conceptual graphs remains a topic of the current literature. For example, Mineau (2000) presents an extensional semantics of the fragment of conceptual graphs that contains neither nested graphs nor negation. For the complete current word on conceptual graphs, see the author's Web site, http://www.bestweb.net/∼sowa/cg/.

Appendix A.3 (pp. 489–491) is a very short introduction to the Knowledge Interchange Format, KIF, mostly by means of five example English sentences that are translated into conceptual graphs, predicate logic, and KIF. KIF is a machine-readable version of predicate logic designed for sharing knowledge bases among programs. Its principal reference is at http://logic.stanford.edu/kif/.

Appendix A is supplemented by an index of special symbols following the name index and the subject index. The format of the entries in the special-symbol index is: English gloss; symbol; pages where the symbol is discussed or used. This index is very useful, but, unfortunately, incomplete.

Chapter 2 (pp. 51–131) is about ontology: "Ontology . . . is the study of existence, of all the kinds of entities—abstract and concrete—that make up the world . . . The two sources of ontological categories are observation and reasoning . . . A choice of ontological categories is the first step in designing a database, a knowledge base, or an object-oriented system" (p. 51). As he did for logic, Sowa introduces the topic of ontology through historical sources, including Heraclitus, Aristotle, Kant, Hegel, Peirce, Husserl, Whitehead, and Heidegger. He then goes on to develop an ontology illustrated by trees, multidimensional matrices (as in Figure 3), and lattices. There is very little use of knowledge representation formalisms, just an occasional conceptual graph, some algebra, some set theory (which is introduced in the chapter, including the definition of such basic terms as *subset*, *union*, and *intersection*), and some predicate logic. There is, however, an extensive discussion of the categories and the distinctions contained in the ontology. This is extremely worthwhile, and includes discussions of roles and adjectives ("happy physicist" vs. "nuclear physicist" vs. "former senator"

	Physical		Abstract	
	Continuant	*Occurrent*	*Continuant*	*Occurrent*
Independent	Object	Process	Schema	Script
Relative	Juncture	Participation	Description	History
Mediating	Structure	Situation	Reason	Purpose

Figure 3
Three-dimensional matrix of twelve of Sowa's categories (p. 75).

vs. "alleged thief" [p. 82]); the related, but different, terms *set, collection, type*, and *category*; mereology (the study of parts and wholes); space and time; and granularity. The historical philosophers and their terminology are cited and used much more than in other knowledge representation texts, which is good or annoying, depending on the reader's attitude.

Chapter 2 is summarized in Appendix B, "Sample Ontology" (pp. 492–512), with diagrams, English explanations of the categories, and many examples of English sentences with conceptual graph representations. This latter style is used especially in a list of nineteen thematic roles, which are used and mentioned elsewhere in the book, but discussed in detail only here in Appendix B.4.

Chapter 3, titled "Knowledge Representations" (pp. 132–205), begins with a section titled "Knowledge Engineering." "Knowledge engineering is the application of logic and ontology to the task of building computable models of some domain for some purpose" (p. 132). But the section is basically an introduction to knowledge representation organized by the five principles of Davis, Schrobe, and Szolovits (1993): a knowledge representation is: (1) "a surrogate"; (2) "a set of ontological commitments"; (3) "a fragmentary theory of intelligent reasoning"; (4) "a medium for efficient computation"; (5) "a medium of human expression" (p. 134). These principles are discussed using a traffic-light example (if the auto-switch is on, it goes back and forth between being red for r units of time and green for g units of time) in several knowledge representation notations, including CLIPS (a production system), an imperative-programming pseudo-language, typed predicate logic, Prolog, and conceptual graphs. Sowa mischaracterizes forward-chaining systems as being equivalent to production systems: "the *forward-chaining* systems are also [like backward-chaining systems] based on logic, but they have procedural aspects that resemble the programming loop" (pp. 138–139, italics in the original), followed by a CLIPS example. Later, he continues the confusion with a different, though also incorrect, characterization: "Repeated execution of modus ponens is called *forward chaining*, and repeated modus tollens is called *backward chaining*" (p. 156, italics in the original). (For a discussion of forward vs. backward chaining, and the related distinctions of bottom-up vs. top-down, and data-driven vs. goal-directed, see Shapiro [1987].)

