March 1985

Report No. STAN-CS-85-1065
Also numbered KSL-85-31

Review of Sowa's "Conceptual Structures"

by

William J. Clancey

Department of Computer Science

Stanford University
Stanford, CA 94305

REVIEW OF SOWA'S "CONCEPTUAL STRUCTURES"

William J. Clancey

Stanford Knowledge Systems Laboratory
Department of Computer Science
701 Welch Road, Building C
Palo Alto, CA 94304

The studies reported here were supported (in part) by:

The Office of Naval Research
Personnel and Training Research Programs,
Psychological Sciences Division.
Contract No. N00014-85K-0305

The Josiah Macy, Jr. Foundation
Grant No. B852005
New York City

Conceptual Structures--Information processing in mind and machine

J. F. Sowa
Addison-Wesley Systems Programming Series
Reading, MA, 1984
481 pages, indices and appendices

Conceptual Structures is a bold, provocative synthesis of logic, linguistics, and Artificial Intelligence research. At the very least, Sowa has provided a clean, well-grounded notation for knowledge representation that many researchers will want to emulate and build upon. At its best, Sowa's notation and proofs hint at what a future ***Principia Mathematica*** of knowledge and reasoning may look like. No other AT text achieves so much in breadth, style, and mathematical precision. This is a book that everyone in AT and cognitive science should know about, and that experienced researchers will profit from studying in some detail.

Conceptual Structures is really three books: an encyclopedic survey of philosophical and psychological foundations of AI theory (including an epilogue on the limits of formal reasoning); a mathematical text that develops a knowledge notation called a ***conceptual graph*** and reasoning operators for manipulating it; and examples of how this notation is useful for natural language processing, database inference, and knowledge engineering. The material presented here was evidently honed by years of teaching experience. The bounty of memorable examples, historical summaries, and subtly witty perspectives on AI make us all grateful students. Here is history and science with a personality.

Yet for all this, the book is not perfect. Sowa has an innovative point of view that could have a strong effect on AI research, but it is an angle developed primarily in database research. This experience is the source of strength of Sowa's ideas, but his knowledge of both expert systems and cognitive science issues is not complete. For example, the relation of conceptual graphs to heuristic reasoning is not adequately developed or demonstrated by working programs. This reflects more the state of the theory, rather than being a fault of the book. Sowa synthesizes theoretical work of the past decade that researchers are only beginning to apply to large scale, "knowledge engineering" problems. The goal of this review is to summarize Sowa's theoretical insights, while articulating gaps that may make their application difficult. As a reader's guide, this review will help you find sections of the book to study in detail.

The mind: A survey and grand scheme

The value of the introductory chapters on philosophy and psychology is perhaps best exemplified by the one page discussion of Wittgenstein (page 15). Here the distinction is clearly made between concepts as composites of well-defined primitives, an extreme Aristotelian view presented in Wittgenstein's ***Tractatus,*** and concepts as family resemblances, the view of ***Philosophical Investigations.*** Upon this philosophical discussion Sowa eventually develops a calculus of type definitions and schemas, along with a basic reasoning operator he calls a "maximal join." Sowa unifies Pierce's type/token distinction (79), Aristotle's idea of type inheritance (81), and Leibniz's Universal Characteristic semantic lattice (82), enabling the AI and cognitive science researcher to appreciate the origins and relevance of these sometimes ancient philosophical problems.

In surveying a topic, Sowa typically presents a page or two of high level summary with a layman's introduction and diverse references to seminal work. For example, in discussing the nature of schemata, Sowa presents fascinating examples from epic poems and jazz (46). In lucid, enchanting prose, Sowa surveys the pervasive role of pattern, form, and grammar in communication. The introduction on conceptual relativity ranges from the nature of species to oil well databases, with reference to Jaensch, Whorf, and Searle. Admittedly, such an encyclopedic overview sometimes reads like little more than a list of pointers to readings, with little sense of additional insight, except for the clarity of restatement. So we get one pithy quote from Maturana (346) and no discussion. These historical surveys are well-written and fascinating, but they just begin to develop connections; a researcher should read the original sources for a deeper understanding.

