

Cognitive Architectures For Conceptual Structures

John F. Sowa
VivoMind Research, LLC

Abstract. The book *Conceptual Structures: Information Processing in Mind and Machine* surveyed the state of the art in artificial intelligence and cognitive science in the early 1980s and outlined a cognitive architecture as a foundation for further research and development. The basic ideas stimulated a broad range of research that built on and extended the original topics. This paper reviews that architecture and compares it to four other cognitive architectures with their roots in the same era: Cyc, Soar, Society of Mind, and Neurocognitive Networks. The CS architecture has some overlaps with each of the others, but it also has some characteristic features of its own: a foundation in Peirce's logic and semiotics; a grounding of symbols in Peirce's twin gates of perception and action; and a treatment of logic as a refinement and extension of more primitive mechanisms of language and reasoning. The concluding section surveys the VivoMind Cognitive Architecture, which implements and extends the CS architecture.

This is a slightly revised version of a paper in *Proc. 19th International Conference on Conceptual Structures*, edited by S. Andrews, S. Polovina, R. Hill, & B. Akhgar, LNAI 6828, Heidelberg: Springer, 2011, pp. 35-49.

1. Cognitive Architectures

A *cognitive architecture* is a design for a computational system for simulating some aspect of human cognition. During the past half century, dozens of cognitive architectures have been proposed, implemented, and compared with human performance (Samsonovich 2010). The book *Conceptual Structures* (Sowa 1984) surveyed the state of the art in the early 1980s and proposed a design that has stimulated a broad range of research and development projects. After more than a quarter century, it's time to review the progress in terms of recent developments in cognitive science, artificial intelligence, and computational linguistics. To provide perspective, it's useful to review some related architectures that have also been under development for a quarter century or more: Cyc, Soar, Society of Mind, and Neurocognitive Networks.

The Cyc project, whose name comes from the stressed syllable of *encyclopedia*, was chartered in 1984 as an engineering project. It placed a higher priority on computational efficiency than simulating psycholinguistic theories. The technical foundation was based on the previous decade of research on knowledge-based systems (Lenat & Feigenbaum 1987):

- Lenat estimated that encyclopedic coverage of the common knowledge of typical high-school graduates would require 30,000 articles with about 30 concepts per article, for a total of about 900,000 concepts.
- The Japanese Electronic Dictionary Research Project (EDR) estimated that the knowledge of an educated speaker of several languages would require about 200K concepts represented in each language.
- Marvin Minsky noted that less than 200,000 hours elapses between birth and age 21. If each person adds four new concepts per hour, the total would be less than a million.

All three estimates suggested that human-level cognition could be achieved with a knowledge base of about a million concept definitions. At a cost of \$50 per definition, Lenat and Feigenbaum believed that the project could be finished in one decade for \$50 million and less than two person-centuries of work.

After the first five years, Cyc had become an informal system of frames with heuristic procedures for processing them (Lenat & Guha 1990). But as the knowledge base grew, the dangers of contradictions, spurious inferences, and incompatibilities became critical. The developers decided to design a more structured representation with more systematic and tightly controlled procedures. Eventually, the CycL language and its inference engines evolved as a superset of first-order logic with extensions to support defaults, modality, metalanguage, and higher-order logic. An important innovation was a context mechanism for partitioning the knowledge base into a basic core and an open-ended collection of independently developed *microtheories* (Guha 1991).

After the first 25 years, Cyc grew far beyond its original goals: 100 million dollars had been invested in 10 person-centuries of work to define 600,000 concepts by 5 million axioms organized in 6,000 microtheories. Cyc can also access relational databases and the Semantic Web to supplement its own knowledge base. For some kinds of reasoning, Cyc is faster and more thorough than most humans. Yet Cyc is not as flexible as a child, and it can't read, write, or speak as well as a child. It has not yet reached the goal of acquiring new knowledge by reading a textbook and generating rules and definitions in CycL.

Unlike the engineering design for Cyc, the Soar design was based on "a unified theory of cognition" (Newell 1990), which evolved from four decades of earlier research in AI and cognitive science: the General Problem Solver as "a program that simulates human thought" (Newell & Simon 1961) and production rules for simulating "human problem solving" (Newell & Simon 1972). The foundations for Soar are based on the earlier mechanisms: production rules for procedural knowledge; semantic networks for declarative knowledge; and learning by building new units called *chunks* as assemblies of earlier units. Declarative knowledge can be stored in either long-term memory (LTM) or short-term (working) memory. It can represent semantic knowledge about concept definitions or episodic knowledge about particular instances of objects or occurrences. More recent extensions (Laird 2008) have added support for emotions and iconic memory for uninterpreted imagery.

In the books *Society of Mind* and *Emotion Engine*, Minsky (1986, 2006) presented a cognitive architecture that he had developed in five decades of research and collaboration with students and colleagues. In a review of Minsky's theories, Singh (2003) compared the Society of Mind to the Soar architecture:

To the developers of Soar, the interesting question is what are the least set of basic mechanisms needed to support the widest range of cognitive processes. The opposing argument of the Society of Mind theory is that the space of cognitive processes is so broad that no particular set of mechanisms has any special advantage; there will always be some things that are easy to implement in your cognitive architecture and other things that are hard. Perhaps the question we should be asking is not so much how do you unify all of AI into one cognitive architecture, but rather, how do you get several cognitive architectures to work together?

That question is the central theme of Minsky's books, but Singh admitted that the complexity of the ideas and the lack of detail has discouraged implementers: "While Soar has seen a series of implementations, the Society of Mind theory has not. Minsky chose to discuss many aspects of the theory but left many of the details for others to fill in. This, however, has been slow to happen."

Neurocognitive networks were developed by the linguist Sydney Lamb (1966, 1999, 2004, 2010), who had written a PhD dissertation on native American languages, directed an early project on machine

translation, developed a theory of *stratificational grammar*, and spent five decades in studying and collaborating with neuroscientists. Lamb's fundamental assumption is that all knowledge consists of connections in networks and all reasoning is performed by making, strengthening, or weakening connections. That assumption, with variations, was the basis for his linguistic theories in the 1960s and his most recent neurocognitive networks. Lamb avoided the symbol-grounding problem by a simple ploy: he didn't assume any symbols — the meaning of any node in a network is purely determined by its direct or indirect connections to sensory inputs and motor outputs. Harrison (2000) implemented Lamb's hypothesis in the PureNet system and showed that it made some cognitively realistic predictions.