Section 3.2 is titled "Representing Structure in Frames" (pp. 143–156), but, after historical notes about frames and a subsection on mapping frames to conceptual graphs (titled "Mapping Frames to Logic" (pp. 147–149)), the discussion switches to description logics (though not by that now-commonly-used name) of the KL-ONE family, without distinguishing the two approaches. What I take to be the defining feature of frames, procedural attachment, is only mentioned once, briefly, without adequate discussion or example, and one of the chief features of description logics, automatic classification, is mentioned in part of a brief subsection titled "Classification" (pp. 153–154), without a mention of the crucial topic of necessary and sufficient conditions.

The theme of Section 3.3, "Rules and Data" (pp. 156–169), is that "the convergence of [rule-based expert systems and relational databases] results from their common logical foundations: they both store data in the existential-conjunctive (EC) subset of logic, and they use the same rules of inference to answer questions, perform updates, and check constraints" (p. 156). The three languages cited frequently in this section are Microplanner, Prolog, and SQL, though SQL is, by far, the most illustrated of the three, CLIPS's notation is used instead of Microplanner's, and conceptual graphs are both used and argued for: "By representing logic in a form that is close to natural language, conceptual graphs can serve as an intermediate language for mapping to lower-level languages like SQL" (p. 161). The pattern-action structure of production rules is discussed more completely here than it was in Section 3.1, and is used as a lead-in to a discussion of the new and similar structure of SQL triggers. The final subsection, "Implementing Logic in Practical Systems" (pp. 168–169), is a discussion of efficiency with no details. Unification is mentioned as "a powerful pattern-matching technique" (p. 168), which is better than its previous, incorrect, characterization as "a rule of inference" (p. 148), but nowhere in the book is the unification algorithm presented in any detail. Similarly, rete networks are mentioned, along with their benefits, but without enough details to implement one. The final statement of this section is "logic serves as the common foundation for database and expert systems . . . Microplanner was inefficient even on the toy databases in SHRDLU. With optimization, SQL database systems routinely manage terabytes of data while responding to requests from thousands of users around the world" (p. 169).

Section 3.4, "Object-Oriented Systems" (pp. 169–178), begins with a discussion of object-oriented programming languages, including a couple of Java examples, and the concept of encapsulation, but quickly moves to conceptual graphs: "To represent the encapsulated objects of object-oriented systems, logic must support *contexts* . . . In conceptual graphs, contexts are represented by concept boxes that contain nested graphs that describe the referent of the concept" (p. 173, italics in the original). This is illustrated by a birthday-party context, which has several levels of nested conceptual graphs, and is now treated as a graphical user interface: "At the bottom of the box in Figure 3.5 is another concept [Process]. By clicking on that box, a person could expand it to a context that shows the steps in the process" (p. 175). The semantics of contexts, however, is not discussed until Chapter 5.

Section 3.5, "Natural Language Semantics" (pp. 178–186), provides a brief overview of natural language processing, mostly by means of an example from DANTE (Velardi, Pazienza, and Giovanetti 1988), including one sentence, a parse tree of its subject, and a conceptual graph representing the sentence. There is brief discussion of morphology, syntax, semantics, thematic relations, ambiguity, question answering, and inference.

Section 3.6, "Levels of Representation" (pp. 186–196), contains short discussions of several kinds of levels. These include the semantic network levels of Brachman (1979), the levels of robotic competence of Brooks (1986), and the design levels of Zachman (1987), as well as the sequences of levels of object to name-as-character-string to binary-representation-of-character string, and object to concept-of-object to representation-of-concept to concept-of-representation.

Chapter 4, "Processes" (pp. 206–264), discusses the nature and representation of times, events, situations, procedures, and processes, mostly via conceptual graphs, but with a major discussion and presentation of Petri nets and of a technique for mapping Petri nets into conceptual graphs. This chapter also contains discussions of process synchronization, data flow, message passing, constraint satisfaction, the frame problem, and the Yale shooting problem. Yet, reading this chapter, I lost track of the point. Is it to implement an intelligent agent that can carry out these procedures? Or reason about them? Or is it to write a program that can simulate these procedures? Or for a human to analyze the procedures by analyzing the diagrams? Or all of those? Or just to show that conceptual graphs can represent anything?—"As complete representations for logic, conceptual graphs and predicate calculus are general enough to represent everything that can be represented in a Petri net, a timing diagram, or any other notation for discrete processes" (p. 227). Sowa's statements of the theme of this chapter are: "Chapter 4 presents methods for representing dynamically changing processes and events" (p. xiii) and "Chapter 4 showed how logical inferences can simulate any computation that can be performed by a digital computer" (p. 427).