In general, Sowa appears to derive a certain pleasure in citing early sources. For example, the idea of a schema is attributed to Kant and Selz. After nine pages of discussion and examples, the final sentence in the section reads, "In AI, Minsky (1975) showed the importance of schemata which he called 'frames.'"(51) This kind of tongue-in-cheek awareness of AI, impishly shows off Sowa's broad view of history. Thus, production rules are attributed to "Thue (1914)" and semantic nets associated with "Masterman (1961)." This is all very entertaining, but sometimes the book reads like a history of how AI evolved on another planet. Students only exposed to this book might have some difficulty following current lines of research. The irony is less funny when we find that Norman and Rumelhart are only cited by one reference in the suggested reading section and not discussed in the section on schemata at all. All of the 1970's research on story understanding, problem solving, and reasoning by analogy using schemata is ignored. Rosch is cited in the bibliography, but not mentioned in the text, a glaring omission. Thus, despite its claim to be a cognitive science text, this book is more valuable for its historical perspective than for its treatment of current research. Sowa knows about recent work, but he is apparently more familiar with sources in other fields, which often predate AI.

Nevertheless, whether in explaining the evolution of cognitive psychology from behaviorism or in proposing a model of an intelligent assistant he calls "Superclerk" (353), Sowa's text is comprehensively clear and instructive, and sometimes profound. Sowa makes startlingly bold statements, with a kind of sermonic clarity that rings of truth and revelation in your mind for days afterwards.

For example, to demolish the misconception that "special symbols and abbreviations are not a part of natural language" (343), Sowa gives examples from accounting textbooks and chemistry to show that "what is natural depends upon the topic." He speaks boldly of what we all know, but rarely manage to say at all: "For any subject, natural language is the form of expression that two experts in a field commonly use in speaking or writing to each other." In arguing that no artificial language could be more precise than English, Sowa concludes with a resounding QED: "Whatever can only be stated vaguely in English cannot be stated at all in a formal language." This book abounds with strong and simple sentences, the mark of a clear thinker. Combined with its breadth and daring attempt to synthesize so much research, the clarity of this work makes it a perfect starting point for discussion. Some good lectures could be lifted from this book verbatim.

Chapter two, on psychology, introduces what I would call a "grand scheme" for how the mind works. This analysis is more complete than anything I have seen elsewhere, made up of '*sensory icons," an "associative comparator," an "assembler," etc. Sowa summarizes the argument, in his soritical style, with a bulleted list of linked statements: "... images could arise from either sensory stimulation or from internal processes... internally generated images have the same nature as sensory icons... concrete concepts with associated percepts can be mapped to images that are accessible to consciousness... conscious reflection is the use of perceptual mechanisms to reanalyze and reinterpret inner speech.'* (61)

To restate, the mind can assemble "percepts" from memory into internal images that are experienced (can be thought about) exactly as images arising from the senses. This model is elegant because it provides a uniform basis for perception and abstract thought. It is perhaps best illustrated by the description of dreaming as a process of story understanding in which the mind feeds upon its own constructed images: The language of thought is tied to images, so the interpretations of images are further images. (34)

Sowa's grand scheme is a framework for all of reasoning. Like the model proposed by Newell and Simon, Sowa has a place for patterns (schemata) and production rules (associative comparator). But he goes a level deeper, speaking of sensory icons, percepts in memory, mental images, and conceptual graphs. "Percepts are fragments of images that fit together like the pieces of a jigsaw puzzle. A conceptual graph describes the way percepts are assembled." (71) Sowa distinguishes a conceptual graph from the term "semantic network": "Each conceptual graph asserts a single proposition. The semantic network is much larger. It includes a defining node for each type of concept, subtype links between defining nodes, and links to perceptual and motor mechanisms." ***(78)*** A concept ***interprets*** a percept; a percept is the ***image*** of a concept.