The *Conceptual Structures* book discussed early work by the developers of these four systems, but the influences were stronger than mere citations. The first version of conceptual graphs was written in 1968 as a term paper for Minsky's AI course at MIT. Among the topics in that course were the General Problem Solver and the semantic networks by Quillian (1966), whose advisers were Newell and Simon. The early cognitive influences evolved from another term paper written in 1968 for a psycholinguistics course at Harvard taught by David McNeill (1970). The first published paper on conceptual graphs (Sowa 1976) was written at IBM, but influenced by the research at Stanford that led to Cyc. One of the early implementations of CGs (Sowa & Way 1986) used software that evolved from the dissertation by Heidorn (1972), whose adviser was Sydney Lamb. The goal for conceptual structures was to synthesize all these sources in a psychologically realistic, linguistically motivated, logically sound, and computationally efficient cognitive architecture.

2. The CS Cognitive Architecture

The cognitive architecture of the *Conceptual Structures* book overlaps some aspects of each of the four architectures reviewed in Section 1. That is not surprising, since the founders of each had a strong influence on the book. But the CS architecture also has some unique features that originated from other sources:

- The first and most important is the logic and semiotics of Charles Sanders Peirce, who has been called “the first philosopher of the 21st century.” His ideas and orientation have influenced the presentation and organization of every aspect of the book and every feature that makes it unique.
- The second feature, which follows from Peirce and which is shared with Lamb, is to ground the symbolic aspects of cognition in the “twin gates” of perception and action. Chapter 2 begins with perception, and Chapter 3 treats conceptual graphs as a special case of perceptual graphs. The ultimate goal of all reasoning is purposive action.
- The third, which also originates with Peirce, is to treat logic as a branch of semiotics. Although some sentences in language can be translated to logic, the semantic foundation is based on prelinguistic mechanisms shared with the higher mammals. (Sowa 2010)
- The fourth, which originated in skepticism about AI before I ever took a course in the subject, is a critical outlook on the often exaggerated claims for the latest and greatest technology. It appears in a strong preference for Wittgenstein's later philosophy, in which he criticized his first book and the assumptions by his mentors Frege and Russell. That skepticism is the basis for the concluding Chapter 7 on “The Limits of Conceptualization.” It also appears in later cautionary lectures and writings about “The Challenge of Knowledge Soup” (Sowa 2005).
- Finally, my preference for a historical perspective on every major topic helps avoid passing

fads. Some so-called innovations are based on ideas that are as old as Aristotle and his sources, many of which came, directly or indirectly, from every civilization in Asia and Africa.

Figure 1 is a copy of Figure 2.2 in the CS book. It illustrates the hypothesis that the mechanisms of perception draw upon a stock of previous *percepts* to interpret incoming sensory *icons*. Those icons are uninterpreted input in the sensory projection areas of the cortex. The percepts are stored in LTM, which is also in an area of the cortex close to or perhaps identical with the same projection area. Percepts may be exact copies of earlier icons or parts of icons. But they could also be copies or parts of copies of a previous *working model*, which is assembled as an interpretation of the current sensory input.

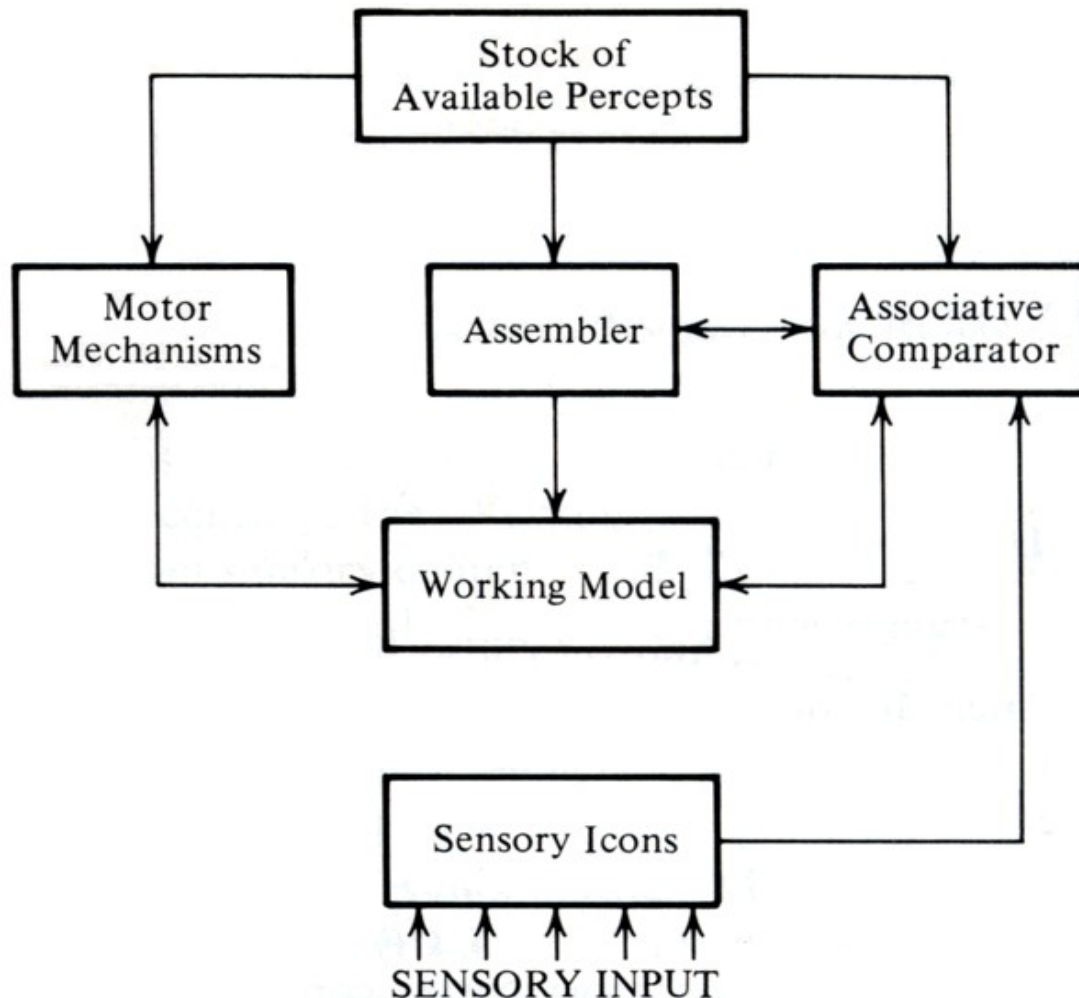


Figure 1. Mechanisms of Perception

The working model in Figure 1 is either an interpretation of sensory input or a mental model that has the same neural representation. The following quotation explains Figure 1:

- “The associative comparator searches for available percepts that match all or part of an incoming sensory icon. Attention determines which parts of a sensory icon are matched first or which classes of percepts are searched.
- The assembler combines percepts from long-term memory under the guidance of schemata. The result is a working model that matches the sensory icons. Larger percepts assembled from smaller ones are added to the stock of percepts and become available for future matching by the associative comparator.

- Motor mechanisms help the assembler construct a working model, and they, in turn, are directed by a working model that represents the goal to be achieved” (Sowa 1984:34).