Chapter 5 is titled "Purposes, Contexts, and Agents" (pp. 265–347). "In Peirce's terms, purpose is the Thirdness that relates some mind or mindlike entity (first), which directs the course of a process (second) toward some goal (third)" (p. 265). Most of the chapter, however, is about contexts, which were previously defined in Section 3.4 (see above), and are again defined here: "In CGs, a context is defined as a concept whose referent field contains nested conceptual graphs" (pp. 276–277). There is a long subsection on "Semantics of Contexts" (pp. 284–297), in which the theories of Peirce, McCarthy, and Barwise and Perry are discussed. However, there is no crisp statement of the semantics of contexts in conceptual graphs. There are, however, many other interesting discussions in this chapter, which may be inferred from some of the section and subsection titles: "Verbs as Nexus" (pp. 270–272) ("In Peirce's terms, a verb is a syntactic unit for representing the mediating Thirdness that relates the participants" [p. 270]); "Defining Verb Senses" (pp. 272–274); "Discourse Representation Theory" (pp. 278–279) ("Although Kamp had not been aware of Peirce's existential graphs, his DRSs were structurally equivalent to Peirce's EGs" [p. 278]); "Resolving Indexicals" (pp. 279–280); "Conversational Implicatures" (pp. 282–284); "Meaning in Natural Languages" (pp. 293–294). Section 5.4, "First-Order Reasoning in Contexts" (pp. 297–307), is mostly about the proof theory of Peirce's existential graphs, which struck me as an interesting system. Section 5.5, "Modal Reasoning in Contexts" (pp. 307–321), is mostly a summary of modal logic. Section 5.6, "Encapsulating Objects in Contexts" (pp. 321–329), relates conceptual graphs to object-oriented programming languages. Section 5.7, "Agents" (pp. 330–339), includes short discussions of consciousness, reactive agents, and BDI (beliefs, desires, intentions) agent models. As was the case for Chapter 4, I found many of the discussions in Chapter 5 interesting, but I had a hard time keeping track of the thread, which Sowa says is "how purpose and context affect knowledge representation and the various theories of modal and intentional logic" (p. xiii) and "how logic can represent distributed agents, encapsulated objects, and message passing between agents and objects" (p. 427).

Given my comments about interesting discussions without a firm thread, one might suppose that Chapter 6, "Knowledge Soup" (pp. 348–407), is a collection of left-over topics, but that is not the point:

> Some of the knowledge in people's heads may be represented in propositions, more of it in imagelike forms, and the rest of it in habits, vague intuitions, and "gut feelings" that are never verbalized or visualized. Whatever its form, the knowledge is far too complex and disorganized to be called a knowledge base. Its fluid, heterogeneous, ever changing, and often inconsistent nature could be better characterized as *knowledge soup*. (p. 248, italics in the original)

This chapter includes sections on: "Vagueness, Uncertainty, Randomness, and Ignorance" (pp. 348–356), including a paragraph-long subsection on the paradox of the heap (p. 353), and a short subsection on liquids (pp. 353–354); "Limitations of Logic" (pp. 356–364), including subsections on "Case-Based Reasoning" (pp. 358–359) and "Propositions and Continuous Models" (pp. 363–364); "Fuzzy Logic" (pp. 364–373); "Nonmonotonic Logic" (pp. 373–383); "Theories, Models, and the World" (pp. 383–394); and "Semiotics" (pp. 394–402).