Sowa lays out an all-encompassing model of cognition that seems seductively real in his presentation. But all of the straightforward talk about brain functions and sensory processing made me uneasy. I kept stopping to wonder, "Do we really know these things?" Sowa acknowledges that the nature of mental imagery is controversial (7). But after stating Kosslyn's findings, Sowa describes the "central controller" as if he were saying what is known to everyone. In summarizing his model, he appears to claim too much: "With emotions to set the goals and with the associative comparator and assembler as the major processing units, the chunks, working registers, schemata, expectancy waves, control marks, and closures provide the mechanisms for an intelligent processor*' (64). Sowa admits that his model is far from complete, but it is bothersome that so much speculative synthesis is stated as established fact. Why is there not even one paragraph in the book where Sowa reflects on what he has attempted to do? The style is very strange. If this is a book of science, why does Sowa present a controversial model as if it is obvious? Used as a textbook, students may get the wrong impression. The grand scheme is daring and is based on familiar components, but it claims more than many scientists are ready to accept.

In a rare slip, we catch Sowa reaching for more than can be said. In support of his belief that psychological experiments and current AI approaches support each other, he states that ***psychological evidence*** for "markers" is their use in programs: "In computer systems the simplest way of identifying entities is by assigning each a unique marker" (85). Thus, he reveals a lurking-behind-the-corners desire to believe too much, that our computational models really ***are*** how the mind works. The historical introductions are similarly strewn with bizarre, unexpected facts, revealing Sowa's broad reading and proclivity to relate specific findings to his grand scheme: "The thalamus generates a six-per-second rhythm that apparently serves as a pacemaker for speech rhythms....'* (216) In describing the principles of natural language (arbitrary standards, structuralism, family resemblances, and open texture), Sowa concludes that because these principles appear at the level of phonology as well as semantics, "they must result from fundamental mechanisms of the brain" (216). Sowa may be right, but the necessity of his statements--"this is how things must be"--is sometimes jarring.

As we get into chapter three, where the logic of conceptual graphs is worked out in mathematical detail, none of this speculative psychological model of perception and imagery matters very much. The text systematically alternates between informal summary and formal prose with assumptions, definitions, and theorems. Conceptual graphs are related to first order logic and other knowledge notations, and demonstrated to be useful for problem solving. Many readers will no doubt be fascinated by Sowa's grand scheme. But the psychology of how the mind constructs conceptual graphs from sensory icons is not essential to the points Sowa makes about knowledge representation.

Conceptual graphs and knowledge representation

In this section, the terms type, hierarchy, individual, generic concept and others are defined mathematically. Reading this, I felt real appreciation for Sowa's systematic approach. This precision is rarely found in descriptions of AI programs and knowledge representations, and is similar to the formal treatment of frames we find in Brachman's work.

Sowa defines a conceptual graph to be a combination of concept and relation nodes, where every arc of every relation is linked to a concept. A simple example of a canonical graph is [COLOR] <--- (ATTR) <--- [PHYSOB], translated as "a color is an attribute of a physical object." (Concepts are in brackets, relations in parentheses.) Canonical graphs are not universal definitions, rather they make up the basis set of what some reasoning agent knows about his world. New conceptual graphs can be assembled from an existing set of canonical graphs by "formation rules" in terms of the operators copy, restrict, join, and simplify. Thus, Sowa defines a reasoning calculus in terms of a notation and operators for manipulating it.

The powerful synthesis of Sowa's conceptual graph theory is well-illustrated by his analysis of Chomsky's famous sentence, '*Colorless green ideas sleep furiously** (95). In attempting to map this into a conceptual graph, the following anomalies are found. Rules for forming conceptual graphs act as ***selectional constraints,*** preventing a join between "green" and "ideas" and between "ideas" and "sleep" (the agent of SLEEP must be of type ANIMAL; COLOR must be an attribute of a PHYSOBJ). Rules of logic (referring to ***meaning postulates*** and word intensions) prevent joining "colorless" and "green." Finally, previously constructed and labeled conceptual graphs (schemata), act as ***plausibility heuristics,*** suggesting that a join between "sleep" and "furiously" is unlikely. Thus Sowa provides a notation for expressing knowledge that combines (local, context-free) canonical graph formation rules with (global, context-sensitive) rules of inference and background knowledge about the world.