The world is a continuum, and image-like percepts can preserve that continuity. But the vocabularies of all languages contain a discrete set of words or *morphemes*. The CS book emphasized the need to represent and reason about both: “To deal with language and imagery, concepts must be associated with both words and percepts. David Waltz (1981), who has done research on both computer vision and natural language processing, has been seeking a uniform underlying representation. He cited the following examples:

My dog bit the mailman’s leg.
My dachshund bit the mailman’s ear.
My doberman bit the mailman’s ear.

To understand the first sentence, no images are necessary. For the second one, people wonder how the dachshund could reach so high. But the third sentence is reasonable because a doberman is a much larger dog. Waltz argued that the brain must use visual and spatial mechanisms for interpreting such sentences. Although people may not have conscious images of the dachshund and doberman, they must use some sort of spatial processing.” Figure 2 introduces conceptual graphs as “a universal, language-independent deep structure” that relates perception to language. “When a person sees a cat sitting on a mat, perception maps the image into a conceptual graph. A person who is bilingual in French and English may say, in speaking French, *Je vois un chat assis sur une natte*. In describing the same perception in English, the person may say *I see a cat sitting on a mat*. The same conceptual graph, which originates in a perceptual process, may be mapped to either language.” (Sowa 1984:38)

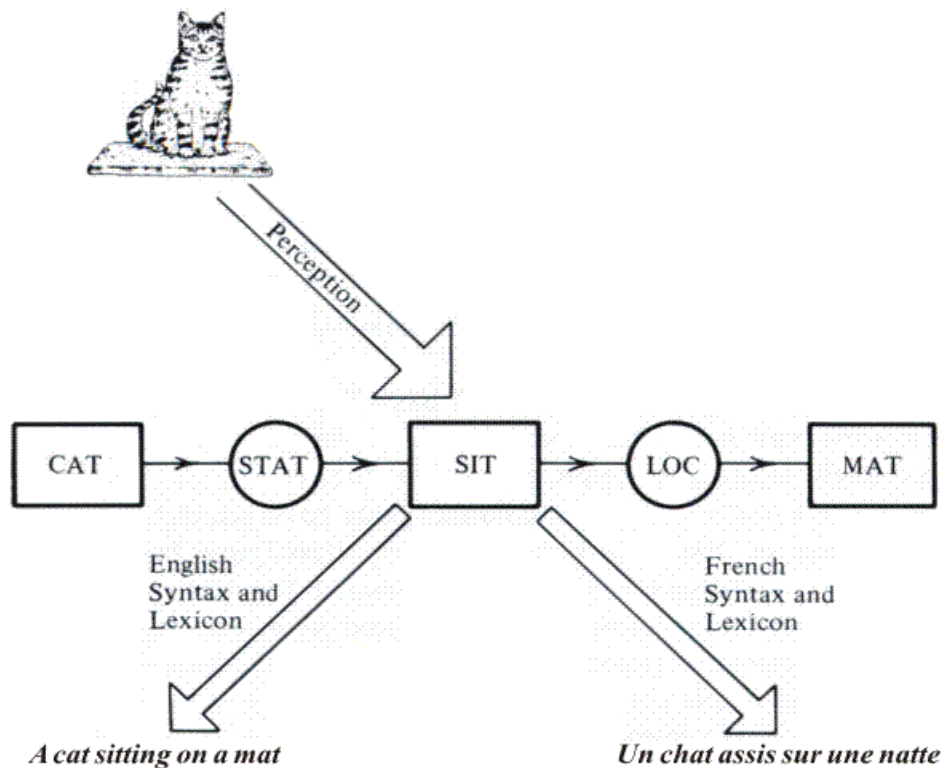


Figure 2. Relating percepts to concepts to languages

The assumption that the same concepts map to and from English and French words requires some qualification. Paradis (2009) pointed out that no two bilingual speakers have exactly the same

experiences with both languages. Figure 2 would be approximately correct for a native English speaker who learned French in school by mapping French words to concepts that were acquired in English.

Instead of assuming distinct mechanisms for propositions and mental imagery, Chapter 3 adds the assumption that the propositional representation in *conceptual graphs* is part of the same construction: “Perception is the process of building a *working model* that represents and interprets sensory input. The model has two components: a sensory part formed from a mosaic of *percepts*, each of which matches some aspect of the input; and a more abstract part called a *conceptual graph*, which describes how the percepts fit together to form the mosaic. Perception is based on the following mechanisms:

- Stimulation is recorded for a fraction of a second in a form called a *sensory icon*.
- The *associative comparator* searches long-term memory for percepts that match all or part of an icon.
- The *assembler* puts the percepts together in a working model that forms a close approximation to the input. A record of the assembly is stored as a conceptual graph.
- Conceptual mechanisms process *concrete concepts* that have associated percepts and *abstract concepts* that do not have any associated percepts.

When a person sees a cat, light waves reflected from the cat are received as a sensory icon s . The associative comparator matches s either to a single cat percept p or to a collection of percepts, which are combined by the assembler into a complete image. As the assembler combines percepts, it records the percepts and their interconnections in a conceptual graph. In diagrams, conceptual graphs are drawn as linked boxes and circles. Those links represent logical associations in the brain, not the actual shapes of the neural excitations.” (Sowa 1984:69-70)

The CS book cited a variety of psychological and neural evidence, which is just as valid today as it ever was. But much more evidence has been gathered, and the old evidence has been interpreted in new ways. The primary hypothesis illustrated by Figure 1 has been supported: the mechanisms of perception are used to build and reason about mental models, and conceptual structures are intimately related to perceptual structures. That assumption has been supported by abundant evidence from both psychological and neural sources (Barsalou 2009). The assumption that percepts can be related to one another by graphs is sufficiently general that it can’t be contradicted. But the more specific assumption that those graphs are the same as those used for logic, language, and reasoning requires further research to fill in the details. The framework is sound, but the developments of the past quarter century have raised more issues to explore and questions to ask.

3. Neural and Psycholinguistic Evidence

Many of the controversies about implementing NLP systems are related to issues about how the human brain processes language. Figure 2 shows the left hemisphere of the brain; the base drawing was copied from Wikipedia, and the labels come from a variety of sources, of which MacNeilage (2008) is the most useful. Broca’s area and Wernicke’s area were the first two areas of the human brain recognized as critical to language. Lesions to Broca’s area impair the ability to generate speech, but they cause only a minor impairment in the ability to recognize speech. Significantly, the impairment in recognition is caused by an inability to resolve ambiguities that depend on subtle syntactic features. Lesions to Wernicke’s area impair the ability to understand language, but they don’t impair the ability to generate syntactically correct speech. Unfortunately, that speech tends to be grammatical nonsense whose semantic content is incoherent.