Chapter 7 is titled "Knowledge Sharing and Acquisition" (pp. 408–466). Section 7.1, "Sharing Ontologies" (pp. 408–417), includes subsections on "Problems of Aligning Ontologies" (pp. 409–412) and "Ontologies and Axioms" (pp. 412–414) that might better have been included in Chapter 2, "Ontology." There are then subsections on natural language and on problem solving that discuss the missionaries-and-cannibals problem but don't seem to be about knowledge sharing or knowledge acquisition. They do, however, contain a conceptual graph for the sentence "Three missionaries wearing flowing robes and three cannibals wearing loincloths are walking together from one village to another" (p. 414), along with a discussion of the problems in representing that sentence. Section 7.2, "Conceptual Schema" (pp. 417–427), is a database systems–oriented section, containing a subsection on the logical inadequacies of a simple natural language database query system, a short introduction to entity-relationship diagrams and a short subsection on query graphs: "conceptual graphs can also express any database query that can be expressed in SQL" (p. 426). Sections 7.3 and 7.4 contain more discussions relating conceptual graphs to various data structures, such as trees, and database systems issues. Section 7.5, "Language Patterns" (pp. 445–452), contains discussions of natural language issues including count nouns vs. mass nouns, stative verbs vs. action verbs, and some thematic roles.

There have been two basic organizing principles for texts on knowledge representation: by knowledge representation formalism, such as that of Reichgelt (1991), and by the subject matter being represented, such as that of Davis (1990). This book follows neither, though it is closer to the latter. Perhaps the word "Foundations" in the subtitle should be taken seriously, and it should be considered not so much a text *about* knowledge representation as a text about the foundations *for* knowledge representation.

Whether you would want to use this book as a text in a knowledge representation course you are teaching probably depends on your attitude toward conceptual graphs and your attitude toward Sowa's style. If you are already an aficionado of conceptual graphs, you will probably like it. If you would rather not use conceptual graphs in your course, you will probably not want to use it. I hope this review and the quotations I've included will give you a flavor of Sowa's style, and help you decide if that would be an asset or a detriment.

For a more experienced audience, however, whether practitioner, researcher, or advanced graduate student, I recommend this book for the many interesting discussions of issues in the foundations of knowledge representation.

The author's Web site for the book, including errata, is http://www.bestweb.net/~sowa/krbook/.

References

Brachman, Ronald J. 1979. On the epistemological status of semantic networks. In Nicholas V. Findler, editor, *Associative Networks: The Representation and Use of Knowledge by Computers*. Academic Press, New York, pages 3–50.

Brooks, Rodney A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23.

Davis, Ernest. 1990. *Representations of Commonsense Knowledge*. Morgan Kaufmann, San Mateo, CA.

Davis, Randall, Howard Schrobe, and Peter Szolovits. 1993. What is a knowledge representation? *AI Magazine*, 14(1):17–33, Spring.

Mineau, Guy W. 2000. The extensional semantics of the conceptual graph formalism. In Bernhard Ganter and Guy W. Mineau, editors, *Conceptual Structures: Logical, Linguistic, and Computational Issues*. Volume 1867 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, pages 221–234.

Reichgelt, Han. 1991. *Knowledge Representation: An AI Perspective*. Ablex, Norwood, NJ.

Shapiro, Stuart C. 1971. A net structure for semantic information storage, deduction and retrieval. In *Proceedings of the Second International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA, pages 512–523.

Shapiro, Stuart C. 1987. Processing, bottom-up and top-down. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*. John Wiley & Sons, New York, pages 779–785. Reprinted in Second Edition, 1992, pages 1,229–1,234.

Sowa, John F. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA.

Velardi, Paola, Maria Teresa Pazienza, and Mario De Giovanetti. 1988. Conceptual graphs for the analysis and generation of sentences. *IBM Journal of Research & Development*, 32(2):251–267.

Zachman, John A. 1987. A framework for information systems architecture. *IBM Systems Journal*, 26(3):276–292.

Stuart C. Shapiro is Professor of Computer Science and Engineering and a member of the Center for Cognitive Science at the University at Buffalo. He is co-editor of *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, (AAAI Press/The MIT Press, 2000), is a Fellow of AAAI, and has served as chair of ACM/SIGART (1991–95) and president of KR, Inc. (1998–2000). His principal research areas are knowledge representation, reasoning, and natural language processing. Shapiro's address is Department of Computer Science and Engineering, University at Buffalo, The State University of New York, 226 Bell Hall, Buffalo, NY 14260-2000, U.S.A.; e-mail: shapiro@cse.buffalo.edu.