Canonical graphs represent an individual's world view. They are formed by perception, the grammatical formation rules, and "insight." Sowa says that insight occurs when a person feels "that existing percepts, concepts, or relations do not adequately describe a situation and may invent a radically new configuration that better describes it" (91). Can we canonicalize any graph we wish? What properties should a starting set of canonical graphs have? Correspondence to the world ("truth") is one issue, efficiency is another. Sowa's five page overview of learning (329) (a reasonable survey, with the usual Sowan references to early work) suggests that learning mechanisms are different from the conceptual graph calculus. In particular, his formal theory leaves out the episodic knowledge that is central to models of memory and learning, such as proposed by Schank. This separation between routine problem solving and learning is a simplification; it is one aspect of the formal theory of conceptual graphs that must be extended.

In defining what a concept is, Sowa makes a basic distinction between type definitions (Aristotelian, with necessary and sufficient conditions) and schemas (Wittgensteinian, with conditions for determining applicability and typical defaults). With typical Sowan matter-of-factness, we are told that "Type definitions are appropriate for some of the formal concepts of science, law, or accounting. Schemata are necessary for the loosely structured concepts of everyday life." (135).

Sowa goes on to formally describe an aggregation (such as CIRCUS-ELEPHANT and HOTEL-RESERVATION), composite individual (instantiated aggregation, e.g., the CIRCUS-ELEPHANT, Jumbo), and prototype (specialization of a composite of schemas, indicating defaults true in a typical case). A prototype is formally defined: "A prototype p for a type t is a monadic abstraction (lambda(a) u) with the following properties: the formal parameter a is of type t; the prototype p is derived by a schematic join of one or more schemata in the schematic cluster for t, with some or all of the concepts in p restricted from generic to individual."

The discussion of Aristotelian definition is simply beautiful. Sowa concludes, "The di-fferentia is the body of a monadic abstraction, and the genus is the type label of the formal parameter."(106) Reading about the operations of aggregation and individuation ("aggregation groups individuals into a composite, and individuation projects a general graph into a composite of individuals"), I realized that this book had completely changed my idea of what knowledge representation is. Rather than thinking in terms of "attributes" and "values," I started to think in terms of concepts described in relation to other concepts, where relations themselves are typed and related to more primitive relations. These ideas have been around in various circles of AI for a decade, but until I read this book, I didn't understand their relevance to heuristic, rule-based programs (see below, "Conceptual graphs and knowledge engineering").

When we get to abstraction and definition, the text becomes a bit complex. The idea of a "maximal join" (103) is very basic, and seems intuitively simple, but I never fully grasped the idea until an example was given in the knowledge engineering chapter. Here is the example: The query graph corresponding to "What was Lee's age when hired?" is merged with the schema for AGE, chosen for merge because of expected relevance as a "relatively rare type." First, we identify the maximal common generalization, which is a subgraph of the AGE schema: [PERSON] ---> (CHRC) ---> [AGE] ---> (PTIM) ---> [TIME] ("a PERSON has a characteristic, AGE, at a point in time, TIME"). Then, we effect a maximal join by replacing the universal quantifier implicit in [PERSON] to give [PERSON: Lee] and replacing the generic concept [TIME] by the universally quantified concept [DATE] (corresponding to the date of hire in the schema for HIRE). Thus, the query is merged with known concepts so that known values can be propagated to compute an answer. Sowa says that maximal joins "form the basis for 'preference semantics' (Wilks, 1975), which encourages maximum connectivity in the generated graphs.'* Maximal joins are equivalent to unification in logic programming (197).

In complete detail, Sowa works out the mathematics of concepts, relations, conceptual graphs, and abstractions. He then groups these into generalization hierarchies and lattices, and defines, where appropriate, operations for maximal or minimal merges, expansions, and contractions of graphs. Most of his ideas have their origin in database query language semantics. He builds upon linguistics and AI work, such as the "selectional constraints" of Katz and Fodor, conceptual dependency graphs of Schank (134), Wilk's preference semantics, Brachman's individuation of concepts (119), Hendrix's partitioned semantic nets (138), among many others.