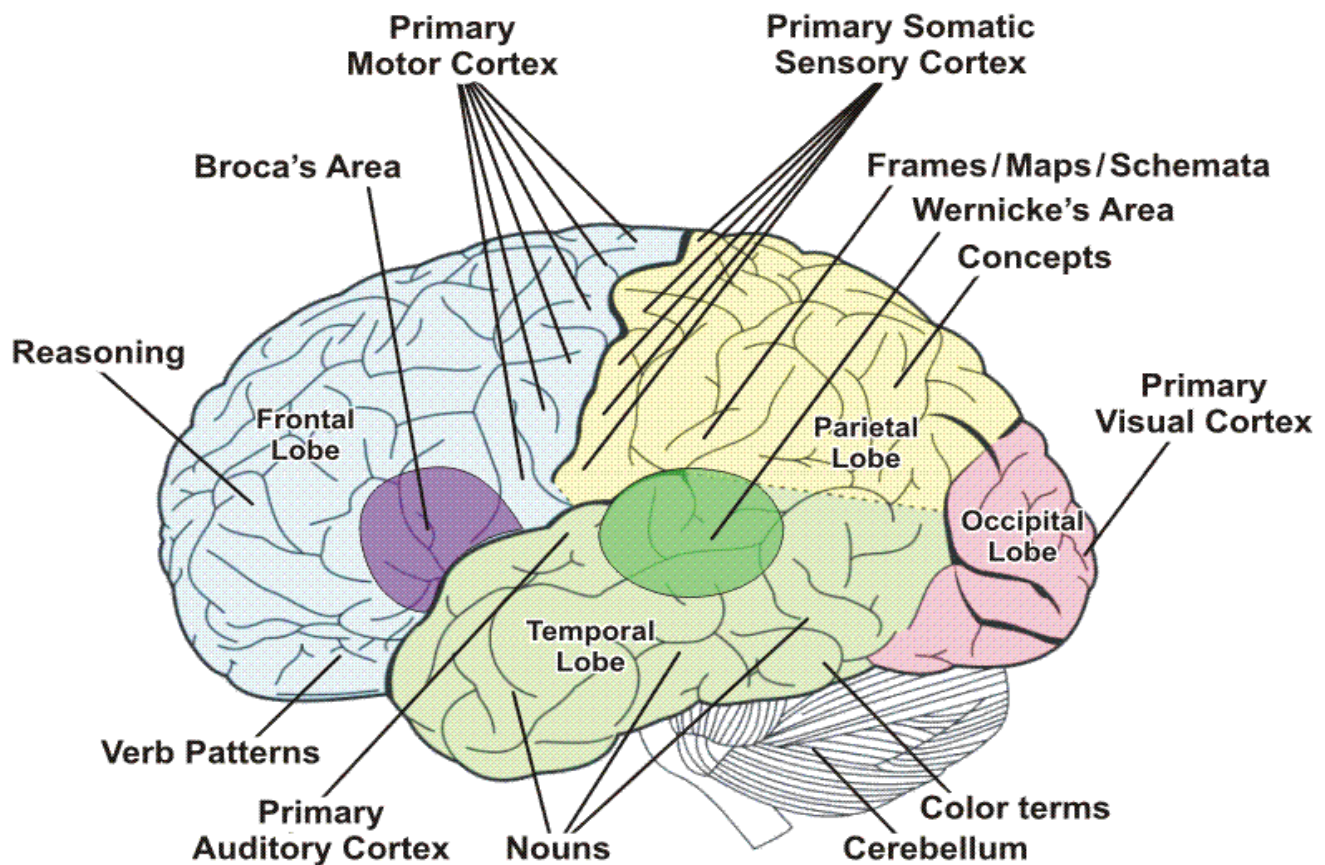


Figure 2. Language areas of the left hemisphere

The neural interconnections explain these observations: Wernicke's area is closely connected to the sensory projection areas for visual and auditory information. Wernicke's area is the first to receive speech input and link it to the store of semantic information derived from previous sensory input. Most of language can be interpreted by these linkages, even if Broca's area is damaged. Broca's area is close to the motor mechanisms for producing speech. It is responsible for fine-grained motions of various kinds, especially the detailed syntactic and phonological nuances in language generation. Lesions in Broca's area make it impossible to generate coherent syntactic structures and phonological patterns. For language understanding, Broca's area is not necessary to make semantic associations, but it can help resolve syntactic ambiguities.

These observations support the CS hypothesis that semantic-based methods are fundamental to language understanding. Wernicke's area processes semantics first, Broca's area operates in parallel to check syntax, and ambiguities in one can be resolved by information from the other. Meanwhile, the right hemisphere interprets pragmatics: emotion, prosody, context, metaphor, irony, and jokes, any of which could clarify, modify, or override syntax and semantics. Conflicts create puzzles that may require conscious attention (or laughter) to resolve.

The evidence also gives some support for the claim that generative syntax is independent of semantics (Chomsky 1957). Lesions in Broca's area impair the ability to generate grammatical speech, and lesions in Wernicke's area cause patients to generate grammatically correct, but meaningless sentences. But there is no evidence for the claim of an innate "universal grammar." Furthermore, the strong evidence for the importance of pragmatics suggests that Chomsky's emphasis on competence is more of a distraction than an aid to understanding cognition.

MacNeilage (2008) and Bybee (2010) argued that the structural support required for language need not be innate. General cognitive abilities are sufficient for a child to learn the syntactic and semantic patterns. Some of the commonalities found in all languages could result from the need to convert the internal forms to and from the linear stream of speech. In evolutionary terms, the various language areas have different origins, and their functions have similarities to the corresponding areas in monkeys and apes. As Figure 2 shows, verbs are closely associated with motor mechanisms while nouns are more closely connected to perception. It suggests that the syntactic structure of verbs evolved from their association with the corresponding actions, but nouns have primarily semantic connections. Deacon (1997, 2004) argued that the cognitive limitations of infants would impose further constraints on the patterns common to all languages: any patterns that a highly distractible infant finds hard to learn will not be preserved from one generation to the next.