Taken as a whole, the idea of a reasoning calculus is startling at first. Mathematically-defined operators working on concepts? A real science or logic of reasoning? Is that possible? Could AI be made as precise as this? Does Sowa bridge the gap between logic and schema-based reasoning? Consideration of problems with standard logic notations and knowledge engineering applications reveals that the answer to these questions is "almost," and Sowa gets a cigar for his efforts.

Conceptual graphs and logic

Chapter one provides a good overview of many of the controversies surrounding the use of logic as a knowledge representation. These problems include: the failure of logic to semantically relate the parts of a conditional statement, the truth of empty extensionality, the non-psychological nature of deductive proof, and a syntax more complicated and difficult to read than natural language. If everyone in AI and Cognitive Science read and understood section 1.6, the field might advance by a great leap in a single day. I showed this material to a specialist in logic programming, and he said, sure, he knows these things and elaborated upon them. Yet in his technical talks and papers he never makes the nature of these controversies clear, only presenting his own point of view, and leaving out deficiencies. Sowa's book is full of the kinds of controversies and multiple perspectives that specialists know, but rarely convey to others in the field.

To a large degree, one purpose of this book is to resolve the conflict between the scruffies (the "network hackers") and the neats (the logicians). Sowa agrees with the scruff ies about "the importance of a smooth mapping to natural language and the heuristic value of schemata." But he sides with the neats in insisting that network notation be grounded in logic. Put the other way, he starts in the logic camp, but agrees with Pierce that a graph notation, resembling Schank's conceptual dependency diagrams, is preferable to the algebraic, linear form of Peano notation. Sowa makes clear that there are alternative forms for displaying conceptual graphs. He illustrates the pros and cons of 2-dimensional graphs, a linear indented form (for terminal output) that resembles a case frame, and first-order predicate calculus formulas. As he mentions in another context, this book shows how to "do logic on graphs."(325)

Using nested graphs, Sowa provides fascinating examples of how quantification can be handled, that, at least from this non-specialist's view, appear to address the problems of scope and coreference. He goes on to demonstrate that conceptual graph notation usually requires fewer symbols and shorter proofs, is more directly mapped to natural language, has direct extensions to modal logic, and can co-exist with other logical notations (149). Later in the book, he argues that putting primary emphasis on nodes that represent individuals avoids the need for duplicate, "scattered" variables that standard logic notation, with its emphasis on predicates, requires (202). Conceptual graphs are usually more concise and therefore easier to read than logical formulas because the arcs on the graphs show connections more directly than variable symbols.

The examples of joins (316) suggest that conceptual graphs provide a more efficient representation than standard logic because they structure the inference process. This is accomplished by the instantiation/specialization rules, network propagation for determining unknowns, plus merging of relevant schemas, bringing in other relations that may be useful for computation or database lookup (illustrated by the date of hire example). For Sowa, a concept is not a data structure used for efficiency, as some might describe frames or units, rather his entire theory of knowledge is concept-centered. Thus, in computing the age at date of hire, the program refers to graphs corresponding to the concepts AGE and HIRE. Coming away from all of this, I had to conclude that if I were going to design a knowledge representation from scratch, Sowa's notation seems like the logical place to begin.

The sections on formal deduction, model theory, tenses and modalities provide advanced theoretical detail that contrast with the encyclopedic terseness of the historical sections. I found these 40 pages to be a rewarding, superb introduction, but some sections (on open worlds, for example) are at the level of detail and rigor of specialized research. The average reader can skip the the proofs, reading the prose in between, and go away grasping most of the material. The discussion of model theory is nothing short of brilliant, starting in typical Sowan style with the first sentence, "A notation by itself has no meaning."(161) A discussion of particular interest contrasts procedural representations (appropriate for the limited requirements of asking questions about single finite models, e.g., a database) with theorem proving/declarative representations (for proving general constraints about all possible models). Sowa argues that conceptual graphs are advantageous in this respect because they provide a common notation for formulas that make statements about a world as well as for structures that represent (model) a possible world. In a detailed discussion, Sowa shows how this approach builds upon Hintikka's (167). He claims that his synthesis (citing a 1979 paper) is similar to Barwise and Perry's ***situation semantics.*** But again, reflecting Sowa's non-mainstream AI point of view, he mentions belief maintenance only in passing and does not discuss circumscription.

In the final section of the reasoning and computation chapter, Sowa develops the idea of ***dataflow graphs*** made up of networks of actors, as a means of representing procedures. Control marks are used to trigger the actors and compute the referents for generic concepts (188), based on the assert/request scheme of Petri nets. Dataflow graphs are bound to conceptual graphs: conceptual relations show the ***roles between entity types*** of the dataflow graph, and actors show their ***functional dependencies.*** Referring to the date of hire example, the actor for <DIFFERENCE-DATE>, cuts across this graph, relating the DATE of birth to the DATE of hire to compute the AGE. Linear and recursive procedures can be defined in this notation, but Sowa does not give primitives for iteration. This section exemplifies the strength of Sowa's analysis in unifying previous work. Later, he summarizes the kinds of knowledge he has brought together in the conceptual graph notation: type hierarchy, functional dependencies, domain roles, definitions, schemata, procedural attachments, and inferences (304).

Conceptual graphs and language

The chapter on language shows Sowa at his most entertaining. Sections on the genesis and strata of language nicely summarize the chimpanzee/ape experiments, human language development, the role of rhythm (inspired by his wife's research), transformational grammar, and so on. Like a good teacher, Sowa shares his favorite examples collected over the years, such as the sentence with 40 different parses, "People who apply for marriage licenses wearing shorts or pedal pushers will be denied licenses." Good examples relate case grammar relations to the conceptual relations of Sowa's graphs. In ten pages, Sowa carefully explains the idea of augmented phrase structure grammar, adapting conceptual graphs to Heidorn's notation (236). The comprehensive summary of parsing methods, including frequent comparisons to Chomsky's approach, and conceptual catalog (appendix of example concepts, relations, and conceptual graphs) make this a valuable text on language processing for the new student and non-specialist researcher alike. And it is just like Sowa to tell us about 'postpositions"--the kind of dry, humorous detail that gives this book a high-intellectual style and makes it fun to read.

Conceptual graphs and knowledge engineering

While Sowa addresses natural language processing in some detail, amply demonstrating the advantages of the conceptual graph notation, the value of conceptual graphs for planning, diagnosis, and configuration is not well-developed. A chapter on knowledge engineering gives brief examples of well-known programs, but Sowa doesn't make proper distinctions or mention deficiencies. In a typical misleading description, he describes Casnet as a model-based program, contrasting it with "surface reasoning," failing to make a distinction between a behavioral state network and a structure/f unction simulation model. Sowa misses a big opportunity here to make his insights understandable by relating them to current research. Moreover, as I will discuss in some detail, the discussion he devotes to procedural knowledge and heuristics is vague and unconvincing.

The main discussion of the use of the conceptual graph notation for problem solving appears not in the knowledge engineering chapter, but in the fourth chapter on reasoning and computation. This very general discussion is a reprise of the conceptual processor model given in the psychology chapter, but now developed with the terminology of conceptual graphs. In a far-ranging and sketchy ten pages, Sowa relates conceptual graphs to demons, blackboards, conflict resolution, heuristics, search, and the proposer/skeptic model of reasoning (206). Sowa frankly admits that his theory has *'unspecified details that must be resolved in a computer implementation" (197).

Sowa's general description of a system architecture, unfortunately buried in these ten pages, is actually quite reasonable:

> For conceptual graphs, heuristics follow from the graph structure. Domain dependencies reside only in schemata and prototypes. Each schema or prototype is a packet of knowledge about some particular domain. The procedures that handle them are general rules or ***metaheuristics*** that apply to any domain. The structural properties of conceptual graphs can aid a system in finding and using large amounts of background knowledge.... (201).

An increasing number of AI programs (e.g., Abel, Neomycin, Dart) clearly separate domain knowledge from explicit reasoning rules. My complaint here is that Sowa suggests in the knowledge engineering chapter that all existing programs are designed this way. In a manner reminiscent of his description of perception and imagery, Sowa fails to distinguish between his idealized view and what most people are doing or believe. For a text like this to be effective, I think the current state of the art needs to be more clearly described and contrasted with the ideal model.