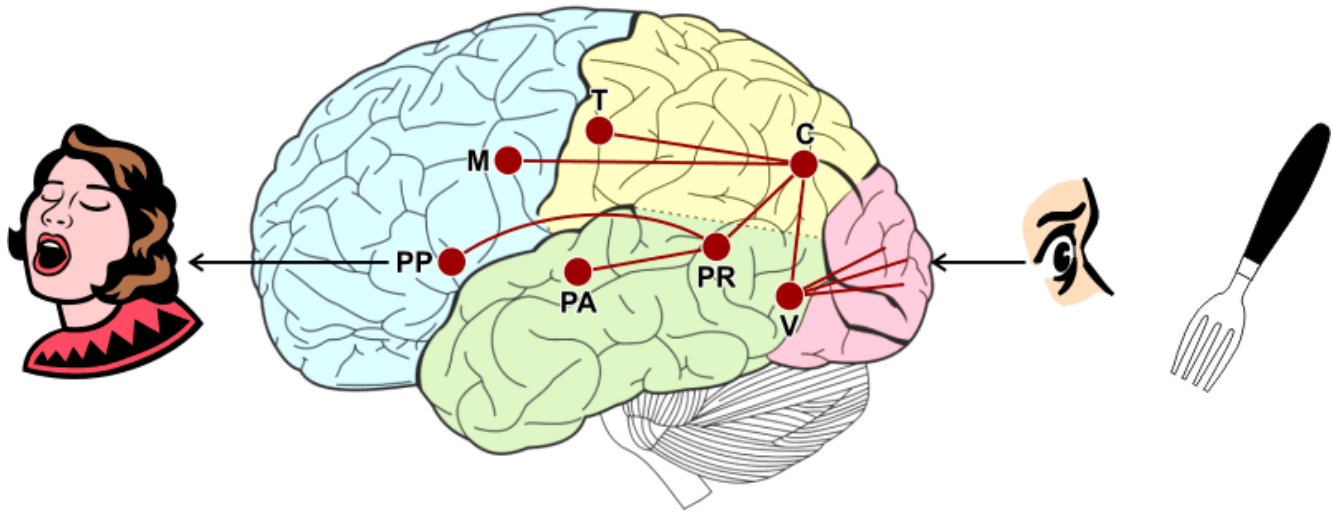


Figure 3. Neurocognitive network for the word *fork*

Figure 3 overlays the base drawing of Figure 2 with a network of connections for the word *fork* as proposed by Lamb (2010). The node labeled C represents the concept of a fork. It occurs in the parietal lobe, which is closely linked to the primary projection areas for all the sensory modalities. For the image of a fork, C is connected to node V, which has links to percepts for the parts and features of a fork in the visual cortex (occipital lobe). For the tactile sensation of a fork, C links to node T in the sensory area for input from the hand. For the motor schemata for manipulating a fork, C links to node M in the motor area for the hand. For the phonology for recognizing the word *fork*, C links to node PR in Wernicke's area. Finally, PR is linked to node PA for the sound /fork/ in the primary auditory cortex and to node PP in Broca's area for producing the sound.

The network in Figure 3 represents *semantic* or *metalevel* information about the links from a concept node C to associated sensory, motor, and verbal nodes. It shows how Lamb solves the symbol-grounding problem. Similar networks can link instance nodes to type nodes to represent *episodic* information about particular people, places, things, and events. Lamb's networks have many similarities to other versions of semantic networks, and they could be represented as conceptual graphs. CGs do have labels on the nodes, but those labels could be considered internal indexes that identify type nodes in Lamb's networks. Those networks, however, cannot express all the logical options of CGs, CycL, and other AI systems. Only one additional feature is needed to support them, and Peirce showed how.

4. Peirce's Logic and Semiotics

To support reasoning at the human level or at the level of Cyc and other engineering systems, a cognitive architecture requires the ability to express the logical operators used in ordinary language. Following are some sentences spoken by a child named Laura at age three (Limber 1973):

Here's a seat. It must be mine if it's a little one.

I want this doll because she's big.

When I was a little girl I could go "geek-geek" like that. But now I can go "this is a chair."

In these sentences, Laura correctly expressed possibility, necessity, tenses, indexicals, conditionals, causality, quotations, and metalanguage about her own language at different stages of life. She had a fluent command of a larger subset of intensional logic than Richard Montague formalized, but it's doubtful that her mental models would involve infinite families of possible worlds.

Lamb's neurocognitive networks can't express those sentences, but Peirce discovered a method for extending similar networks to express all of them. In 1885, he had invented the algebraic notation for predicate calculus and used it to express both first-order and higher-order logic. But he also experimented with graph notations to find a simpler way to express "the atoms and molecules of logic." His first version, called *relational graphs*, could express relations, conjunctions, and the existential quantifier. Following is a relational graph for the sentence *A cat is on a mat*: **Cat—On—Mat.**

In this notation, a bar by itself represents existence. The strings **Cat**, **On**, and **Mat** represent relations. In combination, the graph above says that there exists something, it's a cat, it's on something, and the thing it's on is a mat. Peirce invented this notation in 1883, but he couldn't find a systematic way to express all the formulas he could state in the algebraic notation. In 1897, he finally discovered a simple method: use an oval to enclose any graph or part of a graph that is negated. Peirce coined the term *existential graph* for relational graphs with the option of using ovals to negate any part. Figure 4 shows some examples.



Figure 4. Four existential graphs about pet cats.

The first graph on the left of Figure 4 says that some cat is a pet. The second graph is completely contained in a shaded oval, which negates the entire statement. It says that no cat is a pet. The third graph negates just the pet relation. It says that some cat is not a pet. The fourth graph negates the third graph. The simplest way to negate a sentence is to put the phrase "It is false that" in front of it: *It is false that there exists a cat which is not a pet*. But that combination of two negations can be read in much more natural ways: with a conditional, *If there is a cat, then it is a pet*; or with a universal quantifier, *Every cat is a pet*. Both readings are logically equivalent.

In general, Peirce's relational graphs, when combined with ovals for negation, have the full expressive power of first-order logic. Peirce later experimented with other features to express higher-order logic, modal logic, and metalanguage. With these extensions, existential graphs (EGs) have the full expressive power of CycL and most other AI logics. The CS book adopted Peirce's EGs as the foundation for conceptual graphs. In effect, CGs are typed versions of EGs with some extra features. But every CG can be translated to a logically equivalent EG. For an introduction to EGs, CGs, and their rules of inference, see the article by Sowa (2009); for extensions to metalanguage and modality, see Sowa (2003, 2006).

Even more important than the notation, the EG rules of inference do not require the complex substitutions and transformations of predicate calculus. They perform only two kinds of operations: inserting a graph or subgraph under certain conditions; or the inverse operation of deleting a graph or a subgraph under opposite conditions. These rules are sufficiently simple that they could be implemented on networks like Lamb's with only the operations of making, strengthening, or weakening connections.

Peirce called EGs his "chef d'oeuvre" and claimed that the rules of inference for EGs represent "a moving picture of the mind in thought." After a detailed comparison of Peirce's EGs to current theories about mental models, the psychologist Johnson-Laird (2002) agreed:

Peirce's existential graphs are remarkable. They establish the feasibility of a diagrammatic system of reasoning equivalent to the first-order predicate calculus. They anticipate the theory of mental models in many respects, including their iconic and symbolic components, their eschewal of variables, and their fundamental operations of insertion and deletion. Much is known about the psychology of reasoning... But we still lack a comprehensive account of how individuals represent multiply-quantified assertions, and so the graphs may provide a guide to the future development of psychological theory.