Specifically, the way in which inference is controlled in many rule-based systems by proceduralizing domain knowledge in production rules is mentioned in one fleeting sentence, '*Although production rules are widely used in AI, they frequently lead to ***ad hoc*** systems whose logical basis is obscure."(197) But Sowa never raises this issue in describing Mycin, suggesting by his description that domain knowledge and procedure are separate: "The system asks questions to determine the basic problem; then it applies the inference rules to determine the probable cause and the recommended actions." (283) The separation of asking questions and applying inference rules is not accurate. This might be intended to be a high-level summary, but Sowa will fail to convey his main points if his readers go away thinking that Mycin exemplifies the model.

While Sowa never makes the point very clearly, much of the knowledge now represented in rules in expert systems can be more directly represented in conceptual graphs. Definitions, computational relations, hierarchical relations, and default conclusions can be directly represented and easily reasoned about using Sowa's conceptual graphs. There is no need for rules here.

Rules are also often used to represent the "feature maps" of prototypes (e.g., identifying properties of an organism) or causal relations (e.g., between pathophysiologic states). Here it is less clear if Sowa's calculus inference mechanism is adequate. How do we indicate the order in which to gather information for testing a match? How are partial matches and uncertainty handled? Can causal networks be replaced by schemata describing processes? Again, how do we specify what matches to seek and what ordering to use? Sowa provides a basis for expressing these traditional knowledge engineering issues more precisely, but he only vaguely discusses them.

Some domain-specific rules are ***heuristics*** because they reduce the search for useful conceptual joins. For example, a medical diagnosis rule considering the age of a patient would have '*compiled in" consideration of other facts that would make the age irrelevant for suggesting diseases (for example, a recent trauma). In this sense, domain-specific heuristic rules are compiled conceptual joins; they are programs for bringing in the right schema at the right time (recall the age of hire example). Sowa's strict use of conceptual graphs for domain knowledge would appear to disallow these rules, insisting that (metaheuristic) rules index domain knowledge indirectly through conceptual relations. The implicit metaheuristic in the age rule example is that a statistical correlation (age) is less relevant when there is evidence of an event known to directly cause disorders (trauma). Thus, the relations "statistically correlated with" and "directly causes" organize the domain knowledge; this is what Sowa means when he says that "heuristics follow from the graph structure." Programs like Neomycin express heuristics in just this way, but it is unclear that this indirect, interpretive approach will always be efficient.

If domain-specific rules are necessary to avoid combinatorial search or to avoid time-consuming interpretation of complex general procedures, then the inference mechanisms supplied by Sowa are not sufficient for practical problem solving. We will be left with some ***ad hoc*** rules that leave out conceptual relations and simply state inferential paths. Perhaps these rules should be incorporated as a redundant, compiled form of knowledge, as practice models of chunking suggest. As Sowa says, it's an issue to be resolved (197).

Besides using domain-specific rules to reduce search for conceptual joins, rules are an appropriate representation for procedural knowledge. Most knowledge systems built for some purpose, such as diagnostic consultation, monitoring, or design, are ***programs*** which must interact with a user in some prescribed way, make certain inferences, control consideration of knowledge sources, post/modify partial solutions, print results, and probably cycle through a sequence of such steps. Sowa implies that these programs can be synthesized by the "conceptual processor'* (197), an intriguing way of combining the conceptual calculus with dataflow graphs, using a control marker scheme for managing goals. It is not clear if this proposal is mainly of psychological interest or whether it offers advantages over current AI descriptions of control knowledge.