Although Peirce is best known for his work on logic, he incorporated logic in a much broader theory of signs that subsumes all possible cognitive architectures within a common framework. Every thought, feeling, or perception is a sign. Semiotics includes neural networks because every signal that passes between neurons or within neurons is a sign. Even a single bacterium is a semiotic processor when it swims upstream in following a glucose gradient. But the most fundamental semiotic process in any life form is the act of reproducing itself by interpreting signs called DNA. Figure 5 illustrates the evolution of cognitive systems according to the sophistication of their semiotic abilities.

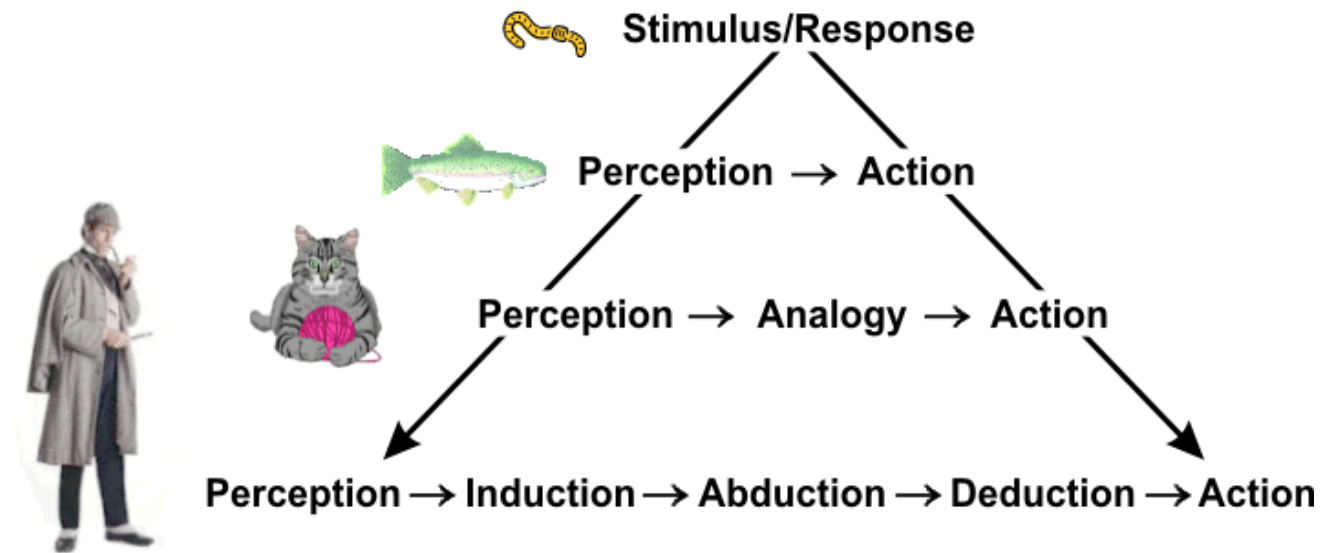


Figure 5. Evolution of cognition

The cognitive architectures of the animals at each stage of Figure 5 build on and extend the capabilities of the simpler stages. The worms at the top have rudimentary sensory and motor mechanisms connected by ganglia with a small number of neurons. A neural net that connects stimulus to response with just a few intermediate layers might be an adequate model. The fish brain is tiny compared to mammals, but it supports rich sensory and motor mechanisms. At the next stage, mammals have a cerebral cortex with distinct *projection areas* for each of the sensory and motor systems. It can support networks with analogies for case-based learning and reasoning. The cat playing with a ball of yarn is

practicing hunting skills with a mouse analog. At the human level, Sherlock Holmes is famous for his ability at induction, abduction, and deduction. Peirce distinguished those three ways of using logic and observed that each of them may be classified as a disciplined special case of analogy.

5. VivoMind Cognitive Architecture

The single most important feature of the VivoMind Cognitive Architecture (VCA) is the high-speed Cognitive Memory™. The first version, implemented in the VivoMind Analogy Engine (VAE), was invented by Arun Majumdar to support the associative comparator illustrated in Figure 1. Another feature, which was inspired by Minsky's Society of Mind, is the distribution of intelligent processing among heterogeneous agents that communicate by passing messages in the Flexible Modular Framework™ (Sowa 2002). Research on bilingualism supports *neurofunctional modularity* for human cognition (Paradis 2009). Practical experience on multithreaded systems with multiple CPUs has demonstrated the flexibility and scalability of a society of distributed heterogeneous agents:

- Asynchronous message passing for control and communication.
- Conceptual graphs for representing knowledge in the messages.
- Language understanding as a knowledge-based perceptual process.
- Analogies for rapidly accessing large volumes of knowledge of any kind.

Learning occurs at every step: perception and reasoning generate new conceptual graphs; analogies assimilate the CGs into Cognitive Memory™ for future use.

The VivoMind Language Processor (VLP) is a semantics-based language interpreter, which uses VAE as a high-speed associative memory and a society of agents for processing syntax, semantics, and pragmatics in parallel (Sowa & Majumdar 2003; Majumdar et al. 2008). During language analysis, thousands of agents may be involved, most of which remain dormant until they are triggered by something that matches their patterns. This architecture is not only computationally efficient, but it produces more accurate results than any single algorithm for NLP, either rule based or statistical.

With changing constraints on the permissible pattern matching, a general-purpose analogy engine can perform any combination of informal analogies or formal deduction, induction, and abduction. At the neat extreme, conceptual graphs have the model-theoretic semantics of Common Logic (ISO/IEC 24707), and VAE can find matching graphs that satisfy the strict constraints of unification. At the scruffy extreme, CGs can represent Schank's conceptual dependencies, scripts, MOPs, and TOPs. VAE can support case-based reasoning (Schank 1982) or any heuristics used with semantic networks. Multiple reasoning methods — neat, scruffy, and statistical — support combinations of heterogeneous theories, encodings, and algorithms that are rarely exploited in AI.

The Structure-Mapping Engine (SME) pioneered a wide range of methods for using analogies (Falkenhainer et al. 1989; Lovett et al. 2010). But SME takes N^3 time to find analogies in a knowledge base with N options. For better performance, conventional search engines can reduce the options, but they are based on an unordered bag of words or other labels. Methods that ignore the graph structure cannot find graphs with similar structure but different labels, and they find too many graphs with the same labels in different structures.

Organic chemists developed some of the fastest algorithms for representing large labeled graphs and efficiently finding graphs with similar structure and labels. Chemical graphs have fewer types of labels and links than conceptual graphs, but they have many similarities. Among them are frequently occurring subgraphs, such as a benzene ring or a methyl group, which can be defined and encoded as

single types. Algorithms designed for chemical graphs (Levinson & Ellis 1992) were used in the first high-speed method for encoding, storing, and retrieving CGs in a generalization hierarchy. More recent algorithms encode and store millions of chemical graphs in a database and find similar graphs in logarithmic time (Rhodes et al. 2007). By using a measure of graph similarity and locality-sensitive hashing, their software can retrieve a set of similar graphs with each search.