Sowa provides an interesting perspective on knowledge acquisition that everyone interested in knowledge engineering will want to read. Sowa opens the knowledge engineering chapter with the remark, "A knowledge-based system keeps track of the meaning of the data and performs inferences to determine what information is needed even when it has not been explicitly requested."(277) This definition clearly reveals his experience with database query languages, the source of his fresh, stimulating point of view. He offers a neat and maybe prescient solution to the problem of training knowledge engineers: "The knowledge engineers of tomorrow will be today's systems analysts who have taken additional training...."(320). In fact, the knowledge acquisition section is really about translation of expert knowledge into conceptual graphs or equivalent languages. To Sowa, knowledge acquisition is concept definition, nicely putting the emphasis on knowledge, not implementation. However, he has oddly made conceptual analysis a separate section, and does not discuss pragmatic issues: interviews, problem formulation, prototype systems, and validation.

In short, while the rest of Sowa's book provides a fine foundation for putting knowledge engineering on a theoretical footing, the discussion of knowledge engineering practice is misleading and may be self-defeating. Sowa does not clearly describe how procedures and heuristics are encoded in today's programs, and he gives no examples of expert systems that use a conceptual graph approach. I am concerned that most readers will find the conceptual processor model to be obscure, never understand the general conception of abstract procedures operating on graph structures, and even go away thinking that the Mycin-like, common rule-based approach is what Sowa has in mind.

The following two sections on database semantics and inference provide some of the best examples in the book of the usefulness of conceptual graphs and are a superb introduction to these topics. The idea that a knowledge-based system does ***database retrieval*** by filling in background knowledge and making plausible inferences illustrates one way in which our current conception of expert systems is likely to evolve.

Conclusions

Hidden away in one suggested reading section, Sowa editorializes a bit, summarizing his contribution: "Although many forms of these networks are used in AI, the philosophical and logical questions underlying them have often been ignored.... (Analysis shows) the sloppy formulations of many theories in the field."(126) He correctly points out that rule-based systems may be harder to prove correct than ordinary programs....(198) As often happens in science, neither side has the full story: Sowa has given AI hackers a notation for describing the knowledge in their programs. The AI hackers' methodology of constructing programs to test theories would help Sowa to demonstrate the completeness and practicality of his ideas.

In spite of Sowa's failure to apply his ideas to difficult applications--outside of natural language and database query applications--the main contributions of this book to knowledge representation ("conceptual structures") should not be lost:

- the unification of logic, plausibility, and meaning constraints, set in a formal notation, with full definitions, proofs, and algorithms for plausible reasoning (conceptual graph formation rules);

- a good philosophical survey of the type/schema problem;

- a daring psychological synthesis, if a bit broad, of the reasoning process and the nature of concepts.

Sowa's insights are clear, but their application is complicated and not worked out. Nevertheless, my recommendation is definite: Every AI and Cognitive Science researcher should study the conceptual graph notation and understand its foundation in logic, database, and knowledge representation research. Specialists in knowledge representation and inference will profit by relating the conceptual graph notation to their own schemes. This book could have its greatest impact on specialists in fields such as cognitive anthropology, who might get a new perspective on knowledge and reasoning, and who could use conceptual graphs for constructing models. As a course text, the book is appropriate for a graduate seminar taught by someone who is familiar with mainstream AI of the past decade, or who intends to relate the book to some other field, such as philosophy. Given the historical bias and lack of development of current research, the experienced AI researcher can use this book most confidently and to the greatest advantage--as a source of new ideas and perspectives, and as a synthesis of research he has heard about, but previously couldn't relate to his own work.

Conceptual Structures closes, appropriately enough, with a detailed chapter entitled the "limits of conceptualization." Here are fascinating surveys on cybernetics, expressive power, relativity, intelligence, the mythology of science, and problems for cognitive science. I must admit, it was the paragraph on Zen Buddhism that led me to buy this book. The section on conceptual relativity is one to come back to again and again: "The only things that can be represented accurately in concepts are man-made structures that once originated as concepts in some person's mind."(345)

In the history of AI, controversies and misunderstandings have often split the community into camps--probably none more intensely argued than the role of logic or formal methods in knowledge representation and reasoning. Sowa bridges the gap with daring, humor, and an eclectic's ability to relate and resolve problems. In this methodologically self -conscious field, it behooves us to follow Sowa's example, to stop demanding that the other fellow prove he is right, and to instead reach out and find something of value in other points of view.