The original version of VAE used algorithms related to those for chemical graphs. More recent variations have led to a family of algorithms that encode a graph in a *Cognitive Signature*[™] that preserves both the structure and the ontology. The encoding time is polynomial in the size of a graph. With a semantic distance measure based on both the structure of the graphs and an ontology of their labels, locality-sensitive hashing can retrieve a set of similar graphs in $\log(N)$ time, where N is the total number of graphs in the knowledge base. With this speed, VAE can find analogies in a knowledge base of any size without requiring a search engine as a preliminary filter. For examples of applications, see the slides by Sowa and Majumdar (2009).

The distributed processing among heterogeneous agents supports Peirce's cycle of pragmatism, as illustrated in Figure 6. That cycle relates perception to action by repeated steps of induction, abduction, reasoning, and testing. Each step can be performed by an application of analogy or by a wide variety of specialized algorithms.

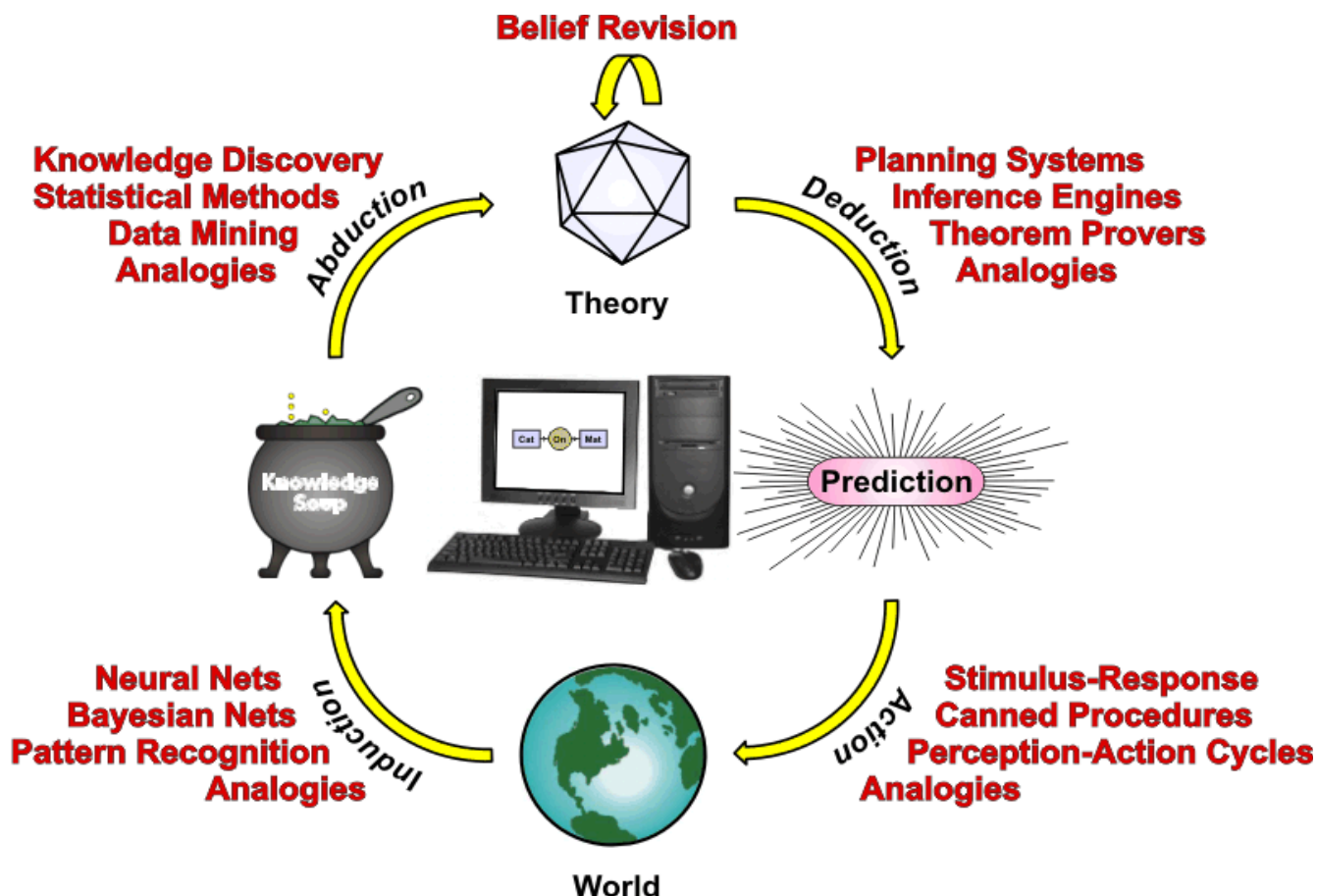


Figure 6. Cycle of Pragmatism

The cycle of pragmatism shows how the VivoMind architecture brings order out of a potential chaos (or Pandemonium). The labels on the arrows suggest the open-ended variety of heterogeneous algorithms, each performed by one or more agents. During the cycle, the details of the internal

processing by any agent is irrelevant to other agents. It could be neat, scruffy, statistical, or biologically inspired. The only requirement is the conventions on the interface to the FMF. An agent that uses a different interface could be enclosed in a wrapper. The overall system is fail soft: a failing agent that doesn't respond to messages is automatically replaced by another agent that can answer the same messages, but perhaps in a very different way. Agents that consistently produce more useful results are rewarded with more time and space resources. Agents that are useless for one application might be rewarded in another application for which their talents are appropriate.

The society of agents can have subsocieties that traverse the cycle of pragmatism at different speeds. Societies devoted to low-level perception and action may traverse each cycle in milliseconds. Societies for reasoning and planning may take seconds or minutes. A society for complex research might take hours, days, or even years.

References

- Anderson, John R., & Gordon H. Bower (1980) *Human Associative Memory: A Brief Edition*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Barsalou, Lawrence W. (2009) Simulation, situated conceptualization, and prediction, *Philosophical Transactions of the Royal Society B* **364**, 1281-1289.
- Bybee, Joan (2010) *Language, Usage, and Cognition*, Cambridge: University Press.
- Chomsky, Noam (1957) *Syntactic Structures*, Mouton, The Hague.
- Deacon, Terrence W. (1997) *The Symbolic Species: The Co-evolution of Language and the Brain*, W. W. Norton, New York.
- Deacon, Terrence W. (2004) Memes as signs in the dynamic logic of semiosis: Beyond molecular science and computation theory, in K. E. Wolff, H. D. Pfeiffer, & H. S. Delugach, *Conceptual Structures at Work*, LNAI 3127, Springer, Berlin, pp. 17-30.
- Falkenhainer, Brian, Kenneth D. Forbus, Dedre Gentner (1989) The structure mapping engine: algorithm and examples, *Artificial Intelligence* **41**, 1-63.
- Harrison, Colin James (2000) PureNet: A modeling program for neurocognitive linguistics, PhD dissertation, Rice University.
- Heidorn, George E. (1972) *Natural Language Inputs to a Simulation Programming System*, Report NPS-55HD72101A, Naval Postgraduate School, Monterey, CA.
- ISO/IEC (2007) *Common Logic (CL) — A Framework for a family of Logic-Based Languages*, IS 24707, International Organisation for Standardisation, Geneva.
- Johnson-Laird, Philip N. (2002) Peirce, logic diagrams, and the elementary processes of reasoning, *Thinking and Reasoning* **8:2**, 69-95.
- Laird, John E. (2008) Extending the Soar cognitive architecture, in P. Wang, B. Goertzel, and S. Franklin, eds. *Artificial General Intelligence 2008*, Amsterdam: IOS Press, pp. 224-235.
- Lamb, Sidney M. (1966) *Outline of Stratificational Grammar*, Georgetown University Press, Washington, DC.
- Lamb, Sydney M. (1999) *Pathways of the Brain: The Neurocognitive Basis of Language*, Amsterdam: John Benjamins.
- Lamb, Sydney M. (2004) *Language and Reality*, London: Continuum.
- Lamb, Sydney M. (2010) Neurolinguistics, Lecture Notes for Linguistics 411, Rice University. <http://www.owl.net.rice.edu/~ling411>
- Lenat, Douglas B., & Edward A. Feigenbaum (1987) On the thresholds of knowledge, *Proc. IJCAI'87*, pp. 1173-1182.

- Lenat, D. B., & R. V. Guha (1990) *Building Large Knowledge-Based Systems*, Reading, MA: Addison-Wesley.
- Limber, John (1973) The genesis of complex sentences, in T. Moore, ed., *Cognitive Development and the Acquisition of Language*, New York: Academic Press, pp. 169-186.
- Lovett, Andrew, Kenneth Forbus, & Jeffrey Usher (2010) A structure-mapping model of Raven's Progressive Matrices, *Proceedings of CogSci-10*, pp. 2761-2766.
- MacNeilage, Peter F. (2008) *The Origin of Speech*, Oxford: University Press.
- Majumdar, Arun K., John F. Sowa, & John Stewart (2008) Pursuing the goal of language understanding, in P. Eklund LNAI 5113, Springer, Berlin, 2008, pp. 21-42. <http://www.jfsowa.com/pubs/pursuing.pdf>
- Majumdar, Arun K., & John F. Sowa (2009) Two paradigms are better than one and multiple paradigms are even better, in S. Rudolph, F. Dau, and S.O. Kuznetsov, eds., <http://www.jfsowa.com/pubs/paradigm.pdf>
- McNeill, David (1970) *The Acquisition of Language*, Harper & Row, New York.
- Minsky, Marvin (1986) *The Society of Mind*, Simon & Schuster, New York.
- Minsky, Marvin Lee (2006) *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*, Simon & Schuster, New York.
- Newell, Allen (1990) *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.
- Newell, Allen, & Herbert A. Simon (1961) GPS, a program that simulates human thought, reprinted in Feigenbaum & Feldman (1963) 279-293.
- Newell, Allen, & Herbert A. Simon (1972) *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ.
- Paradis, Michel (2009) *Declarative and Procedural Determinants of Second Languages*, Amsterdam: John Benjamins.
- Peirce, Charles Sanders (CP) *Collected Papers of C. S. Peirce*, ed. by C. Hartshorne, P. Weiss, & A. Burks, 8 vols., Harvard University Press, Cambridge, MA, 1931-1958.
- Quillian, M. Ross (1966) *Semantic Memory*, Report AD-641671, Clearinghouse for Federal Scientific and Technical Information.
- Samsonovich, Alexei V. (2010) Toward a unified catalog of implemented cognitive architectures, in A. V. Samsonovich et al., eds., *Biologically Inspired Cognitive Architectures 2010*, Amsterdam: IOS Press, pp. 195-244.
- Singh, Push (2003) Examining the society of mind, *Computing and Informatics* **22**, 521-543.
- Sowa, John F. (1976) Conceptual graphs for a data base interface, *IBM Journal of Research and Development* **20:4**, 336-357. <http://www.jfsowa.com/pubs/cg1976.pdf>
- Sowa, John F. (1984) *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA.
- Sowa, John F. (2002) Architectures for intelligent systems, *IBM Systems Journal* **41:3**, 331-349. <http://www.jfsowa.com/pubs/arch.htm>
- Sowa, John F. (2003) Laws, facts, and contexts: Foundations for multimodal reasoning, in *Knowledge Contributors*, edited by V. F. Hendricks, K. F. Jørgensen, and S. A. Pedersen, Kluwer Academic Publishers, Dordrecht, pp. 145-184. <http://www.jfsowa.com/pubs/laws.htm>
- Sowa, John F. (2005) The challenge of knowledge soup, in J. Ramadas & S. Chunawala, *Research Trends in Science, Technology, and Mathematics Education*, Homi Bhabha Centre, Mumbai, pp. 55-90. <http://www.jfsowa.com/pubs/challenge.pdf>
- Sowa, John F. (2006) Worlds, Models, and Descriptions, *Studia Logica*, Special Issue *Ways of Worlds II*, **84:2**, 323-360. <http://www.jfsowa.com/pubs/worlds.pdf>
- Sowa, John F. (2009) Conceptual Graphs for Conceptual Structures, in P. Hitzler & H. Schärfe, eds., *Conceptual Structures in Practice*, Chapman & Hall/CRC Press, pp. 102-136. <http://www.jfsowa.com/pubs/cg4cs.pdf>

Sowa, John F. (2010) The role of logic and ontology in language and reasoning, Chapter 11 of *Theory and Applications of Ontology: Philosophical Perspectives*, edited by R. Poli & J. Seibt, Berlin: Springer, pp. 231-263. <http://www.jfsowa.com/pubs/rolelog.pdf>

Sowa, John F., & Eileen C. Way (1986) Implementing a semantic interpreter using conceptual graphs, *IBM Journal of Research and Development* **30:1**, 57-69. <http://www.jfsowa.com/pubs/cg1986.pdf>

Sowa, John F., & Arun K. Majumdar (2003) Analogical reasoning, in A. de Moor, W. Lex, & B. Ganter, eds., *Conceptual Structures for Knowledge Creation and Communication*, LNAI 2746, Springer-Verlag, Berlin, pp. 16-36. <http://www.jfsowa.com/pubs/analog.htm>

Sowa, John F., & Arun K. Majumdar (2009) Slides of VivoMind applications, <http://www.jfsowa.com/talks/pursue.pdf>